

ECE 445S

Real-Time Digital Signal Processing Laboratory

Prof. Brian L. Evans

Dept. of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712

bevans@ece.utexas.edu

<http://www.ece.utexas.edu/~bevans>

Spring 2025

Table of Contents

0. Introduction
 1. Sinusoidal Generation
 - “Designing Averaging Filters”
 - “Discrete-Time Periodicity”
 - “Sampling Unit Step Function”
 2. Introduction to Digital Signal Processors
 - “Trends in Multicore DSPs”
 - “Comparing Fixed- and Floating-Point DSPs”
 3. Signals and Systems
 4. Sampling and Aliasing
 5. Finite Impulse Resp. Filters
 6. Infinite Impulse Response Filters
 - “BIBO Stability”
 - “Parallel and Cascade Forms”
 7. Interpolation & Pulse Shaping
 8. Quantization
 9. *TMS320C6000 Digital Signal Processors (DSPs)*
 10. Data Conversion – Part I
 11. *Data Conversion – Part II*
 12. Channel Impairments
 13. Digital Pulse Amplitude Modulation (PAM)
 14. Matched Filtering
 15. Quadrature Amplitude Modulation Transmitter
 16. QAM Receiver
 17. *Fast Fourier Transform.*
- Review for Midterm #2

Other Handouts

- A. Course Description
- B. Instructional Staff/Resources
- C. ECE IT Support
- D. Matlab
- E. Convolution Example
- F. Theorem of Linear Systems
- G. Raised Cosine Pulse
- H. Modulation Example
- I. Modulation Summary
- J. Noise-Shaped Feedback
- K. Sample Midterm Exams
- L. Direct Sequence Spreading
- M. Symbol Timing Recovery
- N. Tapped Delay Line on C6700
- O. All-Pass Filters
- P. QAM vs. PAM
- Q. Four Ways To Filter
- R. Intro to Fourier Transforms
- S. Adding Random Variables
- T. Fourier Transform Property of Multiplication in Time
- U. Time Invariance Property
- V. LTI Filters and Frequency Selectivity
- W. Time Invariance for an Integrator

ECE445S Real-Time DSP Lab Course

Lecture and Labs

This course is a four-credit course, with three hours of lecture and three hours of lab per week.

Lecture will be in ECJ 1.318 on Mondays and Wednesdays, from 10:30am to 12:00pm, from Aug. 26th to Dec. 9th. [Laboratory sections](#) will meet from Sep. 2nd to Dec. 6th, in EER 1.810.

This course does not require a semester project nor does it have a final examination. Final grades will consist of pre-lab quizzes, lab reports, homework assignments, in-lecture work, and exams. Exams will be based on material covered in lecture, homework, lab sessions and reading assignments.

[Lecture slides](#) (30 MB) and the [course reader](#) (120 MB) are available for download.

Schedule of lecture/lab topics and reading assignments follows. JSK means Johnson, Sethares & Klein, *Software Receiver Design*. UYG means Unsalan, Yucel & Gurhan, *DSP Using Arm Cortex-M Microcontrollers*.

Week	Monday Lecture	Wednesday Lecture	Major Assignment Due	Lab	Reading Assignment
Aug. 26th	Introduction	Sinusoidal Generation		Review of Prerequisites	Wednesday: JSK ch. 1-2 Reader handouts A-D & R
Sep. 2nd	LABOR DAY HOLIDAY	Sinusoidal Generation	Homework #0 (Friday 11pm)	Introduction Tools	Monday: Pre-lab Reading and JSK 3.1-3.3, 3.6 & 5.2 Wednesday: JSK 3.4, 3.5, 3.7; Reader handouts H & I ; Common Signals in Matlab
Sep. 9th	Sinusoidal Generation	Signals and Systems	Lab #1 Report (Friday 11pm)	Sine Wave Generation	Monday: Pre-lab Quiz Monday: JSK 4.1-4.5, app. A.2, A.4, G.1 & G.2; Reader handouts E & F Wednesday: JSK ch. 3-4
Sep. 16th	Signals and Systems	Signals and Systems	Homework #1 (Friday 11pm)	Sine Wave Generation	Monday: JSK 5.1-5.2, 7.1-7.2 & app. G
Sep. 23rd	Finite Impulse Response Filters	FIR Filters	Lab #2 Report (Friday 11pm)	Digital Filters	Monday: Pre-lab Quiz Monday: JSK 7.1-7.2, app. F & G
Sep. 30th	FIR Filters	Infinite Impulse Response Filters	Homework #2 (Friday 11pm)	Digital Filters	Monday: JSK 5.1-5.2, 6.1-6.3, app. A.3; Wednesday: Reader handout O
Oct. 7th	IIR Filters	IIR Filters	Homework #3 (Friday 11pm)	Digital Filters	
Oct. 14th	Sampling and Aliasing	Midterm #1	Midterm #1 (Wednesday noon)	NO LAB SECTIONS	

There are several [in-lecture assignments](#) throughout the semester.

Spring break is March 9-16, 2024.

For the second half of the semester, the schedule of lecture/lab topics follows.

Week	Monday Lecture	Wednesday Lecture	Major Assignments Due	Lab	Reading
Oct. 21st	Interpolation and Pulse Shaping	Interpolation and Pulse Shaping	Lab #3 Report (Friday 11pm)	Data Scramblers	Monday: Pre-lab Quiz Monday: JSK 6.4-6.6 and Reader handout G
Oct. 28th	Digital Pulse Amplitude Modulation	Digital Pulse Amplitude Modulation	Lab #4 Report (Friday 11pm) Homework #4 (Friday 11pm)	Pulse Amplitude Modulation	Monday: Pre-lab Quiz Monday: JSK 4.6, 8.1 and 8.2; Wednesday: JSK 9.1-9.4, app. B & E; Reader handout S
Nov. 4th	Digital Pulse Amplitude Modulation	Channel Impairments	Homework #5 (Friday 11pm)	Pulse Amplitude Modulation	Wednesday: JSK 8.3-8.5 and ch. 11 & 12; Haykin 4.6, 4.9-4.12
Nov. 11th	Matched Filtering	Matched Filtering	Homework #6 (Friday 11pm)	Pulse Amplitude Modulation	Monday: JSK 5.3-5.4 Wednesday: JSK 16.1-16.2; Reader handout I and handout M
Nov. 18th	Quadrature Amplitude Modulation (QAM) Transmitter	QAM Receiver	Lab #5 Report (Friday 11pm)	Quadrature Amplitude Modulation	Monday: Pre-lab Quiz Wednesday: JSK 6.7-6.8, 10.1-10.4, 13.1-13.3, 16.3-16.8; Reader handout P
Nov. 25th	THANKSGIVING	HOLIDAY		<i>Happy</i>	<i>Holidays</i>
Dec. 2nd	Quantization	Review	Homework #7 (Friday 11pm)	Guitar Special Effects	Monday: Pre-lab Quiz Monday: JSK 2.13, 3.5 & 8.4 Wednesday: Reader handout J
Dec. 9th	Midterm #2	FINAL EXAM STUDY DAY	Midterm #2 (Monday noon)	NO LAB	

 [Playlist of lectures recorded in spring 2014](#)

The following lectures are not scheduled to be presented this semester: [Digital Signal Processors - TMS320C6000 DSP - Advanced Data Conversion](#) - [Fast Fourier Transforms](#) - [DSL Modems](#) - [Analog Sinusoidal Modulation](#) - [Wireless OFDM Systems](#) - [WiMAX](#) - [Spread Spectrum Communications](#) - [Modern DSP Processors](#) - [Native Signal Processing](#) - [Algorithm Interoperability](#) - [System-level Design](#) - [Synchronization in ADSL Modems](#) - [Wireless 1000x](#)

ECE445S Real-Time Digital Signal Processing Laboratory - Overview

[Prof. Brian L. Evans](#)

The learning objectives for the Real-Time Digital Signal Processing Lab course are to

- build intuition for applying signal processing concepts
- design algorithms with tradeoffs in signal quality vs. run-time complexity in mind

The first objective is to help students use critical thinking via inductive reasoning to put the pieces together from the pre-requisite required courses to form a big picture of the field. Signal processing concepts enable a wide variety of applications across science and engineering; the same signal processing concepts apply to speech, audio, image, video, and communication systems, as well as biomedical instrumentation and geophysical analysis. This undergraduate course provides practical insights into waveform generation and filtering, system training and calibration, and communication transmission and reception.

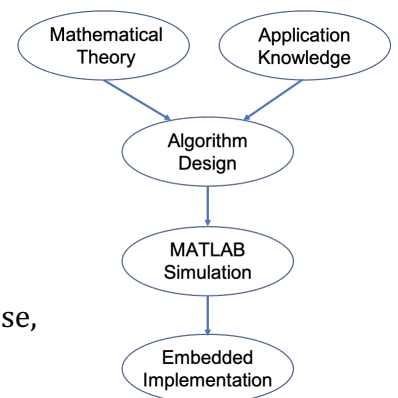
What changes across applications in how one measures the quality of the signal. When humans are the ultimate consumer of the information, speech and audio quality is about the perceived auditory quality. Likewise, image and video quality is about the perceived visual quality. When machine interpretation of the data is ultimate consumer, the signal quality relates to the accuracy and precision of the machine interpretation.

The second objective relates to the critical thinking via deductive reasoning in designing signal processing algorithms. Each algorithm can achieve a certain signal quality but at a certain implementation cost. For implementation cost, this course focuses on how fast the algorithm executes or runs on the input data to produce output data, which is called *run-time complexity*. For a given application, students would evaluate multiple algorithms to pick the one with the best tradeoff. This is a critical thinking exercise.

The second objective introduces students to the design and implementation of signal processing systems. This introduction will help them in the two-semester senior design sequence, for which the Real-Time Digital Signal Processing Lab course is one of seven lab course pre-requisites. The introduction will also help them in courses across several application spaces, including EE 360K Digital Communications, EE 371Q Digital Image Processing, and EE 374K Biomedical Instrumentation, as well as courses in system design and implementation, such as EE 445M Embedded and Real-Time Systems.

As shown on the right, students will use mathematical theory to model applications. From those models, they will design algorithms, simulate them in MATLAB, and map them to embedded software in C. They will quantify tradeoffs in signal quality vs. run-time complexity in the algorithms. Students will connect application theory from their signals and systems course (EE 313/BME 343) and implementation skills from their courses in programming (EE 312) and embedded systems (EE 319K). After the 50% point in the semester, students will begin using concepts from probability (EE 351K) to model thermal noise, quantization error, and other statistical phenomenon in applications.

The specific topics in this course follow:



System Design Flow

Digital signal processing algorithms; simulation and real-time implementation of audio and communication systems; filters; pulse shaping and matched filters; modulation and demodulation; adaptive filters; carrier recovery; symbol synchronization; equalization; quantization.

In the lab and lecture, the course will cover

- *digital signal processing*: signals, sampling, filtering, quantization, oversampling, noise shaping, and data converters.
- *digital communications*: Analog/digital modulation, analog/digital demodulation, pulse shaping, pseudo-noise sequences, ADSL transceivers, and wireless LAN transceivers.
- *digital signal processor architectures*: Harvard architecture, special addressing modes, parallel instructions, pipelining, and real-time programming.

In particular, we will quantify algorithm tradeoffs in signal quality vs. run-time complexity.

In the lab component, students implement transceiver subsystems in C on a Texas Instruments TMS320C6748 floating-point programmable digital signal processor. The C6000 family is used in DSL modems, wireless LAN modems, cellular basestations, and video conferencing systems. For professional audio systems, the C6700 floating-point sub-family empowers guitar effects and intelligent mixing boards. Students test their implementations using Texas Instruments Code Composer Studio software and rack equipment. A [voiceband transceiver reference design and simulation](#) is available.

In addition to learning about transceiver design in the lab and lecture, students will also learn in lecture about the design of modern analog-to-digital and digital-to-analog converters, which employ oversampling, filtering, and dithering to obtain high resolution. Whereas the voiceband modem is a single carrier system, lectures will also cover modern multicarrier modulation systems, as used in cellular LTE communications and Wi-Fi systems. Last, we spend several lectures on digital signal processor architectures, esp. the architectural features adopted to accelerate digital signal processing algorithms.

For the lab component, I chose a floating-point DSP over a fixed-point DSP. The primary reason was to avoid overwhelming the students with the severe fixed-point precision effects so that the students could focus on the design and implementation of real-time digital communications systems. That said, floating-point DSPs are used in industry to prototype algorithms, e.g. to see if real-time performance can be met. If the prototype is successful, then it might be modified for low-volume applications or it might be mapped onto a fixed-point DSP for high-volume applications (where the engineering time for the mapping can potentially be recovered).

Texas is a worldwide epicenter for microprocessors for control, signal processing, and communication systems. Texas Instruments (Dallas, TX) and NXP (Austin, TX) are market leaders in the embedded programmable microcontroller market, esp. for the automotive sector. Texas Instruments and NXP design their microcontrollers and digital signal processors in Texas. Qualcomm designs their Snapdragon processors in Austin, Texas, among other places. Cirrus Logic designs the audio playback subsystem (audio data converter and audio digital signal processor chipsets) and microphone subsystems for the iPhone in Austin. Apple designs its application processors for the iPhone in Austin. To boot, Austin is a worldwide leader in ARM-based digital VLSI design centers.

Introduction

Prof. Brian L. Evans

**Dept. of Electrical and Computer Engineering
The University of Texas at Austin**

Lecture 0 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Welcome

Instructional Staff

- **Support learning & growth to reach goals**
Lecture/lab discussions and assignments
Friday 12-2pm coffee/advising hours (EER O's Café)
In-class and Canvas announcements
- **Prof. Brian L. Evans (he/him)**
Office hours: W 1:00-3:00pm (EER 6.882 & Zoom)
Available in person immediately after lecture



TAs Mr. Faraz Barati (he/him)



Lab M 5:00-8:00pm
T 3:30-6:30pm
Office W 4:30-6:00pm
TH 2:00-3:30pm

Mr. Yongjin Eun (he/him)



Lab W 6:30-9:30pm
F 1:00-4:00pm
Office TH 3:30-5:00pm
F 4:00-5:30pm

0-2

2

Welcome

De-stress Course to Improve Learning

- **Review pre-requisite course info first 3-4 weeks**
- **Predictable weekly schedule and workload**
Balanced workload week-to-week (one major deliverable)
Homework & pre-lab quizzes help you prepare for lab work
Excused absences: religious observance, military orders, COVID
Drop lowest homework, lab report, pre-lab quiz, and in-lecture assignment to balance workload and handle unexpected events
- **Feedback on assignments by next class meeting**
Always know equivalent letter grade: 90+ A, 86.67-89.99 A-, 83.34-86.66 B+, 80.00-83.33 B, 76.67-79.99 B-, etc.
- **Access to previous homework & midterm solutions**

0-3

3

Welcome

Getting the Most Out of This Course

- **Student advice (during Friday coffee hours)**
Remove distractions preventing efficiency in work and play
Write questions on your phone when they arise & ask them
Find a mutually beneficial study group of 2-3 persons
- **Professor advice**
Workload: Choose courses for work-life balance (next slide)
Efficiency: Attend lecture and instructor/TA office hours for summaries of hours of reading into minutes of coverage
Lecture: Better focus when in person; use recordings for review
Assignments: start when assigned and make progress each day: having more calendar time will allow you to learn info better **

** D. Brewer-Deluce, "3 ways to study better, according to cognitive research", *The Conversation*, Aug. 2020

4

Course Overview

Related Undergraduate Courses

Comm/SigProc/Networks
 Digital Image Processing
 Digital Signal Processing

Comp Arch/Emb. Proc.
 Embedded & Real-Time Sys.
 Digital Sys. Design (FPGAs)

Data Science & Info. Proc.
 Data Science Lab
 Data Science Principles

Fields, Waves, Emag Sys.
 Antennas & Wireless Prop.
 Microwave & RF Eng.

Undergrad ECE courses with highest workload ratings?

4.70	460N Computer Arch	F21
4.47	445L Emb Sys Des Lab	Sp22
4.13	445L Emb Sys Des Lab	F21
4.05	461S Operating Systems	F21
3.88	460N Computer Arch	Sp22
3.88	461S Operating Systems	Sp22
3.80	325 Electromagnetic Eng.	Sp22
3.78	422C SW Des & Impl II **	F21
3.77	460R Intro to VLSI Design	F21

↑ Workload rating (1-5 with 5 highest)





** Averaged across all sections that semester. **

5

Course Overview

Real-Time Digital Signal Processing

- **Signal carries information**
- **Signal processing** [signalprocessingsociety.org]
 - Generation, transformation, extraction of information
 - Algorithms with associated architectures and implementations
 - Applications related to processing information
- **System**
 - Manipulates, changes, records or transmits signals
 - Converts a signal into another
- **Real-time systems** [Prof. Yale N. Patt, UT Austin]
 - Guarantee delivery of data by a specific time

6

Course Overview

Real-Time DSP Algorithms in Products



Consumer audio



Communications



Biomedical



Pro-audio



Multimedia



Biomedical

0-7

7

Course Overview

33 PhD, 13 MS and 215 BS alumni

Prof. Evans' Research Projects

System	Contribution	SW release	Prototype	Funding
Cellular (LTE)	full duplex commun.	Matlab		AT&T Labs
	machine learning	Python		NVIDIA
	reconfigurable surface	Matlab		6G@UT
Smart grid commun.	interference reduction real-time testbeds	Matlab	Freescale & TI modems	IBM, NXP, TI
Wi-Fi	interference reduction	Matlab	NI FPGA	Intel, NI
Underwater	large receive array	Matlab	Lake testbed	ARL:UT
Camera	image acquisition	Matlab	DSP/C	Intel, Ricoh
	video acquisition	Matlab	Android	TI
Display	image halftoning	Matlab	C	HP, Xerox
	video halftoning	Matlab	C	Qualcomm
Design tools	distributed computing	Linux/C++	Navy sonar	Navy, NI

DSP Digital Signal Processor FPGA Field Programmable Gate Array 0-8

8

Course Objectives

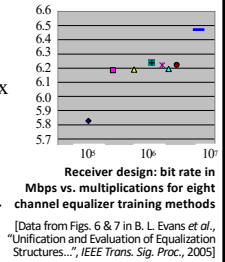
- **Course Objectives**
Build intuition for signal processing concepts
Explore design tradeoffs in *signal quality* vs. *run-time implementation complexity*
- **Lecture: breadth (3 hours/week)**
Building blocks (sinusoids, filters, etc.)
Digital signal processing (DSP) algorithms
Digital communication systems
- **Laboratory: depth (3 hours/week)**
Translate DSP concepts into software
Design/implement data transceiver
Test/validate implementation

Measures of signal quality?
Run-time implementation complexity?

9

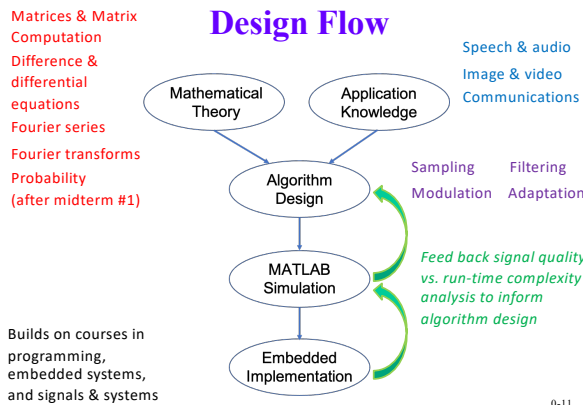
Design Tradeoffs

- **Course Objectives**
Build intuition for signal processing concepts
Explore design tradeoffs in *signal quality* vs. *run-time implementation complexity*
- **Asymmetric Digital Subscriber Line (ADSL) Receiver Design**
Channel equalizer (filter) gives up to 10x increase in bit rate vs. not having one
ADSL transmitter sends training data
Eight adaptive design methods for the channel equalizer in receiver on right
What are the methods with best tradeoff in bit rate vs. run-time complexity?



10

Design Flow



11

Required Textbooks

Software Receiver Design, Oct. 2011

Design of digital communication systems
Convert algorithms into Matlab simulations



Rick Johnson
(Cornell)



Bill Sethares
(Wisconsin)



Andy Klein
(West. Wash.)



Cem Ünsalan
(Marmara University)



M. Erkin Yucel
(Yeditepe)



H. Deniz Gurhan
(Yeditepe)

Digital Signal Processing using Arm Cortex-M based Microcontrollers Sep. 2018

Mapping algorithms to C

0-12

12

Grading

- **Calculation of numeric grades**

20% midterm #1

20% midterm #2

14% homework (*drop lowest grade of eight*)

5% in-lecture assignments (*drop lowest grade*)

5% pre-lab quizzes (*drop lowest grade of six*)

36% lab reports (*drop lowest grade of seven*)

- **20% for each midterm exam ****

In-person exam (75 minutes to complete a 50-minute exam)

Algorithm tradeoffs in signal quality vs. run-time complexity

Based on in-lecture discussions and homework/lab assignments

Dozens of old exams online and in reader (most with solutions)

Average GPA
has been ~3.5

No final
exam

** Be sure to submit your own independent work.

0-13

13

Grading

- **14% homework – 8 assignments (*drop lowest*) ****

Translate signal processing concepts into Matlab simulations

Evaluate tradeoffs in signal quality vs. run-time complexity

- **5% in-lecture assignments (*drop lowest grade*)**

Helps prepare for the homework assignment due Friday

- **5% pre-lab quizzes – for labs 2-7 (*drop lowest*) ****

10 questions on lecture/lab info taken individually on Canvas

- **36% lab reports – for labs 1-7 (*drop lowest*)**

Work *individually* on labs 1&7 and *in team of two* on labs 2-6

Attendance/participation in lab section required and graded

- **Course rank in grad school rec letters when helpful**

** Be sure to submit your own independent work.

0-14

14

ECE 445S Real-Time Digital Signal Processing Lab Spring 2025

Generating Sinusoidal Signals

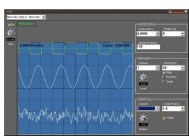
Prof. Brian L. Evans
Dept. of Electrical and Computer Engineering
The University of Texas at Austin

Lecture 1 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

- **Bandwidth**
- **Sinusoidal amplitude modulation**
- **Sinusoidal generation**
- **Design tradeoffs**
- **Communication systems**
 - Wireless communication systems
 - Software transceiver design



1-2

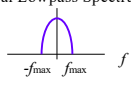
2

Bandwidth

Bandwidth

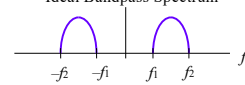
- **Non-zero extent in positive frequencies**

Ideal Lowpass Spectrum



Bandwidth f_{max}

Ideal Bandpass Spectrum



Bandwidth $W = f_2 - f_1$

- **Applies to continuous-time & discrete-time signals**
- **In practice, spectrum won't be ideally bandlimited**
 - Thermal noise has "flat" power spectrum from 0 to 10^{15} Hz
 - Finite observation time of signal leads to infinite bandwidth
- **Alternatives to "non-zero extent"?**

1-3

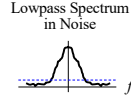
3

Bandwidth

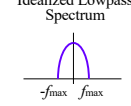
Bandwidth of Lowpass Signal in Noise

- **How to determine f_{max} ?**
 - Apply threshold and eyeball it
 - OR–
 - Estimate f_{max} that captures certain percentage (say 90%) of energy

Lowpass Spectrum in Noise



Idealized Lowpass Spectrum



approximate

where Energy = $\int_0^{\infty} |H(f)|^2 df$

In practice, (a) use large frequency in place of ∞ and (b) integrate measured spectrum numerically

Baseband signal: energy in frequency domain concentrated around DC

1-4

4

Bandwidth

Bandwidth of Bandpass Signal in Noise

- **How to find $W = f_2 - f_1$?**
 Apply threshold and eyeball it
 -OR-
 Assume knowledge of f_c and estimate $f_1 = f_c - W/2$ and $f_2 = f_c + W/2$ that capture percentage of energy

$\min_W \int_{f_c - \frac{1}{2}W}^{f_c + \frac{1}{2}W} |H(f)|^2 df \geq 0.9 \text{ Energy}$

where Energy = $\int_0^\infty |H(f)|^2 df$ and $0 \leq W \leq 2f_c$

In practice, (a) use large frequency in place of ∞ and (b) integrate measured spectrum numerically

1-5

5

Sinusoidal Amplitude Modulation

Amplitude Modulation by Cosine

- $y_I(t) = x_I(t) \cos(\omega_c t)$ See Slides 1-27, 3-6 and 15-6

$$Y_I(\omega) = \frac{1}{2\pi} X_I(\omega) * (\pi \delta(\omega + \omega_c) + \pi \delta(\omega - \omega_c)) = \frac{1}{2} X_I(\omega + \omega_c) + \frac{1}{2} X_I(\omega - \omega_c)$$

Assume $x_I(t)$ is ideal lowpass signal with bandwidth $\omega_1 < \omega_c$

Amplitude modulation doubles baseband bandwidth of ω_1
 $Y_I(\omega)$ is real-valued if $X_I(\omega)$ is real-valued

- What are applications of amplitude modulation?

1-6

6

Sinusoidal Amplitude Modulation

Amplitude Demodulation by Cosine

- How to recover $x_I(t)$ from $y_I(t) = x_I(t) \cos(\omega_c t)$?

$$F\{y_I(t) \cos(\omega_c t)\} = \frac{1}{2} Y_I(\omega + \omega_c) + \frac{1}{2} Y_I(\omega - \omega_c)$$

Homework 0.2

1-7

7

Sinusoidal Amplitude Modulation

Amplitude Modulation by Sine

- $y_2(t) = x_2(t) \sin(\omega_c t)$ See Slides 1-15, 3-6 and 15-6

$$Y_2(\omega) = \frac{1}{2\pi} X_2(\omega) * (j\pi \delta(\omega + \omega_c) - j\pi \delta(\omega - \omega_c)) = \frac{j}{2} X_2(\omega + \omega_c) - \frac{j}{2} X_2(\omega - \omega_c)$$

Assume $x_2(t)$ is ideal lowpass signal with bandwidth $\omega_2 < \omega_c$

Amplitude modulation doubles baseband bandwidth of ω_2
 $Y_2(\omega)$ is imaginary-valued if $X_2(\omega)$ is real-valued

Homework 0.1

1-8

8

Sinusoidal Amplitude Modulation

Amplitude Demodulation by Sine

- How to recover $x_2(t)$ from $y_2(t) = x_2(t) \sin(\omega_c t)$?

$$F\{y_2(t) \sin(\omega_c t)\} = j \frac{1}{2} Y_2(\omega + \omega_c) - j \frac{1}{2} Y_2(\omega - \omega_c)$$

9

Sinusoidal Amplitude Modulation

Sinusoidal Mod/Demod Demo #1

- Sinusoid of fixed frequency can model musical note**
Middle 'A' note on Western scale is at 440 Hz
Signal model $x_1(t) = \cos(2\pi f_1 t)$ where $f_1 = 440$ Hz
- Modulate middle 'A' note**
 $y_1(t) = x_1(t) \cos(2\pi f_c t)$
 f_c is three octaves above f_1
Creates audio effect
- Demodulate to try to recover original 'A' note**
Listen for frequencies in output signal that are not in original note

10

Sinusoidal Amplitude Modulation

Sinusoidal Mod/Demod Demo #1

11

Sinusoidal Amplitude Modulation

Sinusoidal Mod/Demod Demo #1

```

%% upconvert.m
%%
%% Demonstrates upconversion and downconversion
%%
%% by Prof. Brian L. Evans, bevans@ece.utexas.edu
%% The University of Texas at Austin
%%
%% August 28, 2015
%%
%% This Matlab script will play three sounds:
%% (1) low-frequency signal (baseband signal)
%% (2) modulated signal centered at a higher frequency,
%% (3) the demodulated signal.
%% The demodulated signal consists of the baseband
%% signal plus some higher frequency artifacts.

Fs = 44100;           %% Sampling rate
Ts = 1/Fs;           %% Sampling time
tmax = 10;           %% Signals last tmax seconds
t = 0 : Ts : tmax;   %% Vector of time samples

%% Modulate/upconvert baseband signal
%% to be centered at frequency fc
fc = 8*f1;           %% 3 octaves above f1
carrier = cos(2*pi*f_c*t);
modulated = basebandInput .* carrier;

%% Demodulate/downconvert modulated signal
%% 1. Modulate using the same carrier signal
%% (causes gain to be reduced by 1/2)
%% 2. Apply lowpass filter to extract baseband
%% (and multiply by 2 of compensate for step 1)
%% Use same carrier frequency fc as in modulation
carrier = cos(2*pi*f_c*t);
modulateAgain = modulated .* carrier;

FIRlength = floor(fs/(2*f1));
lowpassCoeffs = ones(1, FIRlength) / FIRlength;
basebandOutput = 2*filter(lowpassCoeffs, 1,
    modulateAgain);

%% Playback signals
sound(basebandInput, fs);
pause(tmax*1);
sound(modulated, fs);
pause(tmax*1);
sound(basebandOutput, fs);
    
```

See handout on designing averaging filters

12

Sinusoidal Amplitude Modulation

Sinusoidal Mod/Demod Demo #2

```

%% upconvertGong.m
%%
%% Demonstrates upconversion and downconversion
%%
%% by Prof. Brian L. Evans, bevans@ece.utexas.edu
%% The University of Texas at Austin
%%
%% Original version: August 28, 2015
%% Current version: February 5, 2016
%%
%% This Matlab script will play three sounds:
%% (1) low-frequency signal (baseband signal)
%% (2) upconverted signal centered at a higher
    frequency.
%% (3) the downconverted signal.
%% The downconverted signal consists of the baseband
%% signal plus some higher frequency artifacts.
%%
%% Note: The gong.wav file must be on your path.
addpath('/Users/bl.evans/Documents/courses/SBRBook/SBR
-MatlabFiles/SBR - MatlabFiles');
    
```

```

%% Read gong signal
%% Dominant frequencies 520 (C5), 630, and 660 Hz
(E5) from
%% plotting spectrum: plotspec(basebandInput, 1/fs)
%% Maximum frequency of interest, f1, is 660 Hz.
basebandInput = audioread('gong.wav');
f1 = 660;

Ts = 1/fs;          %% Sampling time
numSamples = length(basebandInput);
n = 1 : numSamples;
t = n * Ts;
tmax = Ts * numSamples;

%% Upconvert using sinusoidal amplitude modulation
%% to be centered at frequency fc
fc = 4*f1;
carrier = cos(2*pi*fc*t);
modulated = basebandInput .* carrier;
    
```

1-13

13

Sinusoidal Amplitude Modulation

Sinusoidal Mod/Demod Demo #2

```

%% Downconvert by sinusoidal amplitude demodulation
%% 1. Modulate using the same carrier signal
%% (causes gain to be reduced by 1/2).
%% Assume knowledge of carrier frequency fc
%% used in modulation/upconversion.
%% 2. Apply lowpass filter to extract baseband
%% (and multiply by 2 of compensate for step 1).
%% Use averaging lowpass filter of length M
%% which has null bandwidth B = 2 pi / M.
%% In sinusoidal amplitude demodulation, baseband
%% bandwidth in Hz is f1. We can solve for
%% M in B = 2 pi / M = 2 pi f1 / fs, which gives
%% M = fs / f1 (converted to integer).
carrier = cos(2*pi*fc*t);
modulateAgain = modulated .* carrier;

FIRlength = floor(fs/f1);
lowpassCoeffs = ones(1, FIRlength) / FIRlength;
basebandOutput = 2*filter(lowpassCoeffs, 1, modulateAgain);

%% Playback signals
sound(basebandInput, fs);
pause(tmax*1);
sound(modulated, fs);
pause(tmax*1);
sound(basebandOutput, fs);
    
```

See handout on
designing
averaging filters

Baseband
signal (gong)

Modulated
signal

Demodulated
signal

1-14

14

Sinusoidal Amplitude Modulation

How to Use Bandwidth Efficiently?

- **Sinusoidal mod doubles transmission bandwidth**
- **Quadrature Amplitude Modulation (QAM)**
Send two baseband signals in same transmission bandwidth
Used in LTE, Wi-Fi, cable, ...

$$S(\omega) = \frac{1}{2}(X_1(\omega + \omega_c) + X_1(\omega - \omega_c)) - j \frac{1}{2}(X_2(\omega + \omega_c) - X_2(\omega - \omega_c))$$

- **Cosine modulated signal is in theory orthogonal to sine modulated signal at transmitter**
Receiver separates $x_1(t)$ and $x_2(t)$ through demodulation

1-15

15

Sinusoidal Generation

Lab 2: Sinusoidal Generation

- **Sampling**
- **Compute sinusoidal waveform**
Math library function call
Difference equation
Lookup table
- **Output waveform**
Digital-to-analog (D/A) conversion
- **Expected outcomes are to understand**
Signal quality vs. run-time complexity tradeoffs

1-16

16

Sinusoidal Generation

Sampling

- Two-sided discrete-time cosine (or sine) signal with fixed-frequency ω_0 in rad/sample has form $\cos(\omega_0 n)$
- One-sided continuous-time analog-amplitude cosine of frequency f_0 in Hz

$$\cos(2\pi f_0 t)$$
 Sample at rate f_s by substituting $t = n T_s = n / f_s$

$$\cos(2\pi (f_0 / f_s) n)$$
 Discrete-time frequency $\omega_0 = 2\pi f_0 / f_s$ in units of rad/sample
 Example: $f_0 = 1200$ Hz and $f_s = 8000$ Hz, $\omega_0 = 3/10 \pi$

Homework 0.3

Sampling Theorem?

1-17

17

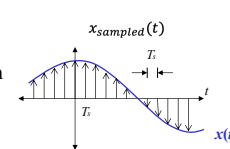
Sinusoidal Generation

Sampling Theorem

- Continuous-time signal $x(t)$ with frequencies no higher than f_{max} can be reconstructed from its samples $x(n T_s)$ if samples taken at rate $f_s > 2 f_{max}$
- Realistic that $x(t)$ has no freq. content above f_{max} ?
- Sampling: Time-Domain Views
 - Discrete-Time Output $x[n] = x(n T_s)$
 - Sampled Analog Output

Models opening/closing of switch as multiplication by impulse train

$$x_{sampled}(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - n T_s)$$



HW 0.3

1-18

18

Sinusoidal Generation

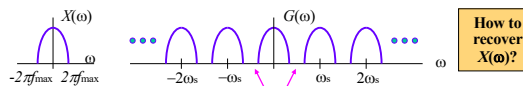
Sampled Analog: Frequency Domain

- Sampling replicates spectrum of continuous-time signal at integer multiples of sampling frequency
- Fourier series of impulse train $\delta_{T_s}(t)$ with period T_s

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n T_s) = \frac{1}{T_s} (1 + 2 \cos(\omega_s t) + 2 \cos(2 \omega_s t) + \dots)$$

$$g(t) = x(t) \delta_{T_s}(t) = \frac{1}{T_s} (x(t) + 2 x(t) \cos(\omega_s t) + 2 x(t) \cos(2 \omega_s t) + \dots)$$

Modulation by $\cos(\omega_s t)$
Modulation by $\cos(2 \omega_s t)$



How to recover $X(\omega)$?

gap if and only if $2\pi f_{max} < 2\pi f_s - 2\pi f_{max} \Leftrightarrow f_s > 2 f_{max}$

HW 0.3

1-19

19

Design Tradeoffs in Generating Sinusoidal Signals

Algorithm #1: C Math Function cos

- Uses double-precision floating-point arithmetic
- No standard in C for internal implementation
- Appropriate for desktop computing
 - On desktop computer, accuracy is a primary concern, so additional computation is often used in C math libraries
 - In embedded scenarios, implementation resources generally at premium, so alternate methods are typically employed
- GNU Scientific Library (GSL) cosine function
 - Function `gsl_sf_cos_e` in file `specfunc/trig.c`
 - Version 1.8 uses 11th order polynomial over 1/8 of period
 - 20 multiply, 30 add, 2 divide and 2 power calculations per output value (additional operations to estimate error)

HW 0.3

1-20

20

Design Tradeoffs in Generating Sinusoidal Signals

Efficient Polynomial Implementation

- **Use 11th-order polynomial**
 Direct form $a_{11} x^{11} + a_{10} x^{10} + a_9 x^9 + \dots + a_0$
 Horner's form minimizes number of multiplications

$$a_{11} x^{11} + a_{10} x^{10} + a_9 x^9 + \dots + a_0 =$$

$$(\dots ((a_{11} x + a_{10}) x + a_9) x \dots) + a_0$$
- **Comparison**

Realization	Multiply Operations	Addition Operations	Memory Usage
Direct form	66	10	13
Horner's form	11	10	12

1-21

21

Design Tradeoffs in Generating Sinusoidal Signals

Algorithm #2: Difference Equation

- **Difference equation with input $x[n]$ and output $y[n]$**

$$y[n] = (2 \cos \omega_0) y[n-1] - y[n-2] + x[n] - (\cos \omega_0) x[n-1]$$
 From inverse z-transform of z-transform of $\cos(\omega_0 n) u[n]$
 Impulse response gives $\cos(\omega_0 n) u[n]$
 Similar difference equation for $\sin(\omega_0 n) u[n]$
- **Implementation complexity**
 Computation: 2 multiplications and 3 additions per cosine value
 Memory Usage: 2 coefficients, 2 previous values of $y[n]$ and 1 previous value of $x[n]$
- **Drawbacks**
 Buildup in error as n increases due to feedback
 Fixed value for ω_0

Initial conditions are all zero

HW 0.4

1-22

22

Design Tradeoffs in Generating Sinusoidal Signals

Differential Equation Drawbacks

- **If implemented with exact precision coefficients and arithmetic, output would have perfect quality**
- **Accuracy loss as n increases due to feedback from**
 Coefficients $\cos(\omega_0)$ and $2 \cos(\omega_0)$ are irrational, except when $\cos(\omega_0)$ is equal to -1, -1/2, 0, 1/2, and 1
 Truncation/rounding of multiplication and addition results
- **Reboot filter after each period of samples by resetting filter to its initial state**
 Reduce loss from truncating/rounding multiplication-addition
 Adapt/update ω_0 if desired by changing $\cos(\omega_0)$ and $2 \cos(\omega_0)$

1-23

23

Design Tradeoffs in Generating Sinusoidal Signals

Algorithm #3: Lookup Table

- **Precompute samples offline and store them in table**
- **Cosine frequency $\omega_0 = 2 \pi f_0 / f_s = 2 \pi N / L$**
 Remove common factors between integers N & L
 $\cos(2 \pi f_0 t)$ has continuous-time period $T_0 = 1 / f_0$
 $\cos(2 \pi (N / L) n)$ has discrete-time period of L samples

Handout on discrete-time periodicity

$N=3$ continuous-time periods Discrete-time period of $L=20$ samples $\omega_0 = 2\pi \frac{1200 \text{ Hz}}{8000 \text{ Hz}} = 2\pi \frac{3}{20}$

24

Design Tradeoffs in Generating Sinusoidal Signals

Design Tradeoffs

- Signal quality vs. run-time complexity in generating $\cos(\omega_0 n)$ where $\omega_0 = 2\pi f_0 / f_s = 2\pi N / L$

Algorithm	MACs/sample	ROM (words)	RAM (words)	Quality in floating pt.	Quality in fixed point
<i>C math library call</i>	30	22	1	Second Best	N/A
<i>Difference equation</i>	2	2	3	Worst	Second Best
<i>Lookup table</i>	0	L	0	Best	Best

MAC Multiplication-accumulation
 RAM Random Access Memory (writeable) ROM Read-Only Memory

1-25

25

Communication Systems

Communication System Structure

- Information sources**
 Voice, music, images, video, and data (message signal $m(t)$)
 Have power concentrated near DC (called *baseband signals*)
- Baseband processing in transmitter**
 Lowpass filter message signal (e.g. AM/FM radio broadcast)
Digital: Add redundancy to message bit stream to aid receiver in detecting and possibly correcting bit errors

TRANSMITTER CHANNEL RECEIVER

1-26

26

Wireless Communication Systems

Transmitter Analog/RF Front End

Mixer: Reduce circuit complexity by replacing multiplication by cosine with sampling device (single transistor)
 BPF selects replica of baseband spectrum at f_c

Antenna Length? Wavelength $\lambda = c / f_c$

1-27

27

Wireless Communication Systems

Communication Channel Modeling

Channel Impairments

- Each path has its own attenuation and delay
- Frequency distortion due to multiple paths
- Additive thermal noise & interference
- More in Lecture 12

TRANSMITTER CHANNEL RECEIVER

1-28

28

Wireless Communication Systems

Receiver Analog/RF Front End

Mixer: Reduce circuit complexity by replacing multiplication by cosine with sampling device (single transistor)
 LPF selects replica of bandpass spectrum at baseband

1-29

29

Software Transceiver Design

Software Transceiver Design

- **Design/implement transceiver**
 Design/evaluate algorithms for each subsystem
 Translate algorithms into real-time software
 Test implementations using signal generators & oscilloscopes

Laboratory	Transceiver Subsystems
1 introduction	Matlab; DSP board & tools
2 sinusoidal generation	sinusoidal mod/demodulation
3(a) finite impulse response filter	pulse shaping, 90° phase shift
3(b) infinite impulse response filter	transmit and receive filters, carrier detection, clock recovery
4 pseudo-noise generation	training sequences
5 pulse amplitude mod/demodulation	training during modem startup
6 quadrature amplitude mod (QAM)	data transmission
7 digital audio effects	not applicable

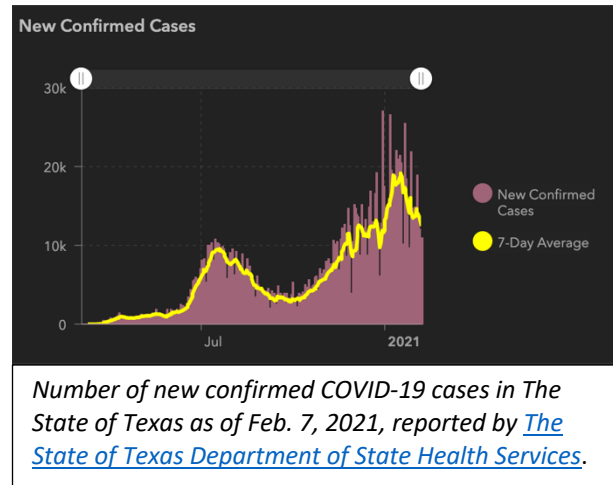
general signal processing building blocks

30

Designing Averaging Filters for Mild Lowpass Filtering and Removal of Harmonics

Prof. Brian L. Evans
The University of Texas at Austin
February 8, 2021

In data analysis, e.g. number of new confirmed COVID-19 cases in Texas each day as shown on the right, we'll often see a moving average to help us discern an underlying trend. The trend in certain segments of the data might show a decrease or increase in what's being measured. A moving average averages out, or smooths out, rapid fluctuations in the data. This is equivalent to keeping low frequency information and attenuating high frequency information in the data.



For a data sequence $x[n]$, a seven-sample moving average can be computed as follows:

$$y[n] = \frac{1}{7} (x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4] + x[n-5] + x[n-6])$$

for $n \geq 0$. The output $y[n]$ is the average of the current data point $x[n]$ and the six previous data points. This difference equation is also known as a seven-coefficient averaging filter.

Analysis. We analyze the frequency response for a linear time-invariant (LTI) system using the discrete-time Fourier transform. A necessary condition for a system to be LTI is that the system must initially be “at rest”. In this case, it means that the initial conditions must be zero. We can identify the initial conditions in several ways. One way is to start computing output values:

$$y[0] = \frac{1}{7} (x[0] + x[-1] + x[-2] + x[-3] + x[-4] + x[-5] + x[-6])$$

The six initial conditions correspond to the six memory locations storing the previous six input values. When $n = 0$, there are no previous input values. We initialize the six memory locations $\{x[n-1], x[n-2], x[n-3], x[n-4], x[n-5], x[n-6]\}$ to zero to make the system be initially at rest. Initializing data variables and arrays to zero is a common programming practice.

There are many ways to find the discrete-time Fourier transform of the seven-coefficient LTI averaging filter. One way is to find the impulse response and then find its discrete-time Fourier transform. The impulse response is the response of a system to a discrete-time impulse:

$$h[n] = \frac{1}{7} (\delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3] + \delta[n-4] + \delta[n-5] + \delta[n-6])$$

The impulse response $h[n]$ is a rectangular pulse that is seven samples long with amplitude $1/7$ as shown on the next page. We can find the Fourier transform from the z-transform:

$$H(z) = \frac{1}{7} (1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6}) \quad \text{for } z \neq 0$$

Since the region of convergence $z \neq 0$ includes the unit circle, we can substitute $z = e^{j\omega}$:

$$H_{freq}(\omega) = H(e^{j\omega}) = \frac{1}{7} (1 + e^{-j\omega} + e^{-2j\omega} + e^{-3j\omega} + e^{-4j\omega} + e^{-5j\omega} + e^{-6j\omega})$$

We can factor out the middle term (due to the phase shift corresponding to the mid-point of the impulse response at sample index $(L-1)/2$) to reveal that the phase is linear in frequency:

$$H_{freq}(\omega) = \frac{1}{7} e^{-j3\omega} (e^{j3\omega} + e^{j2\omega} + e^{j\omega} + 1 + e^{-j\omega} + e^{-j2\omega} + e^{-j3\omega})$$

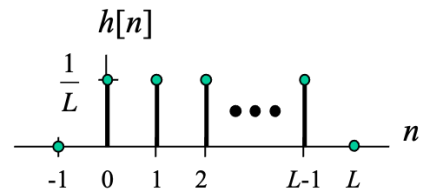
$$H_{freq}(\omega) = \frac{1}{7} (1 + 2 \cos(\omega) + 2 \cos(2\omega) + 2 \cos(3\omega)) e^{-j3\omega} = A(\omega) e^{-j3\omega}$$

Amplitude $A(\omega)$ is a sinc with period 2π and the phase is -3ω which is a line with slope of -3 . Please note that $A(\omega)$ has positive, zero, and negative values, and is not the magnitude.

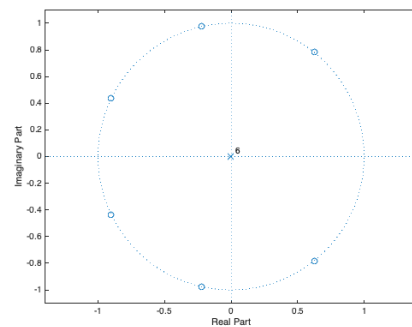
The impulse response (on right) is a rectangular pulse of 7 samples with amplitude $1/7$. The pole-zero plot for $H(z)$ shows zeros at angles (frequencies) that are multiples of $2\pi/7$ rad/sample except at zero. Magnitude response is lowpass with a null bandwidth of $2\pi/7$ rad/sample and frequencies at multiples of $2\pi/7$ rad/sample are zeroed out except at 0 rad/sample. Phase response is linear except for the discontinuities at frequencies that are zeroed out (not passed) by the filter at multiples of $2\pi/7$ rad/sample except at 0 rad/sample.

Design a lowpass discrete-time LTI filter with a null bandwidth of 60 Hz that eliminates multiples of the powerline frequency 60 Hz up to half of the sampling rate. The sampling rate is 420 Hz.

Answer: The null bandwidth in discrete-time frequency is $\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{60 \text{ Hz}}{420 \text{ Hz}} = \frac{2}{7}\pi$ rad/sample. The null bandwidth for an LTI averaging filter with L coefficients is $2\pi/L$; hence, $L = 7$. Averaging filter with 7 coefficients will eliminate 60 Hz, 120 Hz, and 180 Hz for a sampling rate of 420 Hz.

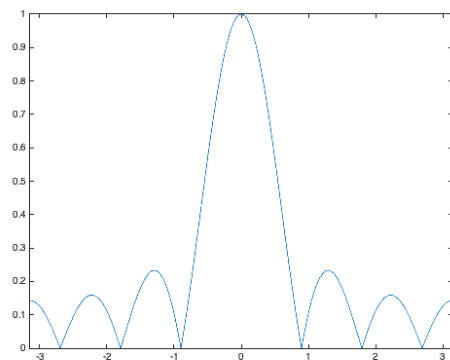


Impulse response for LTI averaging filter with L coefficients ($L = 7$ here)

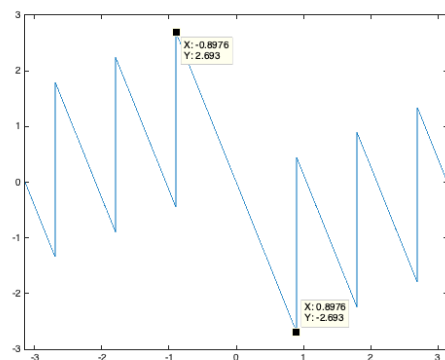


Poles & zeros of $H(z)$ for LTI averaging filter with 7 coefficients in Matlab:

`h = (1/7)*ones(1, 7); zplane(h)`



Magnitude response for LTI averaging filter with 7 coefficients. Magnitude of a periodic sinc. Zeros at multiples of $2\pi/7$ except 0.



Phase response for LTI averaging filter with 7 coefficients in Matlab. Using the highlighted points, the slope is estimated to be -3.0002 .

```
w = -pi : (2*pi/100000) : pi;
H = (1/7)*(1 + exp(-j*w) + exp(-j*2*w) + exp(-j*3*w) + exp(-j*4*w) + exp(-j*5*w) + exp(-j*6*w));
figure; plot(w, abs(H)); xlim( [-pi, pi] ); %% plot the magnitude response
figure; plot(w, phase(H)); xlim( [-pi, pi] ); %% plot the phase response
```

Discrete-Time Periodicity

Prof. B. L. Evans

A discrete-time signal $x[n]$ is periodic if

$$x[n + N_0] = x[n] \quad \text{for all } n. \quad N_0 \text{ is positive.}$$

The smallest value of N_0 is the fundamental period. For a two-sided cosine signal,

$$x[n] = \cos(\omega_0 n) \quad \text{where } \omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{N}{L}$$

where N and L are relatively prime integers,

and f_0 is the continuous-time frequency and

f_s is the sampling rate.

$$x[n + N_0] = \cos\left(2\pi \frac{N}{L} (n + N_0)\right)$$

$$= \cos\left(2\pi \frac{N}{L} n + 2\pi \frac{N}{L} N_0\right)$$

$$= \cos\left(2\pi \frac{N}{L} n\right) = x[n]$$

if $2\pi \frac{N}{L} N_0$ is an integer multiple of 2π , i.e.

if $\frac{N}{L} N_0$ is an integer. The smallest value

of N_0 is $N_0 = L$. Fundamental period is L .

There are N continuous-time periods in the fundamental discrete-time period.

Spring 2014

$$\int_{-\infty}^{\infty} x(at) dt = ?$$

For $a > 0$, let $u = at \Rightarrow t = \frac{1}{a}u$
 $dt = \frac{1}{a} du$

limits $t \rightarrow \infty \Rightarrow u \rightarrow \infty$
 $t \rightarrow -\infty \Rightarrow u \rightarrow -\infty$

$$\int_{-\infty}^{\infty} x(u) \frac{1}{a} du = \frac{1}{a} \int_{-\infty}^{\infty} x(u) du$$

For $a < 0$, let $u = at \Rightarrow t = \frac{1}{a}u$
 $dt = \frac{1}{a} du$

limits $t \rightarrow \infty \Rightarrow u \rightarrow -\infty$

$t \rightarrow -\infty \Rightarrow u \rightarrow \infty$

$$\int_{\infty}^{-\infty} x(u) \frac{1}{a} du = -\frac{1}{a} \int_{-\infty}^{\infty} x(u) du$$

Hence, $\int_{-\infty}^{\infty} x(at) dt = \frac{1}{|a|} \int_{-\infty}^{\infty} x(u) du$
for $a \neq 0$

$$u(t) \Big|_{t=nT_s} = u(nT_s) = \frac{1}{T_s} u[n]$$

INTRODUCTION TO DIGITAL SIGNAL PROCESSORS

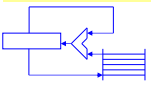
Prof. Brian L. Evans

Contributions by
Dr. Niranjana Damera-Venkata and
Mr. Magesh Valliappan

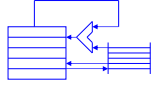
Embedded Signal Processing Laboratory
The University of Texas at Austin

<http://signal.ece.utexas.edu/>

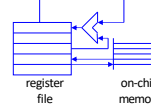
Accumulator architecture



Memory-register architecture



Load-store architecture



register file on-chip memory

1

Outline

- Embedded processors and systems
- Signal processing applications
- TI TMS320C6000 digital signal processor
- Conventional digital signal processors
- Pipelining
- RISC vs. DSP processor architectures
- Conclusion

2-2

2

Embedded Processors and Systems

- Embedded system works
 - ▶ On application-specific tasks
 - ▶ "Behind the scenes" (little/no direct user interaction)
- Units of consumer products shipped worldwide 2017

1472M smart phones	↓ 0.1%	72M DVD/Blu-ray players	↓ 9%
259M PCs/laptops	↓ 0.3%	55M digital media streamers	↑ 10%
164M tablets	↓ 6%	47M game consoles	↑ 24%
138M smart TVs	↑ 20%	25M digital still cameras	↑ 3%
79M cars/ltr trucks	↑ 2%		

1B/year smart phones since 2014 and connected TVs since 2018
- How many embedded processors are in each?
- How much should an embedded processor cost?
 - ▶ 2015: iPhone6 \$676 (16GB) \$942 (128GB) w/o contract

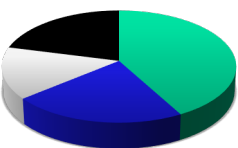
2-3

3

Smart Phone Application Processors

- Standalone app processors (Apple, Samsung)
- Integrated baseband-app processors (Qualcomm)

2017 Smartphone AP Market



Source: Statista and Strategy Analytics
42% Qualcomm, 22% Apple, 15% MediaTek, 21% Others.
Samsung LSI and Spreadtrum had next two largest shares.
Others include Broadcom (Avago), HiSilicon (Huawei), Intel.
40% have eight cores and 14% in 10nm CMOS process

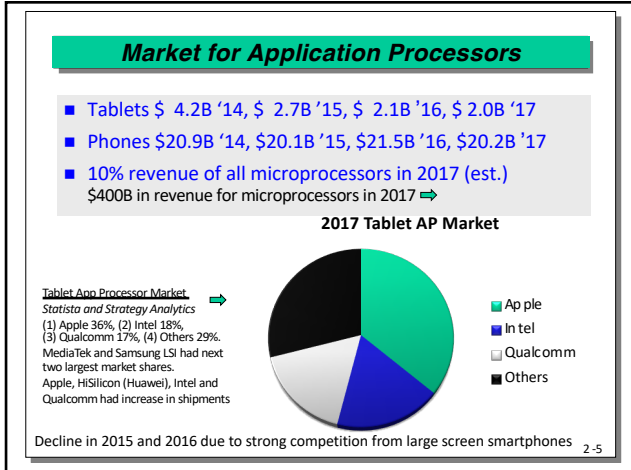
iPhone5 (10+ cores)

- Touchscreen: Broadcom (probably 2 ARM cores)
- Apps: Samsung (2 ARM + 3 GPU cores)
- Audio: Cirrus Logic (1 DSP core + 1 codec)
- Wi-Fi: Broadcom
- Baseband: Qualcomm
- Inertial sensors: STMicroelectronics

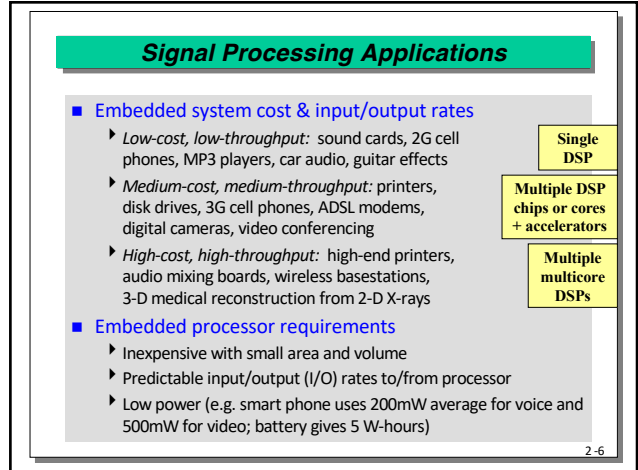
"iPhone 5 Tear Down" →

2-4

4



5



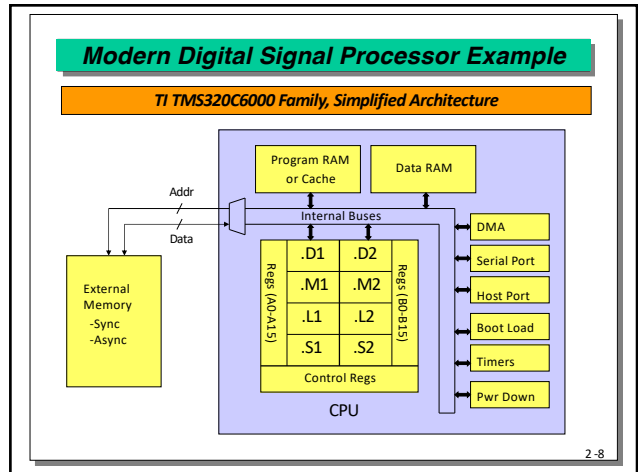
6

Type of Digital Signal Processor?

	Fixed-Point	Floating-Point
Per unit cost	\$2 and up	\$2 and up
Prototyping time	Long	Short
Power consumption	10 mw - 1 W	1-3 W
Battery-powered products	Cell phones Digital cameras	Very few
Other products	DSL modems Cellular basestations	Pro & car audio Medical imaging
Sales volume	High	Low
Prototyping	Convert floating- to fixed-point; use non-standard C extensions; redesign algorithms	Reuse desktop simulations; feasibility check before investing in fixed-point design

2-7

7



8

Modern DSP: TI TMS320C6000 Architecture

- Very long instruction word (VLIW) of 256 bits
 - ▶ Eight 32-bit functional units with one cycle throughput
 - ▶ One instruction cycle per clock cycle
- Data word size and register size are 32 bits
 - ▶ 16 (32 on C6400) registers in each of two data paths
 - ▶ 40 bits can be stored in adjacent even/odd registers
- Two parallel data paths
 - ▶ Data unit - 32-bit address calculations (modulo, linear)
 - ▶ Multiplier unit - 16 bit × 16 bit with 32-bit result
 - ▶ Logical unit - 40-bit (saturation) arithmetic/comparisons
 - ▶ Shifter unit - 32-bit integer ALU and 40-bit shifter

2-9

9

Modern DSP: TI TMS320C6000 Architecture

- Families: All support same C6000 instruction set
 - C6200 fixed-pt. 150- 300 MHz printers, DSL (*obsolete*)
 - C6400 fixed pt. 500-1200 MHz video, DSL
 - C6600 floating 1000-1250 MHz basestations (8 cores)
 - C6700 floating 150-1,000 MHz medical imaging, audio
- TMS320C6748 OMAP-L138 Experimenter Kit
 - 375-MHz CPU (750 million MACs/s, 3000 RISC MIPS)
 - On-chip: 8 kword program, 8 kword data, 64 kword L2
 - On-board memory: 32 Mword SDRAM, 2 Mword ROM

2-10

10

Modern DSP: TMS320C6000 Instruction Set

C6000 Instruction Set by Functional Unit

<u>.S Unit</u>	<u>.L Unit</u>	<u>.D Unit</u>
ADD NEG	ABS NOT	ADD ST
ADDK NOT	ADD OR	ADDA SUB
ADD2 OR	AND SADD	LD SUBA
AND SET	CMPEQ SAT	MV ZERO
B SHL	CMPGT SSUB	NEG
CLR SHR	CMPLT SUB	
EXT SSSL	LMBD SUBC	<u>.M Unit</u>
MV SUB	MV XOR	MPY SMPY
MVC SUB2	NEG ZERO	MPYH SMPYH
MVK XOR	NORM	
MVKH ZERO		<u>Other</u>
		NOP IDLE

Six of the eight functional units can perform integer add, subtract, and move operations

2-11

11

Modern DSP: TMS320C6000 Instruction Set

<u>Arithmetic</u>	<u>Logical</u>	<u>Data Management</u>
ABS	AND	LD
ADD	CMPEQ	MV
ADDA	CMPGT	MVC
ADDK	CMPLT	MVK
ADD2	NOT	MVKH
MPY	OR	ST
MPYH	SHL	
NEG	SHR	<u>Program Control</u>
SMPY	SSHL	B
SMPYH	XOR	IDLE
SADD		NOP
SAT	<u>Bit Management</u>	
SSUB	CLR	
SUB	EXT	
SUBA	LMBD	
SUBC	NORM	
SUB2	SET	
ZERO		

C6000 Instruction Set by Category
(un)signed multiplication
saturation/packed arithmetic

2-12

12

C5000 vs. C6000 Addressing Modes

	TI C5000	TI C6000
Immediate Operand part of instruction	ADD #0Fh	movk .D1 15, A1
Register Operand specified in a register	(implied)	add .L1 A1, A6, A6
Direct Address of operand is part of the instruction (added to imply memory page)	ADD 010h	not supported
Indirect Address of operand is stored in a register	ADD *	ldw .D1 *A5++[8], A1

2-13

13

C6700 Extensions

C6700 Floating Point Extensions by Unit

<u>.S Unit</u>		<u>.L Unit</u>	
ABSDP	CMPLTSP	ADDDP	INTSP
ABSSP	RCPDP	ADDSP	SPINT
CMPEQDP	RCPDP	DPINT	SPTRUNC
CMPEQSP	RSARDP	DPSP	SUBDP
CMPGTDP	RSQRSP	DPTRUNC	SUBSP
CMPGTSP	SPDP	INTDP	
CMPLTDP			
<u>.D Unit</u>		<u>.M Unit</u>	
ADDAD	LDDW	MPYDP	MPYID
		MPYI	MPYSP

Four functional units perform IEEE single-precision (SP) and double-precision (DP) floating-point add, subtract, and move.
Operations beginning with R are reciprocal (i.e. 1/x) calculations.

2-14

14

Selected TMS320C6700 Floating-Point DSPs

Part	MHz	MIPS	Data (kbits)	Program (kbits)	Level2 (kbits)	Price	Applications
AVVQ	150	1200	512	512	0	\$ 88	C6701 EVM board
AVVQ	167	1336	512	512	0	\$141	
AVVQ	150	1200	32	32	512	n/a	C6711 DSK board
AVVQ	250	2000				\$ 18	
AVVQR	150	1200	32	32	512	\$ 14	
AVVQS	167	1336	32	32	1000	\$ 19	
AVVRS	RRU	QXPP	SR	SR	QPPP	D RU	AVVQS A E C I N I O I
AVVRR	300	2400	32	32	1000	\$ 33	
AVVRV	250	2000	1000	3072	256	\$ 10	Professional audio
AVVRW	266	2128	2000	3072	256	\$ 15	Professional audio
AVVWX	300	2400	2000	3072	256	\$ 22	C6727 EVM board
AVVWX	350	2800	2000	3072	256	\$ 30	Professional audio
AVVWX	300	2400	256	256	2048	\$ 18	Pro-audio and video
AVVWX	SWU	SPPP	RUV	RUV	RPTX	D RP	AVVWX G C F A E D I N I O I

DSK: DSP Starter Kit. EVM: Evaluation Module.
Unit price for 100 units. Prices effective February 1, 2009.
For more information: <http://www.ti.com>

2-15

15

Selected TMS320C6000 Fixed-Point DSPs

Part	MHz	MIPS	Data (kbits)	Program (kbits)	Level 2 (kbits)	Price	Applications
AVRPR	250	2000	1000	2000		\$ 66	
AVRPS	300	2400				\$ 79	
AVRPT	250	2000	4000	3000		\$ 84	modems banks ADSL1
AVRPT	300	2400				\$ 84	modems
AVRPT	200	1600	512	512		\$ 11	
AVTQV	720	5760	128	128	8000	\$114	ADSL2 modems
AVTQV	1000	8000	128	128	8000	\$227	3G basestations
AVTQX	500	4000	128	128	5000	\$ 49	
AVTQX	600	4800	128	128	5000	\$ 49	
AVVTQ	500	4000	128	128	1000	\$ 28	Video conferencing
AVVTQ	600	4800	128	128	1000	\$ 31	
AVVTR	500	4000	128	128	2000	\$ 37	Video conferencing
AVVTR	720	5760	128	128	2000	\$ 57	
AVVTX	900	7200	512	512	4000	\$ 64	Video conferencing

C6416 has Viterbi and Turbo decoder coprocessors.
Unit price is for 100 units. Prices effective February 1, 2009.
For more information: <http://www.ti.com>

2-16

16

C6000 Reference Information for Lab Work

- Code Composer Studio v5
http://processors.wiki.ti.com/index.php/CCSv4
- C6000 Optimizing C Compiler 7.4
http://focus.ti.com/lit/ug/spru187u/spru187u.pdf
- C6000 Programmer's Guide
http://www.ti.com/lit/ug/spru198k/spru198k.pdf
- C674x DSP CPU & Instruction Set Ref. Guide
http://focus.ti.com/lit/ug/sprufe8b/sprufe8b.pdf
- C6748 Board
Logic PD's ZOOM OMAP-L138 Experimenter Kit
http://www.logicpd.com/products/development-kits/zoom-omap-l138-experimenter-kit

TEXAS INSTRUMENTS

Ti software development environment

Download them for reference

2-17

17

Conventional Digital Signal Processors

- Low cost: as low as \$2/processor *in volume*
- Deterministic interrupt service routine latency guarantees predictable input/output rates
 - On-chip direct memory access (DMA) controllers
 - Processes streaming input/output separately from CPU
 - Sends interrupt to CPU when frame read/written
 - Ping-pong buffering
 - CPU reads/writes buffer 1 as DMA reads/writes buffer 2
 - After DMA finishes buffer 2, roles of buffers switch
- Low power consumption: 10-100 mW
 - TI TMS320C54: 0.48 mW/MHz → 76.8 mW at 160 MHz
 - TI TMS320C5504: 0.15 mW/MHz → 45.0 mW at 300 MHz
- Based on conventional (pre-1996) architecture

2-18

18

Conventional Digital Signal Processors

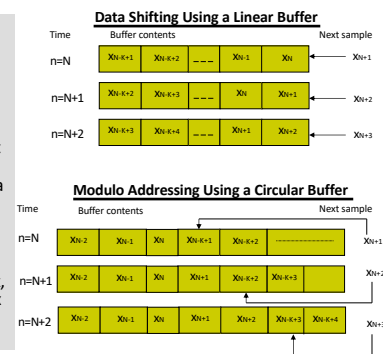
- Multiply-accumulate in one instruction cycle
- Harvard architecture for fast on-chip I/O
 - Separate data memory/bus and program memory/bus
 - 1 read from program memory per instruction cycle
 - 2 reads/writes from/to data memory per inst. cycle
- Instructions to keep pipeline (3-6 stages) full
 - Zero-overhead looping (one pipeline flush to set up)
 - Delayed branches
- Special addressing modes in hardware
 - Bit-reversed addressing (fast Fourier transforms)
 - Modulo addressing for circular buffers (e.g. filters)

2-19

19

Conventional Digital Signal Processors

- Buffers
 - Used in processing streaming data
- Linear buffer
 - Sort by time index
 - Update: discard oldest data, copy old data left, insert new data
- Circular buffer
 - Oldest data index
 - Update: insert new data at oldest index, update oldest index



2-20

20

Conventional Digital Signal Processors

	Fixed-Point	Floating-Point
Á NÓÓÈ NÍÓ	\$2 - \$79	\$2 - \$381
À ÒÌ KÌÁÌ ÓÒÒ	Accumulator	load-store or memory-register
È Í J LÓÒ ÓÒ	2-4 data 8 address	8 or 16 data 8 or 16 address
À Ì Á Ę NÒÌ Ó	16 or 24 bit integer and fixed-point	32 bit integer and fixed/floating-point
É NÁ KÛN	2-64 kwords data	8-64 kwords data
Ð Ì L NÓÓ	2-64 kwords program	8-64 kwords program
À Ì Í ÓÒ ÓÒ	16-128 kw data	16 Mw - 4Gw data
È NÌ Í J	16-64 kw program	16 Mw - 4 Gw program
Á Ñ L NÌ ÓÒ	C, C++ compilers; poor code generation	C, C++ compilers; better code generation
À Ò Ì L NÌ Ó	TI TMS320C5000; Freescale DSP56000	TI TMS320C30; Analog Devices SHARC

2-21

21

Conventional Digital Signal Processors

- Different on-chip configurations in each family
 - Size and map of data and program memory
 - A/D, input/output buffers, interfaces, timers, and D/A
- Drawbacks to conventional digital signal processors
 - No byte addressing (needed for images and video)
 - Limited on-chip memory
 - Limited addressable memory on fixed-point DSPs (exceptions include Freescale 56300 and TI C5409)
 - Non-standard C extensions for fixed-point data type

2-22

22

Pipelining

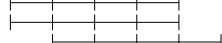
Sequential (Freescale 56000)



Pipelined (Most conventional DSPs)



Superscalar (Pentium)



Superpipelined (TI C6000)



Pipelining

- Process instruction stream in stages (as stages of assembly in manufacturing line)
- Increase throughput

Managing Pipelines

- Compiler or programmer
- Pipeline interlocking

2-23

23

Pipelining: Operation

Time-stationary pipeline model

Programmer controls each cycle

Example: Freescale DSP56001 (has X/Y data memories/registers)

MAC X0, Y0, A X: (R0) +, X0 Y: (R4) -, Y0

Data-stationary pipeline model

Programmer specifies data operations

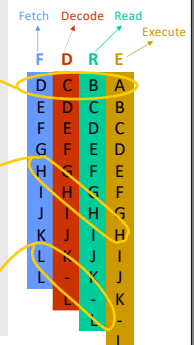
Example: TI TMS320C30

MPYF *++AR0 (1), *++AR1 (IR0), R0

Interlocked pipeline

"Protection" from pipeline effects

May not be reported by simulators:
inner loops may take extra cycles



MAC means multiplication-accumulation.

2-24

24

Pipelining: Control and Data Hazards

- A control hazard occurs when a branch instruction is decoded
 - ▶ Processor "flushes" the pipeline, or
 - ▶ Delayed branch (expose pipeline)
- A data hazard occurs because an operand cannot be read yet
 - ▶ Intended by programmer, or
 - ▶ Interlock hardware inserts "bubble"
 - ▶ TI TMS320C5000 (20 CPU & 16 I/O registers, one accumulator, and one address pointer ARP implied by *)

```

LAR AR2, ADDR ; load address reg.
LACC *-      ; load accumulator w/
              ; contents of AR2
          
```

LAR: 2 cycles to update AR2 & ARP; need NOP after it

2-25

25

Pipelining: Avoiding Control Hazards

High throughput performance of DSPs is helped by on-chip dedicated logic for looping (downcounters/looping registers)

```

; repeat TBLR inst. COUNT-1 times
RPT COUNT
TBLR **
          
```

- A repeat instruction repeats one instruction or block of instructions after repeat
- The pipeline is filled with repeated instruction (or block of instructions)
- Cost: one pipeline flush only

2-26

26

Pipelining: TI TMS320C6000 DSP

- C6000 has deep pipeline
 - ▶ 7-11 stages in C6200: fetch 4, decode 2, execute 1-5
 - ▶ 7-16 stages in C6700: fetch 4, decode 2, execute 1-10
 - ▶ Compiler and assembler must prevent pipeline hazards
- Only branch instruction: delayed unconditional
 - ▶ Processor executes next 5 instructions after branch
 - ▶ Conditional branch via conditional execution: [A2] B loop
 - ▶ Branch instruction in pipeline disables interrupts
 - ▶ Undefined if both shifters take branch on same cycle
 - ▶ Avoid branches by conditionally executing instructions

Pentium IV pipeline has more than 20 stages

Contributions by Sundararajan Sriram (TI) 2-27

27

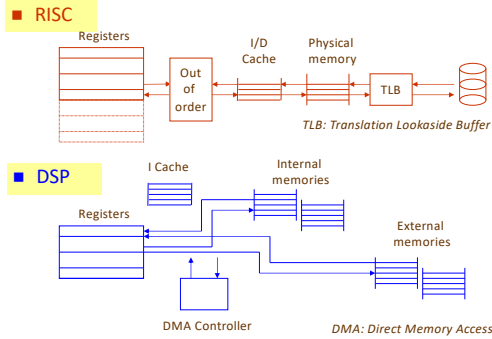
RISC vs. DSP: Instruction Encoding

- RISC: Superscalar, out-of-order execution
- DSP: Horizontal microcode, in-order execution

2-28

28

RISC vs. DSP: Memory Hierarchy



2-29

29

Concluding Remarks

- Conventional digital signal processors
 - High performance vs. power consumption/cost/volume
 - Excel at one-dimensional processing
 - Per cycle: 1 16 × 16 MAC & 4 16-bit RISC instructions
- TMS320C6000 VLIW DSP family
 - High performance vs. cost/volume
 - Excel at multidimensional signal processing
 - Per cycle: 2 16×16 MACs & 4 32-bit RISC instructions
- Get the best of both worlds
 - Assembly language for computational kernels (possibly wrapped in C callable functions)
 - C for main program (control code, interrupt definition)

2-30

30

References

- Unit shipments worldwide (2017)
 - Blu-ray & DVD players:** <https://www.prnewswire.com/news-releases/global-blu-ray-media-and-players-market-2018-2023---a-dying-market-300602693.html>
 - Cars & light trucks:** https://www.scotiabank.com/content/dam/scotiabank/sub-brands/scotiabank-economics/english/documents/GAR_2018-03-29.pdf
 - Digital media streamers:** <https://www.broadbandtvnews.com/2018/08/14/research-globally-more-than-one-billion-connected-tv-devices/>
 - Digital still cameras:** <https://www.businessinsider.com/digital-camera-shipments-15-years-decline-and-rise-2018-2>
 - iPhone5:** <http://www.ifixit.com/TearDown/iPhone-5-TearDown/10525/>
 - PCs/laptops:** <https://www.statista.com/statistics/272595/global-shipments-forecast-for-tablets-laptops-and-desktop-pcs/>
 - Smart phones:** <https://www.idc.com/getdoc.jsp?containerId=prUS43548018>
 - Smart TVs:** <https://technology.ihc.com/604804/smart-tv-share-jumps-to-70-percent-of-tv-shipments-in-2018-from-less-than-50-percent-in-2015-ihc-market-says>
 - Tablets:** <https://venturebeat.com/2018/02/05/idc-tablet-shipments-decline-for-13th-straight-quarter-amazon-overtakes-samsung-for-second-place/>
 - Video game consoles:** <https://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/>

2-31

31

References

- Unit shipments worldwide (2016)
 - Blu-ray & DVD players:** <https://www.futureresource-consulting.com/reports/report/r/futuresource-worldwide-home-video-market-report/i/412362>
 - Cars:** http://www.gbm.scotiabank.com/scpt/gbm/scotiaeconomics63/GAR_2017-02-07.pdf
 - Digital media streamers:** <https://www.strategyanalytics.com/access-services/devices/connected-home/consumer-electronics/reports/report-detail/global-connected-tv-device-vendor-share-q3-2016>
 - Digital still cameras:** <http://promuser.com/markets/2017/global-digital-camera-market-report-january-2017>
 - iPhone5:** <http://www.ifixit.com/TearDown/iPhone-5-TearDown/10525/>
 - PCs/laptops:** <https://www.gartner.com/newsroom/id/3568420>
 - Smart phones:** <http://www.gartner.com/newsroom/id/3609817>
 - Tablets:** <https://www.idc.com/getdoc.jsp?containerId=prUS42272117>
 - Video game consoles:** <https://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/>
- Embedded processor resources
 - Embedded Microproc. Benchmark Consortium: <http://www.eembc.org>
 - Embedded processing comparison: <http://www.embeddedinsights.com/directory.php>

2-32

32

TRENDS IN MULTI-CORE DSP PLATFORMS

Lina J. Karam* Ismail AlKamal* Alan Gatherer** Gene A. Frantz** David V. Anderson† Brian L. Evans‡

*Electrical Engineering Dept., Arizona State University, Tempe, AZ 85287-5706, {ismail.alkamal, karam}@asu.edu

**Texas Instruments, 12500 TI Boulevard MS 8635, Dallas, TX 75243, {genf, gatherer}@ti.com

†School of Electrical and Computer Engineering, Georgia Tech, Atlanta, GA 30332-0250, dva@ece.gatech.edu

‡Electrical & Computer Engineering Dept., The University of Texas at Austin, Austin, TX 78712, bevans@ece.utexas.edu

1. INTRODUCTION

MULTI-CORE Digital Signal Processors (DSPs) have gained significant importance in recent years due to the emergence of data-intensive applications, such as video and high-speed Internet browsing on mobile devices, which demand increased computational performance but lower cost and power consumption. Multi-core platforms allow manufacturers to produce smaller boards while simplifying board layout and routing, lowering power consumption and cost, and maintaining programmability.

Embedded processing has been dealing with multi-core *on a board, or in a system*, for over a decade. Until recently, size limitations have kept the number of cores *per chip* to one, two, or four but, more recently, the shrink in feature size from new semiconductor processes has allowed single-chip DSPs to become multi-core with reasonable on-chip memory and I/O, while still keeping the die within the size range required for good yield. Power and yield constraints, as well as the need for large on-chip memory have further driven these multi-core DSPs to become systems-on-chip (SoCs). Beyond the power reduction, SoCs also lead to overall cost reduction because they simplify board design by minimizing the number of components required.

The move to multi-core systems in the embedded space is as much about integration of components to reduce cost and power as it is about the development of very high performance systems. While power limitations and the need for low-power devices may be obvious in mobile and hand-held devices, there are stringent constraints for non-battery powered systems as well. Cooling in such systems is generally restricted to forced air only, and there is a strong desire to avoid the mechanical liability of a fan if possible. This puts multi-core devices under a serious hotspot constraint. Although a fan cooled rack of boards may be able to dissipate hundreds of Watts (ATCA carrier card can dissipate up to 200W), the density of parts on the board will start to suffer when any individual chip power rises above roughly 10W. Hence, the cheapest solution at the board level is to restrict the power dissipation to around 10W per chip and then pack these chips densely on the board.

The introduction of multi-core DSP architectures presents several challenges in hardware architectures, memory organization and management, operating systems, platform software, compiler designs, and tooling for code development and debug. This article presents an overview of existing multi-core DSP architectures as well as

programming models, software tools, emerging applications, challenges and future trends of multi-core DSPs.

2. HISTORICAL PERSPECTIVES: FROM SINGLE-CORE TO MULTI-CORE

The concept of a Digital Signal Processor came about in the middle of the 1970s. Its roots were nurtured in the soil of a growing number of university research centers creating a body of theory on how to solve real world problems using a digital computer. This research was academic in nature and was not considered practical as it required the use of state-of-the-art computers and was not possible to do in real time. It was a few years later that a Toy by the name of Speak N Spell™ was created using a single integrated circuit to synthesize speech. This device made two bold statements:

-Digital Signal Processing can be done in real time.

-Digital Signal Processors can be cost effective.

This began the era of the Digital Signal Processor. So, what made a Digital Signal Processor device different from other microprocessors? Simply put, *it was the DSP's attention to doing complex math while guaranteeing real-time processing*. Architectural details such as dual/multiple data buses, logic to prevent over/underflow, single cycle complex instructions, hardware multiplier, little or no capability to interrupt, and special instructions to handle signal processing constructs, gave the DSP its ability to do the required complex math in real time.

"If I can't do it with one DSP, why not use two of them?" That is the answer obtained from many customers after the introduction of DSPs with enough performance to change the designer's mind set from "how do I squeeze my algorithm into this device" to "guess what, when I divide the performance that I need to do this task by the performance of a DSP, the number is small." The first encounter with this was a year or so after TI introduced the TMS320C30 – the first floating-point DSP. It had significantly more performance than its fixed-point predecessors. TI took on the task of seeing what customers were doing with this new DSP that they weren't doing with previous ones. The significant finding was that none of the customers were using only one device in their system. They were using multiple DSPs working together to create their solutions.

As the performance of the DSPs increased, more sophisticated applications began to be handled in real time. So, it went from voice to audio to image to video processing. Fig. 1 depicts this evolution. The four lines in

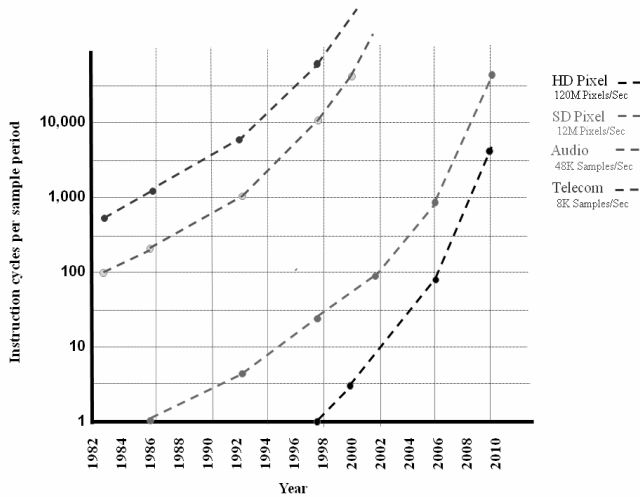


Fig. 1. Four examples of the increase of instruction cycles per sample period. It appears that the DSP becomes useful when it can perform a minimum of 100 instructions per sample period. Note that for a video system the pixel is used in place of a sample.

Fig. 1 represent the performance increases of Digital Signal Processors in terms of instruction cycles per sample period. For example, the sample rate for voice is 8 kHz. Initial DSPs allowed for about 625 instructions per sample period, barely enough for transcoding. As higher performance devices began to be available, more instruction cycles became available each sample period to do more sophisticated tasks. In the case of voice, algorithms such as noise cancellation, echo cancellation and voice band modems were able to be added as a result of the increased performance made available. Fig. 2 depicts how this increase in performance was more the result of multi-processing rather than higher performance single processing elements. Because Digital Signal Processing algorithms are Multiply-Accumulate (MAC) intensive, this chart shows how, by adding multipliers to the architecture, the performance followed an aggressive growth rate. Adding multiplier units is the simplest form of doing multiprocessing in a DSP device.

For TI, the obvious next step was to architect the next generation DSPs with the communications ports necessary to matrix multiple DSPs together in the same system. That device was created and introduced as the TMS320C40. And, as one might suspect, a follow up (fixed-point) device was created with multiple DSPs on one device under the management of a RISC processor, the TMS320C80.

The proliferation of computationally demanding applications drove the need to integrate multiple processing elements on the same piece of silicon. This led to a whole new world of architectural options: homogeneous multi-processing, heterogeneous multi-processing, processors versus accelerators, programmable versus fixed function, a mix of general purpose processors and DSPs, or system in a

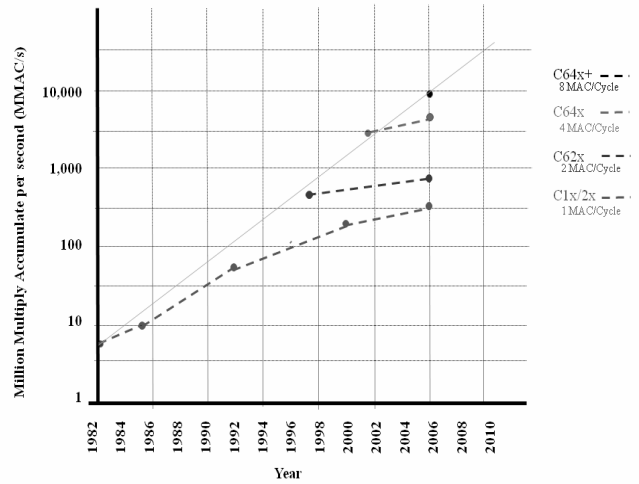


Fig. 2. Four generations of DSPs show how multi-processing has more effect on performance than clock rate. The dotted lines correspond to the increase in performance due to clock increases within an architecture. The solid line shows the increase due to both the clock increase and the parallel processing.

package versus System on Chip integration. And then there is Amdahl's Law that must be introduced to the mix [1-2]. In addition, one needs to consider how the architecture differs for high performance applications versus long battery life portable applications.

3. ARCHITECTURES OF MULTI-CORE DSPs

In 2008, 68% of all shipped DSP processors were used in the wireless sector, especially in mobile handsets and base stations; so, naturally, development in wireless infrastructure and applications is the current driving force behind the evolution of DSP processors and their architectures [3]. The emergence of new applications such as mobile TV and high speed Internet browsing on mobile devices greatly increased the demand for more processing power while lowering cost and power consumption. Therefore, multi-core DSP architectures were established as a viable solution for high performance applications in packet telephony, 3G wireless infrastructure and WiMAX [4]. This shift to multi-core shows significant improvements in performance, power consumption and space requirements while lowering costs and clocking frequencies. Fig. 3 illustrates a typical multi-core DSP platform.

Current state-of-the-art multi-core DSP platforms can be defined by the type of cores available in the chip and include homogeneous and heterogeneous architectures. A homogeneous multi-core DSP architecture consists of cores that are from the same type, meaning that all cores in the die are DSP processors. In contrast, heterogeneous architectures contain different types of cores. This can be a collection of DSPs with general purpose processors (GPPs), graphics processing units (GPUs) or micro controller units (MCUs).

Another classification of multi-core DSP processors is by the type of interconnects between the cores.

More details on the types of interconnect being used in multi-core DSPs as well as the memory hierarchy of these multiple cores are presented below, followed by an overview of the latest multi-core chips. A brief discussion on performance analysis is also included.

3.1 Interconnect and Memory Organization

As shown in Fig. 4, multiple DSP cores can be connected together through a hierarchical or mesh topology. In hierarchical interconnected multi-core DSP platforms, data transfers between cores are performed through one or more switching units. In order to scale these architectures, a hierarchy of switches needs to be planned. CPUs that need to communicate with low latency and high bandwidth will be placed close together on a shared switch and will have low latency access to each others' memory. Switches will be connected together to allow more distant CPUs to communicate with longer latency. Communication is done by memory transfer between the memories associated with the CPUs. Memory can be shared between CPUs or be local to a CPU. The most prominent type of memory architecture makes use of Level 1 (L1) local memory dedicated to each core and Level 2 (L2) which can be dedicated or shared between the cores as well as Level 3 (L3) internal or external shared memory. If local, data is moved off that memory to another local memory using a non CPU block in charge of block memory transfers, usually called a DMA. The memory map of such a system can become quite complex and caches are often used to make the memory look "flat" to the programmer. L1, L2 and even L3 caches can be used to automatically move data around the memory hierarchy without explicit knowledge of this movement in the program. This simplifies and makes more portable the software written for such systems but comes at the price of

uncertainty in the time a task needs to complete because of uncertainty in the number of cache misses [5].

In a mesh network [6-7], the DSP processors are organized in a 2D array of nodes. The nodes are connected through a network of buses and multiple simple switching units. The cores are locally connected with their "north", "south", "east" and "west" neighbors. Memory is generally local, though a single node might have a cache hierarchy. This architecture allows multi-core DSP processors to scale to large numbers without increasing the complexity of the buses or switching units. However, the programmer generally has to write code that is aware of the local nature of the CPU. Explicit message passing is often used to describe data movement.

Multi-core DSP platforms can also be categorized as Symmetric Multiprocessing (SMP) platforms and Asymmetric Multiprocessing (AMP) platforms. In an SMP platform, a given task can be assigned to any of the cores without affecting the performance in terms of latency. In an AMP platform, the placement of a task can affect the latency, giving an opportunity to optimize the performance by optimizing the placement of tasks. This optimization comes at the expense of an increased programming complexity since the programmer has to deal with both space (task assignment to multiple cores) and time (task scheduling). For example, the mesh network architecture of Fig. 4 is AMP since placing dependent tasks that need to heavily communicate in neighboring processors will significantly reduce the latency. In contrast, in a hierarchical interconnected architecture, in which the cores mostly communicate by means of a shared L2/L3 memory and have to cache data from the shared memory, the tasks can be assigned to any of the cores without significantly affecting the latency. SMP platforms are easy to program but can result in a much increased latency as compared to AMP platforms.

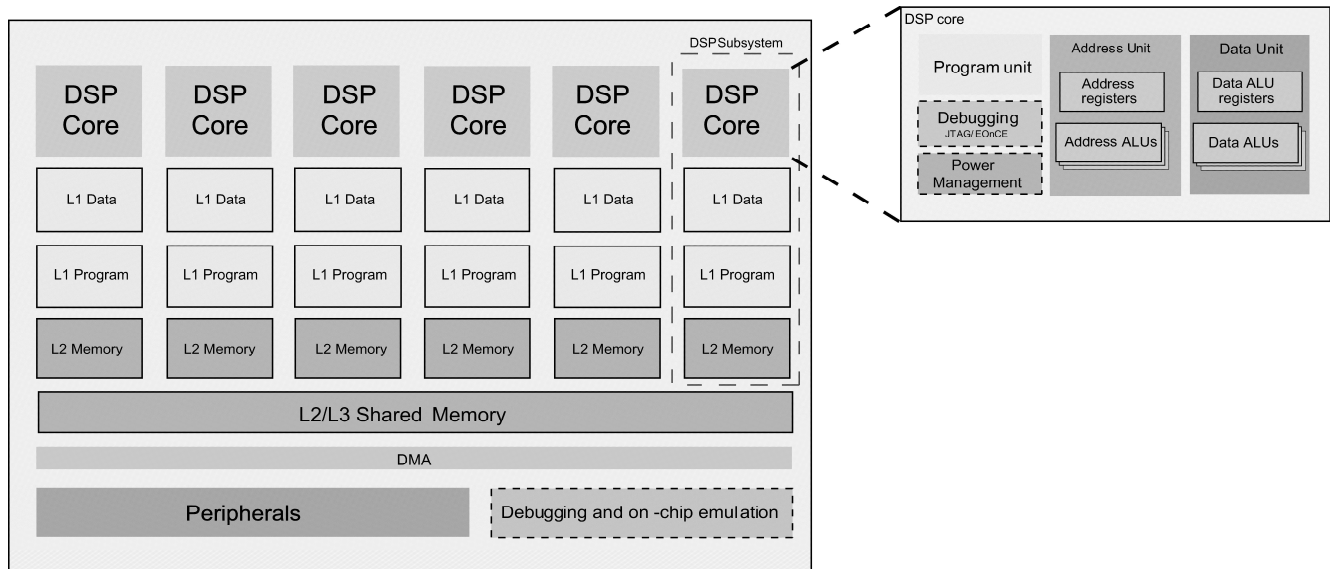


Fig.3. Typical multi-core DSP platform.

Table 1: Multi-core DSP platforms.

	TI [8]	Freescale [9]	picoChip [10]	Tilera [11]	Sandbridge [12-13]
Processor	TNETV3020	MSC8156	PC205	TILE64	SB3500
Architecture	Homogeneous	Homogeneous	Heterogeneous	Homogeneous	Heterogeneous
No. of Cores	6 DSPs	6 DSPs	248 DSPs 1 GPP	64 DSPs	3 DSPs 1 GPP
Interconnect Topology	Hierarchical	Hierarchical	Mesh	Mesh	Hierarchical
Applications	Wireless Video VoIP	Wireless	Wireless	Wireless Networking Video	Wireless

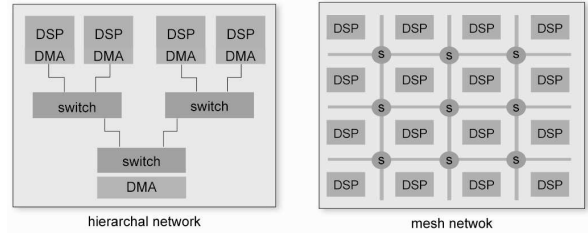


Fig. 4. Interconnect types of multi-core DSP architectures.

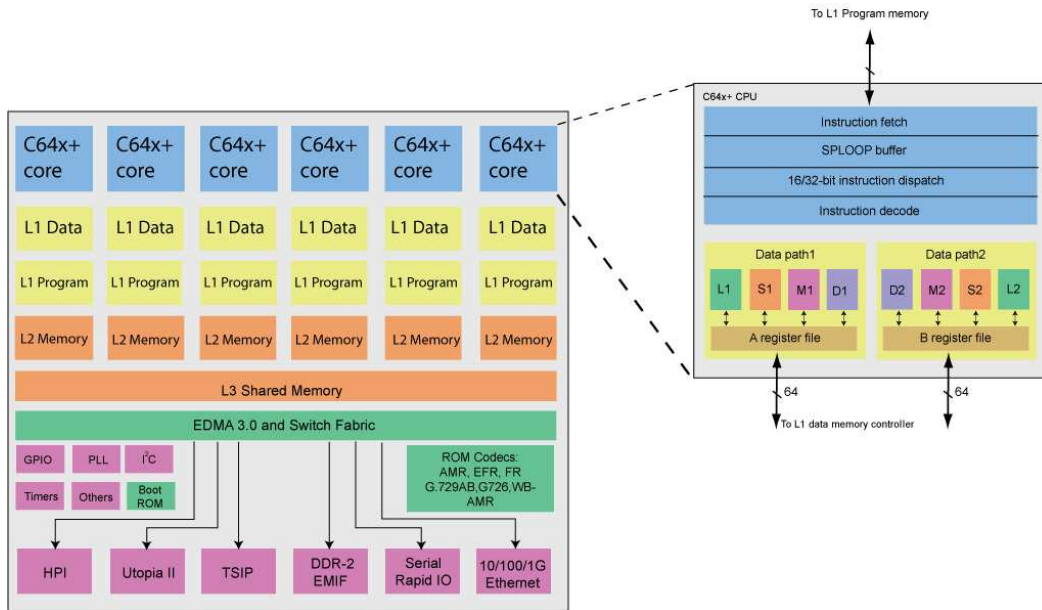


Fig. 5. Texas Instruments TNETV3020 multi-core DSP processor.

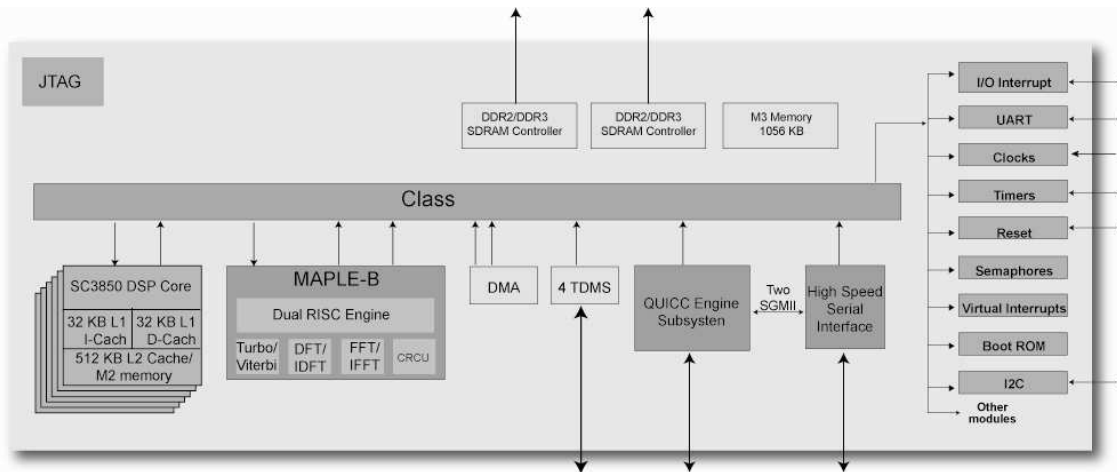


Fig. 6. Freescale 8156 multi-core DSP processor.

3.2 Existing Vendor-Specific Multi-Core DSP Platforms

Several vendors manufacture multi-core DSP platforms such as Texas Instruments (TI) [8], Freescale [9], picoChip [10], Tilera [11], and Sandbridge [12-13]. Table 1 provides an overview of a number of these multi-core DSP chips.

Texas Instruments has a number of homogeneous and heterogeneous multi-core DSP platforms all of which are

based on the hierarchical-interconnect architecture. One of the latest of these platforms is the TNETV3020 (Fig. 5) which is optimized for high performance voice and video applications in wireless communications infrastructure [8]. The platform contains six TMS320C64x+ DSP cores each capable of running at 500 MHz and consumes 3.8 W of power. TI also has a number of other homogeneous multi-core DSPs such as the TMS320TCI6488 which has three

1 GHz C64x+ cores and the older TNETV3010 which contains six TMS320C55x cores, as well as the TMS320VC5420/21/41 DSP platforms with dual and quad TMS320VC54x DSP cores.

Freescale's multi-core DSP devices are based on the StarCore 140, 3400 and 3850 DSP subsystems which are included in the MSC8112 (two SC140 DSP cores), MSC8144E (four SC3400 DSP cores) and its latest MSC8156 DSP chip (Fig. 6) which contains six SC3850 DSP cores targeted for 3G-LTE, WiMAX, 3GPP/3GPP2 and TD-SCDMA applications [9]. The device is based on a homogeneous hierarchical interconnect architecture with chip level arbitration and switching system (CLASS).

PicoChip manufactures high performance multi-core DSP devices that are based on both heterogeneous (PC205) and homogeneous (PC203) mesh interconnect architectures. The PC205 (Fig. 7) was taken as an example of these multi-core DSPs [10]. The two building blocks of the PC205 device are an ARM926EJ-S microprocessor and the picoArray. The picoArray consists of 248 VLIW DSP processors connected together in a 2D array as shown in Fig. 8. Each processor has dedicated instruction and data memory as well as access to on-chip and external memory. The ARM926EJ-S used for control functions is a 32-bit RISC processor. Some of the PC205 applications are in high-speed wireless data communication standards for metropolitan area networks (WiMAX) and cellular networks (HSDPA and WCDMA), as well as in the implementation of advanced wireless protocols.

Tilera manufactures the TILE64, TILEPro36 and TILEPro64 multi-core DSP processors [11]. These are based on a highly scalable homogeneous mesh interconnect architecture.

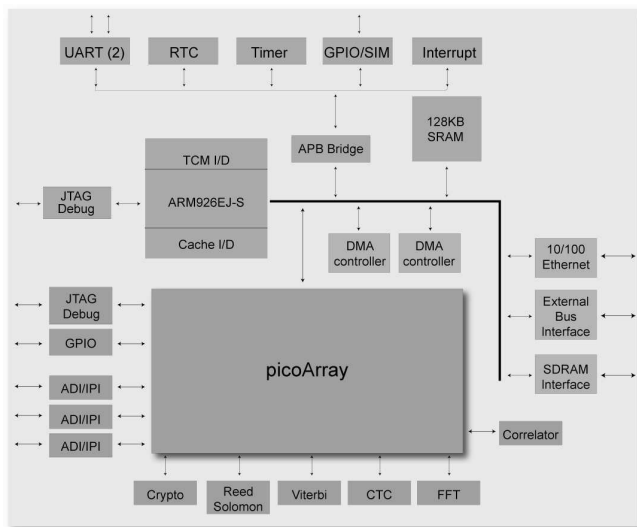


Fig.7. picoChip PC205 multi-core DSP processor.

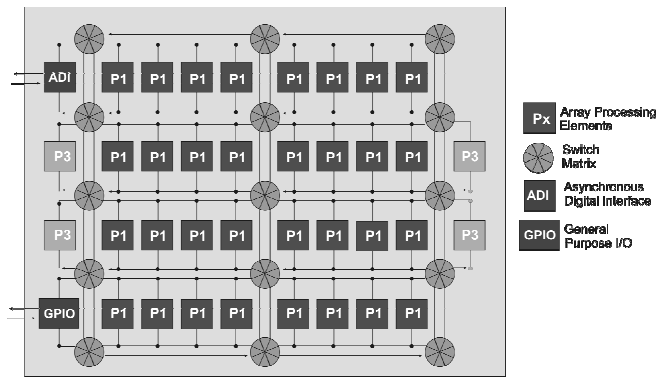


Fig. 8. picoChip picoArray.

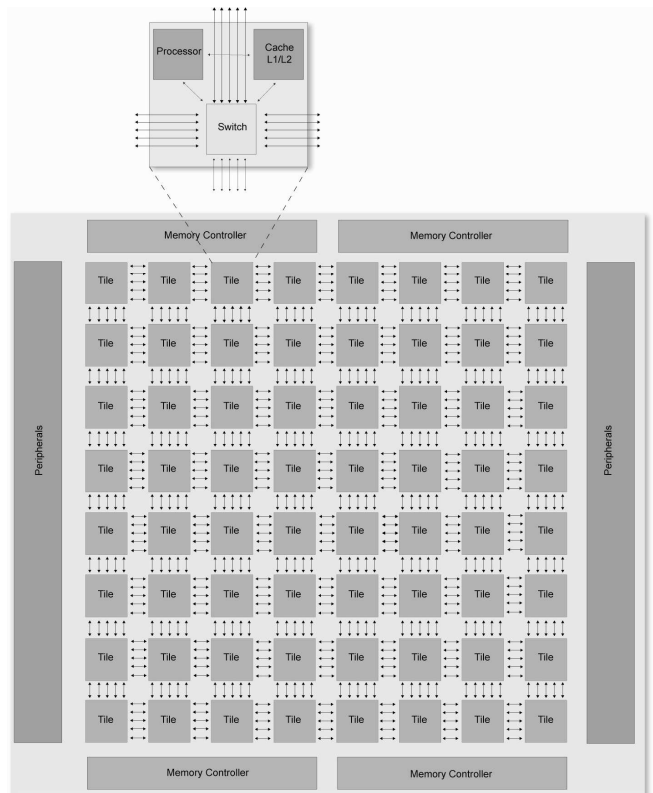


Fig. 9. Tilera TILE64 multi-core DSP processor.

The TILE64 family features 64 identical processor cores (tiles) interconnected using a mesh network of buses (Fig. 9). Each tile contains a processor, L1 and L2 cache memory and a non-blocking switch that connects each tile to the mesh. The tiles are organized in an 8 x 8 grid of identical general processor cores and the device contains 5 MB of on-chip cache. The operating frequencies of the chip range from 500 MHz to 866 MHz and its power consumption ranges from 15 – 22 W. Its main target applications are advanced networking, digital video and telecom.

SandBridge manufactures multi-core heterogeneous DSP chips intended for software defined radio applications. The SB3011 includes four DSPs each running at a minimum of 600 MHz at 0.9V. It can execute up to 32 independent

Table 2: BTDI OFDM benchmark results on various processors for the maximum number of simultaneous OFDM channels processed in real time. The specific number of simultaneous OFDM channels is given in [17].

	<i>Clock (MHz)</i>	<i>DSP cores</i>	<i>OFDM channels</i>
<i>TI TMS320C6455</i>	1200	1	Lowest
<i>Freescale MSC8144</i>	1000	4	Low
<i>Sandbridge SB3500</i>	500	3	Medium
<i>picoChip PC102</i>	160	344	High
<i>Tilera TILE64</i>	866	64	Highest

instruction streams while issuing vector operations for each stream using an SIMD datapath. An ARM926EJ-S processor with speeds up to 300 MHz implements all necessary I/O devices in a smart phone and runs Linux OS. The kernel has been designed to use the POSIX pthreads open standard [14] thus providing a cross platform library compatible with a number of operating systems (Unix, Linux and Windows). The platform can be programmed in a number of high-level languages including C, C++ or Java [12-13].

3.3 Multi-Core DSP Platform Performance Analysis

Benchmark suites have been typically used to analyze the performance among architectures [15]. In practice, benchmarking of multicore architectures has proven to be significantly more complicated than benchmarking of single core devices because multicore performance is affected not only by the choice of CPU but also very heavily by the CPU interconnect and the connection to memory. There is no single agreed-upon programming language for multicore programming and, hence, there is no equivalent of the “out of the box” benchmark, commonly used in single core benchmarks. Benchmark performance is heavily dependent on the amount of tweaking and optimization applied as well as the suitability of the benchmark for the particular architecture being evaluated. As a result, it can be seen that single core benchmarking was already a complicated task when done well, and multicore benchmarking is proving to be exponentially more challenging. The topic of benchmark suites for multicore remains an active field of study [16]. Currently available benchmarks are mainly simplified benchmarks that were mainly developed for single-core systems.

One such a benchmark is the Berkeley Design Technology, Inc (BTDI) OFDM benchmark [17] which was used to evaluate and compare the performance of some single- and multi-core DSPs in addition to other processing engines. The BTDI OFDM benchmark is a simplified digital signal processing path for an FFT-based orthogonal frequency division multiplexing (OFDM) receiver [17]. The path consists of a cascade of a demodulator, finite impulse response (FIR) filter, FFT, slicer, and Viterbi decoder. The benchmark does not include interleaving, carrier recovery,

symbol synchronization, and frequency-domain equalization.

Table 2 shows relative results for maximizing the number of simultaneous non-overlapping OFDM channels that can be processed in real time, as would be needed for an access point or a base station. These results show that the four considered multi-core DSPs can process in real time a higher number of OFDM channels as compared to the considered single-core processor using this specific simplified benchmark.

However, it should be noted that this application benchmark does not necessarily fit the use cases for which the candidate processors were designed. In other words, different results can be produced using different benchmarks since single and multi-core embedded processors are generally developed to solve a particular class of functions which may or may not match the benchmark in use. At the end, what matters most is the actual performance achieved when the chips are tested for the desired customer’s end solution.

4. SOFTWARE TOOLS FOR MULTI-CORE DSPs

Due to the hard real-time nature of DSP programming, one of the main requirements that DSP programmers insist on having is the ability to view low level code, to step through their programs instruction by instruction, and evaluate their algorithms and “see” what is happening at every processor clock cycle. Visibility is one of the main impediments to multi-core DSP programming and to real-time debugging as the ability to “see” in real time decreases significantly with the integration of multiple cores on a single chip. Improved chip-level debug techniques and hardware-supported visualization tools are needed for multi-core DSPs. The use of caches and multiple cores has complicated matters and forced programmers to speculate about their algorithms based on worst-case scenarios. Thus, their reluctance to move to multi-core programming approaches. For programmers to feel confident about their code, timing behavior should be predictable and repeatable [5]. Hardware tracing with Embedded Trace Buffers (ETB) [18] can be used to partially alleviate the decreased visibility issue by storing traces that provide a detailed account of code execution, timing, and data accesses. These traces are collected internally in real-time and are usually retrieved at a later time when a program failure occurs or for collecting useful statistics. Virtual multi-core platforms and simulators, such as Simics by Virtutech [19] can help programmers in developing, debugging, and testing their code before porting it to the real multi-core DSP device.

Operating Systems (OS) provide abstraction layers that allow tasks on different cores to communicate. Examples of OS include SMP Linux [20-21], TI’s DSP BIOS [22], Enea’s OSEck [23]. One main difference between these OS is in how the communication is performed between tasks running on different cores. In SMP Linux, a common set of

tables that reflect the current global state of the system are shared by the tasks running on different cores. This allows the processes to share the same global view of the system state. On the other hand, TI's DSP/BIOS and Enea's OSEck supports a message passing programming model. In this model, the cores can be viewed as "islands with bridges" as contrasted with the "global view" that is provided by SMP Linux. Control and management middleware platforms, such as Enea's dSpeed [23], extend the capabilities of the OS to allow enhanced monitoring, error handling, trace, diagnostics, and inter-process communications.

As in memory organization, programming models in multi-core processors include Symmetric Multiprocessing (SMP) models and Asymmetric Multiprocessing (AMP) models [24]. In an SMP model, the cores form a shared set of resources that can be accessed by the OS.

The OS is responsible for assigning processes to different cores while balancing the load between all the cores. An example of such OS is SMP Linux [18-19] which boasts a huge community of developers and lots of inexpensive software and mature tools. Although SMP Linux has been used on AMP architectures such as the mesh interconnected Tiler architecture, SMP Linux is more suitable for SMP architectures (Section 3.1) because it provides a shared symmetric view. In comparison, TI's DSP/BIOS and Enea's OSE can better support AMP architectures since they allow the programmer to have more control over task assignments and execution. The AMP approach does not balance processes evenly between the cores and so can restrict which processes get executed on what cores. This model of multi-core processing includes classic AMP, processor affinity and virtualization [23].

Classic AMP is the oldest multi-core programming approach. A separate OS is installed on each core and is responsible for handling resources on that core only. This significantly simplifies the programming approach but makes it extremely difficult to manage shared resources and I/O. The developer is responsible for ensuring that different cores do not access the same shared resource as well as be able to communicate with each other.

In processor affinity, the SMP OS scheduler is modified to allow programmers to assign a certain process to a specific core. All other processes are then assigned by the OS. SMP Linux has features to allow such modifications. A number of programming languages following this approach have appeared to extend or replace C in order to better allow programmers to express parallelism. These include OpenMP [25], MPI [26], X10 [27], MCAPI [28], GlobalArrays [29], and Uniform Parallel C [30]. In addition, functional languages such as Erlang [31] and Haskell [32] as well as stream languages such as ACOTES [33] and StreamIT [34] have been introduced. Several of these languages have been ported to multi-core DSPs. OpenMP is an example of that. It is a widely-adopted shared memory parallel programming interface providing high level programming constructs that enable the user to easily expose an application's task and

loop level parallelism in an incremental fashion. Its range of applicability was significantly extended by the addition of explicit tasking features. The user specifies the parallelization strategy for a program at a high level by annotating the program code; the implementation works out the detailed mapping of the computation to the machine. It is the user's responsibility to perform any code modifications needed prior to the insertion of OpenMP constructs. In particular, OpenMP requires that dependencies that might inhibit parallelization are detected and where possible, removed from the code. The major features are directives that specify that a well-structured region of code should be executed by a team of threads, who share in the work. Such regions may be nested. Work sharing directives are provided to effect a distribution of work among the participating threads [35].

Virtualization partitions the software and hardware into a set of virtual machines (VM) that are assigned to the cores using a Virtual Machine Manager (VMM). This allows multiple operating systems to run on single or multiple cores. Virtualization works as a level of abstraction between the OS and the hardware. VirtualLogix employs virtualization technology using its VLX for embedded systems [36]. VLX announced support for TI single and multi-core DSPs. It allows TI's real-time OS (DSP/BIOS) to run concurrently with Linux. Therefore, DSP/BIOS is left to run critical tasks while other applications run on Linux.

5. APPLICATIONS OF MULTI-CORE DSPs

5.1 Multi-core for mobile application processors

The earliest SoC multi-core in the embedded space was the two-core heterogeneous DSP+ARM combination introduced by TI in 1997. These have evolved into the complex OMAP line of SoC for handset applications. Note that the latest in the OMAP line has both multi-core ARM (symmetric multiprocessing) and DSP (for heterogeneous multiprocessing). The choice and number of cores is based on the best solution for the problem at hand and many combinations are possible. The OMAP line of processors is optimized for portable multimedia applications. The ARM cores tend to be used for control, user interaction and protocol processing, whereas the DSPs tend to be signal processing slaves to the ARMs, performing compute intensive tasks such as video codecs. Both CPUs have associated hardware accelerators to help them with these tasks and a wide array of specialized peripherals allows glueless connectivity to other devices.

This multi-core is an integration play to reduce cost and power in the wireless handset. Each core had its own unique function and the amount of interaction between the cores was limited. However, the development of a communications bridge between the cores and a master/slave programming paradigm were important developments that allowed this model of processing to become the most highly used multi-core in the embedded space today [37].

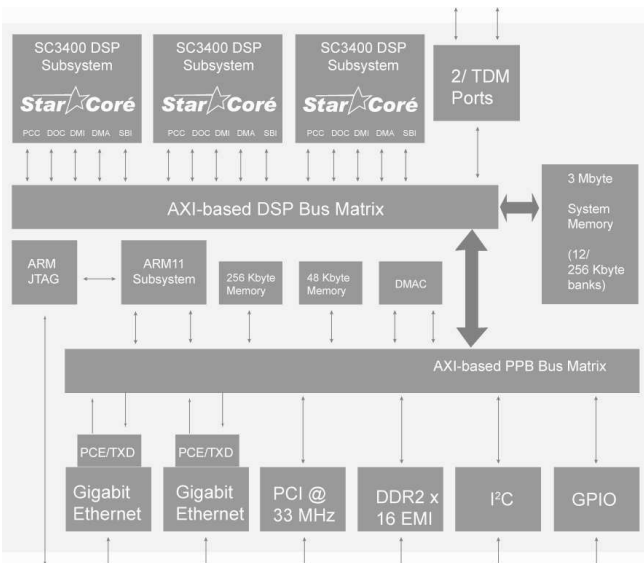


Fig.10. The Agere SP2603.

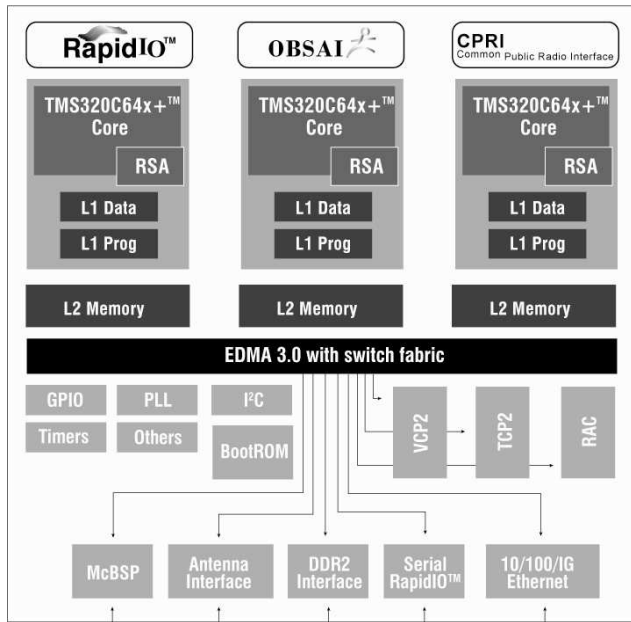


Fig. 11. Texas Instruments TCI6487.

5.2 Multi-core for Core network Transcoding

The next integration play was in the transcoding space. In this space, the master/slave approach is again taken, with a host processor, usually servicing multiple DSPs, that is in charge of load balancing many tasks onto the multi-core DSP. Each task is independent of the others (except for sharing program and some static tables) and can run on a single DSP CPU. Fig. 10 shows the Agere SP2603, a multi-core device used in transcoding applications.

Therefore, the challenge in this type of multi-core SoC is not in the partitioning of a program into multiple threads or the coordination of processing between CPUs, but in the

coordination of CPUs in the access of shared, non CPU, resources, such as DDR memory, Ethernet ports, shared L2 on chip memory, bus resources, and so on. Heterogeneous variants also exist with an ARM on chip to control the array of DSP cores.

Such multi-core chips have reduced the power per channel and cost per channel by an order of magnitude over the last decade.

5.3 Multi-core for Base Station Modems

Finally, the last five years have seen many multi-core entrants into the base station modem business for cellular infrastructure. The most successful have been DSP based with a modest number of CPUs and significant shared resources in memory, acceleration and I/O. An example of such a device is the Texas Instruments TCI6487 shown in Fig. 11.

Applications that use these multi-core devices require very tight latency constraints, and each core often has a unique functionality on the chip. For instance, one core might do only transmit while another does receive and another does symbol rate processing. Again, this is not a generic programming problem. Each core has a specific and very well timed set of tasks to perform. The trick is to make sure that timing and performance issues do not occur due to the sharing of non CPU resources [38].

However, the base station market also attracted new multi-core architectures in a way that neither handset (where the cost constraints and volume tended to favor hardwired solutions beyond the ARM/DSP platform) nor transcoding (where the complexity of the software has kept “standard” DSP multi-core in the forefront) have experienced. Examples of these new paradigm companies include Chameleon, PACT, BOPS, Picochip, Morpho, Morphics and Quicksilver. These companies arose in the late 90s and mostly died in the fallout of the tech bubble burst. They suffered from a lack of production quality tooling and no clear programming model. In general, they came in two types; arrays of ALUs with a central controller and arrays of small CPUs, tightly connected and generally intended to communicate in a very synchronized manner. Fig. 8 shows the picoArray used by picoChip, a proponent of regular, meshed arrays of processors. Serious programming challenges remain with this kind of architecture because it requires two distinct modes of programming, one for the CPUs themselves and one for the interconnect between the CPUs. A single programming language would have to be able to not only partition the workload, but also comprehend the memory locality, which is severe in a mesh-based architecture.

5.4 Next Generation Multi-Core DSP Processors

Current and emerging mobile communications and networking standards are providing even more challenges to DSP. The high data-rates for the physical layer processing,

as well as the requirements for very low power have driven designers to use ASIC designs. However, these are becoming increasingly complex with the proliferation of protocols, driving the need for software solutions.

Software defined radio (SDR) holds the promise of allowing a single piece of silicon to alternate between different modem standards. Originally motivated by the military as a way to allow multinational forces to communicate [39], it has made its way into the commercial arena due to a proliferation of different standards on a single cell phone (for instance GSM, EDGE, WCDMA, Bluetooth, 802.11, FM radio, DVB).

SODA [40] is one multi-core DSP architecture designed specifically for software-defined radio (SDR) applications. Some key features of SODA are the lack of cache with multiple DMA and scratchpad memories used instead for explicit memory control. Each of the processors has a 32x16bit SIMD datapath and a coupled scalar datapath designed to handle the basic DSP operations performed on large frames of data in communication systems.

Another example is the AsAP architecture [41] which relies on the dataflow nature of DSP algorithms to obtain power and performance efficiency. Shown in Fig. 12, it is similar to the Tiler architecture at a superficial glance, but also takes the mesh network principal to its logical conclusion, with very small cores (0.17mm^2) and only a minimal amount of memory per core (128 word program and 128 word data). The cores communicate asynchronously by doubly clocked FIFO buffers and each core has its own clock generator so that the device is essentially clockless. When a FIFO is either empty or full, the associated cores will go into a low power state until they have more data to process. These and other power savings techniques are used in a design that is heavily focused on low power computation. There is also an emphasis on local communication, with each chip connected to its neighbors, in a similar manner to the Tiler multi-core. Even within the core, the connectivity is focused on allowing the core to absorb data rather than reroute it to other cores. The overall goal is to optimize for data flow programming with mostly local interconnect. Data can travel a distance of more than one core but will require more latency to do so. The AsAP chip is interesting as a “pure” example of a tiled array of processors with each processor performing a simple computation. The programming model for this kind of chip is however, still a topic of research. Ambric produced an architecturally similar chip [42] and showed that, for simple data flow problems, software tooling could be developed.

An example of this data flow approach to multi-core DSP design can be found in [43], where the concept of Bulk-Synchronous Processing (BSP), a model of computation where data is shared between threads mostly at synchronization barriers, is introduced. This deterministic approach to the mapping of algorithms to multi-core is in line with the recommendations made in [44] where it is argued that adding parallelism in a non deterministic manner

(such as is commonly done with POSIX threads [14]) leads to systems that are unreasonably hard to test and debug. Fortunately, the parallelization of DSP algorithms can often be done in a deterministic manner using data flow diagrams. Hence, DSP may be a more fruitful space for the development of multi-core than the general purpose programming space.

Sandbridge (see Section 3.2) has also been producing DSPs designed for the SDR space for several years.

6. CONCLUSIONS AND FUTURE TRENDS

In the last 2 years, the embedded DSP market has been swept up by the general increase in interest in multi-core that has been driven by companies such as Intel and Sun.

One of the reasons for this is that there is now a lot of focus on tooling in academia and also a willingness on the part of users to accept new programming paradigms. This industry wide effort will have an effect on the way multi-core DSPs are programmed and perhaps architected. But it is too early to say in what way this will occur. Programming multi-core DSPs remains very challenging. The problem of how to take a piece of sequential code and optimally partition it across multiple cores remains unsolved. Hence, there will naturally be a lot of variations in the approaches taken. Equally important is the issue of debug and visibility. Developing effective and easy-to-use code development and real-time debug tools is tremendously important as the opportunity for bugs goes up significantly when one starts to deal with both time and space.

The markets that DSP plays in have unique features in their desire for low power, low cost and hard real-time processing, with an emphasis on mathematical computation. How well the multi-core research being performed presently in academia will address these concerns remains to be seen.

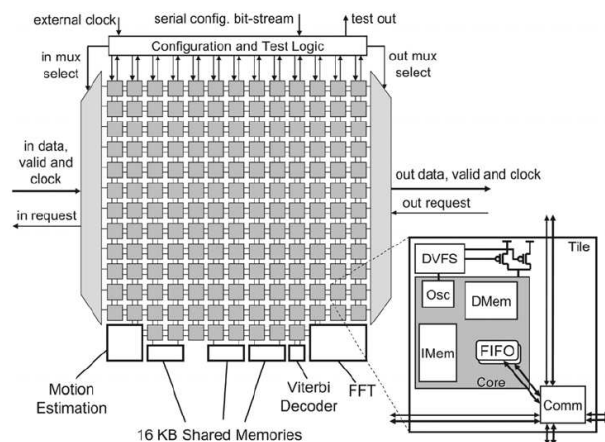


Fig.12. The AsAP processor architecture.

7. REFERENCES

- [1] G.M. Amdahl, “Validity of the single-processor approach to achieving large scale computing capabilities,” in *AFIPS Conference Proceedings*, vol. 30, pp. 483-485, Apr. 1967.

- [2] M.D. Hill and M.R. Marty, "Amdahl's Law in the multicore era," *IEEE Computer Magazine*, vol.41, no.7, pp.33-38, July 2008.
- [3] W. Strauss, "Wireless/DSP market bulletin," Forward Concepts, Feb 2009. [Online] <http://www.fwdconcepts.com/dsp2209.htm>.
- [4] I. Scheiwe, "The shift to multi-core DSP solutions," *DSP-FPGA*, Nov. 2005 [Online] <http://www.dsp-fpga.com/articles/id/?21>.
- [5] S. Bhattacharyya, J. Bier, W. Gass, R. Krishnamurthy, E. Lee, and K. Konstantinides, "Advances in hardware design and implementation of signal processing systems [DSP Forum]," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 175-180, Nov. 2008.
- [6] Practical Programmable Multi-Core DSP, picoChip, Apr. 2007, [Online] <http://www.picochip.com/>.
- [7] Tile Processor Architecture Technology Brief, Tilera, Aug. 2008, [Online] <http://www.tilera.com>.
- [8] TNETV3020 Carrier Infrastructure Platform, Texas Instruments, Jan. 2007, [Online] <http://focus.ti.com/lit/ml/spat174a/spat174a.pdf>
- [9] MSC8156 Product Brief, Freescale, Dec. 2008, [Online] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MSC8156&nodeId=0127950E5F5699
- [10] PC205 Product Brief, picoChip, Apr. 2008, [Online] <http://www.picochip.com/>.
- [11] Tile64 Processor Product Brief, Tilera, Aug. 2008, [Online] <http://www.tilera.com>.
- [12] J. Glossner, D. Iancu, M. Moudgill, G. Nacer, S. Jinturkar, M. Schulte, "The Sandbridge SB3011 SDR platform," *Joint IST Workshop on Mobile Future and the Symposium on Trends in Communications (SymptoTIC)*, pp.ii-v, June 2006.
- [13] J. Glossner, M. Moudgill, D. Iancu, G. Nacer, S. Jinturkar, S. Stanley, M. Samori, T. Raja, M. Schulte, "The Sandbridge Sandblaster Convergence platform," Sandbridge Technologies Inc, 2005. [Online] <http://www.sandbridgetech.com/>
- [14] POSIX, IEEE Std 1003.1, 2004 Edition. [Online] http://www.unix.org/version3/ieee_std.html
- [15] G. Frantz and L. Adams, "The three P's of value in selecting DSPs," *Embedded Systems Programming*, pp. 37-46, Nov. 2004.
- [16] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, K. A. Yelick, "The landscape of parallel computing research: a view from Berkeley," Technical Report No. UCB/EECS-2006-183, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf>, Dec. 2006.
- [17] BDTI, [Online] <http://www.bdti.com/bdtimark/ofdm.htm>
- [18] Embedded Trace Buffer, Texas Instruments eXpressDSP Software Wiki, [Online] http://tiexpressdsp.com/index.php?title=Embedded_Trace_Buffer
- [19] VirtuTech, [Online] http://www.virtutech.com/datasheets/simics_mpc8641d.html
- [20] H. Dietz, "Linux parallel processing using SMP," July 1996. [Online] <http://cobweb.ecn.purdue.edu/~pplinux/ppsm.html>
- [21] M. T. Jones, "Linux and symmetric multiprocessing: unblocking the power of Linux SMP systems," [Online] http://www.ibm.com/developerworks/library/l-linux-smp/TI_DSP/BIOS, [Online] <http://focus.ti.com/docs/toolsw/folders/print/dspbios.html>
- [22] TI DSP/BIOS, [Online] <http://focus.ti.com/docs/toolsw/folders/print/dspbios.html>
- [23] Enea, [Online] <http://www.enea.com/>
- [24] K. Williston, "Multi-core software: strategies for success," *Embedded Innovator*, pp. 10-12, Fall 2008.
- [25] OpenMP, [Online] <http://openmp.org/wp/>
- [26] MPI, [Online] <http://www.mcs.anl.gov/research/projects/mpi/>
- [27] P. Charles, C. Grothoff, V. Saraswat, C. Donawa, A. Kielstra, K. Ebcioglu, C. von Praun, and V. Sarkar, "X10: an object-oriented approach to non-uniform cluster computing," in *ACM OOPSLA*, pp. 519-538, Oct. 2005.
- [28] MCAPI, [Online] <http://www.multicore-association.org/workgroup/comapi.php>
- [29] Global Arrays [Online] <http://www.emsl.pnl.gov/docs/global/>
- [30] Unified Parallel C, [Online] <http://upc.lbl.gov/>
- [31] Erlang, [Online] <http://erlang.org/>
- [32] Haskell [Online] <http://www.haskell.org/>
- [33] ACOTES, [Online] <http://www.hitech-projects.com/euprojects/ACOTES/>
- [34] StreamIT, [Online] <http://www.cag.lcs.mit.edu/streamit/>
- [35] B. Chapman, L. Huang, E. Biscondi, E. Stotzer, A. Shrivastava, and A. Gatherer, "Implementing OpenMP on a high performance embedded multi-core MPSoC," accepted and to appear the *Proceedings of IEEE International Parallel & Distributed Processing Symposium*, 2009.
- [36] VirtualLogix [Online] <http://www.virtuallogix.com/products/vlx-for-embedded-systems/vlx-for-es-supporting-ti-dsp-processors.html>
- [37] E. Heikkila and E. Gulliksen, "Embedded processors 2009 global market demand analysis," VDC Research.
- [38] A. Gatherer, "Base station modems: Why multi-core? Why now?," *ECN Magazine*, Aug. 2008. [Online] http://www.ecnmag.com/supplements-Base-Station-Modems-Why_Multicore.aspx?menuid=580
- [39] Software Communications Architecture, [Online] <http://sca.jpeojtrs.mil/>
- [40] Y. Lin, H. Lee, M. Who, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, K. Flautner, "SODA: A high-performance DSP architecture for Software-Defined Radio," *IEEE Micro*, vol.27, no.1, pp.114-123, Jan.-Feb. 2007
- [41] D. N. Truong et al., "A 167-Processor computational platform in 65nm," *IEEE JSSC*, vol. 44, No. 4, Apr. 2009.
- [42] M. Butts, "Addressing software development challenges for multicore and massively parallel embedded systems," *Multicore Expo*, 2008.
- [43] J. H. Kelm, D. R. Johnson, A. Mahesri, S. S. Lumetta, M. Frank, and S. Patel, "SCHISM: Scalable Cache Incoherent Shared Memory," Tech. Rep. UILU-ENG-08-2212, Univ. of Illinois at Urbana-Champaign, Aug. 2008. [Online] <http://www.crhc.illinois.edu/TechReports/2008reports/08-2212-kelem-tr-with-acks.pdf>
- [44] E. A. Lee, "The problem with threads," UCB Technical Report, Jan. 2006 [Online] <http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-1.pdf>

Comparing Fixed- and Floating-Point DSPs

*Does your design need a fixed- or floating-point DSP?
The application data set can tell you.*



By

*Gene Frantz, TI Principal Fellow, Business Development Manager, DSP
Ray Simar, Fellow and Manager of Advanced DSP Architectures*

System developers, especially those who are new to digital signal processors (DSPs), are sometimes uncertain whether they need to use fixed- or floating-point DSPs for their systems. Both fixed- and floating-point DSPs are designed to perform the high-speed computations that underlie real-time signal processing. Both feature system-on-a-chip (SOC) integration with on-chip memory and a variety of high-speed peripherals to ensure fast throughput and design flexibility. Tradeoffs of cost and ease of use often heavily influenced the fixed- or floating-point decision in the past. Today, though, selecting either type of DSP depends mainly on whether the added computational capabilities of the floating-point format are required by the application.

Different numeric formats

As the terms fixed- and floating-point indicate, the fundamental difference between the two types of DSPs is in their respective numeric representations of data. While fixed-point DSP hardware performs strictly integer arithmetic, floating-point DSPs support either integer or real arithmetic, the latter normalized in the form of scientific notation. TI's TMS320C62x™ fixed-point DSPs have two data paths operating in parallel, each with a 16-bit word width that provides signed integer values within a range from -2^{15} to 2^{15} . TMS320C64x™ DSPs, double the overall throughput with four 16-bit (or eight 8-bit or two 32-bit) multipliers. TMS320C5x™ and TMS320C2x™ DSPs, with architectures designed for handheld and control applications, respectively, are based on single 16-bit data paths.

By contrast, TMS320C67x™ floating-point DSPs divide a 32-bit data path into two parts: a 24-bit mantissa that can be used for either for integer values or as the base of a real number, and an 8-bit exponent. The 16M range of precision offered by 24 bits with the addition of an 8-bit exponent, thus supporting a vastly greater dynamic range than is available with the fixed-point format. The C67x™ DSP can also perform calculations using industry-standard double-width precision (64 bits, including a 53-bit mantissa and an 11-bit exponent). Double-width precision achieves much greater precision and dynamic range at the expense of speed, since it requires multiple cycles for each operation.

Cost versus ease of use

The much greater computational power offered by floating-point DSPs is normally the critical element in the fixed- or floating-point design decision. However, in the early 1990s, when TI released its first floating-point DSP products, other factors tended to obscure the fundamental mathematical issue. Floating-point functions require more internal circuitry, and the 32-bit data paths were twice as wide as those of fixed-point DSPs, which at that time integrated only a single 16-bit data path. These factors, plus the greater number of pins required by the wider data bus, meant a larger die and larger package that resulted in a significant cost premium for the new floating-point devices. Fixed-point DSPs therefore were favored for high-volume applications like digitized voice and telecom concentration cards, where unit manufacturing costs had to be kept low.

Offsetting the cost issue at that time was ease of use. TI floating-point DSPs were among the first DSPs to support the C language, while fixed-point DSPs still needed to be programmed at the assembly code level. In addition, real arithmetic could be coded directly into hardware operations with the floating-point format, while fixed-point devices had to implement real arithmetic indirectly through software routines that added development time and extra instructions to the algorithm. Because floating-point DSPs were easier to program, they were adopted early on for low-volume applications where the time and cost of software development were of greater concern than unit manufacturing costs. These applications were found in research, development prototyping, military applications such as radar, image recognition, three-dimensional graphics accelerators for workstations and other areas.

Today the early differences in cost and ease of use, while not altogether erased, are considerably less pronounced. Scores of transistors can now fit into the same space required by a single transistor a decade ago, leading to SOC integration that reduces the impact of a single DSP core on die size and expense. Many DSP-based products, such as TI's broadband, camera imaging, wireless baseband and OMAP™ wireless application platforms, leverage the advantages of rescaling by integrating more than a single core in a product targeted at a specific market. Fixed-point DSPs continue to benefit more from cost reductions of scale in manufacturing, since they are more often used for high-volume applications; however, the same reductions will apply to floating-point DSPs when high-volume demand for the devices appears. Today, cost has increasingly become an issue of SOC integration and volume, rather than a result of the size of the DSP core itself.

The early gap in ease of use has also been reduced. TI fixed-point DSPs have long been supported by outstandingly efficient C compilers and exceptional tools that

provide visibility into code execution. The advantage of implementing real arithmetic directly in floating-point hardware still remains; but today advanced mathematical modeling tools, comprehensive libraries of mathematical functions, and off-the-shelf algorithms reduce the difficulty of developing complex applications—with or without real numbers—for fixed-point devices. Overall, fixed-point DSPs still have an edge in cost and floating-point DSPs in ease of use, but the edge has narrowed until these factors should no longer be overriding in the design decision.

Floating-point accuracy

As the cost of floating-point DSPs has continued to fall, the choice of using a fixed- or floating-point DSP boils down to whether floating-point math is needed by the application data set. In general, designers need to resolve two questions: What degree of accuracy is required by the data set? and How predictable is the data set?

The greater accuracy of the floating-point format results from three factors. First, the 24-bit word width in TI C67x™ floating-point DSPs yields greater precision than the C62x™ 16-bit fixed-point word width, in integer as well as real values. Second, exponentiation vastly increases the dynamic range available for the application. A wide dynamic range is important in dealing with extremely large data sets and with data sets where the range cannot be easily predicted. Third, the internal representations of data in floating-point DSPs are more exact than in fixed-point, ensuring greater accuracy in end results.

The final point deserves some explanation. Three data word widths are important to consider in the internal architecture of a DSP. The first is the I/O signal word width, already discussed, which is 24 bits for C67x floating-point, 16 bits for C62x fixed-point, and can be 8, 16, or 32 bits for C64x™ fixed-point DSPs. The second word width is that of the coefficients used in multiplications. While fixed-point coefficients are 16 bits, the same as the signal data in C62x DSPs, floating-point coefficients can be 24 bits or 53 bits of precision, depending whether single or double precision is used. The precision can be extended beyond the 24 and 53 bits in some cases when the exponent can represent significant zeroes in the coefficient.

Finally, there is the word width for holding the intermediate products of iterated multiply-accumulate (MAC) operations. For a single 16-bit by 16-bit multiplication, a 32-bit product would be needed, or a 48-bit product for a single 24-bit by 24-bit multiplication. (Exponents have a separate data path and are not included in this discussion.) However, iterated MACs require additional bits for overflow headroom. In C62x fixed-point devices, this overflow headroom is 8 bits, making the total intermediate product word width 40 bits (16 signal + 16 coefficient + 8 overflow). Integrating the same

proportion of overflow headroom in C67x floating-point DSPs would require 64 intermediate product bits (24 signal + 24 coefficient + 16 overflow), which would go beyond most application requirements in accuracy. Fortunately, through exponentiation the floating-point format enables keeping only the most significant 48 bits for intermediate products, so that the hardware stays manageable while still providing more bits of intermediate accuracy than the fixed-point format offers. These word widths are summarized in Table 1 for several TI DSP architectures.

Table 1. Word widths for TI DSPs

TI DSP(s)	Format	Word Width		
		Signal I/O	Coefficient	Intermediate result
C25x	fixed	16	16	40
C5x™/C62x™	fixed	16	16	40
C64x™	fixed	8/16/32	16	40
C3x™	floating	24 (mantissa)	24	32
C67x™(SP)	floating	24 (mantissa)	24	24/53
C67x(DP)	floating	53	53	53

Video and audio data set requirements

The advantages of using the fixed- and floating-point formats can be illustrated by contrasting the data set requirements of two common signal-processing applications: video and audio. Video has a high sampling rate that can amount to tens or even hundreds of megabits per second (Mbps) in pixel data, depending on the application. Pixel data is usually represented in three words, one for each of the red, green and blue (RGB) planes of the image. In most systems, each color requires 8 to 12 bits, though advanced applications may use up to 14 bits per color. Key mathematical operations of the industry-standard MPEG video compression algorithms include discrete cosine transforms (DCTs) and quantization, and there is limited filtering.

Audio, by contrast, has a more limited data flow of about 1 Mbps that results from 24 bits sampled at 48 kilosamples per second (ksps). A higher sampling rate of 192 ksps will quadruple this data flow rate in the future, yet it is still significantly less than video. Operations on audio data include infinite impulse response (IIR) and intensive filtering.

Other application areas

Video thus has much more raw data to process than audio. DCTs and quantization are handled effectively using integer operations, which together with the short data words make video a natural application for C62x and C64x fixed-point DSPs. The massive parallelism of the C64x makes it a excellent platform for applications that run multiple video channels, and some C64x DSP products have been designed with on-chip video interfaces that provide seamless data throughput.

Video may have a larger data flow, but audio has to process its data more accurately. While the eye is easily fooled, especially when the image is moving, the ear is hard to deceive. Although audio has usually been implemented in the past using fixed-point devices, high-fidelity audio today is transistioning to the greater accuracy of the floating-point format. Some C67x DSP products further this trend by integrating a multi-channel audio serial port (McASP) in order to make audio system design easier. As the newest audio innovations become increasingly common in consumer electronics, demand for floating-point DSPs will also rise, helping to drive costs closer to parity with fixed-point DSPs.

The wider words (24-bit signal, 24-bit coefficient, 53-bit intermediate product) of C67x DSPs provide much greater accuracy in audio output, resulting in higher sound quality. Sampling sound with 24 bits of accuracy yields 144 dB of dynamic range, which provides more than adequate coverage for the full amplitude range needed in sound reproduction. Wide coefficients and intermediate products provide a high degree of accuracy for internal operations, a feature that audio requires for at least two reasons.

First, audio typically use cascaded IIR filters to obtain high performance with minimal latency., But, in doing so, each filtering stage propagates the errors of previous stages. So a high degree of precision in both the signal and coefficients are required to minimize the effects of these propagated errors. Second, signal accuracy must be maintained, even as it approaches zero (this is necessary because of the sensitivity of the human ear). The floating-point format by its nature aligns well with the sensitivity of the human ear and becomes more accurate as floating point numbers approach 0. This is the result of the exponent's keeping track of the significant zeros after the binary point and before the significant data in the mantissa. This is in contrast to a fixed point system for very small fractional numbers. All of these aspects of floating-point real arithmetic are essential to the accurate reproduction of audio signals.

Other application areas

The data sets of other types of applications also lend themselves better to either fixed- or floating-point computations. Today, one of the heaviest uses of DSPs is in wired and wireless communications, where most data is transmitted serially in octets that are then

expanded internally for 16-bit processing based on integer operations. Obviously, this data set is extremely well-suited for the fixed-point format, and the enormous demand for DSPs in communications has driven much of fixed-point product development and manufacturing.

Floating-point applications are those that require greater computational accuracy and flexibility than fixed-point DSPs offer. For example, image recognition used for medicine is similar to audio in requiring a high degree of accuracy. Many levels of signal input from light, x-rays, ultrasound and other sources must be defined and processed to create output images that provide useful diagnostic information. The greater precision of C67x signal data, together with the device's more accurate internal representations of data, enable imaging systems to achieve a much higher level of recognition and definition for the user.

Radar for navigation and guidance is a traditional floating-point application since it requires a wide dynamic range that cannot be defined ahead of time and either uses the divide operator or matrix inversions. The radar system may be tracking in a range from 0 to infinity, but need to use only a small subset of the range for target acquisition and identification. Since the subset must be determined in real time during system operation, it would be all but impossible to base the design on a fixed-point DSP with its narrow dynamic range and quantization effects..

Wide dynamic range also plays a part in robotic design. Normally, a robot functions within a limited range of motion that might well fit within a fixed-point DSP's dynamic range. However, unpredictable events can occur on an assembly line. For instance, the robot might weld itself to an assembly unit, or something might unexpectedly block its range of motion. In these cases, feedback is well out of the ordinary operating range, and a system based on a fixed-point DSP might not offer programmers an effective means of dealing with the unusual conditions. The wide dynamic range of a floating-point DSP, however, enables the robot control circuitry to deal with unpredictable circumstances in a predictable manner.

A data set decision

In recent years, as the world of digital signal processing has become much larger, DSPs have become application-driven. SOC integration means that, along with application-specific peripherals, different cores can be integrated on the same device, enabling DSP products to be tailored for the requirements of specific markets. In this environment, floating-point capabilities have become another element in the overall DSP product mix.

A data set decision

There are still some differences in cost and ease of use between fixed- and floating-point DSPs, but these have become less significant over time. The critical feature for designers is the greater mathematical flexibility and accuracy of the floating-point format. For application data sets that require real arithmetic, greater precision and a wider dynamic range, floating-point DSPs offer the best solution. Application data sets that do not require these computational features can normally use fixed-point DSPs. Once the data set requirements have been determined, it should no longer be difficult to decide whether to use a fixed- or floating-point DSP.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

ECE 445S Real-Time Digital Signal Processing Lab Spring 2025

Signals and Systems

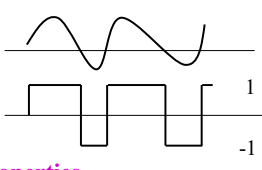
Prof. Brian L. Evans
Dept. of Electrical and Computer Engineering
The University of Texas at Austin

Lecture 3 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

- **Signals**
Continuous-time vs. discrete-time
Analog vs. digital
Unit impulse
- **Continuous-Time System Properties**
- **Sampling**
- **Discrete-Time System Properties**



3-2

2

Review

Many Faces of Signals

- **Function**, e.g. $\cos(t)$ in continuous time or $\cos(\pi n)$ in discrete time, **useful in analysis**
- **Sequence of numbers**, e.g. $\{1,2,3,2,1\}$ or a **sampled triangle function**, **useful in simulation**
- **Set of properties**, e.g. even and causal, **useful in reasoning about behavior**
- **A piecewise representation**, e.g. **useful in analysis**
- **A generalized function**, e.g. $\delta(t)$, **useful in analysis**

$$u(t) = \begin{cases} 1 & \text{for } t > 0 \\ \frac{1}{2} & \text{for } t = 0 \\ 0 & \text{for } t < 0 \end{cases}$$

$$u[n] = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

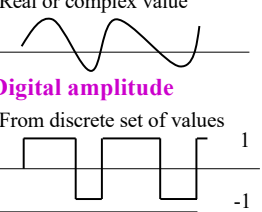
3-3

3

Review

Signals As Functions

- **Continuous-time $x(t)$**
Time, t , is any real value
 $x(t)$ may be 0 for range of t
- **Discrete-time $x[n]$**
 $n \in \{\dots,-3,-2,-1,0,1,2,3,\dots\}$
Integer time index, e.g. n
- **Analog amplitude**
Real or complex value
- **Digital amplitude**
From discrete set of values



Continuous-Time Signal \rightarrow Sampler \rightarrow Discrete-Time Signal
Analog Amplitude Signal \rightarrow Quantizer \rightarrow Digital Amplitude Signal

3-4

4

Review

Continuous-Time Unit Impulse

- **Mathematical idealism for an instantaneous event**
- **Dirac delta as generalized function (a.k.a. functional)**
- **Selected properties**
 - Unit area: $\int_{-\infty}^{\infty} \delta(t) dt = 1$
 - Sifting: $\int_{-\infty}^{\infty} g(t) \delta(t) dt = g(0)$
provided $g(t)$ is defined at $t=0$
- $\delta(t)$ is **infinity or undefined** (both are defensible answers)

$$P_\epsilon(t) = \frac{1}{2\epsilon} \text{rect}\left(\frac{t}{2\epsilon}\right)$$

$$\delta(t) = \lim_{\epsilon \rightarrow 0} P_\epsilon(t)$$

Area = $\lim_{\epsilon \rightarrow 0} \frac{2\epsilon}{2\epsilon} = 1$

3 - 5

5

Review

Continuous-Time Unit Impulse

- $\delta(t)$ under integration
 - $\int_{-\infty}^{\infty} \phi(t) \delta(t) dt = \phi(0)$
 - Assuming $\phi(t)$ is defined at $t=0$
- **Other examples**
 - $\int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1$
 - $\delta(t) * x(t) = \int_{-\infty}^{\infty} \delta(\lambda) x(t-\lambda) d\lambda = x(t)$
 - $\delta(t-T) * x(t) = \int_{-\infty}^{\infty} \delta(\lambda-T) x(t-\lambda) d\lambda = x(t-T)$
- **What about?**
 - $\int_{-\infty}^1 \phi(t) \delta(t) dt = ?$
- **What about?**
 - $\int_{-\infty}^{\infty} \phi(t) \delta(t-T) dt = ?$
 - By substitution of variables, $\int_{-\infty}^{\infty} \phi(\lambda+T) \delta(\lambda) d\lambda = \phi(T)$
- **What about at origin?**
 - $\int_{-\infty}^t \delta(\tau) d\tau = \begin{cases} 1 & t > 0 \\ ? & t = 0 \\ 0 & t < 0 \end{cases} = u(t)$
 - $u(0)$ can take any value
 - Common values: 0, $\frac{1}{2}$, 1

L. B. Jackson, "A correction to impulse invariance," *IEEE Sig. Proc. Letters*, Oct. 2000. 3 - 6

6

Review

Systems

- **Systems operate on signals to produce new signals or new signal representations**

$x(t) \rightarrow T\{\bullet\} \rightarrow y(t)$
 $y(t) = T\{x(t)\}$

$x[n] \rightarrow T\{\bullet\} \rightarrow y[n]$
 $y[n] = T\{x[n]\}$
- **Continuous-time system examples**
 - $y(t) = \frac{1}{2} x(t) + \frac{1}{2} x(t-1)$
 - $y(t) = x^2(t)$

Squaring function can be used in sinusoidal demodulation
- **Discrete-time system examples**
 - $y[n] = \frac{1}{2} x[n] + \frac{1}{2} x[n-1]$
 - $y[n] = x^2[n]$

Average of current input and delayed input is a simple filter

3 - 7

7

Review

Continuous-Time System Properties

- **Let $x(t)$, $x_1(t)$, and $x_2(t)$ be inputs to a continuous-time linear system and let $y(t)$, $y_1(t)$, and $y_2(t)$ be their corresponding outputs**
- **A linear system satisfies**
 - Additivity: $x_1(t) + x_2(t) \Rightarrow y_1(t) + y_2(t)$
 - Homogeneity: $a x(t) \Rightarrow a y(t)$ for any real/complex constant a
- **For time-invariant system, shift of input signal by any real-valued τ causes same shift in output signal, i.e. $x(t - \tau) \Rightarrow y(t - \tau)$, for all τ**

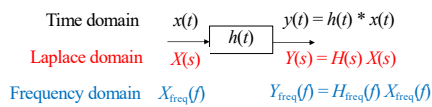
Quick test to identify some nonlinear systems?

3 - 8

8

Why are LTI properties useful?

- An LTI system is uniquely characterized by its impulse response
- Abstract away implementation details by providing an impulse response e.g. to hide intellectual property
- Model unknown system (even though it might not be LTI)
- Fourier transform of the impulse response $h(t)$ is the frequency response of the system $H_{\text{freq}}(f)$



3-9

9

Is a System Linear or Not?

- Example: Squaring block $x(t) \rightarrow \boxed{(\bullet)^2} \rightarrow y(t)$

Does the squaring block pass all-zero input test?

Does the squaring block have homogeneity?

$$\frac{x(t)}{a x(t)} \rightarrow \boxed{(\bullet)^2} \rightarrow \frac{y(t)}{y_{\text{scaled}}(t)} \quad y(t) = x^2(t) \quad y_{\text{scaled}}(t) = (a x(t))^2$$

Check to see if $y_{\text{scaled}}(t) = a y(t)$ for all constant values of a

$$(a x(t))^2 = a x^2(t)$$

$$a^2 x^2(t) = a x^2(t) \quad \text{only for } a = 0 \text{ and } a = 1 \quad \text{No}$$

3-10

10

Is a System Time-Invariant or Not

- Example: Squaring block $x(t) \rightarrow \boxed{(\bullet)^2} \rightarrow y(t)$

Does shift in time for input always give same shift on output?

$$\frac{x(t)}{x(t-t_0)} \rightarrow \boxed{(\bullet)^2} \rightarrow \frac{y(t)}{y_{\text{shifted}}(t)} \quad y(t) = x^2(t) \quad y_{\text{shifted}}(t) = (x(t-t_0))^2$$

Check to see if $y_{\text{shifted}}(t) = y(t-t_0)$ for all real values of t_0

$$(x(t-t_0))^2 = x^2(t-t_0)$$

$$x^2(t-t_0) = x^2(t-t_0) \quad \text{Yes}$$

- All pointwise systems are time-invariant

Output at time t only depends on input at time t

3-11

11

Initial Conditions for Linear Systems

- Observe signals and systems starting at time $t = 0$

- Example: Integrator

$$\frac{x(t)}{x(t)} \rightarrow \int_{-\infty}^t (\bullet) dt \rightarrow \frac{y(t)}{y(t)} \quad y(t) = \int_{-\infty}^t x(u) du = \int_{-\infty}^0 x(u) du + \int_0^t x(u) du$$

- Observe integrator for $t \geq 0$

$$\frac{x(t)}{x(t)} \rightarrow \int_0^t x(u) du + C_0 \rightarrow \frac{y(t)}{C_0} \quad C_0 = \int_{-\infty}^0 x(u) du$$

C_0 is the initial condition w/r to observation

Homogeneity: input $a x(t)$ for output $y_{\text{scaled}}(t)$

Does $y_{\text{scaled}}(t) = a y(t)$ for all values of a ? $y(t) = \int_0^t x(u) du + C_0$

$$y_{\text{scaled}}(t) = \int_0^t a x(u) du + C_0 = a \int_0^t x(u) du + C_0 = a y(t) \quad \text{only if } C_0 = 0$$

- System "at rest" is a necessary condition for linearity

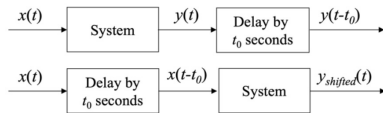
12

Init. Cond. for Time-Invariant Systems?

- Observe system for $t \geq 0^-$

Notation means to include any Dirac delta signals at origin

Does $y_{\text{shifted}}(t) = y(t-t_0)$ for all real-valued t_0 for $t \geq 0^-$?



- Handouts for example systems

Time-Invariance for a (Shift) System Under Observation [link](#)

Time-Invariance for an Integrator [link](#)

3-13

13

Continuous-Time System Properties

- Ideal delay by T seconds



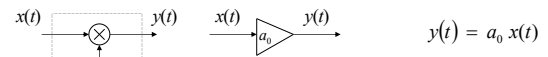
$$y(t) = x(t-T)$$

Role of initial conditions?

Linear? Time-invariant?

- Scale by a constant (a.k.a. gain block)

Two different ways to express it in a block diagram



$$y(t) = a_0 x(t)$$

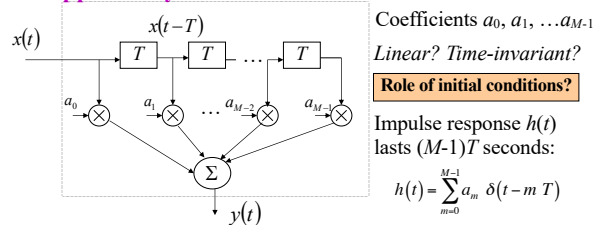
Linear? Time-invariant?

3-14

14

Continuous-Time System Properties

- Tapped delay line



$M-1$ delay blocks

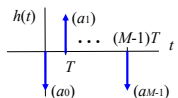
Coefficients a_0, a_1, \dots, a_{M-1}

Linear? Time-invariant?

Role of initial conditions?

Impulse response $h(t)$ lasts $(M-1)T$ seconds:

$$h(t) = \sum_{m=0}^{M-1} a_m \delta(t-mT)$$



$$y(t) = a_0 x(t) + a_1 x(t-T) + \dots + a_{M-1} x(t-(M-1)T)$$

$$y(t) = \sum_{m=0}^{M-1} a_m x(t-mT)$$

15

Continuous-Time System Properties

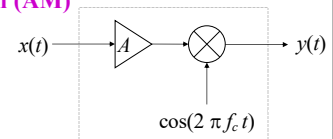
- Amplitude Modulation (AM)

$$y(t) = A x(t) \cos(2\pi f_c t)$$

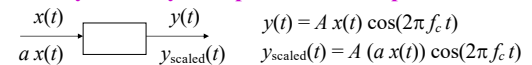
f_c carrier frequency

A is a constant

Linear? Time-invariant?



- Linearity: Does system pass all-zero input test?



$$y(t) = A x(t) \cos(2\pi f_c t)$$

$$y_{\text{scaled}}(t) = A (a x(t)) \cos(2\pi f_c t)$$

$$y_{\text{scaled}}(t) = a (A x(t) \cos(2\pi f_c t)) = a y(t) \text{ for all constants } a$$

- AM radio if $x(t) = 1 + k_a m(t)$ where $m(t)$ is audio to be broadcast and $|k_a m(t)| < 1$ (see lecture 19)

3-16

16

Review

Discrete-Time Signals

- Conversion of signals**
Sampling: Continuous-Time to Discrete-Time
Reconstruction: Discrete-Time to Continuous-Time

3 - 17

17

Review

Generating Discrete-Time Signals

- Many signals originate in continuous time**
Example: Talking on cell phone
- Sample continuous-time signal at equally-spaced points in time to obtain a sequence of numbers**
 $s[n] = s(n T_s)$ for $n \in \{\dots, -1, 0, 1, \dots\}$
 How to choose sampling period T_s ?
- Using a formula**
 Discrete-time impulse $\delta[n]$ on right
 How does $\delta[n]$ look in continuous time?

3 - 18

18

Aliasing Example

- Sample 30 kHz sinusoid using 48 kHz sampling rate**

3 - 19

19

Review

Discrete-Time System Properties

- Let $x_1[n]$, $x_2[n]$ and $x_3[n]$ be inputs to a linear system**
- Let $y_1[n]$, $y_2[n]$ and $y_3[n]$ be corresponding outputs**
- A linear system satisfies**
 Additivity: $x_1[n] + x_2[n] \Rightarrow y_1[n] + y_2[n]$
 Homogeneity: $a x[n] \Rightarrow a y[n]$ for any real/complex constant a
- For a time-invariant system, a shift of input signal by any integer-valued m causes same shift in output signal, i.e. $x[n - m] \Rightarrow y[n - m]$, for all m**
- Role of initial conditions?**

3 - 20

20

Why are LTI properties useful?

- An LTI system is uniquely characterized by its impulse response

Abstract away implementation details by providing an impulse response e.g. to hide intellectual property
Model an unknown system assumed to be LTI

- Fourier transform of the impulse response $h[n]$ is the frequency response of the system $H_{\text{freq}}(\omega)$

$$\begin{array}{l} \text{Time domain} \quad x[n] \xrightarrow{h[n]} y[n] = h[n] * x[n] \\ \text{Z domain} \quad X(z) \xrightarrow{H(z)} Y(z) = H(z) X(z) \\ \text{Frequency domain} \quad X_{\text{freq}}(\omega) \xrightarrow{H_{\text{freq}}(\omega)} Y_{\text{freq}}(\omega) = H_{\text{freq}}(\omega) X_{\text{freq}}(\omega) \end{array}$$

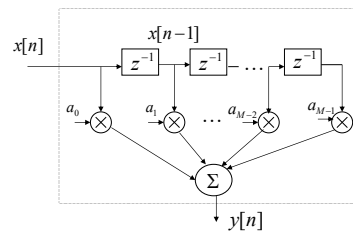
3 - 21

21

Discrete-Time System Properties

- Tapped delay line in discrete time

See also slide 5-4

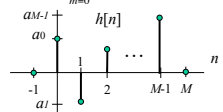


$M-1$ delay blocks where z^{-1} is delay of 1 sample:

$$y[n] = \sum_{m=0}^{M-1} a_m x[n-m]$$

Coefficients a_0, a_1, \dots, a_{M-1} are impulse response:

$$h[n] = \sum_{m=0}^{M-1} a_m \delta[n-m]$$



- Linear?
- Time-invariant?

Role of initial conditions?

22

Averaging Filter

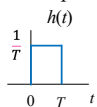
- Continuous time

Averages input signal over previous T seconds

$$y(t) = \frac{1}{T} \int_{t-T}^t x(t) dt$$

Linear? Time-invariant?

Impulse response:



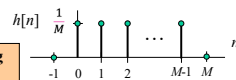
- Discrete time

Averages current and previous $M-1$ samples

$$y[n] = \frac{1}{M} \sum_{m=0}^{M-1} x[n-m]$$

Linear? Time-invariant?

Hint: Tapped delay line with $a_m = 1/M$ for m in $[0, M-1]$



See Designing Averaging Filters Handout

3 - 23

23

First-Order Difference Filter

- Continuous time

$$f(t) \xrightarrow{\frac{d}{dt}} y(t)$$

$$\begin{aligned} y(t) &= \frac{d}{dt} \{f(t)\} \\ &= \lim_{\Delta t \rightarrow 0} \frac{f(t) - f(t - \Delta t)}{\Delta t} \end{aligned}$$

Linear? Time-invariant?

- Discrete time

$$f[n] \xrightarrow{\frac{d}{dt}} y[n]$$

$$\begin{aligned} y[n] &= y(nT_s) = \left. \frac{d}{dt} \{f(t)\} \right|_{t=nT_s} \\ &= \lim_{T_s \rightarrow 0} \frac{f(nT_s) - f(nT_s - T_s)}{T_s} \\ &= f[n] - f[n-1] \end{aligned}$$

Linear? Time-invariant?

Hint: Tapped delay line with $a_0 = 1$ and $a_1 = -1$

See also slide 5-19

3 - 24

24

Sampling and Aliasing

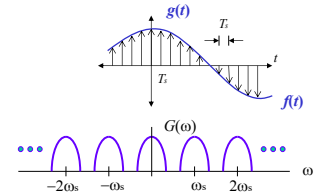
Prof. Brian L. Evans and Mr. Dan Jacobellis
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 4 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

- **Sampling**
 - Time domain views
 - Frequency domain view
 - Sampling theorem
- **Aliasing**
 - Sinusoidal example
 - Bandpass sampling
- **Conclusion**



4-2

2

Sampling – Time Domain View

Sampling

- **Analog-to-Digital Conversion**

Lowpass filter has stopband frequency less than $\frac{1}{2}f_s$ to reduce aliasing at sampler output (enforce sampling theorem)

Today Lecture 8
- **Sampling: Time-Domain Views**

Discrete-Time Output $x[n] = x(nT_s)$

Sampled Analog Output

Models opening/closing of switch as multiplication by impulse train

$$x_{sampled}(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

HW 0.3 4-3

3

Sampling – Frequency Domain View

Sampled Analog: Frequency Domain

- **Sampling replicates spectrum of continuous-time signal at integer multiples of sampling frequency**
- **Fourier series of impulse train $\delta_{T_s}(t)$ with period T_s**

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \frac{1}{T_s}(1 + 2 \cos(\omega_s t) + 2 \cos(2\omega_s t) + \dots)$$

$$g(t) = x(t) \delta_{T_s}(t) = \frac{1}{T_s}(x(t) + 2 x(t) \cos(\omega_s t) + 2 x(t) \cos(2\omega_s t) + \dots)$$

Modulation by $\cos(\omega_s t)$
Modulation by $\cos(2\omega_s t)$

How to recover $X(\omega)$?

gap if and only if $2\pi f_{max} < 2\pi f_s - 2\pi f_{max} \Leftrightarrow f_s > 2f_{max}$

HW 0.3 4-4

4

Sampling Theorem

- Continuous-time signal $x(t)$ with frequencies no higher than f_{max} can be reconstructed from its samples $x(n T_s)$ if samples taken at rate $f_s > 2 f_{max}$
- Nyquist rate = $2 f_{max}$
- Nyquist frequency = $f_s / 2$ maximum frequency captured
- **Unrealistic:** $x(t)$ has no frequency content above f_{max}
- What happens after sampling to $f_{max} = 1/2 f_s$?

$$x(t) = \cos(2 \pi f_{max} t)$$

$$x[n] = \cos\left(2 \pi \frac{f_{max}}{f_s} n\right) = \cos(\pi n) = (-1)^n$$

Highest DT Frequency

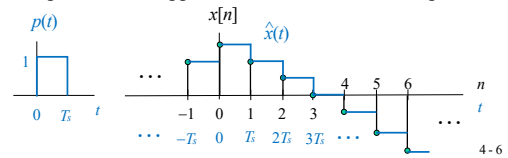
$$y(t) = \sin(2 \pi f_{max} t) \text{ gives } y[n] = \sin(\pi n) = 0$$

Aliasing

5

Sample-and-Hold Reconstruction

- Sampling theorem gives condition for which reconstruction is possible but not how to do it
 - Linear systems approaches for reconstruction
- Ideal lowpass filter with two-sided sinc impulse response
 No unique filtering approach for reconstruction in practice
 Sample-and-hold approach below has efficient implementation

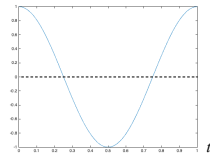


6

Sampling and Oversampling Demo

- As sampling rate increases above Nyquist rate, sampled waveform looks more like original
- Zero crossings: frequency content of a sinusoid

Distance between two zero crossings: one half period
 With sampling theorem satisfied, sampled sinusoid crosses zero right number of times per period
 In some applications, frequency content matters not shape of time-domain waveform



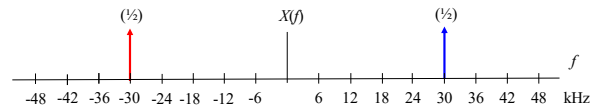
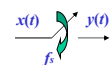
- DSP First, 2nd ed., ch. 4, sampling/interpolation [link](#)

7

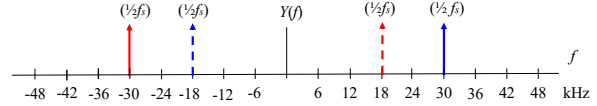
Aliasing: Sinusoidal Example

- Sample $x(t)$ at $f_s = 48$ kHz

$$x(t) = \cos(2 \pi f_0 t) \text{ where } f_0 = 30 \text{ kHz}$$



$Y(f)$ has $X(f)$ plus replicas of $X(f)$ at offsets of multiples of f_s



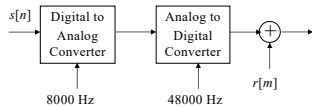
Effective freq. 18 kHz
 Reconstruction uses $-\frac{1}{2} f_s \leq f \leq \frac{1}{2} f_s$

See slides 3-19 & 4-4

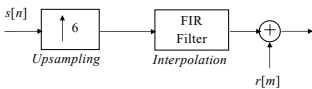
8

Increasing Sampling Rates

- **Consider adding speech clip to an audio track**
 Speech signal $s[n]$ is sampled at 8 kHz
 Audio signal $r[m]$ is sampled at 48 kHz
- **Inefficient approach: Interpolate in continuous time**



- **Efficient approach: Interpolate in discrete time**



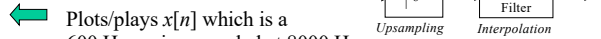
4 - 9

9

Increasing Sampling Rates

- **Upsampling by L**
 Copies input sample to output and appends $L-1$ zeros
 Output has L times as many samples as input samples

- **Audio Demonstration**



Plots/plays $x[n]$ which is a 600 Hz cosine sampled at 8000 Hz

Plots/plays $v[m]$: spectrum is spectrum of $x[n]$ plus $L-1$ replicas
 Interpolation filter fills in inserted zero values in time domain and attenuates replicas in frequency domain due to upsampling
 Rectangular, triangular and truncated sinc FIR filters used

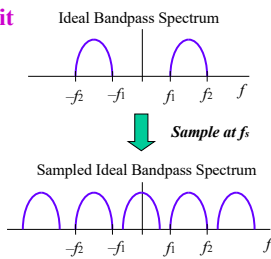
4 - 10

10

Aliasing

Bandpass Sampling

- **Uses aliasing to our benefit**
- **Reduce sampling rate**
 Bandwidth $f_2 - f_1$
 Sampling rate $f_s > f_2 - f_1$
 For replica to be centered at origin after sampling
 $f_{center} = \frac{1}{2}(f_1 + f_2) = k f_s$



- **Practical issues**

Sampling clock tolerance: $f_{center} = k f_s$
 Bandpass and lowpass filter designs
 Effects of noise

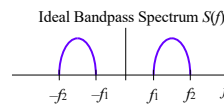
Lowpass filter to extract baseband

4 - 11

11

Aliasing

Bandpass Sampling Example



Bandpass sampling

Bandwidth $f_2 - f_1$
 $f_s > f_2 - f_1$
 $f_{center} = \frac{1}{2}(f_1 + f_2) = k f_s$

Extract IEEE 802.11a Wi-Fi 2.4 GHz Channel 1

$f_1 = 2.401$ GHz
 $f_2 = 2.423$ GHz
 Bandwidth = 0.022 GHz
 $f_c = 2.412$ GHz

Sampling theorem

$f_s > 2f_2$ e.g. $f_s = 4.86$ GHz

Bandpass sampling

$f_s = 0.036$ GHz with $k = 67$
 135x more efficient

12

Software Defined Radio

- Worldwide unlicensed microwave band at 2.4 GHz

Any service can use this band but must follow regulations on transmit power, out-of-band leakage, etc.

Services include Bluetooth, Wi-Fi, wireless mice/keyboards, ZigBee, baby monitors, wireless microphones/speakers

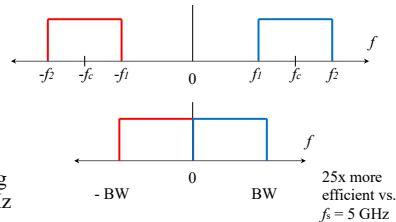
- Extract band for processing

$f_1 = 2.4$ GHz

$f_2 = 2.5$ GHz

BW = 0.1 GHz

Bandpass sampling with $f_s = 0.2$ GHz



13

Conclusion

- Sampling replicates spectrum of continuous-time signal at offsets that are integer multiples of sampling frequency
- Sampling theorem gives necessary condition to reconstruct the continuous-time signal from its samples, but does not say how to do it
- Aliasing occurs due to sampling
 - Noise present at all frequencies
 - A/D converter design tradeoffs to control impact of aliasing
- Bandpass sampling reduces sampling rate significantly by using aliasing to our benefit

4 - 14

14

Rolling Shutter Cameras

- Smart phone and point-and-shoot cameras

No (global) hardware shutter to reduce cost, size, weight

Light continuously impinges on sensor array

Artifacts due to relative motion between objects and camera

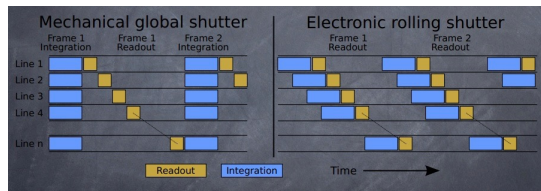


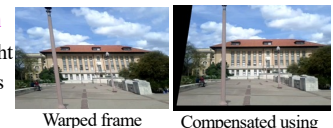
Figure from a tutorial by Forssen et al. at the 2012 IEEE Conf. on Computer Vision & Pattern Recognition

4 - 15

15

Rolling Shutter Artifacts

- Plucked guitar strings – global shutter camera → video
String vibration is (correctly) damped sinusoid vs. time
- “Guitar Oscillations Captured with iPhone 4” → video
← Rolling shutter (sampling) artifacts but not aliasing effects
- Fast camera motion
Pan camera fast left/right
Pole wobbles and bends
Building skewed



C. Jia and B. L. Evans, “Probabilistic 3-D Motion Estimation for Rolling Shutter Video Rectification from Visual and Inertial Measurements,” IEEE Multimedia Signal Proc. Workshop, 2012. Link to article. →

16

Finite Impulse Response Filters

Prof. Brian L. Evans
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 5 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

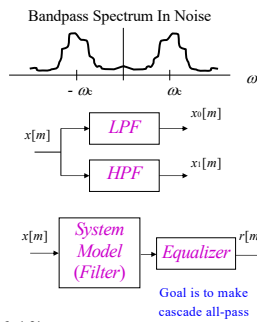
- Many Roles for Filters
- Convolution
- Z-transforms
- Linear time-invariant systems
 - Transfer functions
 - Frequency responses
- Finite impulse response (FIR) filters
 - Filter design
 - Cascading FIR filters demonstration
 - Linear phase

5-2

2

Many Roles for Filters

- **Noise removal**
 Bandpass filtering to suppress out-of-band noise (slide 1-5)
- **Spectral analysis**
 Two-channel filter bank on right
 Used in interpolation (slide 13-9)
- **Spectral shaping**
 Data conversion (lectures 10/11)
 Equalization (slides 6-15 & 16-7)
 Carrier frequency/phase recovery
 Timing recovery (slide 13-17 & 16-10)

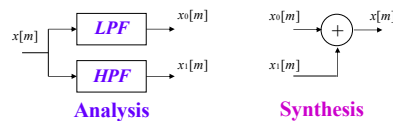


5-3

3

Spectral Analysis

- **Analysis: decompose signal into frequency bands**
Graphic equalizer: apply separate gain to each band
Compression: spend more bits on more important bands
- **Synthesis: combine bands into one signal**
- **Example: Two-band filter bank / wavelet**
 LPF – averaging: $x_0[m] = \frac{1}{2} x[m] + \frac{1}{2} x[m-1]$
 HPF – first-order difference: $x_1[m] = \frac{1}{2} x[m] - \frac{1}{2} x[m-1]$
 Exact reconstruction of $x[m]$ is possible through synthesis

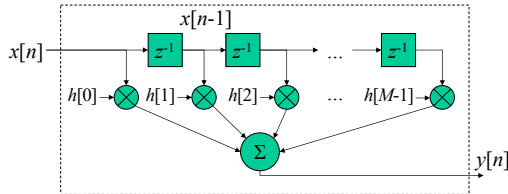


5-4

4

Finite Impulse Response (FIR) Filter

- Same as *discrete-time tapped delay line* (slide 3-22)



- Impulse response $h[n]$ has finite extent $n = 0, \dots, M-1$

$$y[n] = h[0]x[n] + h[1]x[n-1] + \dots + h[M-1]x[n-(M-1)]$$

$$y[n] = \sum_{m=0}^{M-1} h[m] x[n-m]$$

Discrete-time convolution

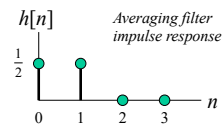
5-5

5

Review

Discrete-time Convolution Derivation

- Output $y[n]$ for input $x[n]$ $y[n] = T\{x[n]\}$
- Any signal can be decomposed into sum of discrete impulses $y[n] = T\left\{\sum_{m=-\infty}^{\infty} x[m]\delta[n-m]\right\}$
- Apply linearity properties $y[n] = \sum_{m=-\infty}^{\infty} x[m]T\{\delta[n-m]\}$
- Apply shift-invariance $y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$
- Apply change of variables $y[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$



$$y[n] = h[0]x[n] + h[1]x[n-1] = (x[n] + x[n-1]) / 2$$

5-6

6

Review

Convolution Comparison

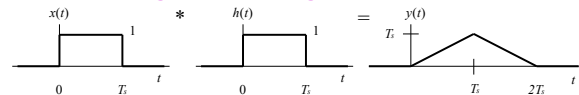
- Continuous-time convolution of $x(t)$ and $h(t)$**
 $y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda = \int_{-\infty}^{\infty} h(\lambda)x(t-\lambda)d\lambda$
 For each t , compute different (possibly) infinite integral
- In discrete-time, replace integral with summation**
 $y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$
 For each n , compute different (possibly) infinite summation
- LTI system**
 Characterized uniquely by its impulse response
 Its output is convolution of the input and its impulse response

5-7

7

Convolution Demos

- Signal Processing First Matlab Demonstrations**
 Continuous-Time <http://dspfirst.gatech.edu/matlab/#cconvdemo> →
 Discrete-Time <http://dspfirst.gatech.edu/matlab/#dconvdemo> →
- Convoluting two signals of finite lengths**
 Continuous-Time $y(t) = x(t) * h(t)$ $L_y = L_x + L_h$
 Discrete-Time $y[n] = x[n] * h[n]$ $L_y = L_x + L_h - 1$
- Convoluting two causal signals?**



What about convoluting two pulses of different lengths? See Appendix E

8

Z-transform Definition

- For discrete-time systems, z -transforms play same role as Laplace transforms do in continuous-time

Bilateral Forward z -transform

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n}$$

Bilateral Inverse z -transform

$$h[n] = \frac{1}{2\pi j} \oint_R H(z) z^{-n+1} dz$$

Inverse transform requires contour integration over closed contour (region) R

Contour integration covered in a Complex Analysis course

- Compute forward and inverse transforms using transform pairs and properties

5-9

9

Three Common Z-transform Pairs

- $h[n] = \delta[n]$

$$H(z) = \sum_{n=-\infty}^{\infty} \delta[n]z^{-n} = \sum_{n=0}^{\infty} \delta[n]z^{-n} = 1$$

Region of convergence: entire z -plane

- $h[n] = \delta[n-1]$

$$H(z) = \sum_{n=-\infty}^{\infty} \delta[n-1]z^{-n} = \sum_{n=1}^{\infty} \delta[n-1]z^{-n} = z^{-1}$$

Region of convergence: entire z -plane except $z = 0$

$$h[n-1] \Leftrightarrow z^{-1} H(z)$$

Finite extent sequences

- $h[n] = a^n u[n]$

$$H(z) = \sum_{n=-\infty}^{\infty} a^n u[n]z^{-n}$$

$$= \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{a}{z}\right)^n$$

$$= \frac{1}{1 - \frac{a}{z}} \quad \text{if } \left|\frac{a}{z}\right| < 1$$

Region of convergence for summation: $|z| > |a|$

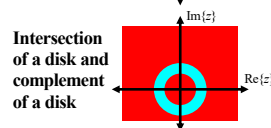
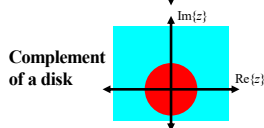
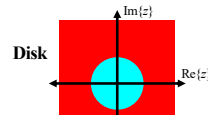
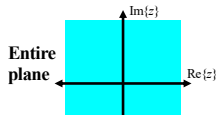
$|z| > |a|$ is the complement of a disk (see next slide)

Infinite extent sequence

10

Region of Convergence

- Region of the complex z -plane for which forward z -transform converges
- Four possibilities ($z = 0$ is special case that may or may not be included)



5-11

11

System Transfer Function

- Z-transform of system's impulse response**
Impulse response uniquely represents an LTI system
- Example: FIR filter with M coefficients (slide 5-5)**
Transfer function $H(z)$ is polynomial in powers of z^{-1}
Region of convergence (ROC) is entire z -plane except $z = 0$
- Since ROC includes unit circle, substitute $z = e^{j\omega}$ into transfer function to obtain frequency response**

$$H(e^{j\omega}) = H(z) \Big|_{z=e^{j\omega}} = \sum_{n=0}^{M-1} h[n]e^{-j\omega n} = h[0] + h[1]e^{-j\omega} + \dots + h[M-1]e^{-j(M-1)\omega}$$

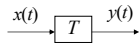
5-12

12

Example: Ideal Delay

Continuous Time

LTI delay by T seconds



$$y(t) = x(t - T)$$

Impulse response

$$h(t) = \delta(t - T)$$

Frequency response

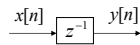
$$H(\Omega) = e^{-j\Omega T}$$

$$|H(\Omega)| = 1$$

$$\angle H(\Omega) = -\Omega T$$

Discrete Time

LTI delay by 1 sample



$$y[n] = x[n - 1]$$

Impulse response

$$h[n] = \delta[n - 1]$$

Frequency response

$$H(\omega) = e^{-j\omega}$$

$$|H(\omega)| = 1$$

$$\angle H(\omega) = -\omega$$

5-13

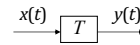
$\omega = \Omega T$
Pass Filter
Linear Phase

13

Phase Shift in Time Domain

Continuous Time

LTI delay by T seconds



$$y(t) = x(t - T)$$

Sinusoidal input signal

$$x(t) = \cos(2\pi f_0 t)$$

Sinusoidal output signal

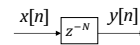
$$y(t) = \cos(2\pi f_0(t - T))$$

$$y(t) = \cos(2\pi f_0 t - 2\pi f_0 T)$$

Phase shift

Discrete Time

LTI delay by N samples



$$y[n] = x[n - N]$$

Sinusoidal input signal

$$x[n] = \cos(\omega_0 n)$$

Sinusoidal output signal

$$y[n] = \cos(\omega_0(n - N))$$

$$y[n] = \cos(\omega_0 n - \omega_0 N)$$

5-14

14

Linear Time-Invariant Systems

• Fundamental Theorem of Linear Systems

If a complex sinusoid were input into an LTI system, then output would be input scaled by frequency response of LTI system (evaluated at complex sinusoidal frequency)

Scaling may attenuate complex sinusoid and shift it in phase (complex sinusoids are eigenfunctions of LTI systems)

Continuous-time example: see handout F (lowpass RC filter)

Discrete-time derivation: Input $x[n] = e^{j\omega n}$ into LTI system

$$y[n] = \sum_{m=-\infty}^{\infty} \underbrace{e^{j\omega(n-m)}}_{x[n] * h[n]} h[m] = e^{j\omega n} \sum_{m=-\infty}^{\infty} \underbrace{h[m] e^{-j\omega m}}_{H(\omega)} = e^{j\omega n} H(\omega)$$

$H(\omega)$ is frequency response (DT Fourier transform of $h[n]$)

15

Frequency Response

• **Continuous-time LTI system** $\frac{e^{j\Omega t}}{\cos(\Omega t)} \rightarrow \boxed{h(t)} \rightarrow \frac{H(j\Omega) e^{j\Omega t}}{|H(j\Omega)| \cos(\Omega t + \angle H(j\Omega))}$

• **Discrete-time LTI system** $\frac{e^{j\omega n}}{\cos(\omega n)} \rightarrow \boxed{h[n]} \rightarrow \frac{H(e^{j\omega}) e^{j\omega n}}{|H(e^{j\omega})| \cos(\omega n + \angle H(e^{j\omega}))}$

• For real-valued impulse response $H(e^{-j\omega}) = H^*(e^{j\omega})$

Input $e^{-j\omega n} + e^{j\omega n} = 2\cos(\omega n)$

Output $H(e^{-j\omega})e^{-j\omega n} + H(e^{j\omega})e^{j\omega n} = H^*(e^{j\omega})e^{-j\omega n} + H(e^{j\omega})e^{j\omega n} =$

$$|H(e^{j\omega})| e^{-j\angle H(e^{j\omega})} e^{-j\omega n} + |H(e^{j\omega})| e^{j\angle H(e^{j\omega})} e^{j\omega n} =$$

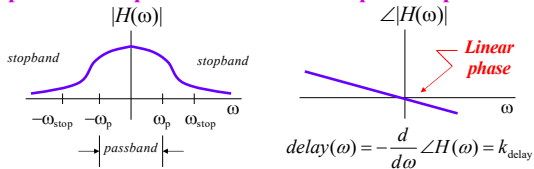
$$2|H(e^{j\omega})| \cos(\omega n + \angle H(e^{j\omega}))$$

5-16

16

Frequency Response

- System response to complex sinusoid $e^{j\omega n}$ for all possible frequencies ω in radians per sample:



Lowpass filter: passes low and attenuates high frequencies
 Linear phase / constant group delay: FIR filter whose impulse response is symmetric or anti-symmetric about its midpoint

- Not all FIR filters exhibit linear phase

5-17

17

FIR Filter Design

- Specify desired piecewise constant magnitude resp.

- Lowpass filter example

$$\omega \in [0, \omega_{\text{pass}}], \text{mag}_{\text{dB}} \in [-A_{\text{pass}}/2, A_{\text{pass}}/2]$$

$$\omega \in [\omega_{\text{stop}}, \pi], \text{mag}_{\text{dB}} \leq A_{\text{stop}}$$

Transition band unspecified but should not amplify

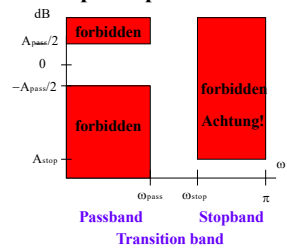
- Symmetric (linear phase) FIR filter design methods

Windowing

Least squares

Remez (Parks-McClellan)

Lowpass Specification



$$\begin{aligned} A_{\text{pass}} & \text{ passband tolerance (dB)} \\ A_{\text{stop}} & \text{ stopband attenuation (dB)} \\ \Delta_{\text{dB}} & = 20 \log_{10}(\Lambda) \end{aligned}$$

5-18

18

Example: Two-Tap Averaging Filter

- Input-output relationship

$$y[n] = \frac{1}{2}x[n] + \frac{1}{2}x[n-1]$$

- Impulse response

$$h[n] = \frac{1}{2}\delta[n] + \frac{1}{2}\delta[n-1]$$

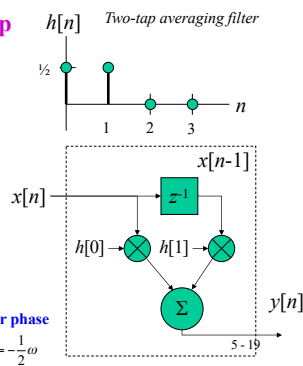
- Frequency response

$$H(\omega) = \frac{1}{2} + \frac{1}{2}e^{-j\omega}$$

$$H(\omega) = \frac{1}{2}e^{-j\frac{1}{2}\omega} \left(e^{+j\frac{1}{2}\omega} + e^{-j\frac{1}{2}\omega} \right)$$

$$H(\omega) = \cos\left(\frac{\omega}{2}\right) e^{-j\frac{1}{2}\omega}$$

magnitude $\angle H(\omega) = -\frac{1}{2}\omega$ linear phase



5-19

19

Example: First-Order Difference

- Input-output relationship

$$y[n] = \frac{1}{2}x[n] - \frac{1}{2}x[n-1]$$

- Impulse response

$$h[n] = \frac{1}{2}\delta[n] - \frac{1}{2}\delta[n-1]$$

- Frequency response

$$H(\omega) = \frac{1}{2} - \frac{1}{2}e^{-j\omega}$$

$$H(\omega) = \frac{1}{2}e^{-j\frac{1}{2}\omega} \left(e^{+j\frac{1}{2}\omega} - e^{-j\frac{1}{2}\omega} \right)$$

$$H(\omega) = \sin\left(\frac{\omega}{2}\right) j e^{-j\frac{1}{2}\omega} = \sin\left(\frac{\omega}{2}\right) e^{j\frac{\pi}{2}} e^{-j\frac{1}{2}\omega} = \sin\left(\frac{\omega}{2}\right) e^{j\left(\frac{\pi}{2} - \frac{1}{2}\omega\right)}$$

generalized linear phase

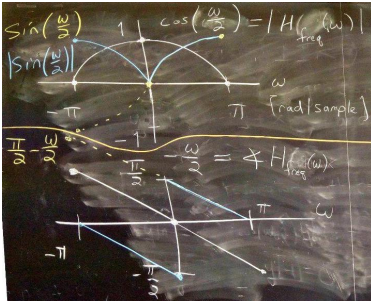
amplitude

5-20

20

Complementary Frequency Responses

Magnitude response



Phase response

Averaging filter (lowpass) in white
 First-order difference filter (highpass) in yellow/blue
 First-order difference has π phase shift at zero mag response frequency

21

Cascading FIR Filters Demo

- **Five-tap discrete-time averaging FIR filter with input $x[n]$ and output $y[n]$**

$$y[n] = x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4]$$

Standard averaging filtering scaled by 5

Lowpass filter (smooth/blur input signal)

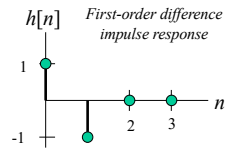
Impulse response of $\{1, 1, 1, 1, 1\}$ removes multiplications

- **First-order difference FIR filter**

$$y[n] = x[n] - x[n-1]$$

Highpass filter (sharpen input signal)

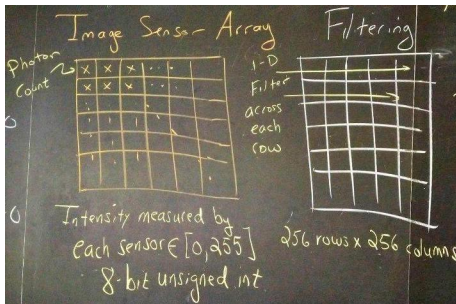
Impulse response is $\{1, -1\}$ removes multiplications



5-22

22

Cascading FIR Filters Demo



Apply 1-D five-tap filter across each row to produce 256 x 260 image and then apply the same 1-D five-tap filter down each column of 256 x 260 image to produce a 260 x 260 image

23

Cascading FIR Filters Demo

- **DSP First, 2nd ed, ch. 6, cascading FIR Filters** →

<http://dspfirst.atech.edu/chapters/06/firfwdemos/blockd/index.html>

From lowpass filter to highpass filter:

original image → blurred image → sharpened/blurred image

From highpass to lowpass filter:

original image → sharpened image → blurred/sharpened image

- **Frequencies that are zeroed out can never be recovered (e.g. DC is zeroed out by highpass filter)**
- **Order of two LTI systems in cascade can be switched under the assumption that computations are performed in exact precision**

5-24

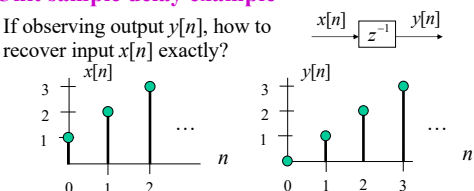
24

Cascading FIR Filters Demo

- **Input image is 256 x 256 matrix**
Each pixel represented by eight-bit number in [0, 255]
0 is black and 255 is white for monitor display
- **Each filter applied along row then column**
Averaging filter adds five numbers to create output pixel which requires 3 extra bits (worst case) for each pass
Difference filter subtracts two numbers to create output pixel which requires 1 extra bit (worst case) for each pass
- **Full output precision 16 bits/pixel is maintained**
Demonstration uses double-precision floating-point data and arithmetic (53 bits of mantissa + sign; 11 bits for exponent)
5 - 25

25

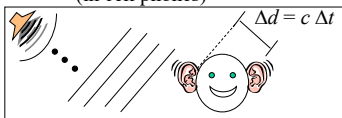
Cascading FIR Filters Demo

- **How to compare images pixel by pixel?**
Input image is 256 x 256 and output image is 261 x 261
- **Unit sample delay example**
If observing output $y[n]$, how to recover input $x[n]$ exactly?

- **FIR filters in demo have constant group delay**
Remove first 2 rows/columns and last 3 rows/columns 5 - 26

26

Importance of Linear Phase


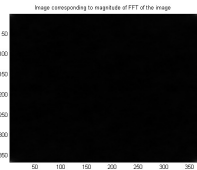
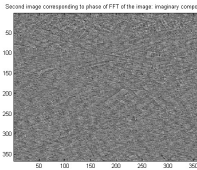
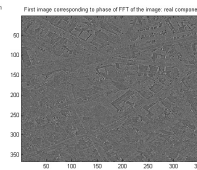
- **Speech signals**
Use phase differences in arrival to locate speaker
Once speaker is located, ears are relatively insensitive to phase distortion in speech from that speaker
Used in speech compression (in cell phones)
- **Linear phase crucial**
Audio
Images
Communication systems
- **Linear phase response**
Need FIR filters
Realizable IIR filters cannot achieve linear phase response over all frequencies




5 - 27

27

Importance of Linear Phase

- **For images, vital visual information in phase** 
- | | | |
|--|---|---|
|  |  |  |
| Image corresponding to magnitude of FFT of the image | Second image corresponding to phase of FFT of the image: imaginary component | First image corresponding to phase of FFT of the image: real component |
| Take FFT of image
Set phase to zero
Take inverse FFT (which is real-valued) | Take FFT of image
Set magnitude to one
Take inverse FFT
Keep imaginary part | Take FFT of image
Set magnitude to one
Take inverse FFT
Keep real part |

- **Original image is from Matlab** 

http://users.ecc.utexas.edu/~bevans/courses/realtime/lectures/05_FIR_Filters/Original%20Image.tif 5 - 28

28

Takeaways

- **Linear time-invariant systems**
 - Uniquely defined by its impulse response
 - Do not create new frequencies
 - System must be initially “at rest” (zero initial conditions)
- **Some FIR filters have linear phase**
 - FIR coefficients (impulse response) must be either even symmetric or odd symmetric about its midpoint
 - Constant group delay in samples for all input frequencies
- **Useful discrete-time FIR filters**
 - Averaging*: Lowpass, linear phase
 - Make all coefficients 1 for efficient multiplier-free version
 - First-order difference*: highpass, linear phase

5 - 29

29

Infinite Impulse Response Filters

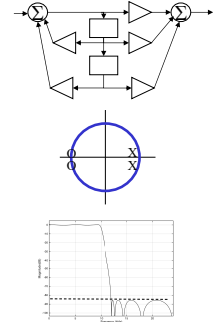
Prof. Brian L. Evans
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 6 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

- IIR biquad filter structures
- Stability
 - Bounded-Input Bounded Output
 - Equalization Example
- IIR filter design
- IIR filter implementation
- Filter design demos
- Conclusion



6-2

2

Discrete-Time IIR Filters

- Infinite Impulse Response (IIR) filter has impulse response of infinite duration, e.g.

$$h[n] = \left(\frac{1}{2}\right)^n u[n] \xleftrightarrow{Z} H(z) = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{1}{2}z^{-1}\right)^n = 1 + \frac{1}{2}z^{-1} + \dots = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

- How to implement the IIR filter by computer?

Let $x[n]$ be the input signal and $y[n]$ the output signal,

$$H(z) = \frac{Y(z)}{X(z)} \quad Y(z) = \frac{1}{1 - \frac{1}{2}z^{-1}} X(z) \quad y[n] - \frac{1}{2}y[n-1] = x[n]$$

$$Y(z) = H(z)X(z) \quad Y(z) - \frac{1}{2}z^{-1}Y(z) = X(z) \quad y[n] = \frac{1}{2}y[n-1] + x[n]$$

Recursively compute output $y[n]$, $n \geq 0$, given $y[-1]$ and $x[n]$ 6-3

3

Many Equivalent Representations

- Difference equation
- Block diagram representation

$$y[n] = \frac{1}{2}y[n-1] + \frac{1}{8}y[n-2] + x[n]$$

Recursive computation needs $y[-1]$ and $y[-2]$

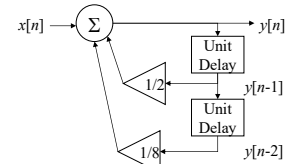
For the filter to be LTI, $y[-1] = 0$ and $y[-2] = 0$

- Transfer function

Assumes LTI system

$$Y(z) = \frac{1}{2}z^{-1}Y(z) + \frac{1}{8}z^{-2}Y(z) + X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2}z^{-1} - \frac{1}{8}z^{-2}} \quad \text{Poles at } -0.183 \text{ and } +0.683 \quad 6-4$$

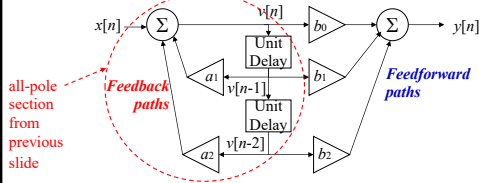


Second-order filter section (a.k.a. biquad) with 2 poles and 0 non-trivial zeros

4

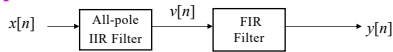
IIR Biquad Block Diagrams

- **Biquad: short for *biquadratic***
Transfer function is ratio of two quadratic polynomials
- **Two poles, and zero, one or two non-trivial zeros**



all-pole section from previous slide

- **Express above as cascade of two familiar filters**

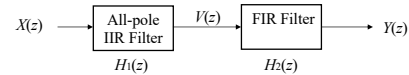


6-5

5

IIR Biquad Transfer Function

- **Assume LTI system (zero initial conditions)**



$$V(z) = H_1(z) X(z) \text{ and } Y(z) = H_2(z) V(z)$$

$$Y(z) = H_1(z) H_2(z) X(z) \text{ and hence } H(z) = H_1(z) H_2(z)$$

- **Overall transfer function**

$$H(z) = \frac{Y(z)}{X(z)} = \left(\frac{V(z)}{X(z)} \right) \left(\frac{Y(z)}{V(z)} \right) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

Real a_1, a_2 : poles are conjugate symmetric ($\alpha \pm j\beta$) or real

Real b_0, b_1, b_2 : zeros are conjugate symmetric or real

6-6

6

IIR Biquad Difference Equation

- **Derive difference equation from transfer function**

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

$$(1 - a_1 z^{-1} - a_2 z^{-2})Y(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2})X(z)$$

Take inverse z-transform of both sides:

$$y[n] - a_1 y[n-1] - a_2 y[n-2] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

Collect terms to isolate $y[n]$ on the left-hand side:

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

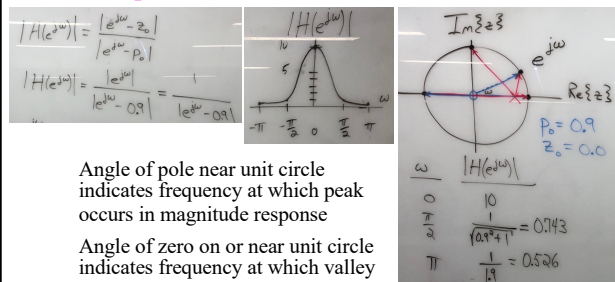
How many multiplications and additions per output sample?

- **How to convert biquad to a first-order IIR section?**

7

First-Order Frequency Response

- **Pole at $p_0 = 0.9$ and zero at $z_0 = 0$**



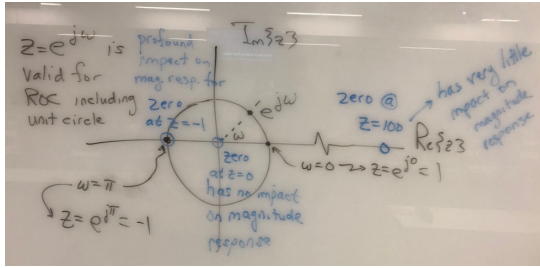
Angle of pole near unit circle indicates frequency at which peak occurs in magnitude response

Angle of zero on or near unit circle indicates frequency at which valley occurs in magnitude response

6-8

8

Impact of Zeros on Magnitude Resp.



$$|H(e^{j\omega})| = |e^{j\omega} - z_0| |e^{j\omega} - z_1| |e^{j\omega} - z_2| \dots$$

6-9

9

IIR Biquad Frequency Responses

Magnitude response

$$H(z) = C \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)}$$

Zeros z_0 & z_1 and poles p_0 & p_1

$|a - b|$ is distance between complex numbers a and b

$$|H(e^{j\omega})| = \left| C \frac{(e^{j\omega} - z_0)(e^{j\omega} - z_1)}{(e^{j\omega} - p_0)(e^{j\omega} - p_1)} \right|$$

$|e^{j\omega} - p_0|$ is distance from point on unit circle $e^{j\omega}$ and pole p_0

$$|H(e^{j\omega})| = |C| \frac{|e^{j\omega} - z_0| |e^{j\omega} - z_1|}{|e^{j\omega} - p_0| |e^{j\omega} - p_1|}$$

Normalizing the magnitude response

C is a degree of freedom and $|C|$ scales magnitude response

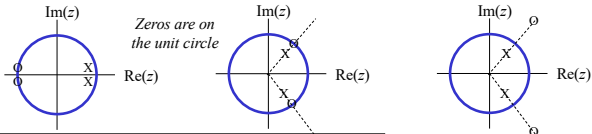
$$H(z) = C \frac{z^2 - (z_0 + z_1)z + z_0 z_1}{z^2 - (p_0 + p_1)z + p_0 p_1}$$

Normalize at DC: Set $H(z) = 1$ at $z = e^{j\omega_0} = e^{j0} = 1$

6-10

10

Frequency Response?



Lowpass, highpass, bandpass, bandstop, allpass, notch?

```

zeroAngle = 15*pi/16;
z0 = exp(j*zeroAngle);
z1 = exp(-j*zeroAngle);
numer = [1 -(z0+z1) z0*z1];

r = 0.9;
poleAngle = pi/16;
p0 = r * exp(j*poleAngle);
p1 = r * exp(-j*poleAngle);
denom = [1 -(p0+p1) p0*p1];

% Normalization
C = (denom * [1 1 1]) / (numer * [1 1 1]);
freqz(C*numer, denom);

zeroAngle = pi/4;
z0 = exp(j*zeroAngle);
z1 = exp(-j*zeroAngle);
numer = [1 -(z0+z1) z0*z1];

r = 0.9;
poleAngle = pi/4;
p0 = r * exp(j*poleAngle);
p1 = r * exp(-j*poleAngle);
denom = [1 -(p0+p1) p0*p1];

% Normalization
C = (denom * [1 1 1]) / (numer * [1 1 1]);
freqz(C*numer, denom);

zeroAngle = pi/4;
z0 = 1/0.8;
z1 = r0*exp(-j*zeroAngle);
numer = [1 -(z0+z1) z0*z1];

r = 1/zz;
poleAngle = pi/4;
p0 = r * exp(j*poleAngle);
p1 = r * exp(-j*poleAngle);
denom = [1 -(p0+p1) p0*p1];

% Normalization
C = (denom * [1 1 1]) / (numer * [1 1 1]);
freqz(C*numer, denom);
    
```

11

Demonstrations

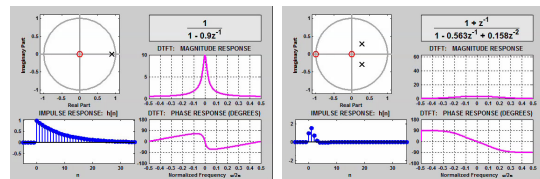
DSP First, 2nd ed., ch. 10



IIR filtering tutorial ([Link](#))

Connection between the Z and frequency domains ([Link](#))

Time/frequency/z domain movies for IIR Filters ([Link](#))



IIR filter with one pole and a zero at the origin

Movement of Poles

6-12

12

Stability

- A discrete-time LTI system is **bounded-input bounded-output (BIBO) stable** if for any bounded input $x[n]$ such that $|x[n]| \leq B_1 < \infty$, then the filter response $y[n]$ is also bounded $|y[n]| \leq B_2 < \infty$
- **Proposition:** A discrete-time filter with an impulse response of $h[n]$ is BIBO stable if and only if

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$



Every finite impulse response LTI system (even after implementation) is BIBO stable

A causal infinite impulse response LTI system is BIBO stable if and only if its poles lie inside the unit circle

13

BIBO Stability

- **Rule #1:** For a causal sequence, poles are inside the unit circle (applies to z -transform functions that are ratios of two polynomials) OR
- **Rule #2:** Unit circle is in the region of convergence. (In continuous-time, imaginary axis would be in region of convergence of Laplace transform.)
- **Example:** $a^n u[n] \leftrightarrow \frac{1}{1 - a z^{-1}}$ for $|z| > |a|$

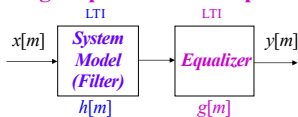
Stable if $|a| < 1$ by rule #1 or equivalently

Stable if $|a| < 1$ by rule #2 because $|z| > |a|$ and $|a| < 1$

14

Equalization

- Design equalizer to compensate frequency distortion



$$Y(z) = H(z) G(z) X(z)$$

$$\frac{Y(z)}{X(z)} = H(z) G(z)$$

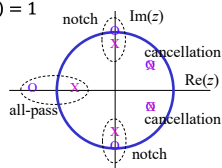
Applications: Audio, image and communication systems

Goal: Make cascade all-pass: $H(z) G(z) = 1$

- **Example:** $H(z)$ is an FIR filter

$$G(z) = \frac{1}{H(z)} \quad \text{Stability issues?}$$

Zeros of $H(z)$ become poles of $G(z)$



15

IIR Filter Design

- Classical IIR filter design algorithms

Design Method	Passband	Stopband
Butterworth	Monotonic	Monotonic
Chebyshev type I	Monotonic	Ripples
Chebyshev type II	Ripples	Monotonic
Elliptic	Ripples	Ripples

- Manual placement of poles and zeros
- Automated adaptation of pole and zero locations
Center frequency of bandpass resonator HW 1.1(d) & 2.1(d)
Notch filter with parameterized notch frequency HW 3.1(c)

16

IIR Filter Implementation

- **Cascade of biquads for N poles**
 N even: $N/2$ conjugate pairs of poles. Cascade of $N/2$ biquads.
 N odd: One real pole and $(N-1)/2$ conjugate pairs of poles.
 Cascade of first-order filter (real pole) and $(N-1)/2$ biquads
 For M biquads, $M!$ orderings of conjugate pairs of poles and for each ordering, $M!$ orderings of conjugate pairs of zeros
Exhaustive search: simulate all combinations to pick best one
Quick pairing: Order biquads in ascending order of quality factors (see next slide) and for each each biquad, choose the conjugate zero pair that is closest in Euclidean distance
- **Single section (single difference equation)**
 Could make a BIBO stable filter become BIBO unstable

17

Quality Factors for Digital Biquads

- **Measures sensitivity in pole locations to perturbations**
- **For poles at $a \pm j b = r e^{\pm j \theta}$, where $r = \sqrt{a^2 + b^2}$ is pole radius ($r < 1$ for stability), with $y = -2 a$:**

$$Q = \frac{\sqrt{(1+r^2)^2 - y^2}}{2(1-r^2)} \text{ where } \frac{1}{2} \leq Q < \infty$$

Real poles: $b = 0$ and $-1 < a < 1$, so $r = |a|$ and $y = \pm 2 a$ and $Q = 1/2$ (impulse response is $C_0 a^n u[n] + C_1 n a^n u[n]$)

Poles on unit circle: $r = 1$ so $Q = \infty$ (oscillatory response)

Imaginary poles: $a = 0$ so $r = |b|$ and $y = 0$, and $Q = \frac{1}{2} \frac{1+r^2}{1-r^2} = \frac{1}{2} \frac{1+b^2}{1-b^2}$

Processors with 16-bit x 16-bit multipliers and 40-bit accumulators: $Q_{\max} \approx 40$

Filter design programs use r as approximation of quality factor

18

Quality Factors for Analog Biquads

- **Second-order filter section**
 Conjugate symmetric poles $a \pm j b$, where $a < 0$
 With no zeros, impulse response is $h(t) = C e^{at} \cos(b t + \theta)$
 which is pure decay when $b = 0$ and pure sinusoid when $a = 0$
- **Quality factor: measures sensitivity of pole locations to perturbations**
 $Q = \frac{\sqrt{a^2 + b^2}}{-2a}$
 Real poles: $b = 0$ so $Q = 1/2$ (exponential decay response)
 Imaginary poles: $a = 0$ so $Q = \infty$ (oscillatory response)
 Maximum Q values: 25 for board-level RC circuits, 40 for switched capacitor designs, and 80 for analog IC designs
- **Classical designs have biquads with high Q factors**

6-19

19

IIR Filter as Single Section

- **IIR filters have rational transfer functions**

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 - a_1 z^{-1} - \dots - a_M z^{-M}} \Rightarrow Y(z) \left(1 - \sum_{m=1}^M a_m z^{-m} \right) = X(z) \sum_{k=0}^N b_k z^{-k}$$

- **Direct form realization**

Dot product of vector of $N+1$ coefficients and vector of current input and previous N inputs (FIR section)

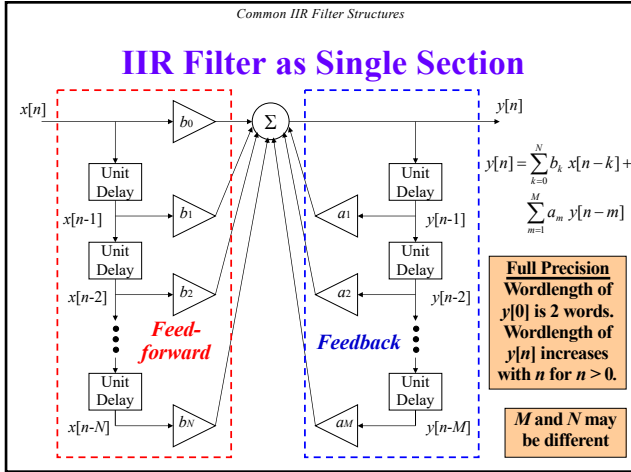
Dot product of vector of M coefficients and vector of previous M outputs ("FIR" filtering of previous output values)

Computation: $M + N + 1$ multiply-accumulates (MACs)

Memory: $M + N$ words for previous inputs/outputs and $M + N + 1$ words for coefficients

6-20

20



21

Stability As Single Section

- **IIR filter design to meet a specification**
Manual or automatic placement of poles and zeros
Poles are inside unit circle for stability
- **Factored transfer function in poles, zeros and gain**
Expanded to obtain coefficients for difference equation
Expansion for feedback coefficients can lead to instability
- **Second-order example: zeros z_0 & z_1 , poles p_0 & p_1**
Each addition and multiplication in 64-bit floating point

$$H(z) = b_0 \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} = b_0 \frac{1 - (z_0 + z_1)z^{-1} + z_0 z_1 z^{-2}}{1 - (p_0 + p_1)z^{-1} + p_0 p_1 z^{-2}}$$

worst-case loss of one bit
worst-case loss of 53 bits

6-22

22

Stability As Single Section

19 x 53 = 1007 mantissa bits lost due to truncating each pairwise multiplication result to 64-bit IEEE floating-point format (with 53 mantissa bits and 11 exponent bits)

See In-Lecture Assignment #1 on IIR filter design and analysis

6-23

23

Cascade of Biquads

MATLAB Filter Designer Demo #1

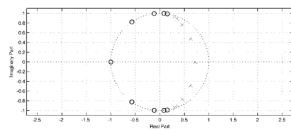
- **Filter design/analysis**
- **Lowpass filter design specification (all demos)**
fpass = 9600 Hz
fstop = 12000 Hz
fsampling = 48000 Hz
Apass = 1 dB
Astop = 80 dB
- **FIR filter – equiripple**
Also called Remez Exchange or Parks-McClellan design
Minimum order is 50
Change Wstop to 80
Order 100 gives Astop 100 dB
Order 200 gives Astop 175 dB
Order 300 does not converge – how to get higher order filter?
- **FIR filter – Kaiser window**
Minimum order 101 meets spec
- **Under analysis menu**
Show magnitude, phase and group delay responses

24

MATLAB Filter Designer Demo #2

- IIR filter – elliptic**
 - Use second-order sections
 - Filter order of 8 meets spec
 - Achieved Astop of ~80 dB
 - Poles/zeros separated in angle
 - Zeros on or near unit circle indicate stopband
 - Poles near unit circle indicate passband
 - Two poles very close to unit circle but still BIBO stable
 - Show group delay response

- IIR filter – elliptic**
 - Use second-order sections
 - Increase filter order to 9
 - Eight complex symmetric poles and one real pole:



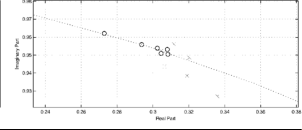
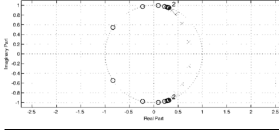
Same observations on left

6 - 25

25

MATLAB Filter Designer Demo #3

- IIR filter – elliptic**
 - Use second-order sections
 - Increase filter order to 20
 - Two poles very close to unit circle but BIBO stable
- Single section (Edit menu)**
 - Oscillation frequency ~9 kHz appears in passband
 - BIBO unstable: two pairs of poles outside unit circle

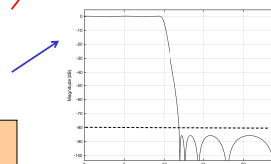
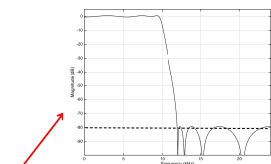


IIR filter design algorithms return poles-zeroes-gain (PZK format): Impact on response when expanding polynomials in transfer function from factored to unfactored form (see slide 6-25)

26

MATLAB Filter Designer Demo #4

- IIR filter - constrained least pth-norm design**
 - Use second-order sections
 - Limit pole radii ≤ 0.92
 - Order 8 does **not** meet both passband/stopband specs for different weights
 - Order 9 does indeed meet specs using $W_{pass} = 1.5$ and $W_{stop} = 800$



Filter order might increase but worth higher complexity for being more robust perturbations in pole locations

27

Conclusion

	FIR Filters	IIR Filters
Implementation complexity (1)	Higher	Lower (sometimes by factor of four)
Minimum order design algorithm	Parks-McClellan (Remez exchange) algorithm (2)	Elliptic design algorithm
BIBO stable?	Always	May become unstable when implemented (3)
Linear phase	If impulse response is symmetric or anti-symmetric about midpoint	No, but phase may be made approximately linear over passband (or other band)

- (1) For same piecewise constant magnitude specification
- (2) Algorithm to estimate minimum order for Parks-McClellan algorithm by Jim Kaiser may be off by 10%. Search for minimum order is often needed.
- (3) Choice of IIR filter structure matters -- use cascade of biquads instead of a single section. Some IIR filter design algorithms can constrain pole locations.

28

Conclusion

- **Keep poles computed by filter design algorithms**
Polynomial deflation (rooting) reliable in floating-point
Polynomial inflation (expansion) may degrade roots
- **Direct form IIR structures expand zeros and poles**
May become unstable for large order filters (order > 12) due to degradation in pole locations from polynomial expansion
- **Cascade of biquads (second-order sections)**
Only poles and zeros of second-order sections expanded
Biquads placed in order of ascending quality factors
Optimal ordering of biquads requires exhaustive search and each ordering can be simulated in parallel for speed

6-29

29

$$y[n] = x[n] * h[n]$$

$$y[n] = \sum_{m=-\infty}^{\infty} h[m] x[n-m]$$

Derivation of
 $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$
 condition

$$\begin{aligned} |y[n]| &= \left| \sum_{m=-\infty}^{\infty} h[m] x[n-m] \right| \\ &\leq \sum_{m=-\infty}^{\infty} |h[m] x[n-m]| \\ &= \sum_{m=-\infty}^{\infty} |h[m]| |x[n-m]| \end{aligned}$$

Since $|x[n]| \leq B_1$,

$$|y[n]| \leq B_1 \sum_{m=-\infty}^{\infty} |h[m]|$$

EE 445S
 Real-Time DSP Lab
 Prof. Brian L. Evans
 The Univ. of Texas at Austin
 Spring 2014

Bounded-Input
 Bounded-Output
 Stability of a
 Linear Time-Invariant
 System. Let input
 $|x[n]| \leq B_1$ for all n .

Special case: FIR Filter that has M coefficients,

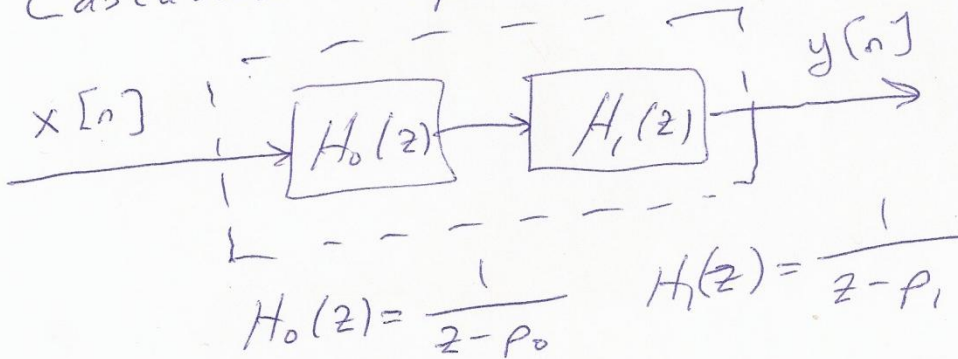
$$\sum_{m=-\infty}^{\infty} |h[m]| = \sum_{m=0}^{M-1} |h[m]| < \infty$$

provided $|h[m]| < \infty$ for all m

Transfer Function

$$H(z) = \frac{z^{-2}}{(1-p_0 z^{-1})(1-p_1 z^{-1})} = \frac{1}{(z-p_0)(z-p_1)}$$

Cascaded Implementation



$$H(z) = H_0(z)H_1(z)$$

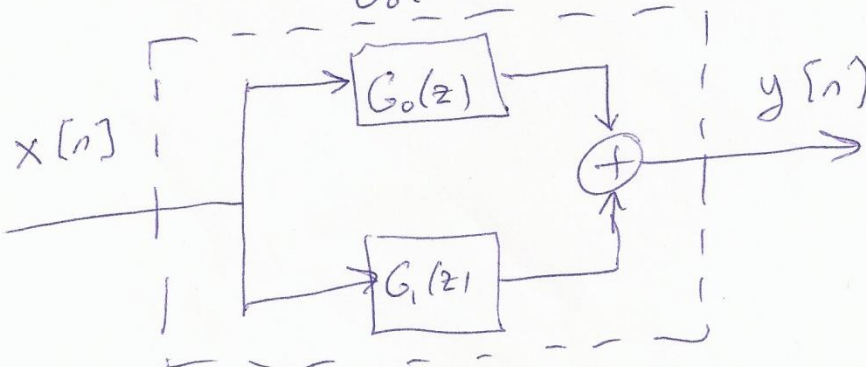
Parallel Implementation

- Use partial fractions decomposition

$$H(z) = \underbrace{\frac{A}{z-p_0}}_{G_0(z)} + \underbrace{\frac{B}{z-p_1}}_{G_1(z)}$$

$$A = \frac{1}{p_0-p_1}$$

$$B = \frac{1}{p_1-p_0}$$



Interpolation and Pulse Shaping

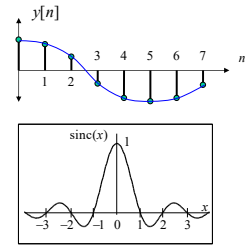
Prof. Brian L. Evans
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 7 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

- **Sampling & reconstruction**
 - Data conversion
 - Interpolation
 - Pulse shapes
 - Demonstration
 - Raised cosine pulse
- **Conclusion**



7-2

2

Sampling & Reconstruction

Continuous-Time Analysis

Sampling $x(t)$

HW 0.3

How to reconstruct $x(t)$?

How to reconstruct $x(t)$ given its sampled version $y(t)$?

Mathematical Model

$x(t) \xrightarrow{f_s} y(t) \xrightarrow{\text{Filter LPF (C=ideal)}} \hat{x}(t)$

$y(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$

$\hat{y}(f) = Y(f) * P(f)$

$P(f) = \sum_{k=-\infty}^{\infty} \delta(f - kf_s)$

$\hat{y}(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$

Lowpass filter has stopband frequency less than $\frac{1}{2}f_s$ to reduce aliasing at sampler output (enforce sampling theorem)

Discrete-to-continuous (time) conversion could be as simple as sample and hold

Lowpass filter has stopband frequency less than $\frac{1}{2}f_s$ to reduce artificial high frequencies

Over what frequency range is the reconstruction valid?
 $-1/2 f_s < f < 1/2 f_s$

7-3

3

Sampling & Reconstruction

Data Conversion

- **Analog-to-Digital Conversion** **Lecture 4** **Lecture 8**

Lowpass filter has stopband frequency less than $\frac{1}{2}f_s$ to reduce aliasing at sampler output (enforce sampling theorem)

Analog Lowpass Filter → Sampler at sampling rate of f_s → Quantizer
- **Digital-to-Analog Conversion** **Lecture 7**

Discrete-to-continuous (time) conversion could be as simple as sample and hold

Discrete to Continuous Conversion → Analog Lowpass Filter

Lowpass filter has stopband frequency less than $\frac{1}{2}f_s$ to reduce artificial high frequencies

7-4

4

Discrete-to-Continuous Conversion

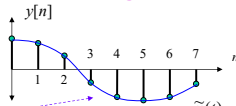
- **Input:** sequence of samples $y[n]$
- **Output:** smooth continuous-time function obtained through interpolation (by “connecting the dots”)

If $f_0 < \frac{1}{2}f_s$, then

$$y[n] = A \cos(2\pi f_0 T_s n + \phi)$$

would be converted to

$$\tilde{y}(t) = A \cos(2\pi f_0 t + \phi)$$



Otherwise, aliasing has occurred, and the converter would reconstruct a cosine wave whose frequency is equal to the aliased positive frequency that is less than $\frac{1}{2}f_s$.

7-5

5

Discrete-to-Continuous Conversion

- **General form of interpolation** $\tilde{y}(t) = \sum_{n=-\infty}^{\infty} y[n] p(t - T_s n)$
is sum of weighted pulses

Sequence $y[n]$ converted into continuous-time signal $\tilde{y}(t)$ that approximates $y(t)$ over frequencies from $-\frac{1}{2}f_s$ to $\frac{1}{2}f_s$.

- **Pulse function $p(t)$**

Rectangular, triangular, sinc, truncated sinc, raised cosine, etc.

Pulses overlap in time when longer than sampling period T_s

Pulses generally have unit amplitude and/or unit area

- **“Mixed-signal” convolution**

Discrete-time signal $y[n]$ and continuous-time signal $p(t)$

Discrete time aligned with continuous time via $T_s n$

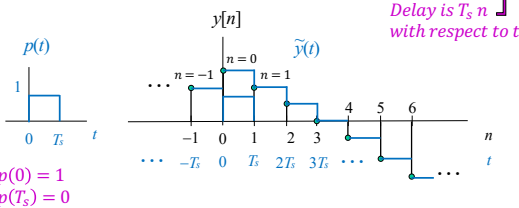
7-6

6

Discrete-to-Continuous Conversion

- **Sample-and-hold reconstruction**

Uses lowpass averaging filter $p(t)$ $\tilde{y}(t) = \sum_{n=-\infty}^{\infty} y[n] p(t - T_s n)$
Has efficient implementation



$$p(0) = 1$$

$$p(T_s) = 0$$

7

Interpolation From Tables

- **Using mathematical tables to estimate a function value**

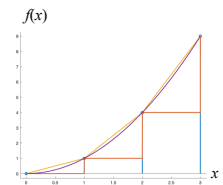
x	$f(x)$
0	0.0
1	1.0
2	4.0
3	9.0

- **Estimate $f(1.5)$ from table**

Zero-order hold: take value to be $f(1)$ to make $f(1.5) = 1.0$ (“stairsteps”)
No multiplications or additions

Linear interpolation: average values of nearest two neighbors: $f(1.5) = 2.5$
2 multiplications and 3 additions per value

Curve fitting: fit four points to cubic polynomial: $f(1.5) = x^2 = 2.25$
43 multiplications to find polynomial coefficients + 1 multiplication per value



8

Curve Fitting

- Interpolate over $1 \leq x \leq 2$ using cubic polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$f(0) = 0 = a_0$$

$$f(1) = 1 = a_0 + a_1 + a_2 + a_3$$

$$f(2) = 4 = a_0 + 2 a_1 + 4 a_2 + 8 a_3$$

$$f(3) = 9 = a_0 + 3 a_1 + 9 a_2 + 27 a_3$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 4 \\ 9 \end{bmatrix}$$

- Matlab code to solve for \vec{a}

$$A = \begin{bmatrix} 1 & 0 & 0 & 0; \\ 1 & 1 & 1 & 1; \\ 1 & 2 & 4 & 8; \\ 1 & 3 & 9 & 27 \end{bmatrix};$$

$$b = \begin{bmatrix} 0; 1; 4; 9 \end{bmatrix};$$

$$a = A \setminus b;$$

$(2/3)n^3$ operations to solve system of n equations

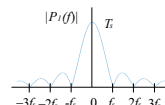
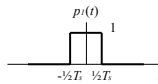
Link

$a_2 = 1$ and $a_0 = a_1 = a_3 = 0$ to give $f(x) = x^2$

9

Pulse Shapes

- Zero-order hold

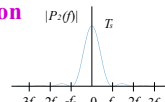
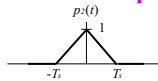


$$P_1(f) = T_s \text{sinc}(f T_s)$$

where

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

- Linear interpolation

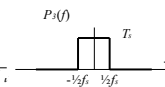
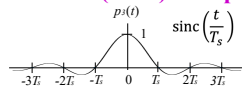


$$P_2(f) = T_s \text{sinc}^2(f T_s)$$

because

$$p_2(t) = \frac{1}{T_s} p_1(t) * p_1(t)$$

- Sinc (ideal) interpolation



$$P_3(f) = T_s \text{rect}\left(\frac{f}{f_s}\right)$$

where

$$\text{rect}(x) = \begin{cases} 1 & \text{for } -\frac{1}{2} \leq x < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

10

Sampling and Interpolation Demo

- DSP First, 2nd ed., ch. 4, Sampling/interpolation

<http://dspfirst.gatech.edu/chapters/04samplin/demos/recon/index.html/>

Sample sinusoidal signal $y(t)$ to form $y[n]$

Reconstruct continuous-time

sinusoidal signal using

rectangular, triangular, or

truncated sinc pulse $p(t)$

$$\tilde{y}(t) = \sum_{n=-\infty}^{\infty} y[n] p(t - T_s n)$$

- Design tradeoffs

Which pulse gives the best reconstruction?

Sinc pulse truncated to 4 sampling periods. Why?

What happens as the sampling rate is increased?

Tradeoffs in signal quality vs. run-time complexity?

7-11

11

Raised Cosine Pulse: Time Domain

- Pulse shaping used in communication systems

$$p(t) = \text{sinc}\left(\frac{t}{T_s}\right) \frac{\cos(2\pi \alpha W t)}{1 - 16 \alpha^2 W^2 t^2}$$

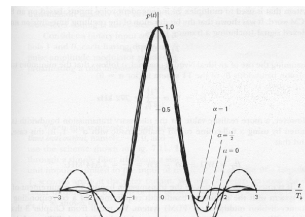
ideal lowpass filter Attenuation by $1/t^2$ for impulse response large t to reduce tail

W is bandwidth of an ideal lowpass response

$\alpha \in [0, 1]$ rolloff factor

Zero crossings at

$$t = \pm T_s, \pm 2 T_s, \dots$$



Simon Haykin, *Communication Systems*, 3rd ed.

- See handout G in reader on raised cosine pulse

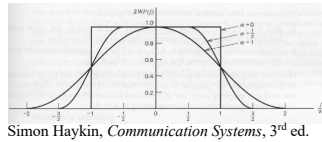
7-12

12

Raised Cosine Pulse Spectra

- **Pulse shaping used in communication systems**

Bandwidth increased by factor of $(1 + \alpha)$:
 $(1 + \alpha) W = 2 W - f_i$
 f_i marks transition from passband to stopband



Simon Haykin, *Communication Systems*, 3rd ed.

$$P(f) = \begin{cases} \frac{1}{2W} & \text{if } 0 \leq |f| < f_i \\ \frac{1}{4W} \left(1 - \sin \left(\frac{\pi(|f| - W)}{2W - 2f_i} \right) \right) & \text{if } f_i \leq |f| < 2W - f_i \\ 0 & \text{otherwise} \end{cases} \quad \left| \begin{array}{l} W = \frac{1}{2T_s} \\ \alpha = 1 - \frac{f_i}{W} \end{array} \right.$$

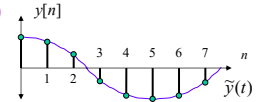
Bandwidth generally scarce in communication systems 7-13

13

Conclusion

- **Discrete-to-continuous time conversion involves interpolating between known discrete-time samples $y[n]$ using pulse shape $p(t)$**

$$\tilde{y}(t) = \sum_{n=-\infty}^{\infty} y[n] p(t - T_s n)$$



- **Common pulse shapes**

Rectangular for same-and-hold interpolation

Triangular for linear interpolation

Sinc for optimal bandlimited linear interpolation but impractical

Truncated sinc or raised cosine for practical interpolation

- **Truncation in time causes smearing in frequency**

7-14

14

Quantization

Prof. Brian L. Evans
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 8 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

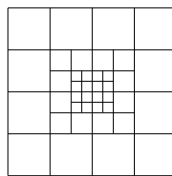
- Resolution
- Quantization
- Quantization error (noise) analysis
- Total harmonic distortion
- Noise immunity in communication systems
- Conclusion
- Digital vs. analog audio (optional)

8 - 2

2

Resolution

- **Human eyes**
 Sample received light on 2-D grid
 Photoreceptor density in retina falls off exponentially away from fovea (point of focus)
 Respond logarithmically to intensity (amplitude) of light



*Foveated grid:
 point of focus in middle*

- **Human ears**
 Respond to frequencies in 20 Hz to 20 kHz range
 Respond logarithmically in both intensity (amplitude) of sound (pressure waves) and frequency (octaves)
 Log-log plot for hearing response vs. frequency

8 - 3

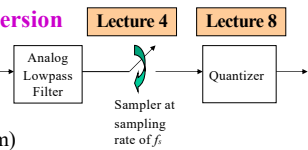
3

Quantization

- Quantization is an interpretation of a continuous quantity by a finite set of discrete values

- **Analog-to-Digital Conversion** Lecture 4 Lecture 8
 Lowpass filter has stopband frequency less than $\frac{1}{2}f_s$ to reduce aliasing at sampler output (enforce sampling theorem)

System properties: *Linearity*
Time-Invariance
Causality
Memory



8 - 4

4

Uniform Amplitude Quantization

- **Round to nearest integer (midtread)**

Quantize amplitude to levels $\{-2, -1, 0, 1\}$

Step size Δ for linear region of operation

Represent levels by $\{00, 01, 10, 11\}$ or $\{10, 11, 00, 01\}$...

Latter is two's complement representation

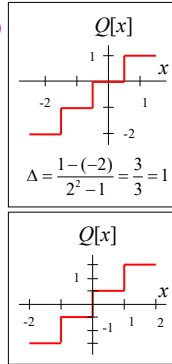
- **Rounding with offset (midrise)**

Quantize to levels $\{-3/2, -1/2, 1/2, 3/2\}$

Represent levels by $\{11, 10, 00, 01\}$...

Step size $\Delta = \frac{3 - (-3)}{2^2 - 1} = \frac{3}{3} = 1$

Used in slide 8-9



5

Audio Compact Discs (CDs)

- **Analog lowpass filter**

Passband 0–20 kHz

Transition band 20–22 kHz

Stopband frequency at 22 kHz (i.e. 10% rolloff)

Designed to control amount of aliasing that occurs at sampler output (and hence called an anti-aliasing filter)

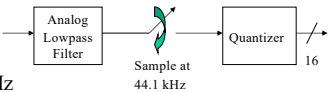
- **Signal-to-noise ratio when quantizing to B bits**

$$1.76 \text{ dB} + 6.02 \text{ dB/bit} * B = 98.08 \text{ dB}$$

Loose upper bound derived in slides 8-9 to 8-13

Audio CDs have dynamic range of about 95 dB

Noise is quantization error (see slide 8-9)



8-6

6

Dynamic Range

- **Signal-to-noise ratio in dB**

$$\begin{aligned} \text{SNR}_{\text{dB}} &= 10 \log_{10} \frac{\text{Signal Power}}{\text{Noise Power}} \\ &= 10 \log_{10} \text{Signal Power} - \\ &\quad 10 \log_{10} \text{Noise Power} \end{aligned}$$

- **For linear systems, dynamic range equals SNR**

- **Lowpass anti-aliasing filter for audio CD format**

Ideal magnitude response of 0 dB over passband

$A_{\text{stopband}} = 0 \text{ dB} - \text{Noise Power in dB} = -98.08 \text{ dB}$

Why $10 \log_{10}$?

For amplitude A ,

$$|A|_{\text{dB}} = 20 \log_{10} |A|$$

With power $P \propto |A|^2$,

$$P_{\text{dB}} = 10 \log_{10} |A|^2$$

$$P_{\text{dB}} = 20 \log_{10} |A|$$

8-7

7

Dynamic Range in Audio

- **Sound Pressure Level (SPL)**

Reference in dB SPL is 20 μPa (threshold of hearing)

40 dB SPL noise in typical living room

120 dB SPL threshold of pain

80 dB SPL resulting dynamic range

- **Estimating dynamic range**

(a) Find maximum RMS output of the linear system with some specified amount of distortion, typically 1%

(b) Find RMS output of system with small input signal (e.g.

-60 dB of full scale) with input signal removed from output

(c) Divide (b) into (a) to find the dynamic range

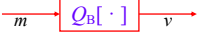
Anechoic room	10 dB
Whisper	30 dB
Rainfall	50 dB
Dishwasher	60 dB
City Traffic	85 dB
Leaf Blower	110 dB
Siren	120 dB

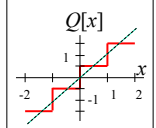
Slide by Dr. Thomas D. Kite, Audio Precision

8-8

8

Quantization Error (Noise) Analysis

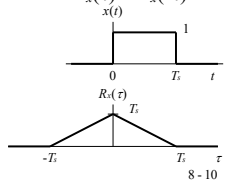
- **Quantization output**
Input signal plus noise
Noise is difference of output and input signals
- **Signal-to-noise ratio (SNR) derivation**
Quantize to B bits


Quantization error (noise)
 $q = Q_B[m] - m = v - m$
- **Uniform midrise quant.**


Example
 $m_{\max} = 2V$
 $B = 2$ bits
 $L = 2^B = 4$ levels
 $\Delta = L-1$ steps
 Input in "linear" region:
 $m \in (-m_{\max}, m_{\max})$
 Quantization error (noise) Q is uniformly distributed
 Quantization levels $L = 2^B$ large enough so $\frac{1}{L-1} \approx \frac{1}{L}$

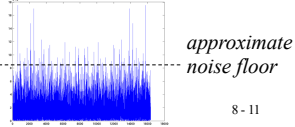
9

Quantization Error (Noise) Analysis

- **Deterministic signal $x(t)$ w/ Fourier transform $X(f)$**
Power spectrum is square of absolute value of magnitude response (phase is ignored)
 $P_x(f) = |X(f)|^2 = X(f) X^*(f)$
Multiplication in Fourier domain is convolution in time domain
Conjugation in Fourier domain is reversal & conjugation in time
 $X(f) X^*(f) = F\{x(\tau) * x^*(-\tau)\}$
 - **Autocorrelation of $x(t)$**
 $R_x(\tau) = x(\tau) * x^*(-\tau)$
Maximum value (when it exists) is at $R_x(0)$
 $R_x(\tau)$ is even symmetric, i.e. $R_x(\tau) = R_x(-\tau)$
- 

10

Quantization Error (Noise) Analysis

- **Two-sided random signal $n(t)$** $P_n(f) = F\{R_n(\tau)\}$
Fourier transform may not exist, but power spectrum exists
 $R_n(\tau) = E\{n(t) n^*(t + \tau)\} = \int_{-\infty}^{\infty} n(t) n^*(t + \tau) dt$ **time-averaged**
 $R_n(-\tau) = E\{n(t) n^*(t - \tau)\} = \int_{-\infty}^{\infty} n(t) n^*(t - \tau) dt = n(\tau) * n^*(-\tau)$
For zero-mean Gaussian random process $n(t)$ with variance σ^2
 $R_n(\tau) = E\{n(t) n^*(t + \tau)\} = 0$ when $\tau \neq 0$
 $R_n(\tau) = E\{n(t) n^*(t + \tau)\} = \sigma^2 \delta(\tau) \iff P_n(f) = \sigma^2$
- **Estimate noise power spectrum in Matlab**
 $N = 16384;$ % finite no. of samples
`gaussianNoise = randn(N,1);`
`plot(abs(fft(gaussianNoise)).^2);`


approximate noise floor
8-11

11

Quantization Error (Noise) Analysis

- **Quantizer step size**
 $\Delta = \frac{2m_{\max}}{L-1} \approx \frac{2m_{\max}}{L}$
- **Quantization error**
 $-\frac{\Delta}{2} \leq q \leq \frac{\Delta}{2}$
 q is sample of zero-mean random variable Q
 Q is uniformly distributed
 $\sigma_Q^2 = E\{Q^2\} - \underbrace{\mu_Q^2}_{\text{zero}}$
 $\sigma_Q^2 = \frac{\Delta^2}{12} = \frac{1}{3} m_{\max}^2 2^{-2B}$
- **Input power: $P_{\text{average},m}$**
 $\text{SNR} = \frac{\text{Signal Power}}{\text{Noise Power}}$
 $\text{SNR} = \frac{P_{\text{average},m}}{\sigma_Q^2} = \left(\frac{3P_{\text{average},m}}{m_{\max}^2} \right) 2^{2B}$
 SNR exponential in B
 Adding 1 bit increases SNR by factor of 4
- **Derivation of SNR in deciBels on next slide**

12

Quantization Error (Noise) Analysis

- SNR in dB = constant + 6.02 dB/bit * B

Loose upper bound

$$\begin{aligned}
 10 \log_{10} \text{SNR} &= 10 \log_{10} \left(\left(\frac{3P_{\text{average,m}}}{m_{\text{max}}^2} \right) 2^{2B} \right) \\
 &= 10 \log_{10} 3 + 10 \log_{10} (P_{\text{average,m}}) - 20 \log_{10} (m_{\text{max}}) + 20 B \log_{10} (2) \\
 &= \underbrace{0.477 + 10 \log_{10} (P_{\text{average,m}}) - 20 \log_{10} (m_{\text{max}})}_{1.76 \text{ and } 1.17 \text{ are common constants used in audio}} + 6.02 B
 \end{aligned}$$

TI Stereo Codec		Signal-to-Noise Ratio	Total Harmonic Distortion
← Cirrus Logic WM8994	ADC	94 dB (15.3 bits)	80 dB (13.0 bits)
	DAC	97 dB (15.8 bits)	82 dB (13.3 bits)
← TLV320AIC3106 (differential mode)	ADC	92 dB (15.0 bits)	91 dB (14.8 bits)
	DAC	99 dB (16.2 bits)	94 dB (15.3 bits)

Using B as the effective number of bits, SNR in dB = 2 + 6 B
 WM8994 is on our STM ARM board; TLV320AIC3106 used in previous TI OMAP-L138 board

13

Total Harmonic Distortion + Noise

- A measure of nonlinear distortion in a system

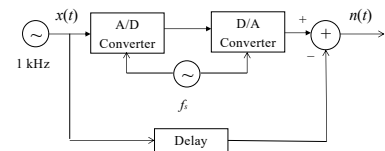
Input is a sinusoidal signal of a single fixed frequency
 From output of system, the input sinusoid signal is subtracted
 Compute SNR

- In audio, sinusoidal signal is often at 1 kHz

In "Sweet spot" for human hearing between 0.8 and 1.2 kHz

- Example

"System" is ADC
 Calibrated DAC
 Signal is $x(t)$
 "Noise" is $n(t)$



14

Noise Immunity at Receiver Output

- Depends on modulation, average transmit power, transmission bandwidth and channel noise
- Analog communications (receiver output SNR)

"When the carrier to noise ratio is high, an increase in the transmission bandwidth B_T provides a corresponding quadratic increase in the output signal-to-noise ratio or figure of merit of the [wideband] FM system."
 – Simon Haykin, *Communication Systems*, 4th ed., p. 147.

- Digital communications (receiver symbol error rate)

"For code division multiple access (CDMA) spread spectrum communications, probability of symbol error decreases exponentially with transmission bandwidth B_T "
 – Andrew Viterbi, *CDMA: Principles of Spread Spectrum Communications*, 1995, pp. 34-36.

8-15

15

Conclusion

- Amplitude quantization approximates its input by a discrete amplitude taken from finite set of values
- Loose upper bound in signal-to-noise ratio of a uniform amplitude quantizer with output of B bits

Best case: 6 dB of SNR gained for each bit added to quantizer
 Key limitation: assumes large number of levels $L = 2^B$

- Best case improvement in noise immunity for communication systems

Analog: improvement quadratic in transmission bandwidth

Digital: improvement exponential in transmission bandwidth

8-16

16

Handling Overflow

- **Example: Consider set of integers {-2, -1, 0, 1}**
 Represented in two's complement system {10, 11, 00, 01}.
 Add $(-1) + (-1) + (-1) + 1 + 1$
 Intermediate computations are $-2, 1, -2, -1$ for wraparound arithmetic and $-2, -2, -1, 0$ for saturation arithmetic
- **Saturation: When to use it?** Native support in MMX and DSPs
 If input value greater than maximum, set it to maximum; if less than minimum, set it to minimum
 Used in quantizers, filtering, other signal processing operators
- **Wraparound: When to use it?** Standard two's complement behavior
 Addition performed modulo set of integers
 Used in address calculations, array indexing

17

Digital vs. Analog Audio

- **An audio engineer claims to notice differences between analog vinyl master recording and the remixed CD version. Is this possible?**
 When digitizing an analog recording, the maximum voltage level for the quantizer is the maximum volume in the track
 Samples are uniformly quantized (to 2^{16} levels in this case although early CDs circa 1982 were recorded at 14 bits)
 Problem on a track with both loud and quiet portions, which occurs often in classical pieces
 When track is quiet, relative error in quantizing samples grows
 Contrast this with analog media such as vinyl which responds linearly to quiet portions

8 - 18

18

Digital vs. Analog Audio

- **Analog and digital media response to voltage v**

$$A(v) = \begin{cases} V_0 + (v - V_0)^{1/3} & \text{for } v > V_0 \\ v & \text{for } -V_0 \leq v \leq V_0 \\ -V_0 - (V_0 - v)^{1/3} & \text{for } v < -V_0 \end{cases} \quad D(v) = \begin{cases} V_0 & \text{for } v > V_0 \\ v & \text{for } -V_0 \leq v \leq V_0 \\ -V_0 & \text{for } v < -V_0 \end{cases}$$
- **For a large dynamic range**
 Analog media: records voltages above V_0 with distortion
 Digital media: clips voltages above V_0 to V_0
- **Audio CDs use delta-sigma modulation**
 Effective dynamic range of 19 bits for lower frequencies but lower than 16 bits for higher frequencies
 Human hearing is more sensitive at lower frequencies

8 - 19

19

Data Conversion

Slides by Prof. Brian L. Evans, Dept. of ECE, UT Austin, and Dr. Thomas D. Kite, Audio Precision, Beaverton, OR tomk@audioprecision.com

Dr. Ming Ding, when he was at the Dept. of ECE, UT Austin, converted slides by Dr. Kite to PowerPoint format

Some figures are from Ken C. Pohlmann, *Principles of Digital Audio*, McGraw-Hill, 1995.

Lecture 10

1

Image Halftoning

- **Handout J on noise-shaped feedback coding**

Different ways to perform one-bit quantization (halftoning)

Original image has 8 bits per pixel original image (pixel values range from 0 to 255 inclusive)

- **Pixel thresholding: Same threshold at each pixel**

Gray levels from 128-255 become 1 (white)

Gray levels from 0-127 become 0 (black)

No noise shaping

- **Ordered dither: Periodic space-varying thresholding**

Equivalent to adding spatially-varying dither (noise) at input to threshold operation (quantizer)

No noise shaping

Example uses 16 different thresholds in a 4×4 mask

Periodic artifacts appear as if screen has been overlaid

10 - 2

2

Image Halftoning

- **Error diffusion: Noise-shaping feedback coding**

Contains sharpened original plus high-frequency noise

Human visual system less sensitive to high-frequency noise (as is the auditory system)

Example uses four-tap Floyd-Steinberg noise-shaping (i.e. a four-tap IIR filter)

- **Image quality of halftones**

Thresholding (low): error spread equally over all freq.

Ordered dither (medium): resampling causes aliasing

Error diffusion (high): error placed into higher frequencies

- **Noise-shaped feedback coding is a key principle in modern A/D and D/A converters**

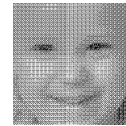
10 - 3

3

Digital Halftoning Methods



Clustered Dot Screening
AM Halftoning



Dispersed Dot Screening
FM Halftoning



Error Diffusion
FM Halftoning 1975



Blue-noise Mask
FM Halftoning 1993



Green-noise Halftoning
AM-FM Halftoning 1992



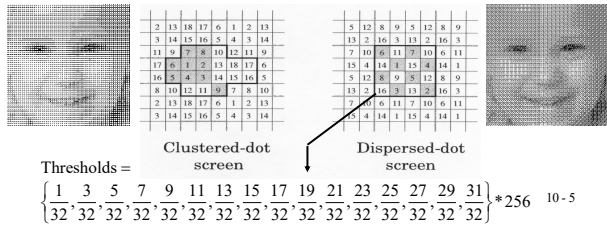
Direct Binary Search
FM Halftoning 1992

10 - 4

4

Screening (Masking) Methods

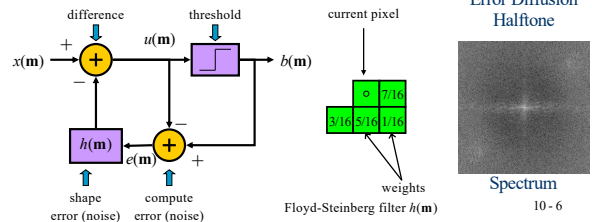
- **Periodic array of thresholds smaller than image**
Spatial resampling leads to aliasing (gridding effect)
Clustered dot screening produces a coarse image that is more resistant to printer defects such as ink spread
Dispersed dot screening has higher spatial resolution



5

Grayscale Error Diffusion

- **Shapes quantization error (noise) into high frequencies**
- **Type of sigma-delta modulation**
- **Error filter $h(m)$ is lowpass**



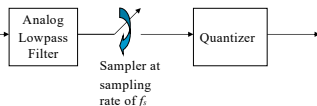
6

Old-Style A/D and D/A Converters

- **Used discrete components (before mid-1980s)**

A/D Converter

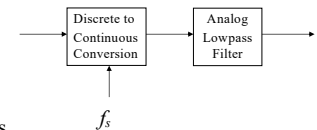
Lowpass filter has stopband frequency of $\frac{1}{2}f_s$ or less



D/A Converter

Lowpass filter has stopband frequency of $\frac{1}{2}f_s$ or less

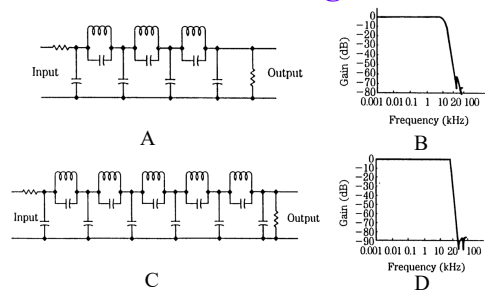
Discrete-to-continuous conversion could be as simple as sample and hold



10 - 7

7

Cost of Multibit Conversion Part I: Brickwall Analog Filters

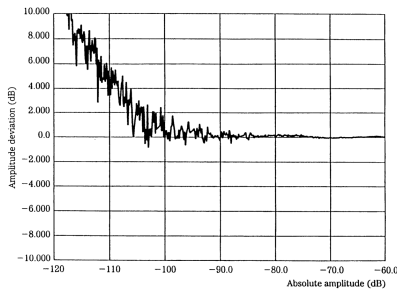


Pohlmann Fig. 3-5 Two examples of passive Chebyshev lowpass filters and their frequency responses. A. A passive low-order filter schematic. B. Low-order filter frequency response. C. Attenuation to -90 dB is obtained by adding sections to increase the filter's order. D. Steepness of slope and depth of attenuation are improved.

10 - 8

8

Cost of Multibit Conversion Part II: Low-Level Linearity



Pohlmann Fig. 4-3 An example of a low-level linearity measurement of a D/A converter showing increasing non-linearity with decreasing amplitude. 10 - 9

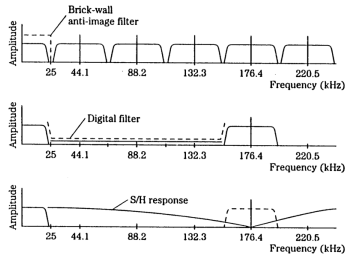
9

Solutions

- **Oversampling eases analog filter design**
Also creates spectrum to put noise at inaudible frequencies
 - **Add dither (noise) at quantizer input**
Breaks up harmonics (idle tones) caused by quantization
 - **Shape quantization noise into high frequencies**
Auditory system is less sensitive at higher frequencies
 - **State-of-the-art in 20-bit/24-bit audio converters**
- | | | | |
|-----------------|---|---|---|
| Oversampling | 64x | 256x | 512x |
| Quantization | 8 bits | 6 bits | 5 bits |
| Additive dither | 2-bit Δ PDF | 2-bit Δ PDF | 2-bit Δ PDF |
| Noise shaping | 5 th / 7 th order | 5 th / 7 th order | 5 th / 7 th order |
| Dynamic range | 110 dB | 120 dB | 120 dB |

10

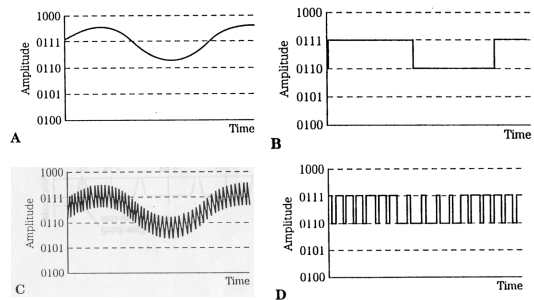
Solution 1: Oversampling



Pohlmann Fig. 4-15 Image spectra of nonoversampled and oversampled reconstruction. Four times oversampling simplifies reconstruction filter. 10 - 11

11

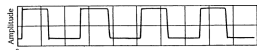
Solution 2: Add Dither



Pohlmann Fig. 2-8 Adding dither at quantizer input alleviates effects of quantization error. A. An undithered input signal with amplitude on the order of one LSB. B. Quantization results in a coarse coding over two levels. C. Dithered input signal. 10 - 12 D. Quantization yields a PWM waveform that codes information below the LSB.

12

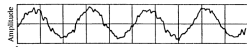
Time Domain Effect of Dither



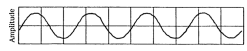
A A 1 kHz sine wave with amplitude of one-half LSB without dither produces a square wave.



B Dither of one-third LSB rms amplitude is added to the sine wave before quantization, resulting in a PWM waveform.



C Modulation carries the encoded sine wave information, as can be seen after 32 averagings.



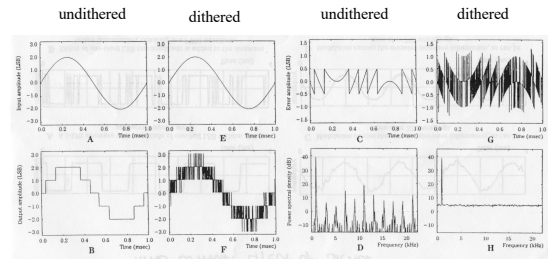
D Modulation carries the encoded sine wave information, as can be seen after 960 averagings.

Pohlmann Fig. 2-9 Dither permits encoding of information below the least significant bit. Vanderkooy and Lipshitz.

10 - 13

13

Frequency Domain Effect of Dither



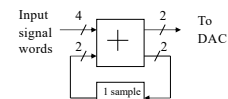
Pohlmann Fig. 2-10 Computer-simulated quantization of a low-level 1-kHz sine wave without, and with dither. A. Input signal. B. Output signal (no dither). C. Total error signal (no dither). D. Power spectrum of output signal (no dither). E. Input signal. F. Output signal (triangular pdf dither). G. Total error signal (triangular pdf dither). H. Power spectrum of output signal (triangular pdf dither) Lipshitz, Wamnamaker, and Vanderkooy

10 - 14

14

Solution 3: Noise Shaping

We have a two-bit DAC and four-bit input signal words. Both are unsigned.

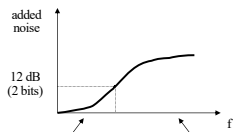


Assume input = 1001 constant

Time	Upper	Lower	Sum	Output to DAC
1	1001	00	1001	10
2	1001	01	1010	10
3	1001	10	1011	10
4	1001	11	1100	11

Average output = $1/4(10+10+10+11)=1001$
 \Rightarrow 4-bit resolution at DC!

Going from 4 bits down to 2 bits increases noise by ~ 12 dB. However, the shaping eliminates noise at DC at the expense of increased noise at high frequency.



If signal is in this band, you are better off! Let's hope this is above the passband! (oversample)

10 - 15

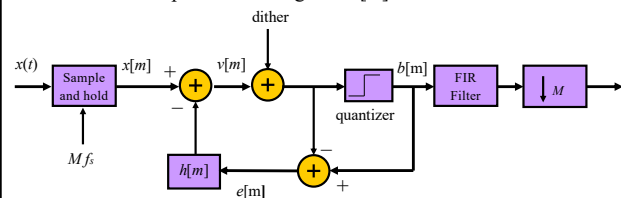
15

Putting It All Together

- A/D converter samples at f_s and quantizes to B bits
- Sigma delta modulator implementation

Internal clock runs at Mf_s

FIR filter expands wordlength of $b[m]$ to B bits



10 - 16

16

Channel Impairments

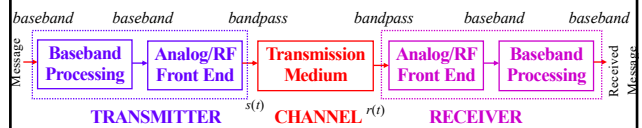
Prof. Brian L. Evans
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 12 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Communication System Structure

- **Information sources**
 Voice, music, images, video, and data (baseband signals)
- **Transmitter**
 Signal processing block lowpass filters message signal
 Carrier circuits block upconverts baseband signal and bandpass filters to enforce transmission band



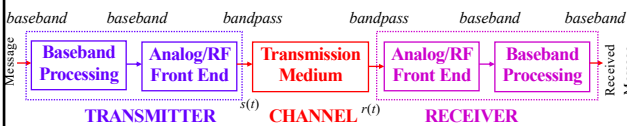
12 - 2

2

Communication Channel

- **Transmission medium**
 Wireline (twisted pair, coaxial, fiber optics)
 Wireless (indoor/air, outdoor/air, underwater, space)
- **Propagating signals degrade over distance**
- **Repeaters can strengthen signal and reduce noise**
 Simple repeater is a bandpass filter and power amplifier

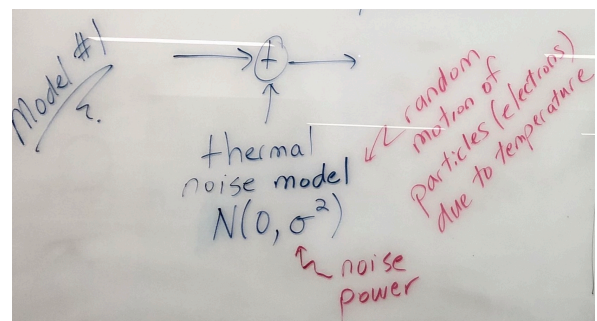
Light painting
 Wi-Fi demo



12 - 3

3

Impairment: Additive Thermal Noise



Models thermal noise in the analog/RF front end of a receiver

12 - 4

4

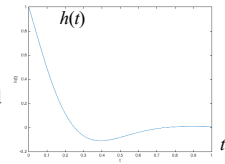
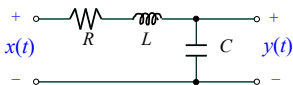
Impairment: LTI Effects

- **Wired linear time-invariant (LTI) effects**

Resistor-inductor-capacitor (RLC) model of wired channel
 LTI system must be "at rest" – initial conditions must be zero
 (initial current across inductors and voltage across capacitors)

- **Second-order case**

Linear constant-coefficient differential equation (with zero initial conditions)
 Impulse response is a damped sinusoid



```
t = 0 : 0.001 : 1;
h = exp(-5*t) .* cos(2*pi*t);
plot(t, h);
xlabel('t'); ylabel('h(t)');
```

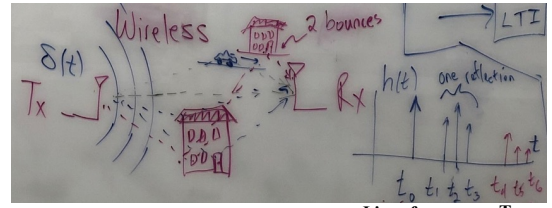
Impulse Response Derivation →

5

Impairment: LTI Effects

- **Wireless LTI effects due to wave propagation**

Reflection, absorption, and scattering



Line of Sight Two Bounces

12-6

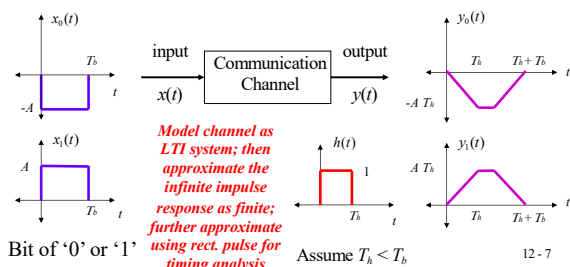
6

Impairment: LTI Effects

- **Linear time-invariant effects**

Distortion in frequency: due to channel frequency response

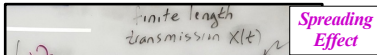
Spreading in time: due to channel impulse response



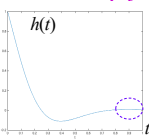
12-7

7

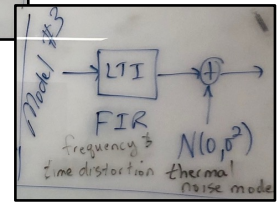
Impairment: LTI Effects



Wired/wireless physical medium is IIR



Model as FIR by truncating impulse response



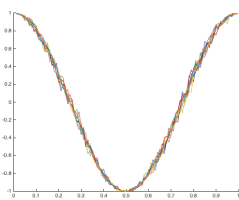
8

Impairment: Phase Jitter

- Linear time-varying effects: **Phase jitter**

Sinusoid at same fixed frequency experiences different phase shifts when passing through channel

Visualize phase jitter in periodic waveform by plotting it over one period, superimposing second period on the first, etc.



```
fc = 1;
Tc = 1 / fc;
fs = 100*fc;
Ts = 1/fs;
t = Ts : Ts : Tc;
numPeriods = 10;
figure;
hold on;
for i = 1 : numPeriods
    phase = 0.05*randn(1, length(t));
    plot(t, cos(2*pi*fc*t + phase));
    pause(1);
end
hold off;
```

12-9

9

Impairment: Phase Jitter

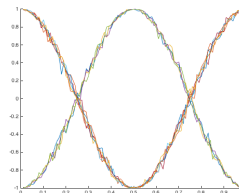
- Linear time-varying effects: **Phase jitter**

Transmission of two-level sinusoidal amplitude modulation

Bit of value '1' becomes an amplitude of +1

Bit of value '0' becomes an amplitude of -1

Visualize by superimposing each period on first (**eye diagram**)



```
fc = 1;
Tc = 1 / fc;
fs = 100*fc;
Ts = 1/fs;
t = Ts : Ts : Tc;
numPeriods = 10;
figure;
hold on;
for i = 1 : numPeriods
    symAmp = round(2*(randn() > 0) - 1);
    phase = 0.05*randn(1, length(t));
    plot(t, symAmp*cos(2*pi*fc*t + phase));
    pause(1);
end
hold off;
```

12-10

10

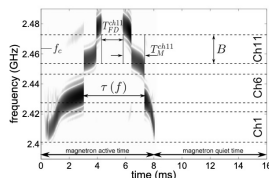
Impairment: Additive Interference

- Nonlinear effects

Harmonics: due to quantization, voltage rectifiers, squaring devices, power amplifiers, etc.

Additive noise: arises from many sources in transmitter, channel, and receiver (e.g. thermal noise)

Additive interference: arises from other systems operating in transmission band

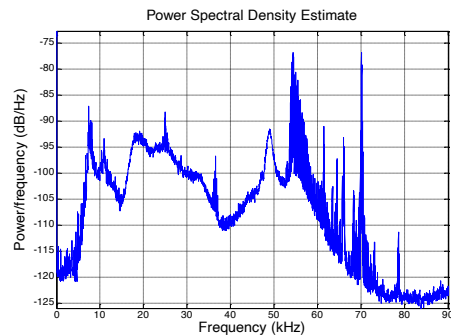


Spectrogram of emissions from a microwave oven in unlicensed 2.4 GHz band sweeping across IEEE 802.11g channels 1, 6, 11 [Nassar, Lin & Evans, 2011]

Keysight Wi-Fi demo →

11

Impairment: Additive Interference



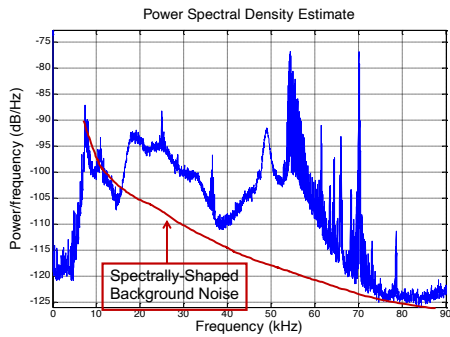
Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011

Home Powerline

12-12

12

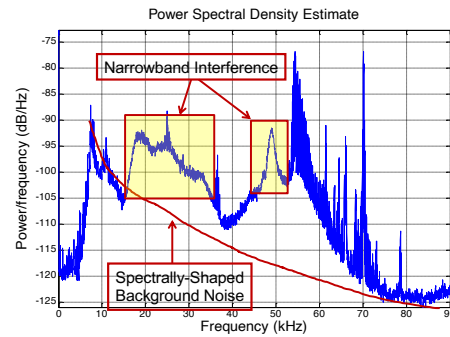
Impairment: Additive Interference



Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011 Home Powerline 12 - 13

13

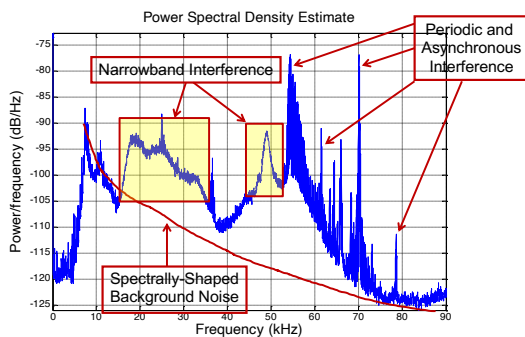
Impairment: Additive Interference



Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011 Home Powerline 12 - 14

14

Impairment: Additive Interference

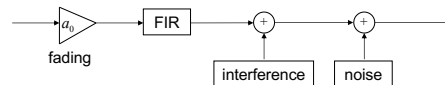


Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011 Home Powerline 12 - 15

15

Impairment: Fading

- Same as wireline channel impairments plus others
- **Fading: multiplicative noise**
Talking on a mobile phone and reception fades in and out
Represented as time-varying gain that follows a particular probability distribution
- **Model #4: fading, LTI effects additive noise, and additive interference**



12 - 16

16

Digital Pulse Amplitude Modulation (PAM)

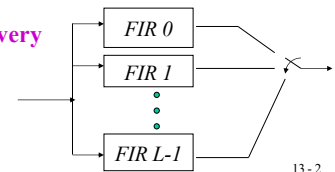
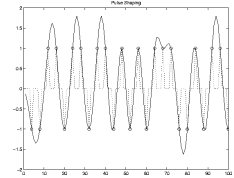
Prof. Brian L. Evans
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 13 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

- Introduction
- Pulse shaping
- Pulse shaping filter bank
- Design tradeoffs
- Symbol clock recovery

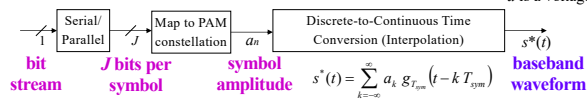
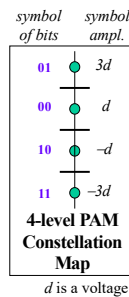


13 - 2

2

Introduction

- **Convert bit stream into pulse stream**
 Group stream of bits into symbols of J bits
 Map symbol of bits to unique amplitude
 Convert from discrete-time to continuous-time
- **M -level PAM or simply M -PAM ($M = 2^J$)**
 Symbol period is T_{sym} and bit rate is Jf_{sym}
 Interpolation pulse shape $g_{T_{sym}}(t)$ per slide 7-6

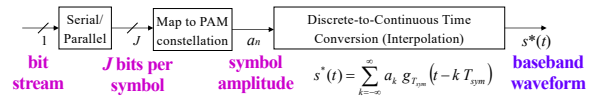
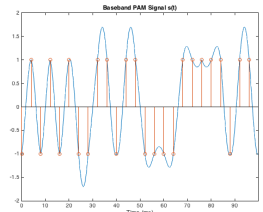


13 - 3

3

2-PAM Transmission

- **2-PAM example (right)**
 Truncated raised cosine pulse $g_{T_{sym}}(t)$ with peak of 1
 What are d and T_{sym} ?
 Max amplitude depends on d ?
- **Highest frequency $\frac{1}{2}f_{sym}$**
 Alternating symbol amplitudes $+d, -d, +d, \dots$ which is $d \cos(\pi n)$



13 - 4

4

Baseband PAM Transmission Code

```

% Baseband PAM Signal Generation
% by Prof. Brian L. Evans
% The University of Texas at Austin
% Spring 2018

% m is sample index
% n is symbol index

% Simulation parameters
N = 25; % Number symbol periods

% Pulse shape g[m]
Ng = 4; % Number symbol periods
L = 20; % Samples/symbol period
f0 = 1/L;
midpt = Ng*L/2;
m = (-midpt):(midpt-1);
g = sinc(f0*m);

% Adjust for group delay
N = N + (Ng/2);
d = 1;
M = 2;
loffset = M + 1;
symAmp = (2*randi(M,1,N) - loffset)*d;

% Discrete-time baseband PAM signal
Nmax = N*L;
v = zeros(1,Nmax);
v(1:L:end) = symAmp; % interpolation
s = conv(v, g); % pulse shaping
length = length(s); % trim result
s = s(midpt+1:length-midpt+1);

% Interpretation in continuous time
Tsym = 4; % Symbol period in sec
fsym = 1/Tsym; % Symbol rate in Hz
fs = L*fsym; % Sampling rate in Hz
Ts = 1/fs; % Sampling time in sec

% Plots
Nmax = length(s);
m = 0:(Nmax-1);
t = m*Ts;
Nmax = Nmax / L;
n = 0:(Nmax-1);
figure;
plot(t,s);
hold on;
stem(n*Tsym,symAmp);
hold off;
xlim([0 (Nmax-(Ng/2))*Tsym-Ts]);
ymax = Ng * (M-1) * d / 2;
ylim([-ymax ymax]);
xlabel('Time (ms)');
title('Baseband PAM Signal s(t)');
    
```

5

4-PAM Transmission

4-PAM example (right)

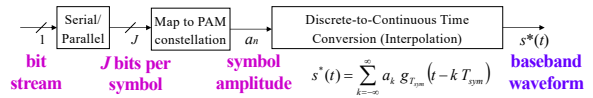
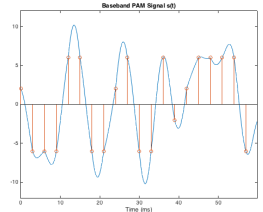
Truncated raised cosine pulse $g_{T_{sym}}(t)$ with peak of 1

What are d and T_{sym} ?

Max amplitude depends on d ?

Highest frequency $\frac{1}{2} f_{sym}$

Alternating symbol amplitudes $+d, -d, +d, \dots$ or $+3d, -3d, +3d, \dots$

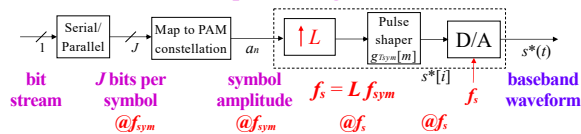


13-6

6

PAM Transmission

Convert baseband processing to discrete time



$$s^*[Ln + m] = \sum_{k=-\infty}^{\infty} a_k g_{T_{sym}}[L(n-k) + m]$$

Sample at sampling time T_s ; let $t = (nL + m) T_s$

L samples per symbol period T_{sym} i.e. $T_{sym} = L T_s$
 n is the index of the current symbol period being transmitted
 m is a sample index within n th symbol (i.e., $m = 0, 1, \dots, L-1$)

7

PAM Transmission

Derivation of $s^*[Ln + m]$ expression on previous slide

Slide 13-7

L samples per symbol period T_s

$f_s = L f_{sym}$

m is the sample index within the n th symbol period: $m \in \{0, 1, \dots, L-1\}$

$t = (nL + m) T_s$ continuous-time pulse shape

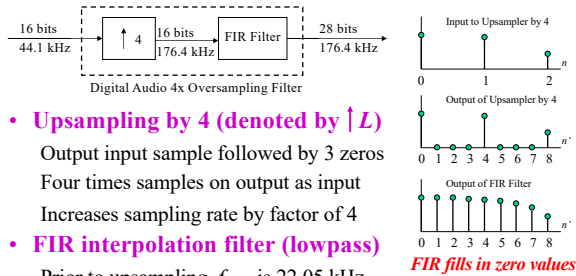
$$s(t) = \sum_{k=-\infty}^{\infty} a_k g_{T_{sym}}(t - k T_{sym}) \rightarrow g_{T_{sym}}((nL+m)T_s - kL T_s) = g_{T_{sym}}(nL T_s + m T_s - kL T_s) = g_{T_{sym}}((n-k)L T_s + m T_s) = g_{T_{sym}}((n-k)L + m)$$

write to sampling at T_s write to sampling rate f_s discrete-time pulse shape

13-8

8

Digital Interpolation Example



- **Upsampling by 4 (denoted by $\uparrow L$)**
Output input sample followed by 3 zeros
Four times samples on output as input
Increases sampling rate by factor of 4
- **FIR interpolation filter (lowpass)**
Prior to upsampling, f_{\max} is 22.05 kHz
At filter input, $\omega_{\max} = 2\pi f_{\max} / f_s = \pi / 4 = \pi / L$
Filter specifications: $\omega_{\text{stop}} < \pi / 4$ and $\omega_{\text{pass}} = 0.9 \omega_{\text{stop}}$

13-9

9

PAM Transmission Example

Fall 2021 Midterm #2

Problem 2.1. Baseband PAM System. 21 points.

Consider a binary phase shift keying (BPSK) system, a.k.a. a two-level pulse amplitude modulation (2-PAM) system.

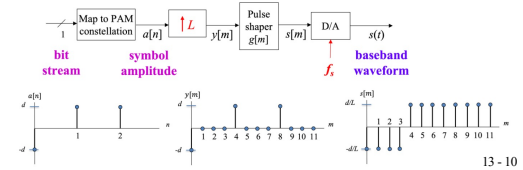
The system parameters are described on the right:

- $J = 1$ bit/symbol
- $L = 4$ samples per symbol period
- Pulse shape $g[m]$ is a rectangular pulse of $L = 4$ samples in duration with amplitude $1/L$.
- A bit of value 0 is mapped to symbol amplitude $-d$, and a bit of value 1 is mapped to symbol amplitude d .

(a) For the BPSK transmitter below, the input bit stream is 011. Plot the discrete-time signals $a[n]$, $y[m]$ and $s[m]$. 9 points.

PAM System Parameters

- $a[n]$ symbol amplitude
- $2d$ constellation spacing
- f_s sampling rate
- f_{sym} symbol rate
- $g[m]$ pulse shape
- $h[m]$ matched filter impulse resp.
- J bits/symbol
- L samples/symbol period
- M levels, i.e. $M = 2^J$
- m sample index
- n symbol index

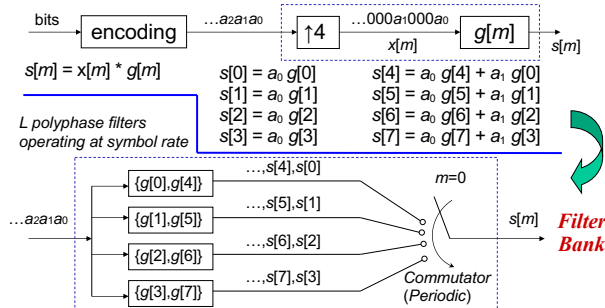


13-10

10

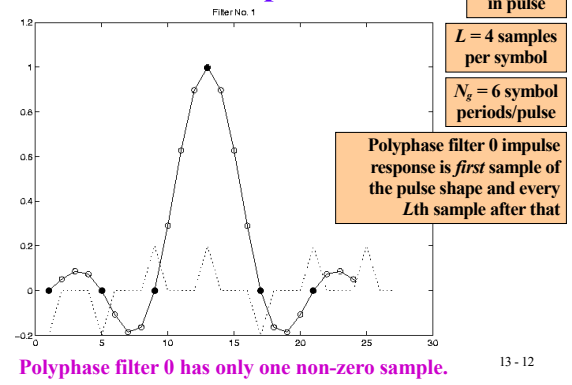
Filter Bank Example #1

- Avoid multiplying by and storing inserted 0 samples
- Pulse shape $g[m]$ lasts 2 symbols (8 samples; $L = 4$)



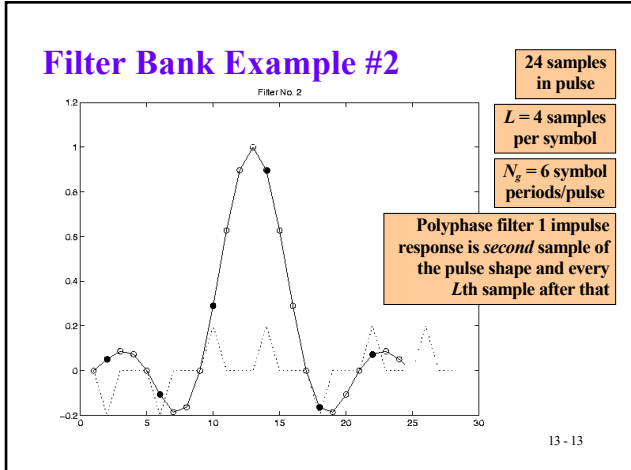
11

Filter Bank Example #2

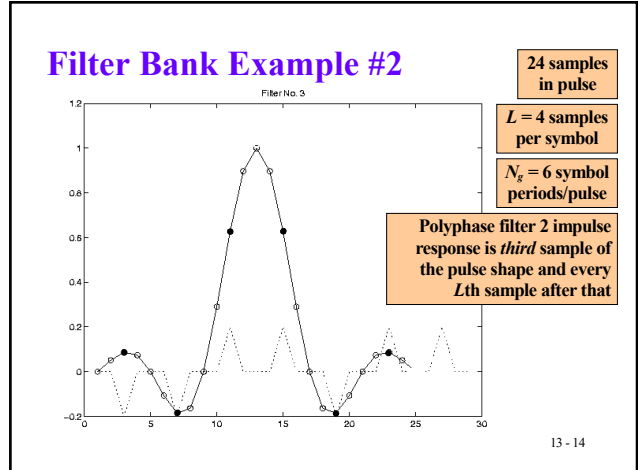


13-12

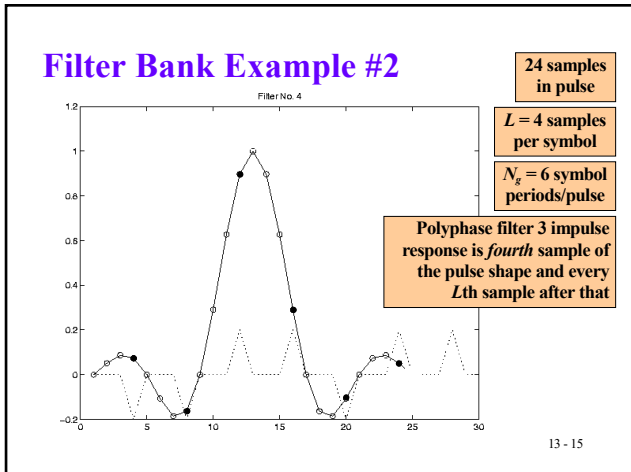
12



13



14



15

Pulse Shaping Design Tradeoffs

	Computation in MACs/s	Memory size in words	Memory reads in words/s	Memory writes in words/s
Direct structure (slide 13-9)	$(L N_g)(L f_{sym})$			
Filter bank structure (slide 13-12)	$L N_g f_{sym}$			

f_{sym} symbol rate
 L samples/symbol
 N_g duration of pulse shape in symbol periods

13 - 16

16

Pulse Shaping Design Tradeoffs

- **FIR filter with N coefficients to compute output**

Read current input value $x[n]$

Read pointer to newest input sample in circular buffer

Read previous $N-1$ input values

Read N filter coefficients h $y[n] = \sum_{m=0}^{N-1} h[m] x[n-m]$

Compute the output value $y[n]$

Write current input value to circular buffer to oldest sample

Write (update) the newest pointer in circular buffer

Write output value

- **N multiplications-additions, $2N + 1$ reads, 3 writes**

13 - 17

17

Pulse Shaping Design Tradeoffs

- **Direct structure (slide 13-9)**

FIR filter has $L N_g$ coefficients ($N = L N_g$)

FIR filter: N multiplications-additions, $2N + 1$ reads, 3 writes

FIR filter runs at sampling rate $L f_{sym}$

Upsampler reads at symbol rate and writes at sampling rate

- **Filter bank structure (slide 13-9)**

L polyphase FIR filters with N_g coefficients each

FIR filter: N_g multiplications-additions, $2N_g + 1$ reads, 3 writes

Each polyphase FIR filter runs at symbol rate f_{sym}

Commutator reads and writes at sampling rate $L f_{sym}$

13 - 18

18

Matched Filtering and Digital Pulse Amplitude Modulation (PAM)

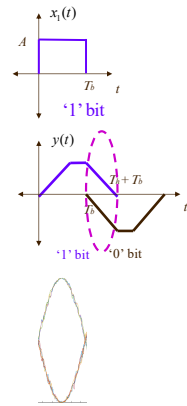
Slides by Prof. Brian L. Evans and Dr. Serene Banerjee
 Dept. of Electrical and Computer Engineering
 The University of Texas at Austin

Lecture 14 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Outline

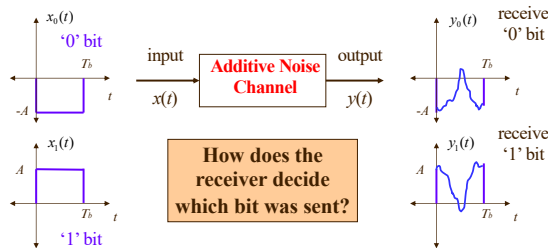
- Transmitting one bit at a time
- Matched filtering
- PAM system
- Intersymbol interference
- Communication performance
 - Bit error probability for binary signals
 - Symbol error probability
- Eye diagram (*in lab*)



2

Transmitting One Bit

- Transmission on communication channels is analog
- Two-level digital pulse amplitude modulation (PAM)

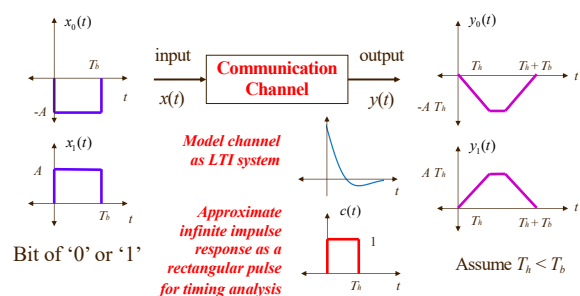


14-3

3

Transmitting One Bit

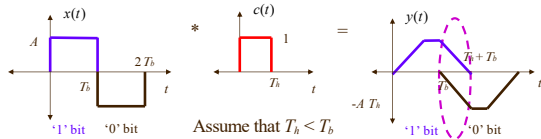
- Two-level digital pulse amplitude modulation over channel that has memory but does not add noise



4

Transmitting Two Bits (Interference)

- Transmitting two bits (pulses) back-to-back will cause overlap (interference) at the receiver



- Sample $y(t)$ at $T_b, 2T_b, \dots$, and threshold with threshold of zero
- How do we prevent intersymbol interference (ISI) at the receiver?

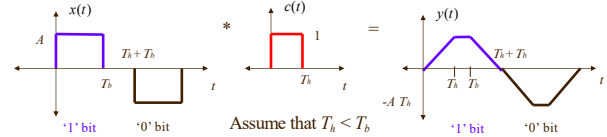
Intersymbol interference

14-5

5

Preventing ISI at Receiver

- Option #1: wait T_h seconds between pulses in transmitter (called guard period or guard interval)

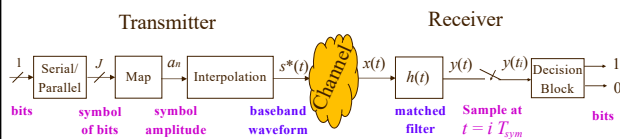


- Option #2: use channel equalizer in receiver
FIR filter designed via training sequences sent by transmitter
Design goal: cascade of channel memory and channel equalizer should give all-pass frequency response

14-6

6

Digital 2-level PAM System



- Transmitted signal $s^*(t) = \sum_{k=-\infty}^{\infty} a_k g(t - k T_{sym})$
Interpolation pulse shape $g(t)$ with $g(0) = 1$ and $g(k T_{sym}) = 0$ for $k \neq 0$ and with lowpass frequency content
- Reliable communication requires synchronization of symbol clocks between transmitter and receiver

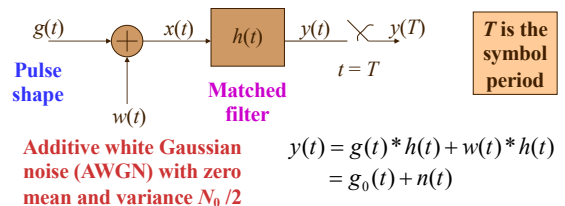
14-7

7

Matched Filter

- Detection of pulse in presence of additive noise

Receiver knows what pulse shape it is looking for
Channel memory ignored (assumed compensated by other means, e.g. channel equalizer in receiver)



14-8

8

Matched Filter Derivation

- **Design of matched filter**

Maximize signal power i.e. power of $g_0(t) = g(t) * h(t)$ at $t = T$

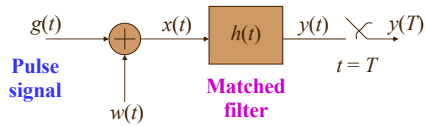
Minimize noise i.e. power of $n(t) = w(t) * h(t)$

- **Combine design criteria**

Maximize η , where η is peak pulse SNR

$$\eta = \frac{|g_0(T)|^2}{E\{n^2(t)\}} = \frac{\text{instantaneous signal power}}{\text{average noise power}}$$

T is the symbol period



14-9

9

Power Spectra

- **Deterministic signal $x(t)$ w/ Fourier transform $X(f)$**

Power spectrum is square of absolute value of magnitude response (phase is ignored)

$$P_x(f) = |X(f)|^2 = X(f) X^*(f)$$

Multiplication in Fourier domain is convolution in time domain

Conjugation in Fourier domain is reversal & conjugation in time

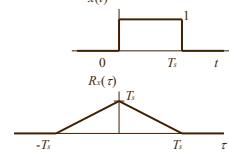
$$X(f) X^*(f) = F\{x(\tau) * x^*(-\tau)\}$$

- **Autocorrelation of $x(t)$**

$$R_x(\tau) = x(\tau) * x^*(-\tau)$$

Maximum value (when it exists) is at $R_x(0)$

$R_x(\tau)$ is even symmetric, i.e. $R_x(\tau) = R_x(-\tau)$



10

Power Spectra

- **Two-sided random signal $n(t)$** $P_n(f) = F\{R_n(\tau)\}$

Fourier transform may not exist, but power spectrum exists

$$R_n(\tau) = E\{n(t) n^*(t+\tau)\} = \int_{-\infty}^{\infty} n(t) n^*(t+\tau) dt \quad \text{time-averaged}$$

$$R_n(-\tau) = E\{n(t) n^*(t-\tau)\} = \int_{-\infty}^{\infty} n(t) n^*(t-\tau) dt = n(\tau) * n^*(-\tau)$$

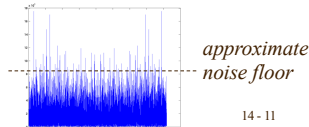
For zero-mean Gaussian random process $n(t)$ with variance σ^2

$$R_n(\tau) = E\{n(t) n^*(t+\tau)\} = 0 \quad \text{when } \tau \neq 0$$

$$R_n(\tau) = E\{n(t) n^*(t+\tau)\} = \sigma^2 \delta(\tau) \iff P_n(f) = \sigma^2$$

- **Estimate noise power spectrum in Matlab**

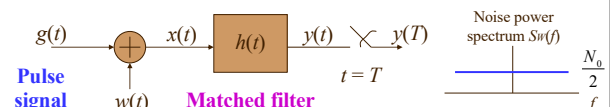
```
N = 16384; % finite no. of samples
gaussianNoise = randn(N,1);
plot(abs(fft(gaussianNoise)).^2);
```



14-11

11

Matched Filter Derivation



- **Noise** $n(t) = w(t) * h(t)$ $S_N(f) = S_w(f) S_H(f) = \frac{N_0}{2} |H(f)|^2$

$$E\{n^2(t)\} = \int_{-\infty}^{\infty} S_N(f) df = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df$$

- **Signal** $g_0(t) = g(t) * h(t)$ $G_0(f) = H(f)G(f)$

$$g_0(t) = \int_{-\infty}^{\infty} H(f) G(f) e^{j2\pi f t} df$$

$$|g_0(T)|^2 = \left| \int_{-\infty}^{\infty} H(f) G(f) e^{j2\pi f T} df \right|^2$$

T is the symbol period

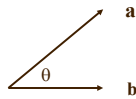
14-12

12

Matched Filter Derivation

- Find $h(t)$ that maximizes pulse peak SNR η

$$\eta = \frac{|\int_{-\infty}^{\infty} H(f) G(f) e^{j2\pi f T} df|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df}$$



- Schwartz's inequality

For vectors: $|\mathbf{a}^T \mathbf{b}^*| \leq \|\mathbf{a}\| \|\mathbf{b}\| \Leftrightarrow \cos\theta = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$

For functions: $\left| \int_{-\infty}^{\infty} \phi_1(x) \phi_2^*(x) dx \right|^2 \leq \int_{-\infty}^{\infty} |\phi_1(x)|^2 dx \int_{-\infty}^{\infty} |\phi_2(x)|^2 dx$

upper bound reached iff $\phi_1(x) = k \phi_2(x) \quad \forall k \in \mathbb{R}$

14 - 13

13

Matched Filter Derivation

Let $\phi_1(f) = H(f)$ and $\phi_2(f) = G^*(f) e^{-j2\pi f T}$

$$\left| \int_{-\infty}^{\infty} H(f) G(f) e^{j2\pi f T} df \right|^2 \leq \int_{-\infty}^{\infty} |H(f)|^2 df \int_{-\infty}^{\infty} |G(f)|^2 df$$

$$\eta = \frac{\left| \int_{-\infty}^{\infty} H(f) G(f) e^{j2\pi f T} df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df} \leq \frac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2 df$$

$$\eta_{\max} = \frac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2 df, \text{ which occurs when}$$

$H_{\text{opt}}(f) = k G^*(f) e^{-j2\pi f T} \quad \forall k$ by Schwartz's inequality

Hence, $h_{\text{opt}}(t) = k g^*(T-t)$

T is the symbol period

14 - 14

14

Matched Filter

- Impulse response is $h_{\text{opt}}(t) = k g^*(T-t)$

Symbol period T , transmitter pulse shape $g(t)$ and gain k
Scaled, conjugated, time-reversed, and shifted version of $g(t)$
Duration and shape determined by pulse shape $g(t)$

- Maximizes peak pulse SNR

$$\eta_{\max} = \frac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2 df = \frac{2}{N_0} \int_{-\infty}^{\infty} |g(t)|^2 dt = \frac{2E_b}{N_0} = \text{SNR}$$

Does not depend on pulse shape $g(t)$

Proportional to signal energy (energy per bit) E_b

Inversely proportional to power spectral density of noise

14 - 15

15

Matched Filter for Rectangular Pulse

- Matched filter for causal rectangular pulse shape

Impulse response is causal rectangular pulse of same duration

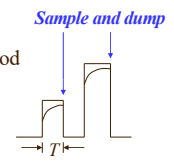
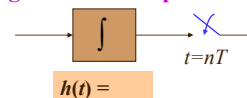
- Convolve input with rectangular pulse of duration T sec and sample result at T sec is same as

First, integrate for T sec

Second, sample at symbol period T sec

Third, reset integration for next time period

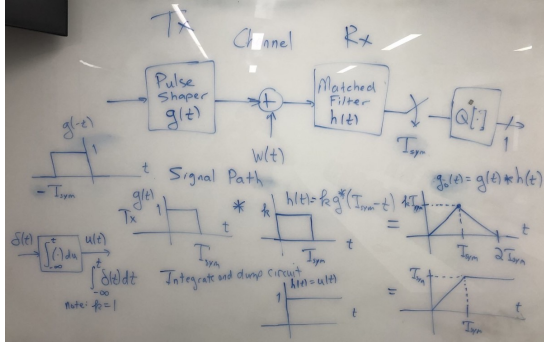
- Integrate and dump circuit



14 - 16

16

Matched Filter for Rectangular Pulse

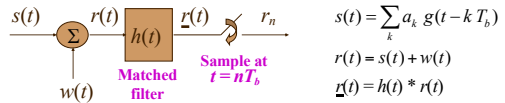


14-17

17

Bit Error Probability for 2-PAM

- T_b is bit period (bit rate is $f_b = 1/T_b$)



$w(t)$ is spectrally flat noise that follows a Gaussian distribution with zero mean and variance σ^2

- Lowpass filtering a Gaussian random process produces another Gaussian random process

Mean scaled by $H(0)$

Variance scaled by twice lowpass filter's bandwidth

- Matched filter's bandwidth is $\frac{1}{2}f_b$

14-18

18

Bit Error Probability for 2-PAM

- Noise power at matched filter output

$$v(nT) = \int_{-\infty}^{\infty} h(\tau)w(nT - \tau)d\tau \quad \text{Filtered noise} \quad T = T_{\text{syn}}$$

$$E\{v^2(nT)\} = E\left\{\left[\int_{-\infty}^{\infty} h(\tau)w(nT - \tau)d\tau\right]^2\right\} \quad \text{Noise power}$$

$$= E\left\{\int_{-\infty}^{\infty} g_T(\tau_1)w(nT - \tau_1)g_T(\tau_2)w(nT - \tau_2)d\tau_1d\tau_2\right\}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_T(\tau_1)g_T(\tau_2)E\{w(nT - \tau_1)w(nT - \tau_2)\}d\tau_1d\tau_2$$

$$= \sigma^2 \int_{-\infty}^{\infty} h^2(\tau)d\tau = \sigma^2 \frac{1}{2\pi} \int_{-\omega_{\text{syn}}/2}^{\omega_{\text{syn}}/2} H^2(\omega)d\omega = \frac{\sigma^2}{T}$$

14-19

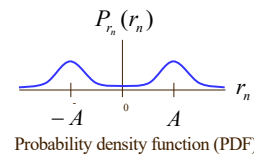
19

Bit Error Probability for 2-PAM

- Symbol amplitudes of $+A$ and $-A$
- Rectangular pulse shape with amplitude 1
- Bit duration (T_b) of 1 second
- Matched filtering with gain of one (see slide 14-15)

Integrate received signal over n th bit period and sample

$$\begin{aligned} r_n &= \int_n^{n+1} r(t) dt \\ &= \pm A + \int_n^{n+1} w(t) dt \\ &= \pm A + v_n \end{aligned}$$



14-20

20

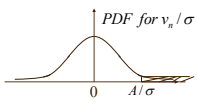
Bit Error Probability for 2-PAM

- Probability of error given that transmitted pulse has amplitude $-A$

$T_b = 1$

$$P(\text{error} | s(nT_b) = -A) = P(-A + v_n > 0) = P(v_n > A) = P\left(\frac{v_n}{\sigma} > \frac{A}{\sigma}\right)$$
- Random variable $\frac{v_n}{\sigma}$ is Gaussian with zero mean and variance of one

PDF for $N(0, 1)$



$$P(\text{error} | s(nT) = -A) = P\left(\frac{v_n}{\sigma} > \frac{A}{\sigma}\right) = \int_{\frac{A}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dv = Q\left(\frac{A}{\sigma}\right)$$

Q function is on next slide

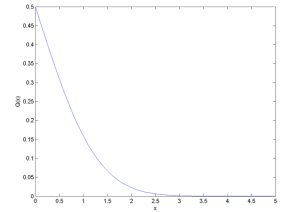
14 - 21

21

Q Function

- Q function
$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-y^2/2} dy$$
- Complementary error function erfc

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$$
- Relationship
$$Q(x) = \frac{1}{2} \text{erfc}\left(\frac{x}{\sqrt{2}}\right)$$



Erfc[x] in Mathematica

erfc(x) in Matlab

14 - 22

22

Bit Error Probability for 2-PAM

- Probability of error given that transmitted pulse has amplitude A

$T_b = 1$

$$P(\text{error} | s(nT_b) = A) = Q(A/\sigma)$$
- Assume that 0 and 1 are equally likely bits
$$P(\text{error}) = P(A)P(\text{error} | s(nT_b) = A) + P(-A)P(\text{error} | s(nT_b) = -A)$$

$$= \frac{1}{2}Q\left(\frac{A}{\sigma}\right) + \frac{1}{2}Q\left(\frac{A}{\sigma}\right) = Q\left(\frac{A}{\sigma}\right) = Q(\sqrt{\rho})$$

where, $\rho = \text{SNR} = \frac{A^2}{\sigma^2}$
- Probability of error exponentially decreases with SNR (see slide 8-16)

$$Q(\sqrt{\rho}) \leq \frac{1}{\sqrt{2\pi}} \frac{e^{-\frac{\rho}{2}}}{\sqrt{\rho}}$$

for large positive ρ

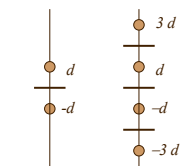
14 - 23

23

PAM Symbol Error Probability

- Set symbol time (T_{sym}) to 1 second
- Average transmitted signal power
$$P_{\text{Signal}} = E\{a_i^2\} \times \frac{1}{2\pi} \int_{-\infty}^{\infty} |G_T(\omega)|^2 d\omega = E\{a_i^2\}$$

$$G_T(\omega) \text{ square root raised cosine spectrum}$$
- M-level PAM symbol amplitudes
$$l_i = d(2i-1), \quad i = -\frac{M}{2} + 1, \dots, 0, \dots, \frac{M}{2}$$
- With each symbol equally likely
$$P_{\text{Signal}} = \frac{1}{M} \sum_{i=-\frac{M}{2}+1}^{\frac{M}{2}} l_i^2 = \frac{2}{M} \sum_{i=1}^{\frac{M}{2}} (d(2i-1))^2 = (M^2 - 1) \frac{d^2}{3}$$



2-PAM 4-PAM
Constellation points with receiver decision boundaries

14 - 24

24

PAM Symbol Error Probability

- Noise power and SNR
- Consider $M-2$ inner levels in constellation

$$P_{Noise} = \frac{1}{2\pi} \int_{-\omega_{sym}/2}^{\omega_{sym}/2} \frac{N_0}{2} d\omega = \frac{N_0}{2}$$

two-sided power spectral density of AWGN

$$SNR = \frac{P_{Signal}}{P_{Noise}} = \frac{2(M^2 - 1)}{3} \times \frac{d^2}{N_0}$$

- Assume ideal channel, i.e. one without ISI

$$r_n = a_n + v_n$$

channel noise after matched filtering and sampling

Error only if $|v_n| > d$
where $\sigma^2 = N_0/2$

Probability of error is

$$P(|v_n| > d) = 2Q\left(\frac{d}{\sigma}\right)$$

- Consider two outer levels in constellation

$$P(v_n > d) = Q\left(\frac{d}{\sigma}\right)$$

14 - 25

25

PAM Symbol Error Probability

- Assuming that each symbol is equally likely, symbol error probability for M -level PAM

$$P_e = \frac{M-2}{M} \left(2Q\left(\frac{d}{\sigma}\right) \right) + \frac{2}{M} Q\left(\frac{d}{\sigma}\right) = \frac{2(M-1)}{M} Q\left(\frac{d}{\sigma}\right)$$

$M-2$ interior points 2 exterior points

- Symbol error probability in terms of SNR

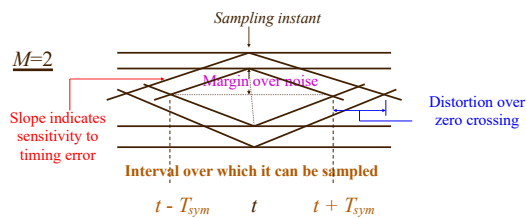
$$P_e = 2 \frac{M-1}{M} Q\left(\left(\frac{3}{M^2-1} SNR\right)^{\frac{1}{2}}\right) \text{ since } SNR = \frac{P_{Signal}}{P_{Noise}} = \frac{d^2}{3\sigma^2} (M^2 - 1)$$

14 - 26

26

Eye Diagram for 2-PAM

- Useful for PAM transmitter and receiver analysis and troubleshooting

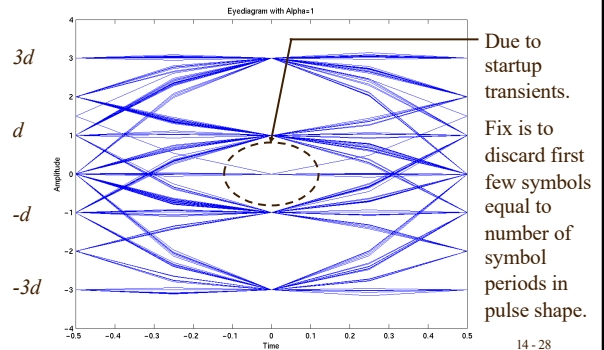


- The more open the eye, the better the reception

14 - 27

27

Eye Diagram for 4-PAM



14 - 28

28

Symbol Clock Recovery

- Transmitter and receiver normally have different oscillator circuits
- Critical for receiver to sample at correct time instances to have max signal power and min ISI
- Receiver should try to synchronize with transmitter clock (symbol frequency and phase)
 - First extract clock information from received signal
 - Then either adjust analog-to-digital converter or interpolate
- Next slides develop adjustment to A/D converter
- Also, see Handout M in the reader

14 - 29

29

Symbol Clock Recovery

- $g_1(t)$ is impulse response of LTI composite channel of pulse shaper, noise-free channel, receive filter

$$q(t) = s^*(t) * g_1(t) = \sum_{k=-\infty}^{\infty} a_k g_1(t - kT_{sym})$$

$s^*(t)$ is transmitted signal

$$p(t) = q^2(t) = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} a_k a_m g_1(t - kT_{sym}) g_1(t - mT_{sym})$$

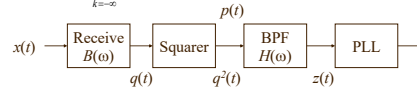
$g_1(t)$ is deterministic

$$E\{p(t)\} = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} E\{a_k a_m\} g_1(t - kT_{sym}) g_1(t - mT_{sym})$$

$E\{a_k a_m\} = a^2 \delta[k-m]$

$$= a^2 \sum_{k=-\infty}^{\infty} g_1^2(t - kT_{sym})$$

Periodic with period T_{sym}



14 - 30

30

Symbol Clock Recovery

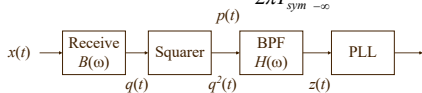
- Fourier series representation of $E\{p(t)\}$

$$E\{p(t)\} = \sum_{k=-\infty}^{\infty} p_k e^{jk\omega_{sym}t}$$

where $p_k = \frac{1}{T_{sym}} \int_0^{T_{sym}} E\{p(t)\} e^{-jk\omega_{sym}t} dt$
- In terms of $g_1(t)$ and using Parseval's relation

$$p_k = \frac{a^2}{T_{sym}} \int_{-\infty}^{\infty} g_1^2(t) e^{-jk\omega_{sym}t} dt = \frac{a^2}{2\pi T_{sym}} \int_{-\infty}^{\infty} G_1(\omega) G_1(k\omega_{sym} - \omega) d\omega$$
- Fourier series representation of $E\{z(t)\}$

$$z_k = p_k H(k\omega_{sym}) = H(k\omega_{sym}) \frac{a^2}{2\pi T_{sym}} \int_{-\infty}^{\infty} G_1(\omega) G_1(k\omega_{sym} - \omega) d\omega$$



14 - 31

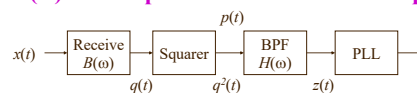
31

Symbol Clock Recovery

- With $G_1(\omega) = X(\omega) B(\omega)$
 - Choose $B(\omega)$ to pass $\pm \frac{1}{2}\omega_{sym} \rightarrow p_k = 0$ except $k = -1, 0, 1$

$$Z_k = p_k H(k\omega_{sym}) = H(k\omega_{sym}) \frac{a^2}{2\pi T_{sym}} \int_{-\infty}^{\infty} G_1(\omega) G_1(k\omega_{sym} - \omega) d\omega$$
 - Choose $H(\omega)$ to pass $\pm\omega_{sym} \rightarrow Z_k = 0$ except $k = -1, 1$

$$E\{z(t)\} = \sum_k Z_k e^{jk\omega_{sym}t} = e^{-j\omega_{sym}t} + e^{j\omega_{sym}t} = 2 \cos(\omega_{sym}t)$$
- $B(\omega)$ is lowpass filter with $\omega_{passband} = \frac{1}{2}\omega_{sym}$
- $H(\omega)$ is bandpass filter with center frequency ω_{sym}



14 - 32

32

Quadrature Amplitude Modulation (QAM) Transmitter

Prof. Brian L. Evans

Dept. of Electrical and Computer Engineering
The University of Texas at Austin

Lecture 15 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Introduction

- **Digital Pulse Amplitude Modulation (PAM)**
Modulates digital information (symbols) onto amplitude of pulse
May be later upconverted (e.g. to radio frequency)
- **Digital Quadrature Amplitude Modulation (QAM)**
Two-dimensional extension of digital PAM
Baseband signal requires sinusoidal amplitude modulation
May be later upconverted (e.g. to radio frequencies)
- **Digital QAM modulates digital information onto pulses that are modulated onto**
Amplitudes of a sine and a cosine, or equivalently
Amplitude and phase of single sinusoid

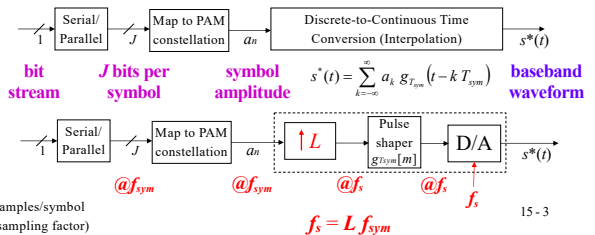
15 - 2

2

Baseband PAM Transmitter

- **Bit stream to continuous-time pulse stream**

Group stream of bits into symbols of J bits
Map symbol of J bits to unique symbol amplitude
Interpolate from discrete-time to continuous-time



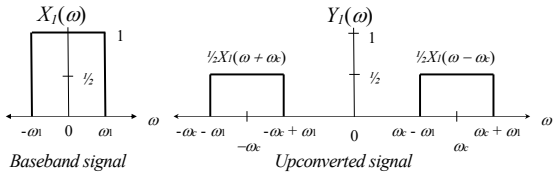
3

Amplitude Modulation by Cosine

$$y_I(t) = x_I(t) \cos(\omega_c t) \quad Y_I(\omega) = \frac{1}{2} X_I(\omega + \omega_c) + \frac{1}{2} X_I(\omega - \omega_c)$$

Assume $x_I(t)$ is an ideal lowpass signal with bandwidth ω_1
Assume $\omega_1 < \omega_c$

$Y_I(\omega)$ is real-valued if $X_I(\omega)$ is real-valued



- **Demodulation: modulation then lowpass filtering**

See Slides 1-6 and 1-7

15 - 4

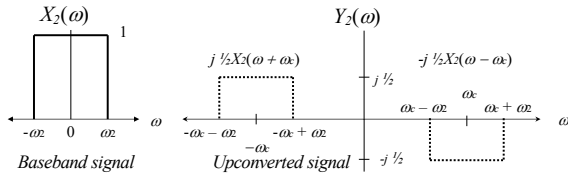
4

Amplitude Modulation by Sine

$y_2(t) = x_2(t) \sin(\omega_c t)$ $Y_2(\omega) = \frac{j}{2} X_2(\omega + \omega_c) - \frac{j}{2} X_2(\omega - \omega_c)$

Assume $x_2(t)$ is an ideal lowpass signal with bandwidth ω_2
 Assume $\omega_2 < \omega_c$

$Y_1(\omega)$ is imaginary-valued if $X_1(\omega)$ is real-valued



- Demodulation: modulation then lowpass filtering

See Slides 1-8 and 1-9

How to Use Bandwidth Efficiently?

- Sinusoidal mod doubles transmission bandwidth

Send two baseband signals in same transmission bandwidth

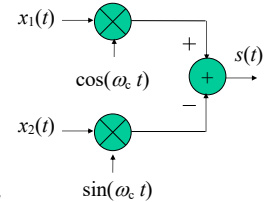
Called Quadrature Amplitude Modulation (QAM)

Used in DSL, cable, Wi-Fi, LTE

$S(\omega) = \frac{1}{2}(X_1(\omega + \omega_c) + X_1(\omega - \omega_c)) - j\frac{1}{2}(X_2(\omega + \omega_c) - X_2(\omega - \omega_c))$

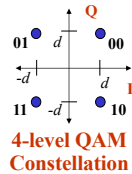
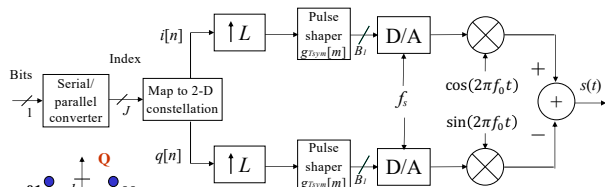
- Cosine modulated signal is in theory orthogonal to sine modulated signal at transmitter

Receiver separates $x_1(t)$ and $x_2(t)$ through demodulation



QAM Transmitter Architecture #1

- Use D/A converter for each baseband PAM channel

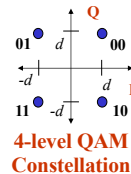
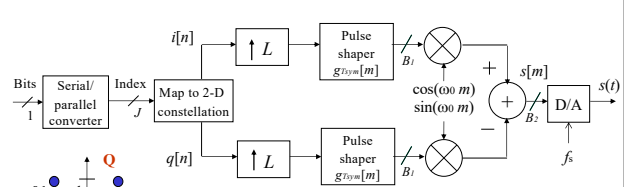


4-QAM is 2-PAM in the in-phase dimension and 2-PAM in the quadrature dimension

D/A power consumption $\propto f_s 2^{B_1}$ where $B_1 \geq \frac{J}{2}$

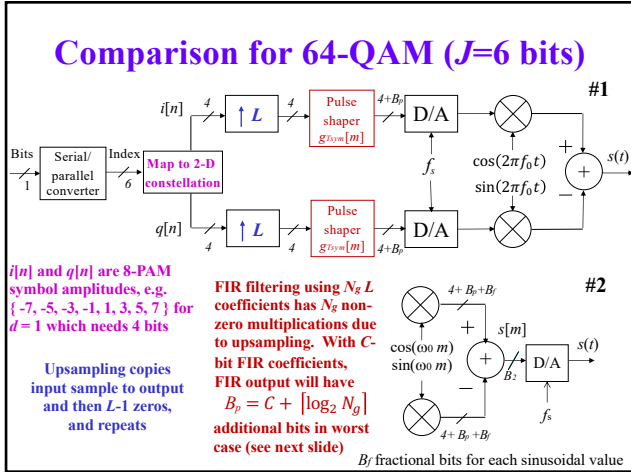
QAM Transmitter Architecture #2

- Use only one D/A converter

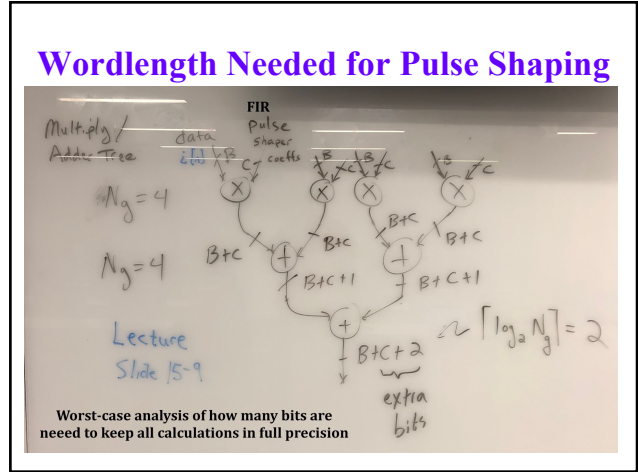


4-QAM is 2-PAM in the in-phase dimension and 2-PAM in the quadrature dimension

D/A power consumption $\propto f_s 2^{B_2}$ where $B_2 \geq J$



9



10

Performance Analysis of PAM

- If we sample matched filter output at correct time instances, nT_{sym} , without any ISI, received signal

$$x(nT_{sym}) = s(nT_{sym}) + v(nT_{sym})$$

$$v(nT) \sim N(0; \sigma^2/T_{sym})$$
 where transmitted signal is

$$s(nT_{sym}) = a_n = (2i-1)d \text{ for } i = -M/2+1, \dots, M/2$$

$$v(t)$$
 output of matched filter $G_r(\omega)$ for input of channel additive white Gaussian noise $N(0; \sigma^2)$

$$G_r(\omega)$$
 passes frequencies from $-\omega_{sym}/2$ to $\omega_{sym}/2$, where $\omega_{sym} = 2\pi f_{sym} = 2\pi / T_{sym}$
- Matched filter has impulse response $g_r(t)$

4-level PAM Constellation

15-11

11

Performance Analysis of PAM

- Decision error for inner points**

$$P_I(e) = P(|v(nT_{sym})| > d) = 2Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$
- Decision error for outer points**

$$P_{O_i}(e) = P(v(nT_{sym}) > d) = Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$

$$P_{O_i}(e) = P(v(nT_{sym}) < -d) = P(v(nT_{sym}) > d) = Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$
- Symbol error probability**

$$P(e) = \frac{M-2}{M}P_I(e) + \frac{1}{M}P_{O_i}(e) + \frac{1}{M}P_{O_o}(e) = \frac{2(M-1)}{M}Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$

8-level PAM Constellation

15-12

12

Performance Analysis of QAM

- If we sample matched filter outputs at correct time instances, nT_{sym} , without any ISI, received signal

$$x(nT_{sym}) = s(nT_{sym}) + v(nT_{sym})$$

- Transmitted signal

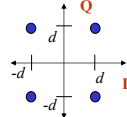
$$s(nT_{sym}) = a_n + j b_n = (2i-1)d + j(2k-1)d$$

where $i, k \in \{-1, 0, 1, 2\}$ for 16-QAM

- Noise $v(nT_{sym}) = v_I(nT_{sym}) + j v_Q(nT_{sym})$

For error probability analysis, assume noise terms independent and each term is Gaussian random variable $\sim N(0; \sigma^2/T_{sym})$

In reality, noise terms have common source of additive noise in channel



4-level QAM Constellation

15 - 13

13

Performance Analysis of 16-QAM

- Type 1 correct detection

$$P_1(c) = P(|v_I(nT_{sym})| < d \text{ \& } |v_Q(nT_{sym})| < d)$$

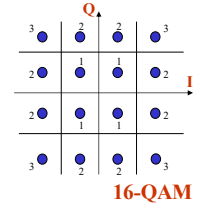
$$= P(|v_I(nT_{sym})| < d) P(|v_Q(nT_{sym})| < d)$$

Assume statistical independence of $v_I(nT_{sym})$ and $v_Q(nT_{sym})$

$$= \left(1 - P(|v_I(nT_{sym})| > d)\right) \left(1 - P(|v_Q(nT_{sym})| > d)\right)$$

$$= \left(1 - 2Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right) \left(1 - 2Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)$$

$$= \left(1 - 2Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)^2$$



16-QAM

1 - interior decision region
2 - edge region but not corner
3 - corner region

14

Performance Analysis of 16-QAM

- Type 2 correct detection

$$P_2(c) = P(v_I(nT_{sym}) < d \text{ \& } |v_Q(nT_{sym})| < d)$$

$$= P(v_I(nT_{sym}) < d) P(|v_Q(nT_{sym})| < d)$$

$$= \left(1 - Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right) \left(1 - 2Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)$$

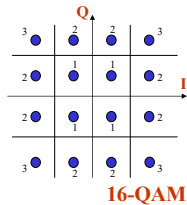
- Type 3 correct detection

$$P_3(c) = P(v_I(nT_{sym}) < d \text{ \& } v_Q(nT_{sym}) > -d)$$

$$= P(v_I(nT_{sym}) < d) P(v_Q(nT_{sym}) > -d)$$

$$= \left(1 - Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)^2$$

1 - interior decision region
2 - edge region but not corner
3 - corner region



16-QAM

15

Performance Analysis of 16-QAM

- Probability of correct detection

$$P(c) = \frac{4}{16} \left(1 - 2Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)^2 + \frac{4}{16} \left(1 - Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)^2$$

$$+ \frac{8}{16} \left(1 - Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right) \left(1 - 2Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)\right)$$

$$= 1 - 3Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right) + \frac{9}{4}Q^2\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)$$

- Symbol error probability (lower bound)

$$P(e) = 1 - P(c) = 3Q\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma\sqrt{T_{sym}}}\right)$$

- What about other rectangular QAM constellations?

16

Examples

- **Determine decision regions for the receiver**

Draw boundaries at midpoint of every pair of constellation points

- **Rectangular constellations**

PAM in I and Q dimensions

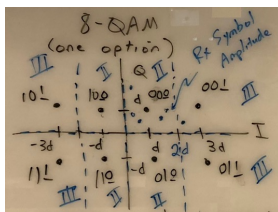
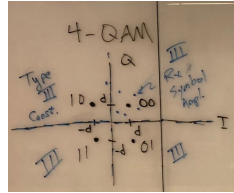
I and Q axes become boundaries

Gray coding always possible

$$P_{4QAM}(e) = 1 - P_{III}(c) = 2q - q^2$$

$$P_{8QAM}(c) = \frac{4}{8}P_{II}(c) + \frac{4}{8}P_{III}(c)$$

$$P_{8QAM}(e) = 1 - P_{8QAM}(c) = \frac{5}{2}q - \frac{3}{2}q^2$$



$$q = Q \left(\frac{d}{\sigma} \sqrt{T_{sym}} \right) \quad 15-17$$

Average Power Analysis

- Assume each symbol is equally likely
- Assume energy in pulse shape is 1
- **4-PAM constellation**

Amplitudes are in set $\{-3d, -d, d, 3d\}$

Total power $9d^2 + d^2 + d^2 + 9d^2 = 20d^2$

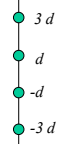
Average power per symbol $5d^2$

- **4-QAM constellation points**

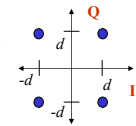
Points are in set $\{-d-jd, -d+jd, d+jd, d-jd\}$

Total power $2d^2 + 2d^2 + 2d^2 + 2d^2 = 8d^2$

Average power per symbol $2d^2$



4-level PAM Constellation



4-level QAM Constellation

17

18

Quadrature Amplitude Modulation (QAM) Receiver

Prof. Brian L. Evans

Dept. of Electrical and Computer Engineering
The University of Texas at Austin

Lecture 16 <http://www.ece.utexas.edu/~bevans/courses/realtime>

1

Introduction

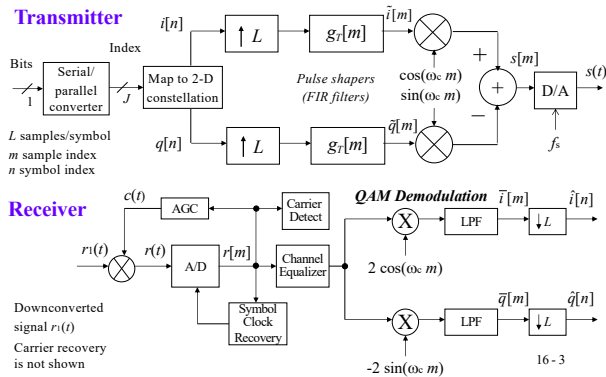
- Channel impairments**
 - Linear and nonlinear distortion of transmitted signal
 - Additive noise (often assumed to be Gaussian)
- Mismatch in transmitter/receiver analog front ends**
- Receiver subsystems to compensate for impairments**

Fading	Automatic gain control (AGC)
Additive noise	Matched filters
Linear distortion	Channel equalizer
Carrier mismatch	Carrier recovery
Symbol timing mismatch	Symbol clock recovery

16 - 2

2

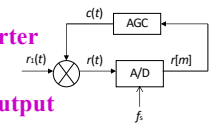
Baseband QAM



3

Automatic Gain Control

- Scales input voltage to A/D converter**
- A/D converter with 8-bit signed output**



Increase/decrease gain for low/high $r_1(t)$

When gain $c(t)$ is zero, A/D output is 0

When gain $c(t)$ is infinity, A/D output is -128 or 127

f_{-128}, f_0, f_{127} represent how frequently outputs -128, 0, 127 occur

$f_i = c_i / N$ where c_i is count of times i occurs in last N samples

Update #1: $c(t) = (1 + 2f_0 - f_{-128} - f_{127}) c(t - \tau)$

Update #2: $c(t) = \frac{2f_0 + \epsilon}{f_{-128} + f_{127} + \epsilon} c(t - \tau) = \frac{2c_0 + \epsilon N}{c_{-128} + c_{127} + \epsilon N} c(t - \tau)$


Constant $\epsilon > 0$ prevents division by zero

16 - 4

4

Channel Equalizer

- **Mitigates linear distortion in channel**
- **When placed after A/D converter**
Time domain: shortens channel impulse response
Frequency domain: compensates channel distortion over entire discrete-time frequency band instead of transmission band
- **Ideal channel**
 Cascade of delay Δ and gain g
Impulse response: impulse delayed by Δ with amplitude g
Frequency response: allpass and linear phase (no distortion)
 Undo effects by discarding Δ samples and scaling by $1/g$

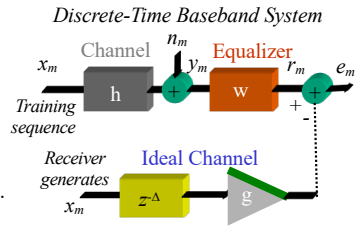


16 - 5

5

Channel Equalizer

- **IIR equalizer**
 Ignore noise n_m
 Set error e_m to zero
 $H(z) W(z) = g z^{-\Delta}$
 $W(z) = g z^{-\Delta} / H(z)$
- **FIR equalizer**
 Adapt equalizer coefficients when transmitter sends training sequence to reduce measure of error, e.g. square of e_m

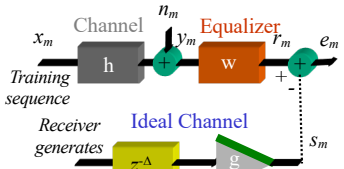


16 - 6

6

Adaptive FIR Channel Equalizer

- **Simplest case: $w[m] = \delta[m] + w_1 \delta[m-1]$**
 Two real-valued coefficients w/ first coefficient fixed at one
- **Derive update equation for w_1 during training**



$$e[m] = r[m] - s[m]$$

$$s[m] = g x[m - \Delta]$$

$$r[m] = y[m] + w_1 y[m-1]$$

$$J_{LMS}[m] = \frac{1}{2} e^2[m]$$

$$w_1[m+1] = w_1[m] - \mu \left. \frac{\partial J_{LMS}[m]}{\partial w_1} \right|_{w_1=w_1[m]}$$

$$w_1[m+1] = w_1[m] - \mu e[m] y[m-1]$$

Using least mean squares (LMS)
 Step size $0 < \mu < 1$

16 - 8

7

Adaptive FIR Channel Equalizer

- **General case: Adapt N real coefficients w_0, w_1, \dots, w_{N-1}**
 $r[m] = w_0 y[m] + w_1 y[m-1] + \dots + w_{N-1} y[m-(N-1)]$
- **Derive coefficient update equations during training**
Ideal: $s[m] = g x[m - \Delta]$ **Error:** $e[m] = r[m] - s[m]$ $J_{LMS}[m] = \frac{1}{2} e^2[m]$
 $w_0[m+1] = w_0[m] - \mu \left. \frac{\partial J_{LMS}[m]}{\partial w_0} \right|_{w_0=w_0[m]} = w_0[m] - \mu e[m] \left. \frac{\partial r[m]}{\partial w_0} \right|_{w_0=w_0[m]}$

$\begin{bmatrix} \bar{w}_0[m] \\ w_0[m] \\ w_1[m] \\ \vdots \\ w_{N-1}[m] \end{bmatrix}$	$\begin{bmatrix} w_0[m+1] = w_0[m] - \mu e[m] y[m] \\ w_1[m+1] = w_1[m] - \mu e[m] y[m-1] \\ \vdots \\ w_{N-1}[m+1] = w_{N-1}[m] - \mu e[m] y[m-(N-1)] \end{bmatrix}$	$\begin{bmatrix} \bar{y}[m] \\ y[m] \\ y[m-1] \\ \vdots \\ y[m-(N-1)] \end{bmatrix}$
--	---	--

- **Vector form:** $\bar{w}[m+1] = \bar{w}[m] - \mu e[m] \bar{y}[m]$

16 - 8

8

Carrier Detection

- **Detect energy of received signal (always running)**
 $p[m] = c p[m-1] + (1-c)r^2[m]$
 c is a constant where $0 < c < 1$ and $r[m]$ is received signal
 Let $x[m] = r^2[m]$. What is the transfer function?
 What values of c to use?
- **If receiver is not currently receiving a signal**
 If energy detector output is larger than a large threshold, assume receiving transmission
- **If receiver is currently receiving signal, then it detects when transmission has stopped**
 If energy detector output is smaller than a smaller threshold, assume transmission has stopped

16-9

9

Symbol Clock Recovery

- **Two single-pole bandpass filters in parallel**
 One tuned to upper Nyquist frequency $\omega_u = \omega_c + 0.5 \omega_{sym}$
 Other tuned to lower Nyquist frequency $\omega_l = \omega_c - 0.5 \omega_{sym}$
 Bandwidth is $B/2$ (100 Hz for 2400 baud modem)
- **A recovery method**
 Multiply upper bandpass filter output with conjugate of lower bandpass filter output and take the imaginary value
 Sample at symbol rate to estimate timing error τ
 $v[n] = \sin(\omega_{sym} \tau) \approx \omega_{sym} \tau$ when $|\omega_{sym} \tau| \ll 1$
 Smooth timing error estimate to compute phase advancement
 $p[n] = \beta p[n-1] + \alpha v[n]$

Pole locations?

See Reader handout M

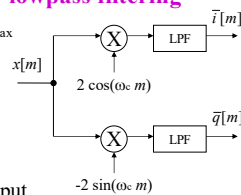
Lowpass IIR filter

16-10

10

Baseband QAM Demodulation

- **Recovers baseband in-phase/quadrature signals**
- **Assumes perfect AGC, equalizer, symbol recovery**
- **QAM modulation followed by lowpass filtering**
 Receiver $f_{max} = 2f_c + B$ and $f_s > 2f_{max}$
- **Lowpass filter has other roles**
 Matched filter
 Anti-aliasing filter
- **Matched filters**
 Maximize SNR at downsampler output
 Hence minimize symbol error at downsampler output



16-11

11

Baseband QAM Demodulation

- **QAM baseband signal** $x[m] = i[m] \cos(\omega_c m) - q[m] \sin(\omega_c m)$
- **QAM demodulation**

Modulate and lowpass filter to obtain baseband signals


$$\begin{aligned} \hat{i}[m] &= 2x[m] \cos(\omega_c m) = 2i[m] \cos^2(\omega_c m) - 2q[m] \sin(\omega_c m) \cos(\omega_c m) \\ &= \underbrace{i[m]}_{\text{baseband}} + \underbrace{i[m] \cos(2\omega_c m) - q[m] \sin(2\omega_c m)}_{\text{high frequency component centered at } 2\omega_c} \end{aligned}$$

$$\begin{aligned} \hat{q}[m] &= -2x[m] \sin(\omega_c m) = -2i[m] \cos(\omega_c m) \sin(\omega_c m) + 2q[m] \sin^2(\omega_c m) \\ &= \underbrace{q[m]}_{\text{baseband}} - \underbrace{i[m] \sin(2\omega_c m) - q[m] \cos(2\omega_c m)}_{\text{high frequency component centered at } 2\omega_c} \end{aligned}$$

$$\cos^2 \theta = \frac{1}{2}(1 + \cos 2\theta) \quad 2 \cos \theta \sin \theta = \sin 2\theta \quad \sin^2 \theta = \frac{1}{2}(1 - \cos 2\theta)$$

16-12

12



Review for Midterm #2

Prof. Brian L. Evans

EE 445S Real-Time Digital Signal Processing Laboratory

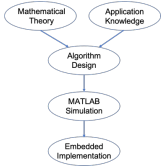
THE UNIVERSITY OF TEXAS AT AUSTIN

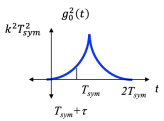
10 January 2025

1

Outline

- Course objectives
- Design flow
- Signal processing
- Communication systems
- Communication system tradeoffs
- Symbol timing recovery

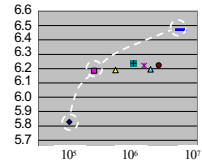




2

Course Overview

- **Objectives**
Build intuition for signal processing concepts
Explore design tradeoffs in *signal quality vs. run-time implementation complexity*
- **Asymmetric Digital Subscriber Line (ADSL) Receiver Design**
FIR channel equalizer gives up to 10x increase in bit rate vs. not having one
ADSL transmitter sends training data
Eight adaptive channel equalizer design methods shown on right
Three have best tradeoffs in bit rate vs. run-time computational complexity

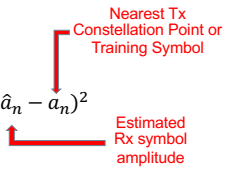


[Data from Figs. 6 & 7 in B. L. Evans et al., "Unification and Evaluation of Equalization Structures...", *IEEE Trans. Sig. Proc.*, 2005]

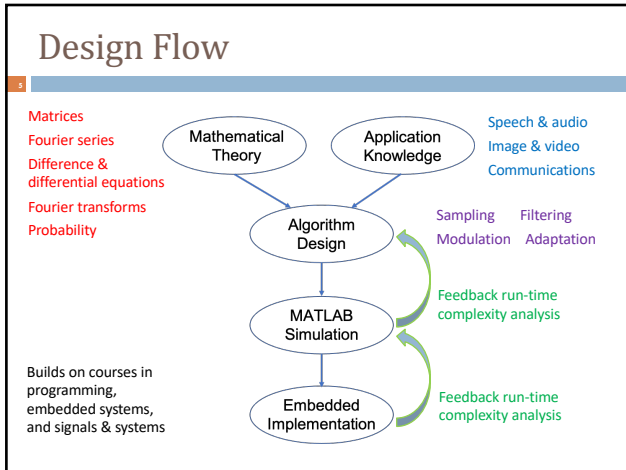
3

Signal Quality Measures

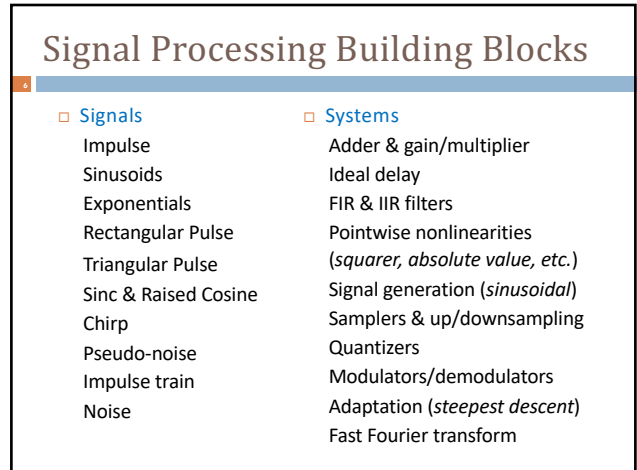
- **Signal-to-noise ratio:** $SNR = \frac{\text{Signal Power}}{\text{Noise Power}}$
Thermal noise (modeled as Gaussian) in Rx analog/RF front end
Quantization noise (modeled as uniform) in data converters
- **Communication Systems**
PAM/QAM Bit rate: $J f_{sym}$
PAM Bit error rate $\propto Q(C_0 \sqrt{SNR})$
PAM Error vector magnitude squared $(\hat{a}_n - a_n)^2$
- **Data Converters**
Signal-to-noise ratio
Total harmonic distortion plus noise (lecture slide 8-14)



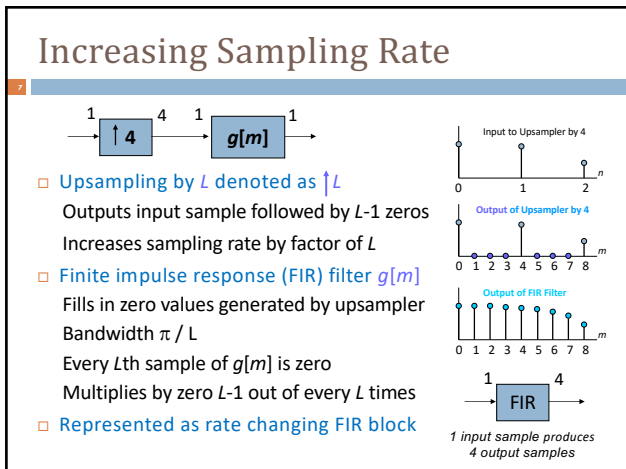
4



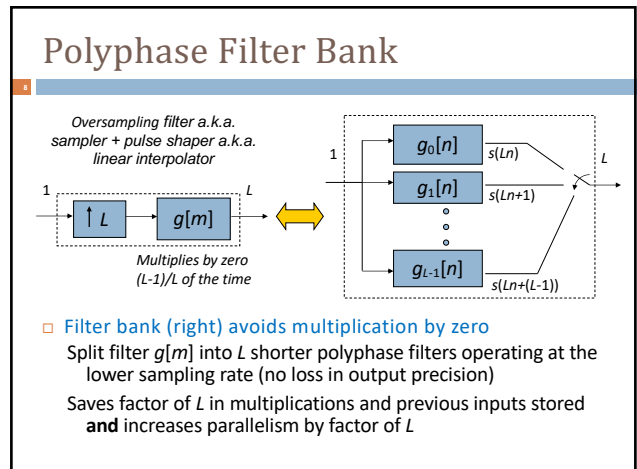
5



6



7



8

Decreasing Sampling Rate

- Finite impulse response (FIR) filter $g[m]$
 - Lowpass filter to enforce sampling theorem
 - Bandwidth is π / L
- Downsampling by L denoted as $\downarrow L$
 - Inputs L samples
 - Outputs first sample and discards $L-1$ samples
 - Decreases sampling rate by factor of L
- Represented as rate changing FIR block
 - 4 input samples produce 1 output sample

9

Polyphase Filter Bank Form

Undersampling filter a.k.a. Matched filter + sampling a.k.a. linear decimator

Outputs discarded $(L-1)/L$ of the time

$$y[1] = v[L] = h[0] s[L] + h[1] s[L-1] + \dots + h[L-1] s[1] + h[L] s[0]$$

- Filter bank only computes values output by downsampler
 - Split filter $h[m]$ into L shorter polyphase filters operating at the lower sampling rate (no loss in output precision)
 - Reduces multiplications and increases parallelism by factor of L

10

Communication Systems

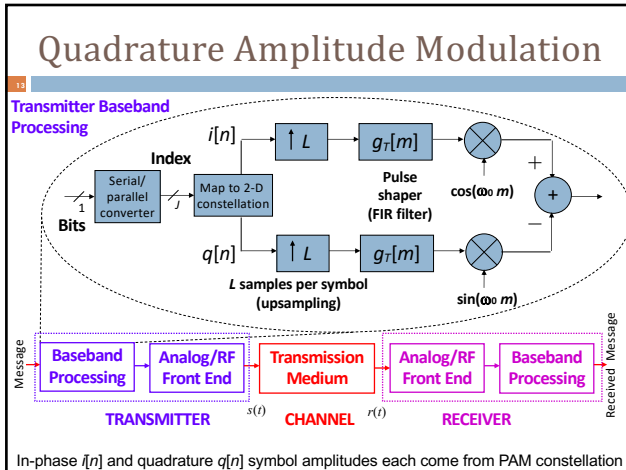
- Message signal is information to be sent
 - Information may be voice, music, images, video, data
 - Low frequency (baseband) signal centered at DC
- Transmitter baseband processing includes lowpass filtering to enforce transmission band
- Transmitter analog/RF front end includes digital-to-analog converter, analog/RF upconverter, and transmit filter

11

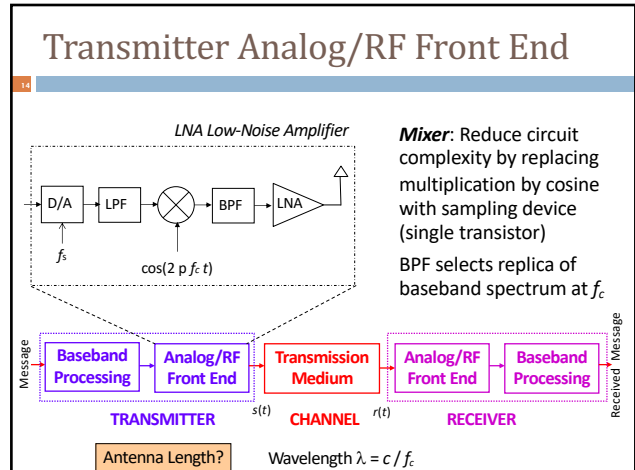
Communication Systems

- Propagating signals experience attenuation & spreading w/ distance
 - Model the environment
- Receiver analog/RF front end includes receive filter, carrier recovery, analog/RF downconverter, automatic gain control and analog-to-digital converter
- Receiver baseband processing extracts/enhances baseband signal

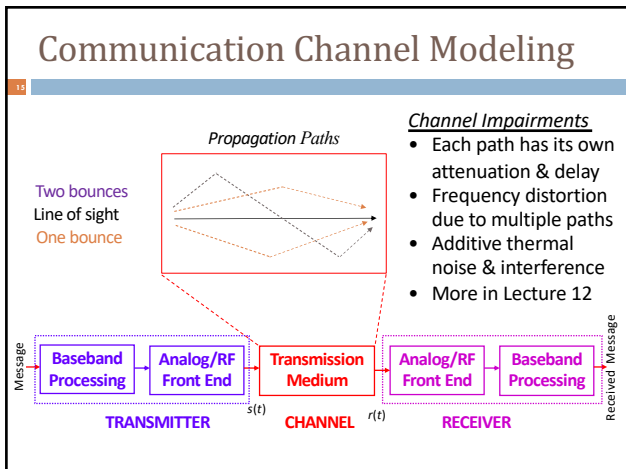
12



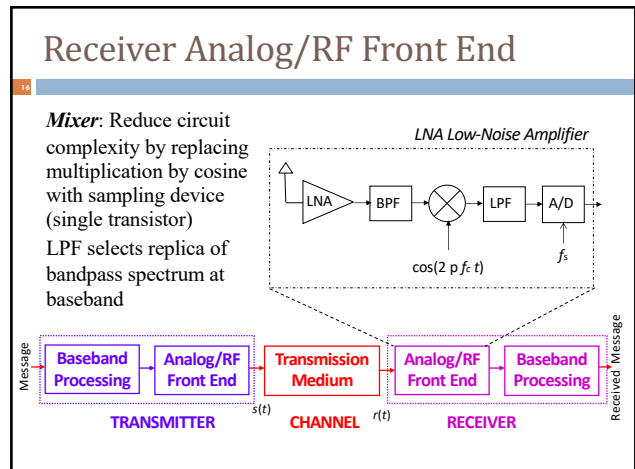
13



14



15



16

Baseband Equivalent Channel

- Baseband discrete-time channel model
 - Combines transmitter carrier circuits, physical channel and receiver carrier circuits
 - One model uses cascade of gain, FIR filter, and additive noise
 - Fast simulation

17

Quad. Amplitude Demodulation

Receiver Baseband Processing

TRANSMITTER: Baseband Processing, Analog/RF Front End, Transmission Medium, Analog/RF Front End, Baseband Processing

CHANNEL: s(t), r(t)

RECEIVER: Baseband Processing, Analog/RF Front End, Baseband Processing

18

QAM Signal Quality

- Assumptions
 - Each symbol is equally likely
 - Channel only consists of additive noise
 - White Gaussian noise with zero mean and variance σ^2 in in-phase and quadrature components
 - Total noise power of $2\sigma^2$
 - Carrier frequency and phase recovery
 - Symbol timing recovery
- Probability of symbol error
 - Constellation spacing of $2d$
 - Symbol duration of T_{sym}

$$P(e) = 3Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$

$\frac{d}{\sigma}\sqrt{T_{sym}}$ is proportional to $\sqrt{\text{SNR}}$

19

Communication System Tradeoffs

What happens to signal quality and run-time implementation complexity if PAM system parameter in first column increases? Indicate **increase**, **decrease**, or **blank** to mean no effect

Fall 2020 Midterm #2.4

Assume other parameters in first column are not changing

Parameter	Transmission Bandwidth	Bit Rate	Symbol Error Rate	Tx Power Consumption	Run-Time Complexity
B bits in A/D output & D/A input		?	?		increase
2d constellation spacing in Volts			decrease	increase	
f_{sym} symbol rate in Hz	increase	increase	increase		increase
J bits/symbol		increase	increase	increase	increase
L samples/symbol			decrease		increase
N_g symbol periods in pulse shape			decrease		increase

20

Spring 2021 Midterm #2.1

Symbol Timing Recovery

- Consider a baseband PAM system using a rect. pulse shape
- Matched filter impulse response is also a rectangular pulse

Tx Channel

Pulse shape

Rx

Matched filter

k can be negative

when there is no noise

- Receiver needs to find the symbol timing offset τ
- Develop adaptive method to update τ in n th symbol period

21

Spring 2021 Midterm #2.1

Symbol Timing Recovery

- Maximize power at received symbol amplitude $y(nT_{sym} + \tau)$
- Objective function: $J(y(t)) = \frac{1}{2} y^2(t)$

$$y(t) = \underbrace{g(t)}_{g_0(t)} * \underbrace{h(t)}_{n(t)} + w(t) * h(t)$$

$n(t)$ is Gaussian random signal with 0 mean and variance $\frac{\sigma^2}{T_{sym}}$

Average noise power $\frac{\sigma^2}{T_{sym}}$ is constant regardless of sampling time

$$\tau[n+1] = \tau[n] + \mu \left. \frac{d}{d\tau} J(y(nT_{sym} + \tau)) \right|_{\tau=\tau[n]}$$

$$\tau[n+1] = \tau[n] + \mu \left. \frac{d}{d\tau} (y(nT_{sym} + \tau))^2 \right|_{\tau=nT_{sym} + \tau[n]}$$

22

Spring 2021 Midterm #2.1

Symbol Timing Recovery

- A differentiator circuit can be as implemented as an RC circuit where the output is tapped across the resistor
- RL circuit where the output is tapped across the inductor
- For $h(t)$ being a rectangular pulse, we simplify derivative of $y(t)$ by using a linear time-invariant model of differentiation with impulse response $v(t)$ per JSK Appendix G.2:

$$\frac{d}{dt} y(t) = v(t) * (h(t) * x(t)) = (v(t) * h(t)) * x(t) = (\delta(t) - \delta(t - T_{sym})) * x(t) = x(t) - x(t - T_{sym})$$

- How to handle many symbols?
Send bits 11
What is $y^2(t)$?

23

Spring 2021 Midterm #2.1

Data Conversion

- Analog-to-Digital**

- Quantize to B bits
- Quantization error = noise
- $SNR_{dB} \approx C_0 + 6.02 B$
- Dynamic range $\approx SNR$
- $SNR_{dB} = 10 \log_{10} \frac{\text{Signal Power}}{\text{Noise Power}}$
- $SNR_{dB} = 10 \log_{10} P^{signal} - 10 \log_{10} P^{noise}$
- $SNR_{dB} = P_{dB}^{signal} - P_{dB}^{noise}$

- Digital-to-Analog**

- A/D and D/A lowpass filter
- $f_{stop} < \frac{1}{2} f_s$
- $f_{pass} \leq 0.9 f_{stop}$
- $A_{stop} = SNR_{dB}$
- $A_{pass} = 20 \log_{10}(1 - \Delta)$ where Δ is quantizer step size: $\frac{2m_{max}}{2^B - 1}$
- m_{max} is max quantizer voltage

24

ECE445S Real-Time Digital Signal Processing Lab (Spring 2024)

Lecture:	MW 10:30am–12:00pm in ECJ 1.312
Instructor:	Prof. Brian L. Evans, bevans@ece.utexas.edu (He/His/Him)
Office Hours:	MW 2:00–3:30pm in EER 6.882 and immediately after lecture
Lab Sections:	M 6:30–9:30pm (Barati) T 5:00–8:00pm (Barati)
(EER 1.810)	W 6:30–9:30pm (Eun) F 1:00–4:00pm (Eun)
TA Office Hours:	Mr. Faraz Barati (he/him), W 4:30–6:00pm, TH 3:00–4:30pm
	Mr. Yongjin Eun (he/him), TH 4:30–6:00pm, F 5:00–6:30pm
TA E-mail:	faraz.barati@utmail.utexas.edu and ye2259@utmail.utexas.edu
Course Web Page:	http://users.ece.utexas.edu/~bevans/courses/realtime

This course covers discrete-time signal processing algorithms and translating them into system simulations and embedded real-time software. A goal is to understand design tradeoffs in signal quality vs. run-time implementation complexity. Applications include audio, communications, and image processing. Great way to build breadth and depth in signal processing.

Prerequisites

ECE312/312H C Programming and 319K/319H Intro to Embedded Systems with a grade of at least C- in each; BME343 or ECE313 Signals and Systems with a grade of at least C-; credit with a grade of at least C- or registration for BME/ECE333T Engineering Communication; and credit with a grade of at least C- or registration for BME335/ECE351K Probability.

Topical Outline

Digital signal processing algorithms; simulation and real-time implementation of audio and communication systems; filters; pulse shaping and matched filters; modulation and demodulation; adaptive filters; carrier recovery; symbol synchronization; equalization; quantization.

Order of Lecture Topics

Sinusoidal Generation - Signals & Systems - Finite Impulse Response Filters - Infinite Impulse Response Filters - Sampling & Aliasing - Interpolation & Pulse Shaping - Digital Pulse Amplitude Modulation (PAM) - Channel Impairments - Matched Filtering - Digital Quadrature Amplitude Modulation (QAM) - Quantization - Data Conversion

Order of Laboratory Topics

Simulation Tools - Sinusoidal Generation - Filters - Pseudonoise Sequences & Data Scramblers - Digital PAM Transceivers - Digital QAM Transceivers - Guitar Effects

Required Textbooks

1. C. R. Johnson Jr., W. A. Sethares and A. G. Klein, *Software Receiver Design*, Oct. 2011, ISBN 978-0521189446. Paperback. Matlab code. Free Download from UT Libraries.
2. C. Unsalan, M. E. Yucel, and H. D. Gurhan, *Digital Signal Processing using Arm Cortex-M based Microcontrollers*, Arm Educational Media, Sep. 14, 2018, ISBN 978-1-911531-15-9, 354 pages. Request free download.
3. B. L. Evans, *ECE445S Real-Time DSP Lab Course Reader*. Free download.

Supplemental Text

4. J. H. McClellan, R. W. Schafer, and M. A. Yoder, *Signal Processing First*, ISBN 978-0130909992, 2003. On-line demonstrations.

Grading

14% Homework, 20% Midterm #1, 20% Midterm #2, 5% In-Lecture Work, 5% Pre-lab quizzes, 36% Lab reports. Lab work is performed in person; lab attendance and participation is required and comprises 25% of the lab report grade. In-class midterm exams are on Wednesday, Mar. 6th, and Monday, Apr. 29th. There is no final exam.

Assignment of letter grades is below; no rounding will be applied. Although there aren't any extra credit assignments, other options for flexibility are described next.

90.00–100.00	A	86.67–89.99	A-	83.34–86.66	B+
80.00–83.33	B	76.67–79.99	B-	73.34–76.66	C+
70.00–73.33	C	66.67–69.99	C-	etc.	

Flexibility

Each student will be able to drop the lowest grade in each of the following categories: homeworks, in-lecture assignments, pre-lab quizzes, and lab reports. This will give flexibility when something unexpected happens not covered by these policies, and allow you to strategically use grade drops to balance your course workload and other commitments. Please let me know if you face difficulties this semester in accessing course resources or completing work.

Critical Thinking

I am interested in the critical thinking process you use when solving a problem. Please provide rationale and justification. For homework and exams, I use the following rubric:

- 3/3 points. Correct rationale/justification and correct results
- 2/3 points. Correct rationale/justification but incorrect results.
- 1/3 points. Some progress; errors in rationale/justification even if results are correct.
- 0/3 points. Answer is blank, or only repeated/reworded the question, or gave an incorrect answer without justification, or never really “got going” towards a solution.

Regrade Requests

Request for regrading an assignment must be made in writing within one (1) week of the graded assignment being made available to students in the class.

Learning and Growth

Throughout the course, your learning and growth in theory and practice of the engineering profession are important to me. We all need accommodations because we all learn differently, and the current pandemic makes accommodations all the more important. If there are aspects of this course that prevent you from learning or exclude you, please let me know as soon as possible. Together we will develop strategies to meet your needs and course requirements. I also encourage you to reach out to the resources available through UT. Many are on this syllabus. I am happy to connect you with a person or Center if you would like.

Use of AI Generated Content

This course encourages students to explore the use of generative artificial intelligence (GAI) tools such as ChatGPT for all assignments and assessments, except for the two midterm

exams. Any such use must be appropriately acknowledged and cited. It is each student’s responsibility to assess the validity and applicability of any GAI output that is submitted; you bear the final responsibility. Violations of this policy will be considered violations of the campus academic integrity. We draw your attention to the fact that different classes at UT Austin could implement different AI policies, and it is the student’s responsibility to conform to expectations for each course. The use of generative artificial intelligence tools on two midterm exams is prohibited.

Examples of Content Input-and Output AI Generation with Associated Tools:

Content Input-Output	Tool
Text In, Text Out	ChatGPT, GPT-3, GPT-4
Text In, Image Out	DALL-E, Midjourney, Stable Diffusion
Image In, Text Out	GPT-4, BLIP
Text In, Video Out	RunwayML, Decorum
Multimedia In, Image Out	Stable Diffusion, Midjourney, Deforum
Parameters In, Audio (Music) Out	Boomy, AIVA

Academic Integrity

Discussion of homework questions is encouraged. Please submit your own independent homework solutions, lab 1 and 7 reports, pre-lab quizzes, and midterm exams. For labs 2–6, students work in teams of two, and each team would submit one report. Collaboration is allowed for in-lecture assignments, but each person must submit their own assignment. For all homework, in-lecture assignments, and lab reports, cite your sources of information. When providing answers on midterm exams, be sure to cite your sources of information as part of your justification for your answers.

Each student is expected to abide by the UT Honor Code: “As a student of The University of Texas at Austin, I shall abide by the core values of the University and uphold academic integrity.” If you use words or ideas that are not your own (or that you have used in a previous class), you must cite your sources. Otherwise, you might be in violation of the university’s academic integrity policies. Please see Student Conduct and Academic Integrity.

Texas Senate Bill 17

Texas Senate Bill 17, the recent law that outlaws diversity, equity, and inclusion programs at public colleges and universities in Texas, does not in any way affect content, instruction or discussion in a course at public colleges and universities in Texas. Expectations and academic freedom for teaching and class discussion have not been altered post-SB 17, and students should not feel the need to censor their speech pertaining to topics including race and racism, structural inequality, LGBTQ+ issues, or diversity, equity, and inclusion.

Use of Electronics

Please focus the use of electronics on the content during lecture and lab sessions to maximize your own learning and support the learning environment for others.

Video Recordings

Video recording of class activities are reserved for students and TAs in this class only for educational purposes and are protected by FERPA laws if any students are identifiable in

the video. Video recordings should not be shared outside the class in any form. Students violating this university policy could face misconduct proceedings.

Food Pantry and Career Clothes Closet

UT Outpost (UA9 Building, 2609 University Avenue) is equipped with a food pantry, and a career clothing closet, to ensure every Longhorn has access to professional clothes for job and internship interviews. Emergencies and financial hardships can interfere with student success beyond the classroom, and this program will serve as an additional resource for students. This resource is from Student Emergency Services in the Office of the Dean of Students.

Disability and Access

The university is committed to creating an accessible and inclusive learning environment consistent with university policy and federal and state law. Please let me know if you experience any barriers to learning so I can work with you to ensure you have equal opportunity to participate fully in this course. If you are a student with a disability, or think you may have a disability, and need accommodations please contact Disability & Access (formerly Services for Students with Disabilities). Here are some examples of the types of diagnoses and conditions that can be considered disabilities: Attention-Deficit/Hyperactivity Disorders (ADHD), Autism, Blind & Visually Impaired, Brain Injuries, Deaf & Hard of Hearing, Learning Disabilities, Medical Disabilities, Physical Disabilities, Psychological Disabilities and Temporary Disabilities. Please refer to the D&A website for contact and more information. If you are already registered with D&A, please deliver your Accommodation Letter to me as early as possible in the semester so we can discuss your approved accommodations and needs in this course.

Mental Health Counseling

College can be stressful and sometimes we need a little help. Luckily, we have a wealth of resources and dedicated people ready to assist you, and treatment does work. The Counseling and Mental Health Center provides counseling, psychiatric, consultation, and prevention services that facilitate academic and life goals and enhance personal growth and well-being. Counselors are available Monday-Friday 8am-5pm by phone (512-471-3515) and Zoom.

Alternatively, you can talk to Ms. Alexandra Okeke, LPC, right here in the College of Engineering. Ms. Okeke is our CARE Counselor and she can be reached at 512-471-3741.

If you are experiencing a mental health crisis (e.g. depression or anxiety), please call the Mental Health Center Crisis line at 512-471-CALL(2255). Call even if you aren't sure you're in a full-blown crisis, but sincerely need help. Staff are there to help you.

Student Rights and Responsibilities

- You have a right to a learning environment that supports mental and physical wellness.
- You have a right to respect.
- You have a right to be assessed and graded fairly.
- You have a right to freedom of opinion and expression.
- You have a right to privacy and confidentiality.
- You have a right to meaningful and equal participation, to self-organize groups to improve your learning environment.

- You have a right to learn in an environment that is welcoming to all people. No student shall be isolated, excluded or diminished in any way.

With these rights come responsibilities, you are responsible for

- taking care of yourself, managing your time, and communicating with the teaching team and others if things start to feel out of control or overwhelming.
- acting in a way worthy of respect and respectful of others.
- creating an inclusive environment and speaking up when someone is excluded.
- holding yourself accountable to these standards, holding each other to these standards, and holding the teaching team accountable as well.

Your experience with this course is directly related to the quality of the energy that you bring to it, and your energy shapes the quality of your peers' experiences.

Personal Pronoun Use

Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, culture, religion, politics, sexual orientation, gender, gender expression, gender variance, and nationalities. Class rosters are provided to the instructor with the student's legal name, unless they have added a "preferred name" with the Gender and Sexuality Center. Canvas provides an opportunity to select a pronoun preference. I will gladly honor your request to address you by a name that is different from what is on the roster, and by the gender pronouns you use (she/he/they/ze, etc).

Official Correspondence

UT Austin considers e-mail as an official mode of university correspondence. You are responsible for following course-related information on the course Canvas site.

Q Drop Policy

If you would like to drop a class after the 12th class day, you'll need to execute a Q drop before the Q-drop deadline, which is Monday, April 4th. Under Texas law, you are only allowed six Q drops while you are in college at any public Texas institution. More information.

Religious Holy Days

In accordance with section 51.911 of the Texas Education code and University policies on class attendance, a student who misses classes or other required activities, including examinations, for the observance of a religious holy day should inform the instructor as far in advance of the absence as possible so that arrangements can be made to complete an assignment within a reasonable period after the absence. A reasonable accommodation does not include substantial modification to academic standards, or adjustments of requirements essential to any program of instruction. Students and instructors who have questions or concerns about academic accommodations for religious observance or religious beliefs may contact the Office for Inclusion and Equity. The University does not maintain a list of religious holy days.

Absence for Military Service

In accordance with section 51.9111 of the Texas Education code and University policies on class attendance, a student is excused from attending classes or engaging in other required activities, including exams, if they are called to active military service of a reasonably brief

duration. The maximum time for which the student may be excused has been defined by the Texas Higher Education Coordinating Board as “no more than 25 percent of the total number of class meetings or the contact hour equivalent (not including the final examination period) for the specific course or courses in which the student is currently enrolled at the beginning of the period of active military service.” The student will be allowed a reasonable time after the absence to complete assignments and take exams.

https://healthyhorns.utexas.edu/coronavirus_vaccination.html,

Safety Information

If you have concerns about the safety or behavior of students, TAs, Professors, or others, call the Behavioral Concerns Advice Line at 512-232-5050. Your call can be anonymous. If something doesn't feel right, it probably isn't. Trust your instincts and share your concerns.

Occupants of buildings are required to evacuate buildings when a fire alarm is activated. Alarm activation or announcement requires exiting and assembling outside.

- Familiarize yourself with all exit doors of each classroom and building you may occupy. The nearest exit door may not be the one you used when entering the building.
- Students requiring assistance in evacuation shall inform their instructor in writing during the first week of class.
- In the event of an evacuation, follow the instruction of faculty or class instructors. Do not re-enter a building unless given instructions by the following: Austin Fire Department, UT Austin Police Department, or Fire Prevention Services.
- Information regarding emergency evacuation routes and emergency procedures.

More safety information.

Sanger Learning Center

More than one-third of undergraduates use the Sanger Learning Center each year to improve their academic performance. All students are welcome to join their classes and workshops and make appointments for their private learning specialists, peer academic coaches, and tutors. For more information, see the Sanger Web site or call 512-471-3614 (JES A332).

Title IX Reporting

Title IX is a federal law that protects against sex and gender-based discrimination, sexual harassment, sexual assault, sexual misconduct, dating/domestic violence and stalking at federally funded educational institutions. UT Austin is committed to fostering a learning and working environment free from discrimination in all its forms where all students, faculty, and staff can learn, work, and thrive. When sexual misconduct occurs in our community, the university can:

1. Intervene to prevent harmful behavior from continuing or escalating.
2. Provide support and remedies to students and employees who have experienced harm or have become involved in a Title IX investigation.
3. Investigate and discipline violations of the university's relevant policies.

Faculty members and certain staff members are considered “Responsible Employees” or “Mandatory Reporters,” which means that they are required to report violations of Title IX to the Title IX Coordinator at UT Austin. **I am a Responsible Employee and must report any Title IX related incidents** that are disclosed in writing, discussion, or one-on-one. Before talking with me, or with any faculty or staff member about a Title IX related incident, be sure to ask whether they are a responsible employee. If you want to speak with someone for support or remedies without making an official report to the university, email advocate@austin.utexas.edu. For more info about reporting options and resources, visit the campus resources page or e-mail the Title IX Office at titleix@austin.utexas.edu.

Campus Carry

“The University of Texas at Austin is committed to providing a safe environment for students, employees, university affiliates, and visitors, and to respecting the right of individuals who are licensed to carry a handgun as permitted by Texas state law.” More information.

Land Acknowledgment

I would like to acknowledge that we are meeting on the Indigenous lands of Turtle Island, the ancestral name for what now is called North America. Moreover, I would like to acknowledge the Alabama-Coushatta, Caddo, Carrizo/Comecrudo, Coahuiltecan, Comanche, Kickapoo, Lipan Apache, Tonkawa and Ysleta Del Sur Pueblo, and all the American Indian and Indigenous Peoples and communities who have been or have become a part of these lands and territories in Texas. (Pronunciation guide)

References

In the section **Use of AI Generated Content**, the wording is from Harvard University, and the table of Generative AI tools is from the UT AI Tools Taskforce - Arts & Design Subcommittee. This syllabus uses wording suggested by Prof. Mary Steinhardt and effective syllabus template from the Faculty Innovation Center at UT Austin. The above Land Acknowledgment was drafted by a faculty Committee on Land Acknowledgment and passed by the UT Austin Faculty Council on September 21, 2020, and approved by the UT Austin President.

Handout B: Instructional Staff and Web Resources

1 Background of the Instructors

Brian L. Evans is Professor of Electrical and Computer Engineering at UT Austin. He is an IEEE Fellow “for contributions to multicarrier communications and image display”. At the undergraduate level, he teaches Linear Systems and Signals and Real-Time Digital Signal Processing Lab. His BSEECs (1987) degree is from the Rose-Hulman Institute of Technology, and his MSEE (1988) and PhDEE (1993) degrees are from the Georgia Institute of Technology. He joined UT Austin in 1996. His first programming experience on digital signal processors was in Spring of 1988.

Teaching assistants (TAs) will run lab sections, grade lab reports, answer e-mail and hold office hours. TAs are Mr. Yunseong Cho and Mr. Dan Jacobellis. Mr. Cho researches in next-generation cellular communication systems, and Mr. Jacobellis researches acoustics, signal processing, pattern recognition, and parallel computing. An undergraduate grader will grade homework assignments.

2 Supplemental Information

Wireless Networking & Communications Seminars.

You can search Google scholar to find papers and patent applications on the topic.

Sometimes, an article found on Google scholar is only available through a specific database, e.g. IEEE Explore. You can access these databases from an on-campus computer. If you are off campus, then you can access these databases by first connecting to www.lib.utexas.edu, then selecting the database under Research Tools, and finally logging in using your UT EID.

INDUSTRIAL

- *Circuit Cellar Magazine* <http://www.circuitcellar.com>
- *Electronic Design Magazine* <http://electronicdesign.com>
- *Embedded Systems Design Magazine* <http://www.eetimes.com/design/embedded>
- *Inside DSP* <http://www.bdti.com/insideDSP>
- *Sensors Magazine* <http://www.sensorsmag.com>
- *Sensors & Transducers Journal* http://www.sensorsportal.com/HTML/DIGEST/New_Digest.htm

ACADEMIC

- *IEEE Communications Magazine*
- *IEEE Computer Magazine*
- *EURASIP Journal on Advances in Signal Processing*
- *IEEE Signal Processing Magazine*
- *IEEE Transactions on Communications*
- *IEEE Transactions on Computers*
- *IEEE Transactions on Signal Processing*
- *Journal on Embedded Systems*
- *Proc. IEEE Real-Time Systems Symposium*
- *Proc. IEEE Workshop on Signal Processing Systems*
- *Proc. Int. Workshop on Code Generation for Embedded Processors*

3 Web Resources (by Ms. Ankita Kaul)

MIT OpenCourseWare:

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-341-discrete-time-signal-processing-fall-2005/>

*Advantages: Exceptional Lecture Notes! The readings are more in depth than lecture material, but still quite fascinating.

*Disadvantages: The homework assignments and solutions were Advantages for practice, but many problems outside the scope of the 445S class

UC-Berkeley DSP Class Page:

<http://www-inst.eecs.berkeley.edu/ee123/fa09/#resources>

*Advantages: The articles and applets under 'Resources' are quite interesting and useful

*Disadvantages: Seemingly no actual Berkeley work actually on website, everything taken from other sources ...

Carnegie Mellon DSP Class Page:

<http://www.ece.cmu.edu/ee791/>

*Advantages: Lectures had a lot of Matlab code for personal demonstration purposes

*Disadvantages: The lecture notes themselves are far more math-y than the context of 445S - still interesting though

Purdue DSP Class Lecture Notes Page:

<http://cobweb.ecn.purdue.edu/ipollak/ee438/FALL04/notes/notes.html>

*Advantages: the notes are super simple and easy to understand

*Disadvantages: only covers first half of 445S coursework

Doing a search on Apple's iTunes U[niversity] for DSP provided numerous FREE lectures from MIT, UNSW, IIT, etc. for download as well.

Youtube Video Resources:

<http://www.youtube.com/watch?v=7H4sJdyDztI&feature=related>

Signal Processing Tutorial: Nyquist Sampling Theorem and Anti-Aliasing (Part 1)

*Advantages: visuals

*Disadvantages: ... a bit slow

<http://www.youtube.com/watch?v=Fy9dJgGCWZI>

Sampling Rate, Nyquist Frequency, and Aliasing

*Advantages: visualization of basic concepts

*Disadvantages: very short, would have liked more explanation

http://www.youtube.com/watch?v=RJrEaTJuX_A&feature=related

Simple Filters Lecture, IIT-Delhi Lecture

*Advantages: explanations of going to and from magnitude/phase

*Disadvantages: watch out for lecturer's accent

<http://www.youtube.com/watch?v=Xl5bJgOkCGU&feature=channel>

FIR Filter Design, IIT-Delhi Lecture

*Advantages: significantly deeper explanations of math than in class

*Disadvantages: lecturer's accent, video gets stuck about 30 seconds in

<http://www.youtube.com/watch?v=vyNyx00DZBc>

Digital Filter Design

*Advantages: information - especially on design TRADEOFFs

*Disadvantages: sound quality, better off just reading slides while he lectures

Handout C: ECE IT Support

ECE Teaching Labs for various courses are located in the basement and first-floor of the Engineering Education and Research Center (EERC) Building. The computing facilities in the ECE teaching labs are available when officially scheduled lab sections are not meeting in them. The ECE teaching labs are open Mondays–Thursdays from 9:00am to 10:00pm, Fridays 9:00am to 7:00pm, and Saturdays 10:00am to 6:00pm. More information about the ECE teaching labs is available at <http://www.ece.utexas.edu/it/labs>.

1 Available Hardware

The ECE Department has about 200 workstations, including Unix workstations and Windows machines, for student use. In addition to the workstations in the ECE teaching labs, several Linux and Windows workstations are available for remote connection. For more information, see <http://www.ece.utexas.edu/it/remote-linux>.

2 Available Software on the Unix Workstations

The following programs are installed on all of the ECE workstations unless otherwise noted.

- **Matlab** is a number crunching tool for matrix-vector calculations which is well-suited for algorithm development and testing. It comes with a signal processing toolbox (FFTs, filter design, etc.). It is run by typing **matlab**. Matlab is licensed to run on the Windows PCs in the ECE LRC, as well as Unix machines `luigi`, `mario` and `princess` in the ECE LRC. On the Unix machines, be sure to type `module load matlab` before running Matlab. For more information about using Matlab, please see Appendix D in this reader.
- **Mathematica** is a environment for solving algebraic equations, solving differential and difference equations in closed-form, performing indefinite integration, and computing Laplace, Fourier, and other transforms. The command-line interface is run by typing **math**. The graphical user interface is run by typing **mathematica**. On ECE LRC machines, Mathematica is only licensed to run on `sunfire1`.
- The GNU C compiler **gcc** and GNU C++ compiler **g++** are available.
- LabVIEW software environment, which is a graphical programming environment that is useful for signal processing and communication systems developed at National Instruments, is also installed. LabVIEW's Mathscript facility can execute many Matlab scripts and functions. We have a site license for LabVIEW that allows faculty, staff and students to install LabVIEW on their personally-owned computers. For more information, see <http://users.ece.utexas.edu/~bevans/courses/realtime/homework/index.html#labview>

Placeholder - please ignore.

Introduction to Computation in Matlab

Prof. Brian L. Evans, Dept. of ECE, The University of Texas, Austin, Texas USA

Matlab's forte is numeric calculations with matrices and vectors. A vector can be defined as

```
vec = [1 2 3 4];
```

The first element of a vector is at index 1. Hence, `vec(1)` would return 1. A way to generate a vector with all of its 10 elements equal to 0 is

```
zerovec = zeros(1,10);
```

Two vectors, **a** and **b**, can be used in Matlab to represent the left hand side and right hand side, respectively, of a linear constant-coefficient difference equation:

$$a(3) y[n-2] + a(2) y[n-1] + a(1) y[n] = b(3) x[n-2] + b(2) x[n-1] + b(1) x[n]$$

The representation extends to higher-order difference equations. Assuming zero initial conditions, we can derive the transfer function. The transfer function can also be represented using the two vectors **a** (negated feedback coefficients) and **b** (feedforward coefficients). For the second-order case, the transfer function becomes

$$H(z) = \frac{b(1) + b(2)z^{-1} + b(3)z^{-2}}{a(1) + a(2)z^{-1} + a(3)z^{-2}}$$

We can factor a polynomial by using the `roots` command.

Here is an example of values for vectors **a** and **b**:

```
a = [ 1  6/8  1/8];  
b = [ 1  2  3 ];
```

For an asymptotically stable transfer function, i.e. one for which the region of convergence includes the unit circle, the frequency response can be obtained from the transfer function by substituting $z = \exp(j\omega)$. The Matlab command `freqz` implements this substitution:

```
[h, w] = freqz(b, a, 1000);
```

The third argument for `freqz` indicates how many points to use in uniformly sampling the points on the unit circle. In this example, `freqz` returns two arguments: the vector of frequency response values **h** at samples of the frequency domain given by **w**. One can plot the magnitude response on a linear scale or a decibel scale:

```
plot(w, abs(h));  
plot(w, 20*log10(abs(h)));
```

The phase response can be computed using a smooth phase plot or a discontinuous phase plot:

```
plot(w, unwrap(angle(h)));  
plot(w, angle(h));
```

One can obtain help on any function by using the `help` command, e.g.

```
help freqz
```

As an example of defining and computing with matrices, the following lines would define a 2 x 3 matrix **A**, then define a 3 x 2 matrix **B**, and finally compute the matrix **C** that is the inverse of the transpose of the product of the two matrices **A** and **B**:

$$\begin{aligned} \mathbf{A} &= [1 \ 2 \ 3; 4 \ 5 \ 6]; \\ \mathbf{B} &= [7 \ 8; 9 \ 10; 11 \ 12]; \\ \mathbf{C} &= \text{inv}((\mathbf{A} * \mathbf{B})'); \end{aligned}$$

Matlab Tutorials, Help and Training

Here are excellent Matlab tutorials:

<https://stat.utexas.edu/training/software-tutorials#matlab>

http://www.mathworks.com/academia/student_center/tutorials/mltutorial_launchpad.html

The following Matlab tutorial book is a useful reference:

Duane C. Hanselman and Bruce Littlefield, *Mastering MATLAB*, ISBN 9780136013303, Prentice Hall, 2011.

A full version of Matlab is available for your personal computers via a university site license

<http://www.ece.utexas.edu/it/software>

and scroll down to “MATLAB”. Although the first few computer homework assignments will help step you through Matlab, it is strongly suggested that you take the free short courses that the Department of Statistics and Data Sciences will be offering. The schedule of those courses is available online at

<https://stat.utexas.edu/training/software-short-courses>

Technical support is provided through [free consulting services](#) from the [Department of Statistics and Data Sciences](#). Simple queries can be e-mailed to stat.consulting@austin.utexas.edu. For more complicated inquiries, please go in person to their offices located in GDC 7.404. You can walk in or schedule an [appointment online](#).

Running Matlab in Unix

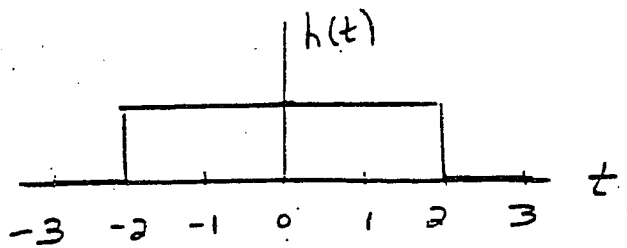
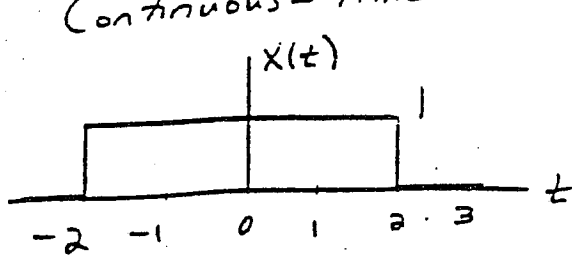
On the Unix machines in the ECE Learning Resource Center, you can run Matlab by typing

```
module load matlab
matlab
```

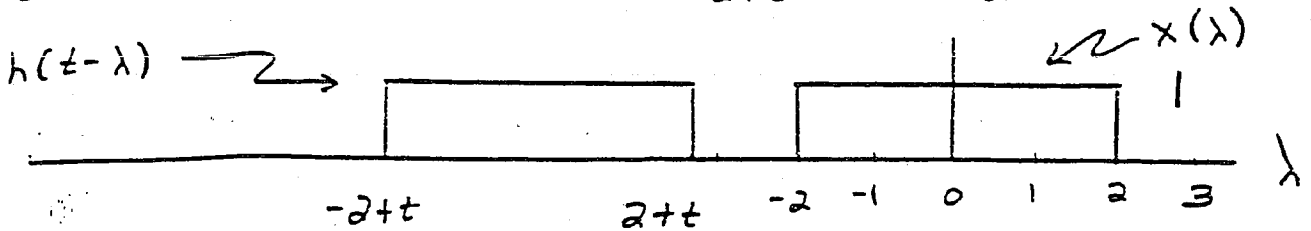
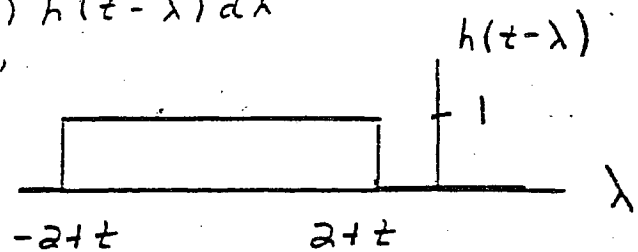
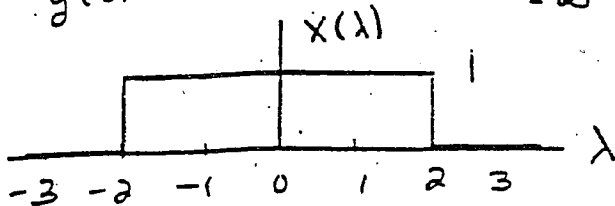
When Matlab begins running, it will automatically execute the commands in your Matlab initialization file, if you have one. On Unix systems, the initialization file must be `~/matlab/startup.m` where `~` means your home directory.

Convolution of Two Rectangular Pulses

1. Continuous-Time Convolution



$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda) h(t-\lambda) d\lambda$$



There is no overlap when $2+t < -2 \Rightarrow t < -4$

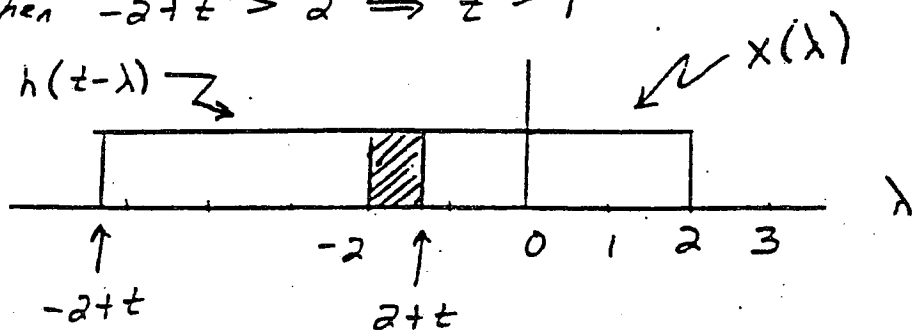
There is no overlap when $-2+t > 2 \Rightarrow t > 4$

For $-4 \leq t \leq 0$,

there is overlap.

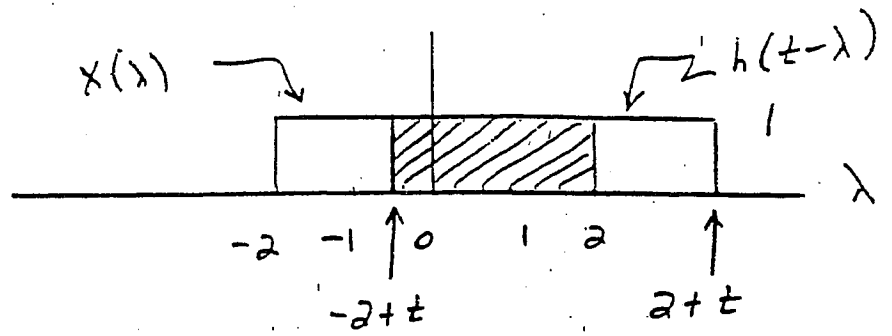
from $\lambda = -2$ to

$$\lambda = 2+t$$



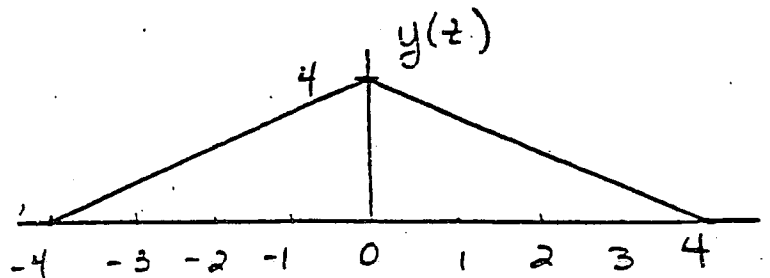
$$y(t) = \int_{-\infty}^{\infty} x(\lambda) h(t-\lambda) d\lambda = \int_{-2}^{2+t} 1 \cdot d\lambda = [\lambda]_{-2}^{2+t} = t+4$$

For $0 < t \leq 4$,
 there is overlap
 from $\lambda = -2 + t$
 to $\lambda = 2$



$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda = \int_{-2+t}^2 1 \cdot d\lambda = [\lambda]_{-2+t}^2 = 4 - t$$

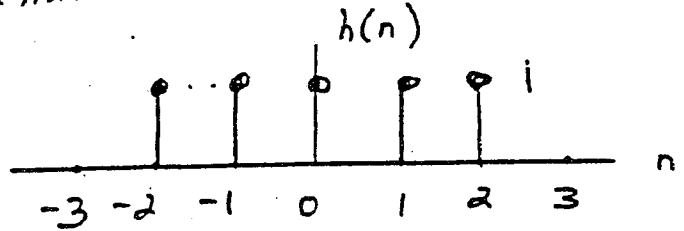
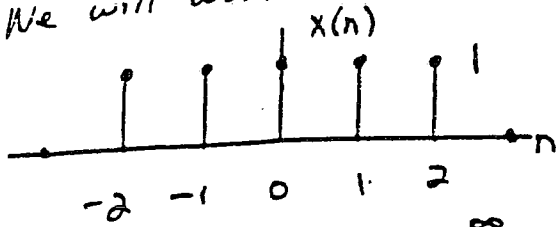
$$y(t) = \begin{cases} 0 & \text{for } t < -4 \\ 4+t & \text{for } -4 \leq t \leq 0 \\ 4-t & \text{for } 0 < t \leq 4 \\ 0 & \text{for } t > 4 \end{cases}$$



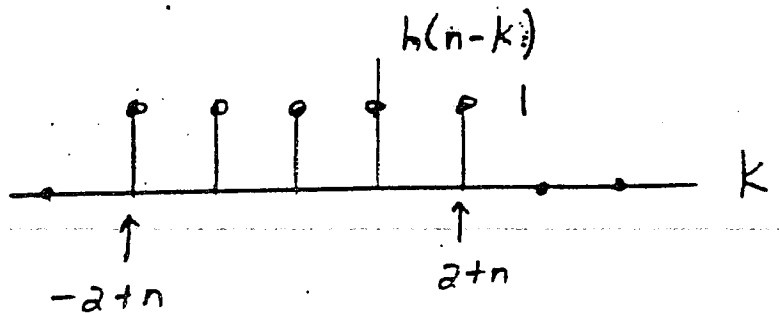
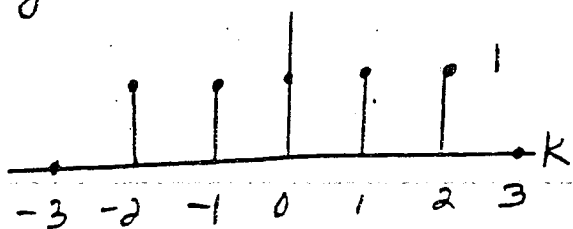
To check a convolution result, check the value of $y(t)$ at the endpoints of each interval - they should agree.

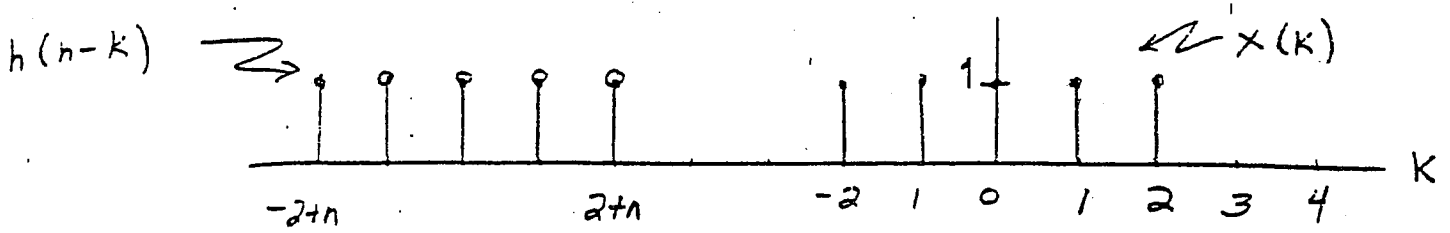
2. Discrete-Time Convolution

We will work a similar convolution in discrete time.



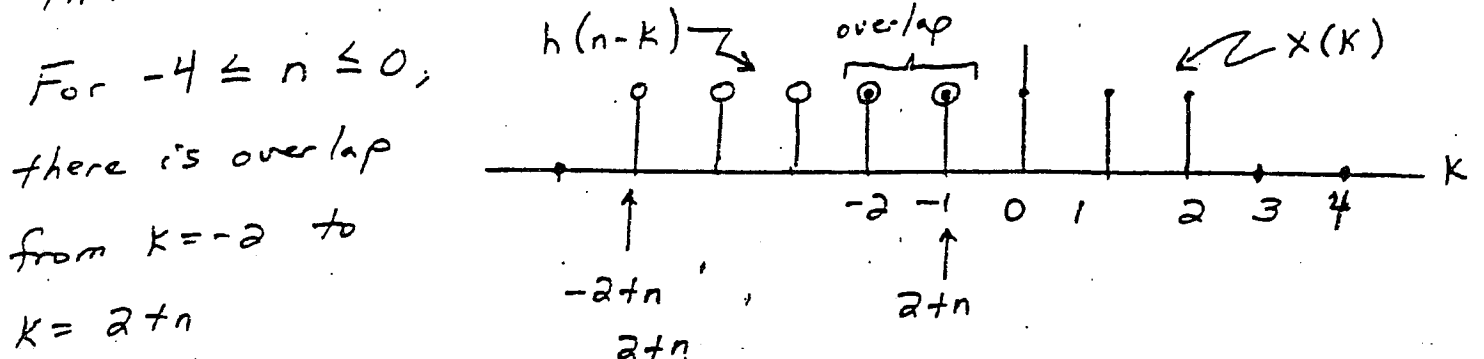
$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k), \text{ so } k \text{ has the same role as } \lambda$$



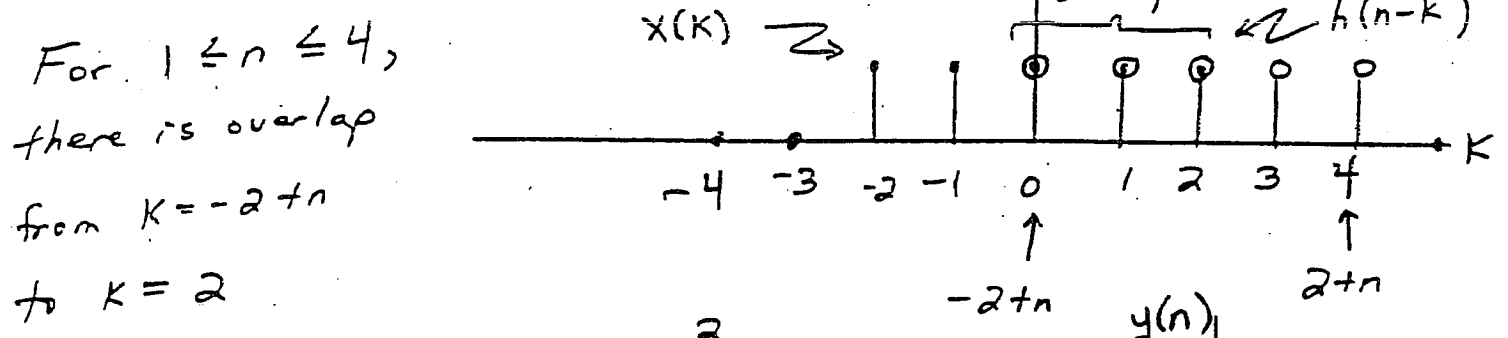


There is no overlap when $2+n < -2 \Rightarrow n < -4$

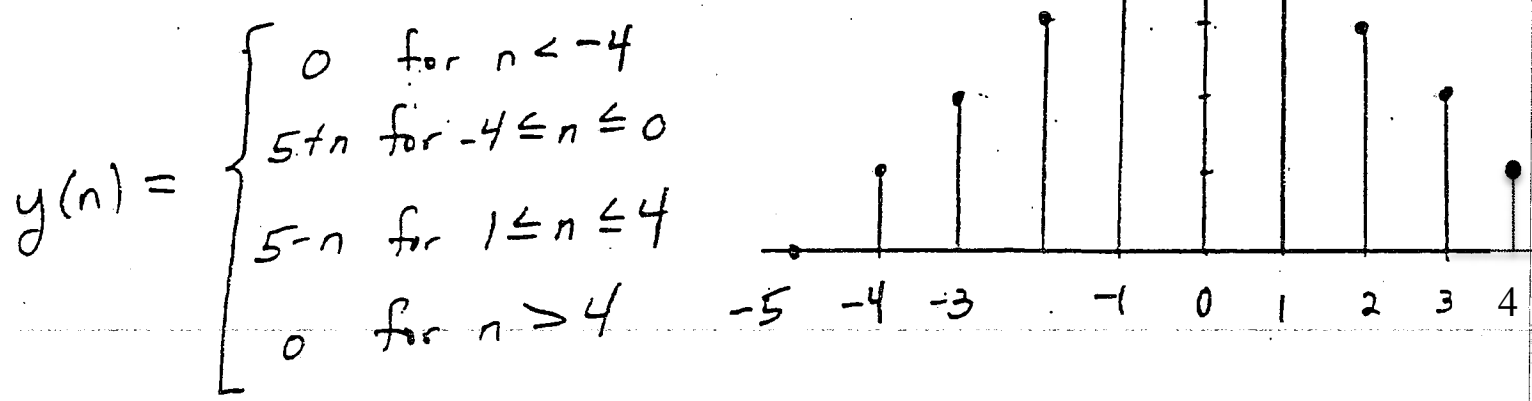
There is no overlap when $-2+n > 2 \Rightarrow n > 4$



$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k) = \sum_{k=-2}^{2+n} 1 = n+5$$



$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k) = \sum_{k=-2+n}^2 1 = 5-n$$



$$y(n) = \begin{cases} 0 & \text{for } n < -4 \\ 5+n & \text{for } -4 \leq n \leq 0 \\ 5-n & \text{for } 1 \leq n \leq 4 \\ 0 & \text{for } n > 4 \end{cases}$$

Handout F: Fundamental Theorem of Linear Systems

Theorem: Let a linear time-invariant system g has an $e_f(t)$ denote the complex sinusoid $e^{j2\pi ft}$. Then, $g(e_f(\cdot), t) = g(e_f(\cdot), 0)e_f(t) = c e_f(t)$.

Example: Analog RC Lowpass Filter

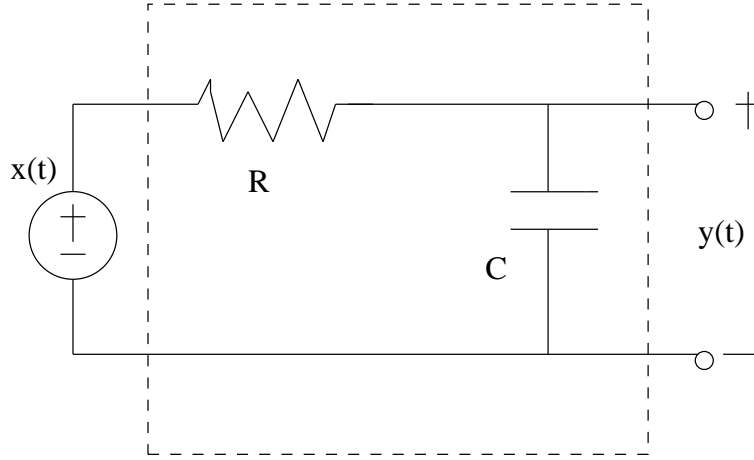


Figure 1: A First-Order Analog Lowpass Filter

The impulse response for the circuit in Fig. 1, i.e. the output measured at $y(t)$ when $x(t) = \delta(t)$, is

$$h(t) = \frac{1}{RC} e^{-\frac{1}{RC}t} u(t)$$

For a complex sinusoidal input, $x(t) = e_f(t) = e^{j2\pi ft}$,

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(t - \lambda) h(\lambda) d\lambda \\ &= \int_{-\infty}^{\infty} e^{j2\pi f(t-\lambda)} \frac{1}{RC} e^{-\frac{1}{RC}\lambda} u(\lambda) d\lambda \\ &= e^{j2\pi ft} \left[\frac{1}{RC} \int_{-\infty}^{\infty} e^{-j2\pi f\lambda} e^{-\frac{1}{RC}\lambda} d\lambda \right] \\ &= \left[\frac{\frac{1}{RC}}{j2\pi f + \frac{1}{RC}} \right] e^{j2\pi ft} \\ &= g(e_f(\cdot), 0) e_f(t) \end{aligned}$$

So, $g(e_f(\cdot), 0) = H(f)$, which is the transfer function of the system.

Placeholder – please ignore.

Handout G: Raised Cosine Pulse

Section 7.5, pp. 431–434, Simon Haykin, *Communication Systems*, 4th ed.

We may overcome the practical difficulties encountered with the ideal Nyquist channel by extending the bandwidth from the minimum value $W = R_b/2$ to an adjustable value between W and $2W$. We now specify the frequency function $P(f)$ to satisfy a condition more elaborate than that for the ideal Nyquist channel; specifically, we retain three terms of (7.53) and restrict the frequency band of interest to $[-W, W]$, as shown by

$$P(f) + P(f - 2W) + P(f + 2W) = \frac{1}{W}, -W \leq f \leq W \quad (1)$$

We may devise several band-limited functions to satisfy (1). A particular form of $P(f)$ that embodies many desirable features is provided by a *raised cosine spectrum*. This frequency characteristic consists of a flat portion and a *rolloff* portion that has a sinusoidal form, as follows:

$$P(f) = \begin{cases} \frac{1}{2W} & \text{for } 0 \leq |f| < f_1 \\ \frac{1}{4W} \left(1 - \sin \frac{\pi(|f| - W)}{2W - 2f_1} \right) & \text{for } f_1 \leq |f| < 2W - f_1 \\ 0 & \text{for } |f| \geq 2W - f_1 \end{cases} \quad (2)$$

The frequency parameter f_1 and bandwidth W are related by

$$\alpha = 1 - \frac{f_1}{W} \quad (3)$$

The parameter α is called the *rolloff factor*; it indicates the *excess bandwidth* over the ideal solution, W . Specifically, the transmission bandwidth B_T is defined by $2W - f_1 = W(1 + \alpha)$.

The frequency response $P(f)$, normalized by multiplying it by $2W$, is shown plotted in Fig. 1 for three values of α , namely, 0, 0.5, and 1. We see that for $\alpha = 0.5$ or 1, the function $P(f)$ cuts off gradually as compared with the ideal Nyquist channel (i.e., $\alpha = 0$) and is therefore easier to implement in practice. Also the function $P(f)$ exhibits odd symmetry with respect to the Nyquist bandwidth W , making it possible to satisfy the condition of (1). The time response $p(t)$ is the inverse Fourier transform of the function $P(f)$. Hence, using the $P(f)$ defined in (2), we obtain the result (see Problem 7.9)

$$p(t) = \text{sinc}(2Wt) \left(\frac{\cos 2\pi\alpha Wt}{1 - 16\alpha^2 W^2 t^2} \right) \quad (4)$$

which is shown plotted in Fig. 2 for $\alpha = 0, 0.5$, and 1. The function $p(t)$ consists of the product of two factors: the factor $\text{sinc}(2Wt)$ characterizing the ideal Nyquist channel and a

second factor that decreases as $1/|t|^2$ for large $|t|$. The first factor ensures zero crossings of $p(t)$ at the desired sampling instants of time $t = iT$ with i an integer (positive and negative). The second factor reduces the tails of the pulse considerably below that obtained from the ideal Nyquist channel, so that the transmission of binary waves using such pulses is relatively insensitive to sampling time errors. In fact, for $\alpha = 1$, we have the most gradual rolloff in that the amplitudes of the oscillatory tails of $p(t)$ are smallest. Thus, the amount of intersymbol interference resulting from timing error decreases as the rolloff factor α is increased from zero to unity.

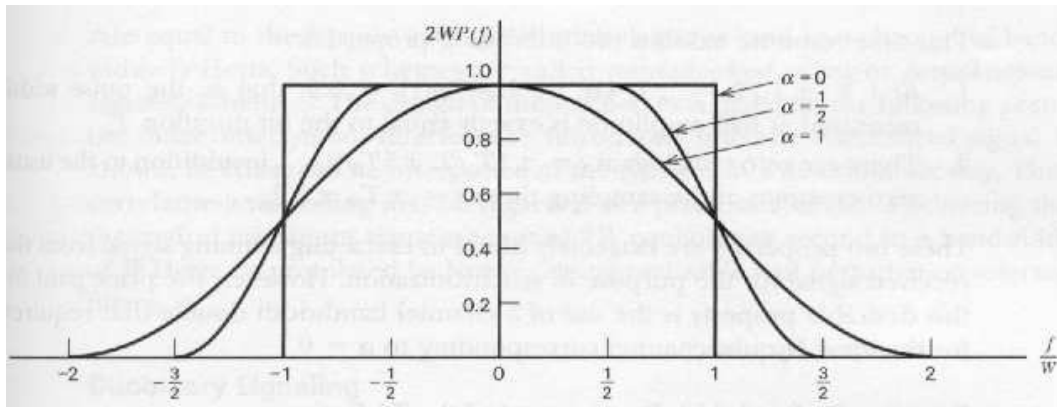


Figure 1: Frequency response for the raised cosine function.

The special case with $\alpha = 1$ (i.e., $f_1 = 0$) is known as the *full-cosine rolloff* characteristic, for which the frequency response of (2) simplifies to

$$P(f) = \begin{cases} \frac{1}{4W} \left(1 + \cos \frac{\pi f}{2W} \right) & \text{for } 0 < |f| < 2W \\ 0 & \text{if } |f| \geq 2W \end{cases}$$

Correspondingly, the time response $p(t)$ simplifies to

$$p(t) = \frac{\text{sinc}(4Wt)}{1 - 16W^2t^2} \quad (5)$$

The time response exhibits two interesting properties:

1. At $t = \pm T_b/2 = \pm 1/4W$, we have $p(t) = 0.5$; that is, the pulse width measured at half amplitude is exactly equal to the bit duration T_b .
2. There are zero crossings at $t = \pm 3T_b/2, \pm 5T_b/2, \dots$ in addition to the usual zero crossings at the sampling times $t = \pm T_b, \pm 2T_b, \dots$

These two properties are extremely useful in extracting a timing signal from the received signal for the purpose of synchronization. However, the price paid for this desirable property is the use of a channel bandwidth double that required for the ideal Nyquist channel corresponding to $\alpha = 0$.

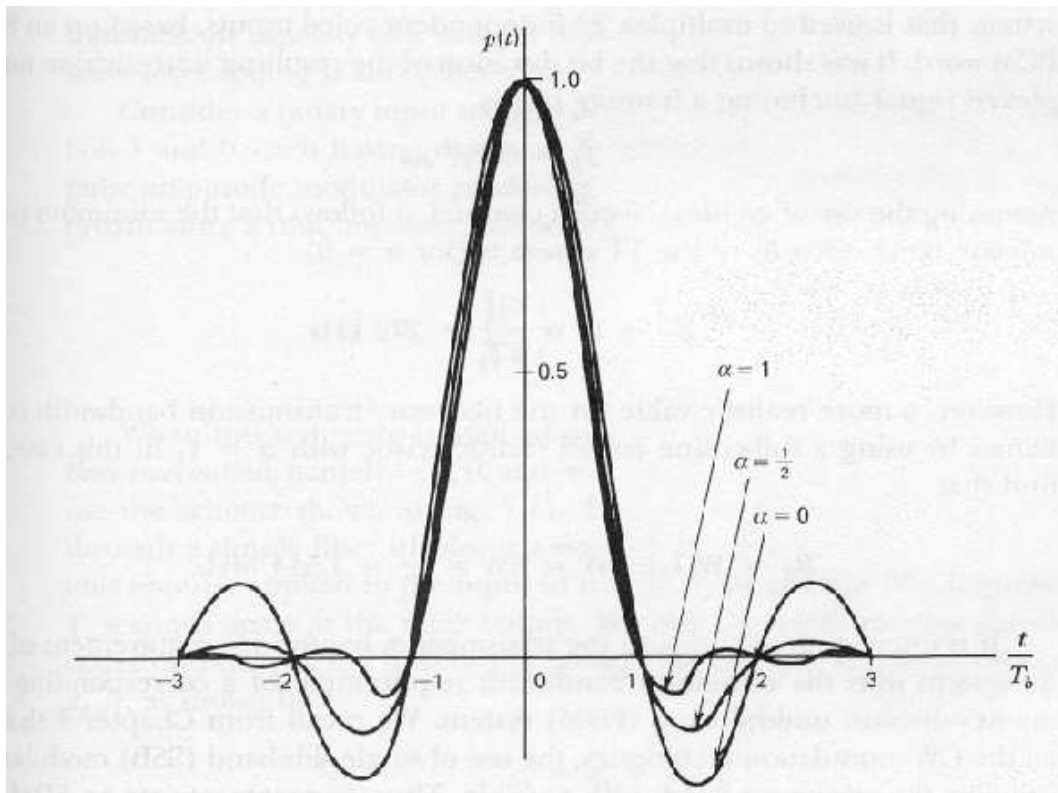


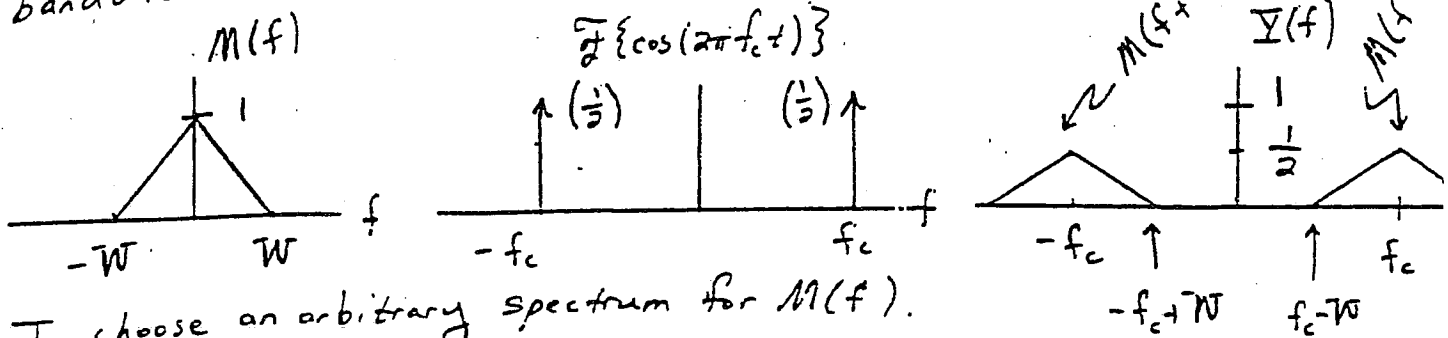
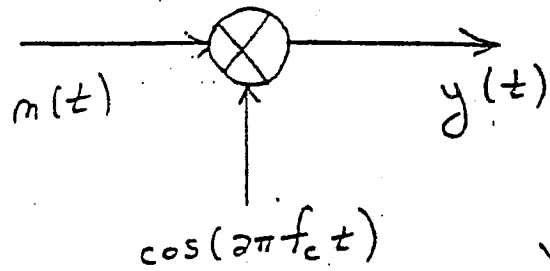
Figure 2: Time response for the raised cosine function.

Placeholder – please ignore.

Amplitude Modulation with $\cos(2\pi f_c t)$

$$y(t) = m(t) \cos(2\pi f_c t)$$

$m(t)$ is lowpass with bandwidth W ($f_c \gg W$)



I choose an arbitrary spectrum for $M(f)$.

$$Y(f) = M(f) * \mathcal{F}\{\cos(2\pi f_c t)\}$$

$$Y(f) = M(f) * \frac{1}{2} (\delta(f+f_c) + \delta(f-f_c))$$

$$Y(f) = \frac{1}{2} \int_{-\infty}^{\infty} (\delta(\lambda+f_c) + \delta(\lambda-f_c)) M(f-\lambda) d\lambda$$

Dir. / has. are non-zero @ $\lambda = -f_c$ and $\lambda = f_c$

$$Y(f) = \frac{1}{2} (M(f+f_c) + M(f-f_c))$$

Example uses the normalized Fourier transform

It is easier to work with amplitude modulation to draw pictures in the frequency domain when analyzing the resulting spectrum for amplitude modulation.

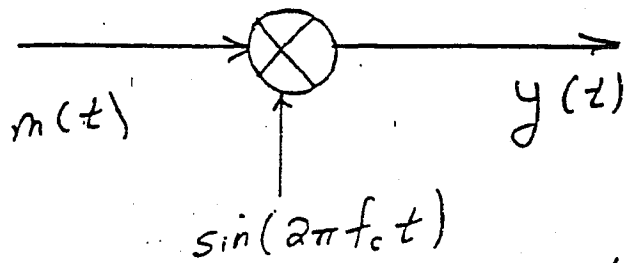
Recall that multiplication in time becomes convolution in the frequency domain.

The bandwidth of $Y(f)$ is $2W$

Amplitude Modulation with $\sin(2\pi f_c t)$

$$y(t) = m(t) \sin(2\pi f_c t)$$

As before, $m(t)$ is lowpass with bandwidth W and $f_c \gg W$.



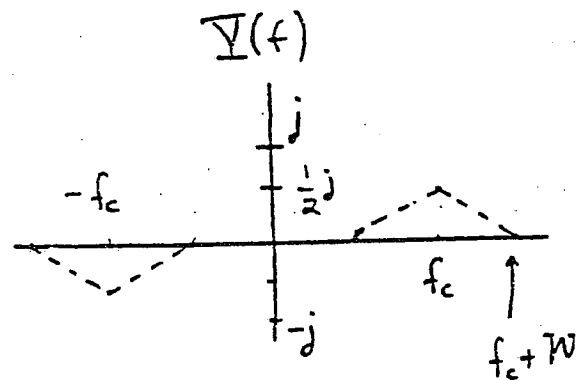
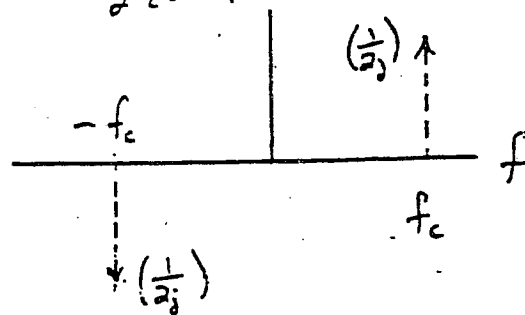
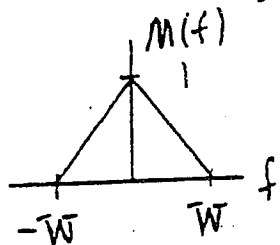
$$Y(f) = M(f) * \mathcal{F}\left\{\frac{1}{2j} \sin(2\pi f_c t)\right\}$$

$$Y(f) = M(f) * \frac{1}{2j} (-\delta(f+f_c) + \delta(f-f_c))$$

$$Y(f) = \frac{1}{2j} \int_{-\infty}^{\infty} (-\delta(\lambda+f_c) + \delta(\lambda-f_c)) M(f-\lambda) d\lambda$$

Deltas are non-zero at $\lambda = -f_c$ and $\lambda = f_c$

$$Y(f) = \frac{1}{2j} (-M(f+f_c) + M(f-f_c))$$



So, the area under the Dirac delta functions for $\mathcal{F}\left\{\frac{1}{2j} \sin(2\pi f_c t)\right\}$ is $-\frac{1}{2j}$, or $+\frac{1}{2j}$, respectively.

So, in the time domain, $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ are orthogonal. In the frequency domain, $\cos(2\pi f_c t)$ becomes real-valued, and $\sin(2\pi f_c t)$ becomes imaginary. The bandwidth of $Y(f)$ is $2W$.

Handout I: Analog Sinusoidal Modulation Summary

Many ways exist to modulate a message signal $m(t)$ to produce a modulated (transmitted) signal $x(t)$. For amplitude, frequency, and phase modulation, modulated signals can be expressed in the same form as

$$x(t) = A(t) \cos(2\pi f_c t + \Theta(t))$$

where $A(t)$ is a real-valued amplitude function (a.k.a. the envelope), f_c is the carrier frequency, and $\Theta(t)$ is the real-valued phase function. Using this framework, several common modulation schemes are described below. In the table below, the amplitude modulation methods are double sideband larger carrier (DSB-LC), DSB suppressed carrier (DSB-SC), DSB variable carrier (DSB-VC), and single sideband (SSB). The hybrid amplitude-frequency modulation is quadrature amplitude modulation (QAM). The angle modulation methods are phase and frequency modulation.

Modulation	$A(t)$	$\Theta(t)$	Carrier	Type	Use
DSB-LC	$A_c [1 + k_a m(t)]$	Θ_0	Yes	Amplitude	AM radio
DSB-SC	$A_c m(t)$	Θ_0	No	Amplitude	
DSB-VC	$A_c m(t) + \epsilon$	Θ_0	Yes	Amplitude	
SSB	$A_c \sqrt{m^2(t) + [m(t) \star h(t)]^2}$	$\arctan(-\frac{m(t) \star h(t)}{m(t)})$	No	Amplitude †	Marine radios
QAM	$A_c \sqrt{m_1^2(t) + m_2^2(t)}$	$\arctan(-\frac{m_2(t)}{m_1(t)})$	No	Hybrid	Satellite
Phase	A_c	$\Theta_0 + k_p m(t)$	No	Angle	Underwater modems
Frequency	A_c	$2\pi k_f \int_0^t m(t) dt$	No	Angle	FM radio TV audio

† $h(t)$ is the impulse response of a bandpass filter or phase shifter to effect a cancellation of one pair of redundant sidebands. For ideal filters and phase shifters, the modulation is amplitude modulation because the phase would not carry any information about $m(t)$.

Each analog TV channel is allocated a bandwidth of 6 MHz. The picture intensity and color information are transmitted using vestigial sideband modulation. Vestigial sideband modulation is a variant of amplitude modulation (not shown above) in which the upper sideband is kept and a fraction of the lower sideband is kept, or vice-versa. In an analog TV signal, the audio portion is frequency modulated.

The following quantity is known as the complex envelope

$$\tilde{x}(t) = A(t) e^{j\Theta(t)} = x_I(t) + j x_Q(t)$$

where $x_I(t)$ is called the in-phase component and $x_Q(t)$ is called the quadrature component. Both $x_I(t)$ and $x_Q(t)$ are lowpass signals, and hence, the complex envelope $\tilde{x}(t)$ is a lowpass signal. An alternative representation for the modulated signal $x(t)$ is

$$x(t) = \Re\{\tilde{x}(t) e^{j2\pi f_c t}\}$$

Halftone Comparison



Simple Threshold



Clustered-dot dither



Error diffusion



Original

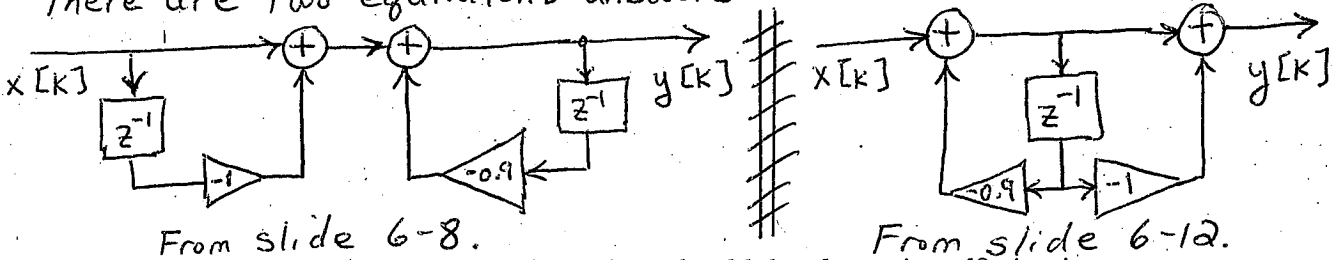
Problem 1.1 Digital Filter Analysis. 20 points. IIR

A causal discrete-time linear time-invariant FIR filter with input $x[k]$ and output $y[k]$ is governed by the following difference equation:

$$y[k] = -0.9 y[k-1] + x[k] - x[k-1]$$

(a) Draw the block diagram for this filter. 4 points.

There are two equivalent answers:



From slide 6-8.

From slide 6-12.

(b) What are the initial conditions and what values should they be assigned? 4 points.

$$y[0] = -0.9 y[-1] + x[0] - x[-1]$$

Initial conditions are $x[-1]$ and $y[-1]$.

Both should be set to zero to ensure causal LTI system.

(c) Find the equation for the transfer function in the z-domain including the region of convergence. 4 points.

$$Y(z) = -0.9 z^{-1} Y(z) + X(z) - z^{-1} X(z)$$

$$Y(z) + 0.9 Y(z) = X(z) - z^{-1} X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{1 + 0.9 z^{-1}} = H(z) \text{ for } |z| > 0.9$$

(See slides 5-21 and 5-22 for the region of convergence.)

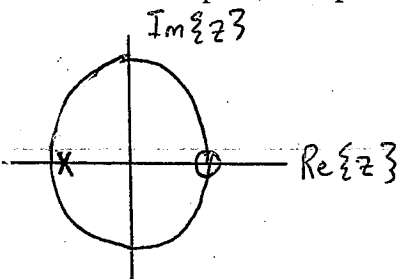
(d) Find the equation for the frequency response of the filter. 4 points.

Since the region of convergence includes the unit circle,

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 - e^{-j\omega}}{1 + 0.9 e^{-j\omega}}$$

(e) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.

(See slide 6-6)



Pole at $1 + 0.9 z^{-1} = 0 \Rightarrow z = -0.9$.

Zero at $1 - z^{-1} = 0 \Rightarrow z = 1$.

$z = 1 \Rightarrow \omega = 0$ (low frequencies)

$z = -1 \Rightarrow \omega = \pi$ (high frequencies)

HIGHPASS FILTER

K-2

Pole near unit circle indicates passband. Zero on or near unit circle indicates stopband.

Problem 1.2 Digital Filter Design. 20 points.

Asymmetric Digital Subscriber Line (ADSL) systems transmit voice and data over a telephone line using frequencies from 0 Hz to 1.1 MHz. ADSL transceivers use a sampling rate of 2.2 MHz.

$$f_s = 2.2 \text{ MHz}$$

Consider an AM radio station that has a carrier frequency of 550 kHz, has a transmission bandwidth of 10 kHz, and is interfering with ADSL transmission.

$$f_c = 550 \text{ kHz}$$

Design a digital IIR filter *biquad* for the ADSL receiver to reject the AM radio station but pass as much of the ADSL transmission band as possible.

- (a) Is the digital IIR filter biquad lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.

The digital IIR filter biquad is a narrowband bandstop filter, a.k.a. a notch filter. The biquad needs to reject as best it can frequencies between 545 kHz and 555 kHz with respect to a sampling rate of 2.2 MHz.

- (b) Give formulas for the pole-zero locations of the biquad. 8 points.

A biquad has 2 poles and 0-2 zeros (see slide 6-4).

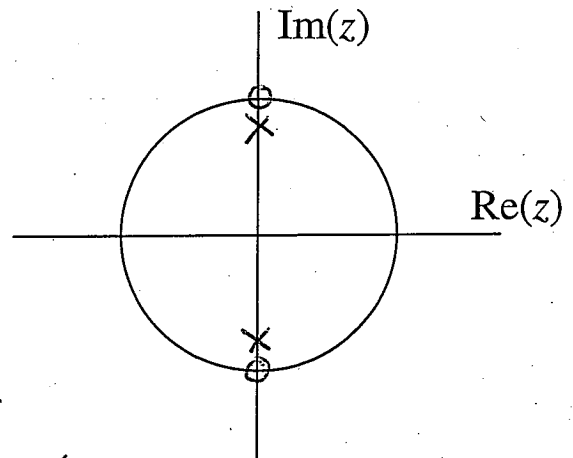
Design a notch filter to notch out (reject) the AM radio station frequency. Pole-zero diagram shown on slide 6-6 (middle plot).

Zero locations $\left\{ \begin{array}{l} z_0 = e^{+j\omega_c} \\ z_1 = e^{-j\omega_c} \end{array} \right.$ Pole locations $\left\{ \begin{array}{l} p_0 = 0.9 e^{+j\omega_c} \\ p_1 = 0.9 e^{-j\omega_c} \end{array} \right.$ $\omega_c = 2\pi \frac{f_c}{f_s} = \frac{\pi}{2}$

- (c) Draw the poles and zeros on the pole-zero diagram on the right. 4 points.

The zero locations are conjugate symmetric to notch out frequencies at $+\omega_c$ and $-\omega_c$ and to ensure real-valued feed forward coefficients.

The pole locations also have angles of $+\omega_c$ and $-\omega_c$ to ensure an all-pass response outside of the narrow stopband.



- (d) Compute the scaling constant (gain) for the filter's transfer function. 4 points.

$$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

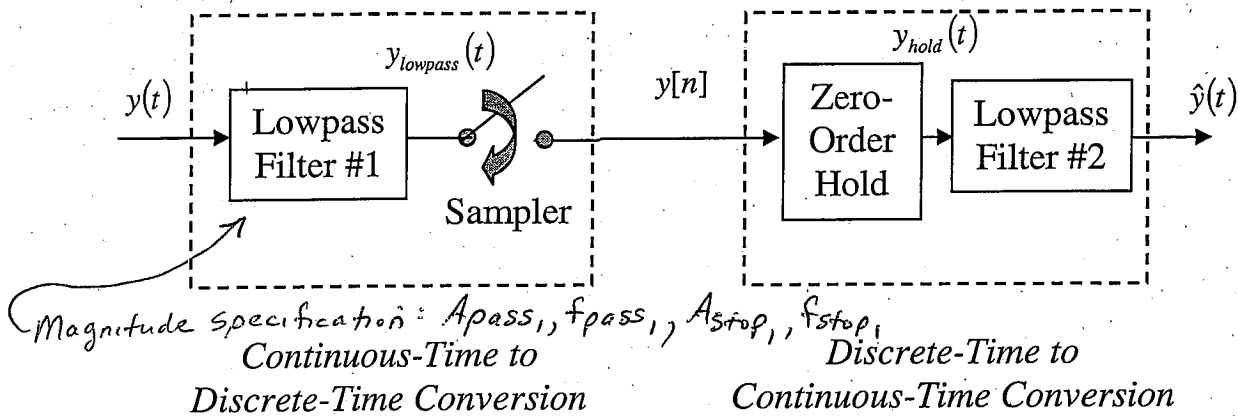
Scaling constant is C (see slide 6-5).

Representing a filter in terms of poles-zeros-gain (PZK) is common. We can arbitrarily set the DC response to 1: $H(e^{j0}) = C \frac{(1 - z_0)(1 - z_1)}{(1 - p_0)(1 - p_1)} = 1$

Problem 1.3. Sampling and Interpolation. 20 points.

This problem explores the many unrealistic assumptions in the Shannon sampling theorem (slides 4-6 and 4-7).

Consider the process of converting a continuous-time signal to a discrete-time signal and the process of converting a discrete-time signal to a continuous-time signal. In this problem, we will *not* include the effects of quantization.



The lowpass filters are analog, continuous-time, infinite impulse response filters. In the application that will use these subsystems, the phases of the signals are not important, but tracking the time-domain waveform closely is important.

- (a) What algorithm would you use to design lowpass filters #1 and #2 to guarantee a minimum number of biquads? 5 points.

Elliptic filter design algorithm (see slides 6-23 and 6-24, and homework problem 1.5).

- (b) How would you choose the sampling rate f_s ? 5 points.

*From the Shannon sampling theorem: $f_s > 2 f_{stop}$ (slides 4-6)
 f_{stop} is chosen to be the f_{max} of interest in $y(t)$ times 1.1 (10% rolloff).
 To track time-domain waveform closely, $f_s > 8 f_{stop}$ (slide 4-10).*

- (c) The sampler and zero-order hold subsystems are being driven by two different oscillators, where each oscillator oscillates at the approximately the sampling rate. Is the accuracy of the oscillators a significant problem? Why or why not? 5 points.

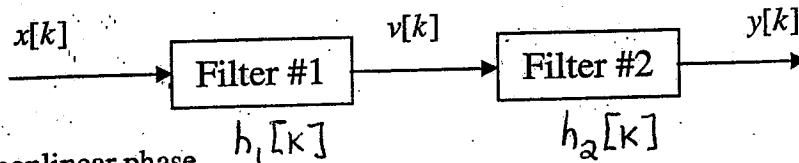
A timing error in the oscillator corresponds to a phase shift in frequency response. Since the phases of the signals in this application are not important, we can tolerate the usual 0.1% accuracy in oscillators.

- (d) What is the pulse shape being used for interpolation in the discrete-time to continuous-time conversion shown above? 5 points.

The pulse shape in the zero-order hold is a rectangular pulse (see slide 7-5). The actual pulse used in the interpolation is the convolution of the rectangular pulse (of duration T_s) with the impulse response of lowpass filter #2.

Problem 1.4. Phase Response. 20 points.

Two stable discrete-time linear time-invariant (LTI) filters are in cascade as shown below.



Filter #1 has nonlinear phase.

Either prove that the cascade of LTI filters always has nonlinear phase, or give a counter-example.

Among stable discrete-time LTI filters that can be implemented, only symmetric FIR filters and anti-symmetric FIR filters have linear phase. The nonlinear phase in filter #1 could result from either an FIR filter without symmetry or anti-symmetry in the coefficients or an IIR filter. Filter #2 is free w/r to phase.

Counter-example #1: FIR filters (from a student solution)

$$\text{Let } h_1[k] = \delta[k] + \frac{1}{2} \delta[k-1]$$

$$h_2[k] = \frac{1}{2} \delta[k] + \delta[k-1]$$

Impulse response of the cascade is

$$h[k] = h_1[k] * h_2[k] = \frac{1}{2} \delta[k] + \frac{5}{4} \delta[k-1] + \frac{1}{2} \delta[k-2]$$

which has linear phase.

Counter-example #2: IIR-FIR cascade

Let filter #1 be an IIR biquad (all-pole) and filter #2 be a three-tap FIR filter.

$$H_1(z) = C \frac{1}{(1-p_0 z^{-1})(1-p_1 z^{-1})} \quad \text{Let } H_2(z) = (1-p_0 z^{-1})(1-p_1 z^{-1})$$

$$\text{where } p_0 = r e^{j\theta} \text{ and } p_1 = r e^{-j\theta}, r \in (0,1). \quad = 1 - (p_0 + p_1) z^{-1} + p_0 p_1 z^{-2}$$

$$= 1 - (2 \cos \theta) z^{-1} + r^2 z^{-2}$$

Any value of $r \in (0,1)$ creates nonlinear phase for filter #2 and #1.

$H(z) = H_1(z) H_2(z) = C$, where C is a constant, and hence the cascade has linear phase.

Problem 1.5. Oscillator Implementation. 20 points.

Consider a casual discrete-time linear time-invariant (LTI) filter whose impulse response is $\cos(\pi/3 k) u[k]$. The relationship in the discrete-time domain between the input signal $x[k]$ and the output signal $y[k]$ is

$$y[k] = x[k] + y[k-1] - y[k-2] - \frac{1}{2}x[k-1]$$

In IEEE floating point arithmetic, $\frac{1}{2}$ is exactly represented using one bit of mantissa.

with initial conditions $y[-1] = 0$, $y[-2] = 0$, and $x[-1] = 0$.

The filter is to be implemented on a TI TMS320C6700 programmable digital signal processor using IEEE single-precision floating-point data and arithmetic. The single-precision IEEE floating-point data format has 1 sign bit, 8 exponent bits, and 23 mantissa bits.

(a) In generating the impulse response $\cos(\pi/3 k) u[k]$ using the above difference equation, how many bits of precision are lost in the calculation of $y[0]$, $y[1]$ and $y[2]$? 4 points.

NO LOSS OF PRECISION

$$\begin{aligned}
 x[k] = \delta[k] &\Rightarrow x[0] = 1 \Rightarrow y[0] = x[0] + y[-1] - y[-2] + \frac{1}{2}x[-1] = x[0] = 1 \\
 &x[1] = 0 \Rightarrow y[1] = x[1] + y[0] - y[-1] - \frac{1}{2}x[0] = y[0] - \frac{1}{2}x[0] = \frac{1}{2} \\
 &x[2] = 0 \Rightarrow y[2] = x[2] + y[1] - y[0] - \frac{1}{2}x[1] = y[1] - y[0] = -\frac{1}{2}
 \end{aligned}$$

(b) How would you restore any lost precision that may result from the above difference equation that would be applicable for any frequency of the oscillator? 4 points.

For the general case of implementing an oscillator of frequency of the form $\omega_0 = 2\pi \frac{N}{L}$, where L is the period, we reset the filter every L th sample by setting the initial conditions to zero and the input to $x[k] = \delta[k]$. (We are assuming that N and L are relatively prime, e.g. $N=6$ and $L=35$.)

(c) Write TI TMS320C6700 assembly to generate the constant $\frac{1}{2}$ as a single-precision floating point number in register A6. 4 points.

Using instructions on slides 1-20, 1-23, and 2-13,

```

MVK .S1 2, A6 ; A6 = 2
INTSP .L1 A6, A6 ; A6 = 2.0
RCPSP .S1 A6, A6 ; A6 = 1/2.0
    
```

(d) Given below is a linear C6700 assembly language program to compute $y[k]$. The address for x is in A4 and the address for y is in B5. 8 points.

1. Assign functional units for each instruction.
2. Show all possible parallel groupings that do not cause pipeline hazards by using the double vertical bar \parallel notation (the \parallel notation is in chapter 3 of Tretter's manual). - page 94

```

; Constant 0.5 stored in A6 in single-precision floating-point format
LDW .D1 *A4[1], A7 ; A7 = x[k-1]
|| LDW .D2 *B5[1], B8 ; B8 = y[k-1]
LDW .D1 *A4, A3 ; A3 = x[k]
|| LDW .D2 *B5[2], B9 ; B9 = y[k-2]
NOP 3
MPYSP .M1 A7, A6, A10 ; A10 = 0.5 x[k-1]
STW .D1 A3, *A4[1] ; Update x[k-1] = x[k]
|| SUBSP .L2x B9, A3, B11 ; B11 = x[k] - y[k-2]
ADDSP .L2 B8, B11, B11 ; B11 = x[k] + y[k-1] - y[k-2]
|| STW .D2 B8, *B5[2] ; Update y[k-2] = y[k-1]
SUBSP .L2x A10, B11, B11 ; B11 = x[k] + y[k-1] - y[k-2] - 0.5 x[k-1]
STW .D2 B11, *B5[1] ; Update y[k-1] = y[k]
    
```

Two out of three registers from B file implies a functional unit from data path #2.

Using slides 1-20, 1-23, 2-5, and 2-6.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 9, 2006

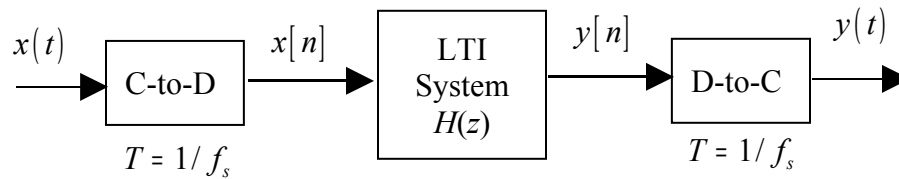
Course: EE 345S Arslan

Name: _____
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network.
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	20		Digital Filter Analysis
2	20		IIR Filter
3	20		Sampling and Reconstruction
4	20		Linear Systems
5	20		Assembly Language
<i>Total</i>	100		

Problem 1.2 IIR filtering. 20 points.
For the system shown below



The input signal $x(t)$ to the Continuous-to-Discrete converter is

$$x(t) = 4 + \cos(500\pi t) - 3\cos\left(\left[\frac{2000\pi}{3}\right]t\right)$$

The transfer function for the linear, time-invariant (LTI) system is $H(z)$

$$H(z) = \frac{(1 - z^{-1})(1 - e^{j\pi/2}z^{-1})(1 - e^{-j\pi/2}z^{-1})}{(1 - 0.9e^{j2\pi/3}z^{-1})(1 - 0.9e^{-j2\pi/3}z^{-1})}$$

If $f_s = 1000$ samples/sec, determine an expression for $y(t)$, the output of the Discrete-to-Continuous converter.

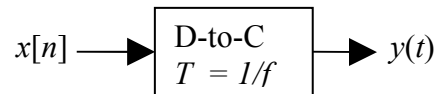
Problem 1.3. Sampling and Reconstruction. 20 points.

Suppose that a discrete-time signal $x[n]$ is given by the formula

$$x[n] = 10 \cos(0.2\pi n - \pi / 7)$$

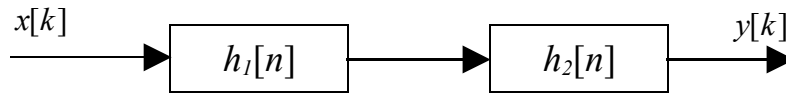
and that it was obtained by sampling a continuous-time signal at a sampling rate of $f_s = 2000$ samples/sec.

- a) Determine two different continuous-time signals $x_1(t)$ and $x_2(t)$ whose samples are equal to $x[n]$; i.e. find $x_1(t)$ and $x_2(t)$ such that $x[n] = x_1(nT_s) = x_2(nT_s)$
- b) If $x[n]$ is given by the equation above, what signal will be reconstructed by an ideal D-to-C converter operating at sampling rate 2000 samples/sec? That is, what is the output $y(t)$ in the following figure if $x[n]$ is as given above?

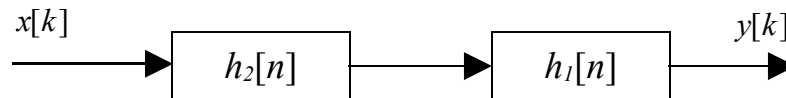


Problem 1.4. Linear Systems. 20 points.

Two stable discrete-time linear time-invariant (LTI) filters are in cascade as shown below.



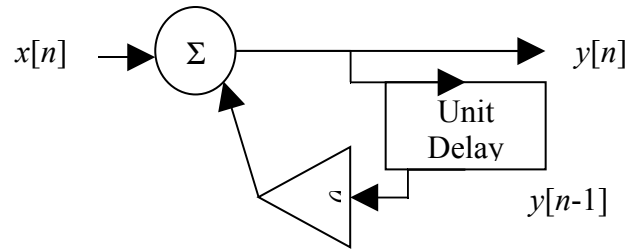
a) Show that the end-to-end system from $x[k]$ to $y[k]$ is equivalent to the following system where the order of the systems have been replaced, or give a counter-example. 10 points



b) What practical considerations have to be taken into account when switching the order of two systems in practice? 10 points

Problem 1.5 Assembly Language. 20 points.

Consider the discrete-time linear time-invariant filter with $x[n]$ and output $y[n]$ shown on the right. Assume that the input signal $x[n]$ and the coefficient a represent complex numbers.



(a) Write the difference equation for this filter. Is this an FIR or IIR filter? 4 point.

(b) Sketch the pole-zero plot for this filter. 4 points.

(c) Write a linear TI C6700 assembly language routine to implement the difference equation. Assume that the address for x is in A4 and the address to y is in A5. Assume that the input and output data as well as the coefficient consist of single precision floating point complex numbers. Assume that the assembler will insert the correct number of no-operation (NOP) instructions to prevent pipeline hazards. 12 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 19, 2006

Course: EE 345S Evans

Name: Set Solution
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network.
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

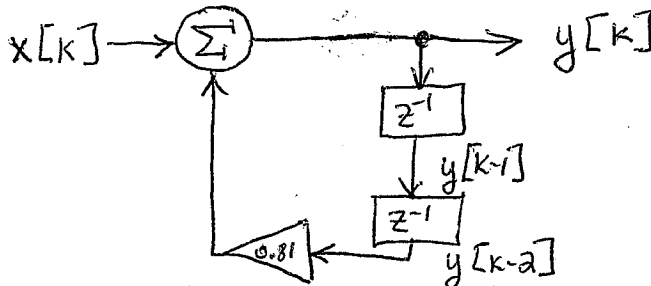
<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	20		Digital Filter Analysis
2	20		Digital Filter Design
3	20		Sampling
4	24		Time-Domain Response
5	16		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 20 points.

A causal discrete-time linear time-invariant filter with input $x[k]$ and output $y[k]$ is governed by the following difference equation:

$$y[k] = 0.81 y[k-2] + x[k]$$

(a) Draw the block diagram for this filter. 4 points.



The flow of data through the block diagram is shown by the arrows.

(b) What are the initial conditions and what values should they be assigned? 4 points.

$$y[0] = 0.81 y[-2] + x[0]$$

$$y[1] = 0.81 y[-1] + x[1]$$

Initial conditions are $y[-1]$ and $y[-2]$. Their values should be zero to guarantee LTI properties

(c) Find the equation for the transfer function in the z-domain including the region of convergence of the filter. 4 points.

Take the z-transform of the difference equation with $y[-1]=0$ and $y[-2]=0$ and isolate the ratio $Y(z)/X(z)$:

$$Y(z) = 0.81 z^{-2} Y(z) + X(z)$$

$$Y(z)(1 - 0.81 z^{-2}) = X(z) \Rightarrow \frac{Y(z)}{X(z)} = \frac{1}{1 - 0.81 z^{-2}} = H(z)$$

(d) Find the equation for the frequency response of the filter. 4 points.

Since the region of convergence (Roc) of the transfer function includes the unit circle, the substitution $z = e^{j\omega}$ is valid:

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1}{1 - 0.81 e^{-j2\omega}}$$

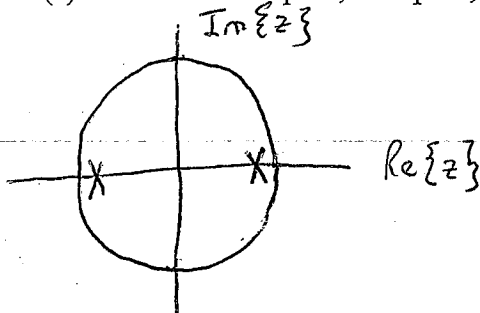
Poles are located at

$$1 - 0.81 z^{-2} = 0$$

$$(1 + 0.9 z^{-1})(1 - 0.9 z^{-1}) = 0$$

$$z = -0.9 \text{ and } z = 0.9$$

(e) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.



Poles near the unit circle (but inside the unit circle) indicate the pass bands of the filter. Pass band at $\omega = 0$. Another pass band at $\omega = \pm \pi$. Bandstop filter.

Roc is $|z| > 0.9$

Problem 1.2-Digital Filter Design. - 20 points.

A dual-tone multiple-frequency (DTMF) signal consists of a sum of two sinusoids of different frequencies.

The table at the right shows the DTMF frequencies for a touchtone landline phone. The sampling rate is 8000 Hz.

Design a digital IIR filter *biquad* to detect a frequency of 1477 Hz but not detect the 1336 Hz frequency.

	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

Alphabet of DTMF symbols for touchtone phones

- (a) Is the digital IIR filter biquad lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.

*A biquad has 2 poles and 0-2 zeros.
We want to pass 1477 Hz and reject 1336 Hz.
This is a bandpass filter with a center frequency of 1477 Hz.*

- (b) Give formulas for the pole-zero locations of the biquad. 8 points.

*$\omega_{1477} = 2\pi \frac{1477 \text{ Hz}}{8000 \text{ Hz}}$; poles located at $p_0 = 0.95e^{j\omega_{1477}}$; $p_1 = 0.95e^{-j\omega_{1477}}$
 $\omega_{1336} = 2\pi \frac{1336 \text{ Hz}}{8000 \text{ Hz}}$; zeros located at $z_0 = e^{j\omega_{1336}}$; $z_1 = e^{-j\omega_{1336}}$*

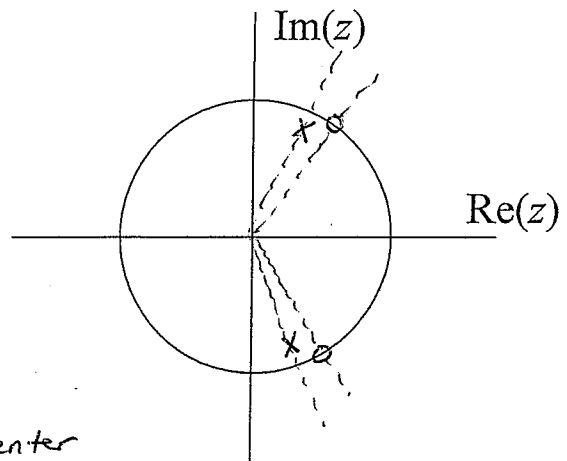
- (c) Draw the poles and zeros on the pole-zero diagram on the right. 4 points.

$\omega_{1336} \approx \frac{\pi}{3}$

*Zeros are conjugate symmetric
poles are conjugate symmetric*

$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$

$\omega_{\text{center}} = 1477 \text{ Hz} \Rightarrow z_{\text{center}} = e^{j\omega_{\text{center}}}$



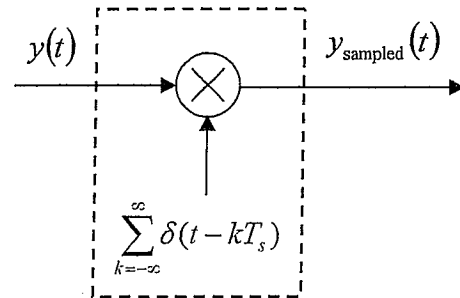
- (d) Compute the scaling constant (gain) for the filter's transfer function. 4 points.

Let $H(z_{\text{center}}) = 1$, i.e. set the filter's response to 1 at 1477 Hz,

$H(z_{\text{center}}) = C \frac{(1 - z_0 z_{\text{center}}^{-1})(1 - z_1 z_{\text{center}}^{-1})}{(1 - p_0 z_{\text{center}}^{-1})(1 - p_1 z_{\text{center}}^{-1})} = 1$ and solve for C.

Problem 1.3. Sampling. 20 points.

One can represent ideal sampling a continuous-time analog signal $y(t)$ at intervals of a sampling period T_s as amplitude modulation by an impulse train to produce a sampled analog signal. We can view the input-output relationship in continuous time as given below:



Further, we know that the Fourier transform of the impulse train is another impulse train:

$$\sum_{k=-\infty}^{\infty} \delta(t - kT_s) \Leftrightarrow f_s \sum_{n=-\infty}^{\infty} \delta(f - n f_s)$$

In practice, we cannot sample at $t = -\infty$. In practice, we would use causal sampling.

- (a) Derive a formula for the Fourier transform of a casual impulse train $u(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s)$ 10 points.

Note: This is identical to letting $y(t) = u(t)$ in the above block diagram.

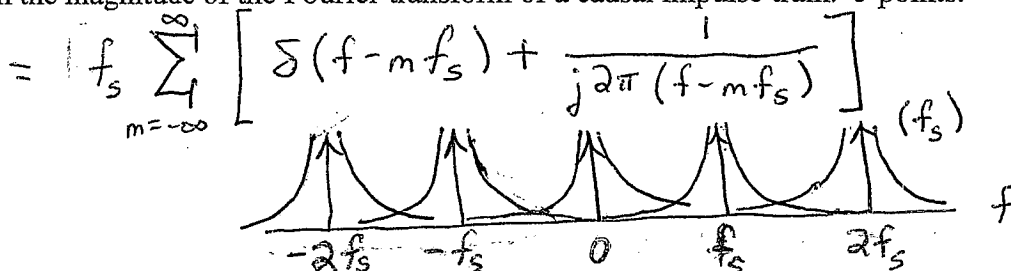
Solution #1: $u(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s) = \sum_{k=0}^{\infty} \delta(t - kT_s)$

$$\mathcal{F}\left\{ \sum_{k=0}^{\infty} \delta(t - kT_s) \right\} = \sum_{k=0}^{\infty} \mathcal{F}\{\delta(t - kT_s)\} = \sum_{k=0}^{\infty} e^{-j2\pi kT_s f}$$

Solution #2: Multiplication in the time domain means convolution in the frequency domain:

$$\begin{aligned} \mathcal{F}\left\{ u(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \right\} &= \mathcal{F}\{u(t)\} * \mathcal{F}\left\{ \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \right\} \\ &= \left[\frac{1}{2} \delta(f) + \frac{1}{j2\pi f} \right] * \left[f_s \sum_{m=-\infty}^{\infty} \delta(f - m f_s) \right] \end{aligned}$$

- (b) Sketch the magnitude of the Fourier transform of a causal impulse train. 5 points.

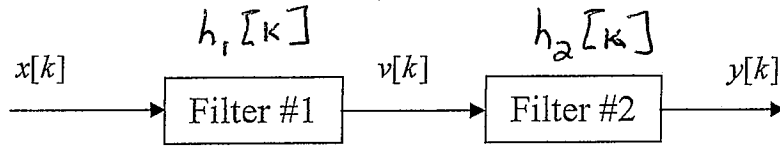


- (c) Describe the difference between the Fourier transforms of the (two-sided) impulse train and the causal (one-sided) impulse train. 5 points.

Both contain Dirac deltas with area f_s replicated every f_s frequency (i.e. an impulse train with spacing between impulses of f_s). The causal version in the frequency domain has an additional periodic term $\frac{1}{j2\pi(f - mf_s)}$ where m is the period index.

Problem 1.4. Time-Domain Response. 24 points.

Two causal, stable, discrete-time, linear time-invariant (LTI) filters are in cascade as shown below.



Filter #1 has an infinite impulse response (IIR). Filter #2 has a finite impulse response (FIR).

Either prove that the cascade of the LTI filters always has an infinite impulse response, or give a counter-example.

The statement is false. Proof by counterexample. Let $|a| < 1$.

Solution in the transform domain - pole-zero cancellation

$$\text{Let } h_1[k] = a h_1[k-1] + x[k] \Rightarrow H_1(z) = \frac{1}{1 - a z^{-1}}$$

$$\text{and } h_2[k] = v[k] - a v[k-1] \Rightarrow H_2(z) = 1 - a z^{-1}$$

$$H(z) = H_1(z) H_2(z) = \left[\frac{1}{1 - a z^{-1}} \right] [1 - a z^{-1}] = 1$$

$$h[k] = \delta[k] \text{ which has a finite impulse response.}$$

(Note: If filter #1 were a communication channel, then filter #2 would be called a channel shortening equalizer because the design of filter #2 shortens the impulse response of the cascade from being infinite in duration to be finite in duration. In addition, the specific examples of $h_1[k]$ and $h_2[k]$ above has $h_2[k]$ as a frequency domain equalizer in that the cascade becomes all-pass.)

Problem 1.5. Potpourri. 16 points.

Please determine whether the following claims are true or false and support each answer with a brief justification. If you give a true or false answer without any justification, then you will be awarded zero points for that answer.

See your notes for in-class discussion of slides 5-17 and 5-18, and end of lecture 6 slides

- (a) The Parks-McClellan algorithm (a.k.a. Remez Exchange) always gives the shortest length FIR filters to meet a piecewise constant magnitude response specification. 4 points.

FALSE The Parks-McClellan algorithm finds the shortest length linear phase FIR filters with floating-point coefficients to meet a piecewise constant magnitude response specification. The Parks-McClellan algorithm is iterative and may fail to converge to a solution, e.g. for very long FIR filters.

- (b) The Laplace transform is a generalized Fourier transform in the sense that every continuous-time signal that has a Fourier transform has a Laplace transform. 4 points.

FALSE The two-sided signal $\cos(2\pi f_0 t)$ has a Fourier transform but not a (two-sided) Laplace transform. (This question appears on slide 5-12.)

- (c) When the order of two linear time-invariant (LTI) filters in cascade is switched, the input-output relationship of the cascade always remains the same. 4 points.

FALSE From slide 5-14, "order of two LTI systems in cascade can be switched under the assumption that computations are performed in exact precision." This is why the order of biquads in a cascaded IIR implementation

- (d) In a communications receiver, you are asked to implement a sinusoidal generator for a downconversion (demodulation) stage in C on a programmable digital signal processor. The sinusoidal generator will run at a frequency provided externally. (The external frequency reference is necessary to adapt to the carrier frequency used by the transmitter and altered by the channel.) The most efficient implementation of sinusoidal generator is to use a C function call to either the cos or sin function. 4 points.

matters, and why the IIR filter structure matters.

Clarification: The value of the frequency ω_0 can change over time. The change is due to variation in the transmitter oscillator over time, and nonlinear and time-varying effects from the analog front ends (e.g. power amplifier) and channel.

FALSE All three sinusoidal generation methods (C function call, lookup table, and difference equation) can support a time-varying value for ω_0 . (the lookup table would require interpolation). Lookup table or difference equation is more efficient than a C call.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 8, 2007

Course: EE 345S Evans

Name: Set Solution
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Please disable all wireless connections on your computer system.**
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	20		Digital Filter Analysis
2	20		Digital Filter Design Part I
3	20		Digital Filter Design Part II
4	20		Phase
5	20		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 20 points.

A causal discrete-time linear time-invariant filter with input $x[k]$ and output $y[k]$ is governed by the following difference equation

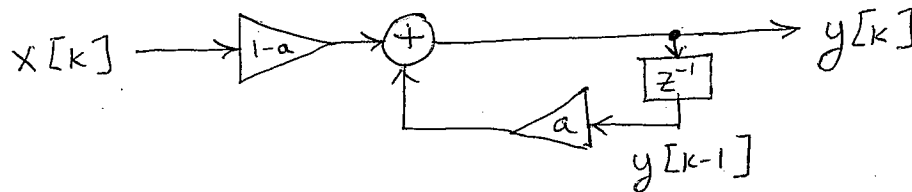
$$y[k] = a y[k-1] + (1-a)x[k]$$

where a is a real-valued constant with $0 < a < 1$.

(a) Is this a finite impulse response filter or infinite impulse response filter? 2 points.

IIR. Difference equation relies on the previous output value (has feedback). The system has a non-zero pole at $z=a$.

(b) Draw the block diagram for this filter. 4 points.



(b) What are the initial conditions and what values should they be assigned? 2 points.

$$y[0] = a y[-1] + (1-a)x[0]$$

Initial condition is $y[-1]$. We set $y[-1]=0$ to ensure LTI properties.

(c) Find the equation for the transfer function in the z -domain including the region of convergence. 4 points.

$$Y(z) = a z^{-1} Y(z) + (1-a) X(z)$$

$$(1 - a z^{-1}) Y(z) = (1-a) X(z)$$

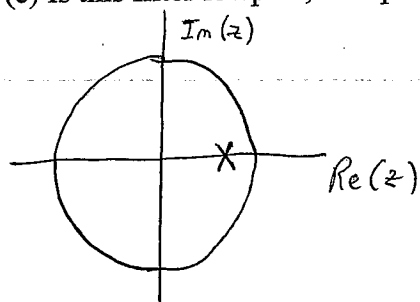
$$\frac{Y(z)}{X(z)} = \frac{1-a}{1 - a z^{-1}} \quad \text{for } |z| > a \text{ due to causality.}$$

(d) Find the equation for the frequency response of the filter. 4 points.

Since the region of convergence for the transfer function contains the unit circle, i.e. the filter is stable,

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1-a}{1 - a e^{-j\omega}}$$

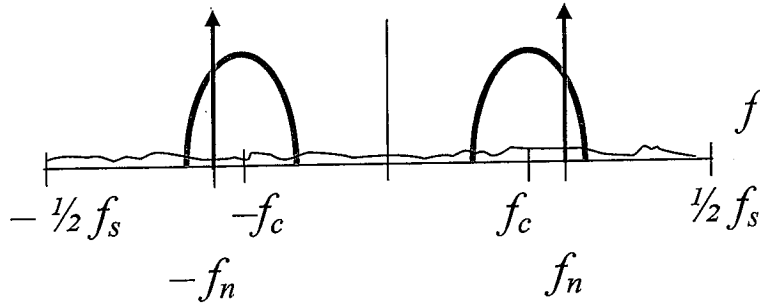
(e) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.



*Filter has one pole at $z=a$, $0 < a < 1$. $\omega_{\text{pole}}=0$.
Filter has no zeros. Pole determines passband.
Filter is lowpass when $a \approx 1$ and essentially allpass when $a \approx 0$.*

Problem 1.2 Digital Filter Design. Part I. 20 points.

Consider a transmission of an amplitude modulated signal centered at carrier frequency f_c over a communication channel with additive noise. There is narrowband interference at frequency f_n . The spectrum of the received signal is shown below:



The first three stages of the receiver (after the antenna) are shown below.



The A/D converter samples at a sampling rate of f_s . Assume that $f_s \gg f_c$, as shown above.

Design filter #1 as a digital IIR biquad to remove the narrowband interference while passing the other frequencies as much as possible by properly placing poles and zeros.

(a) Is the filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.

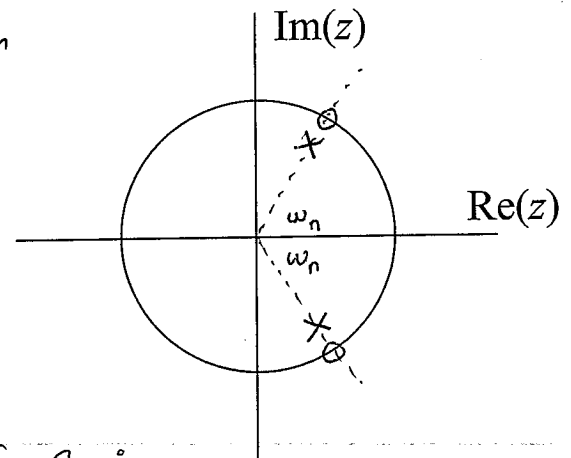
Notch. We seek to eliminate only one frequency (f_n) and pass all other frequencies as much as possible.

(b) Give formulas for the pole-zero locations of the biquad. 8 points.

$$\omega_n = 2\pi \frac{f_n}{f_s}; \quad p_0 = 0.9 e^{-j\omega_n}; \quad p_1 = 0.9 e^{+j\omega_n}$$

$$z_0 = e^{-j\omega_n}; \quad z_1 = e^{+j\omega_n}$$

(c) Draw the poles and zeros on the pole-zero diagram on the right. 4 points.



(d) Compute the scaling constant (gain) for the filter's transfer function. 4 points.

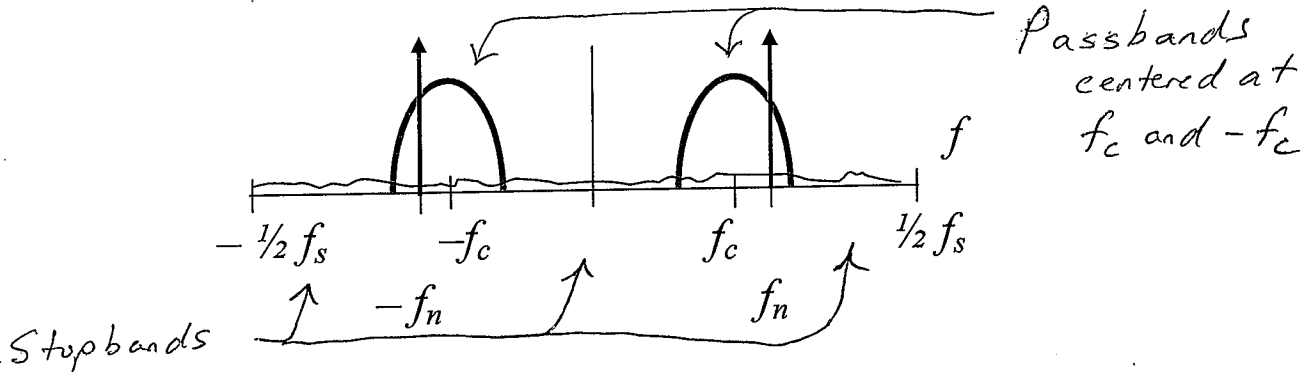
$$H_1(z) = C_1 \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

Set the DC gain to be one and solve for C_1 :

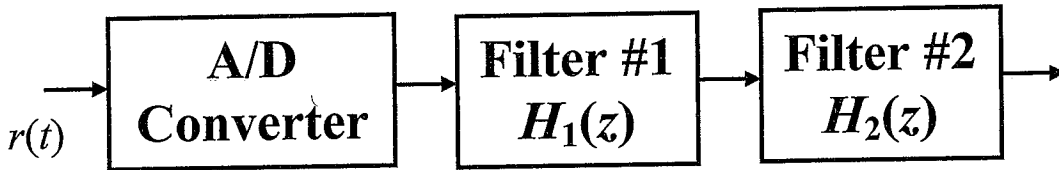
$$H(1) = C_1 \frac{(1 - z_0)(1 - z_1)}{(1 - p_0)(1 - p_1)} = 1$$

Problem 1.3 Digital Filter Design. Part II. 20 points.

As in problem 1.2, consider a transmission of an amplitude modulated signal centered at carrier frequency f_c over a communication channel with additive noise. There is narrowband interference at frequency f_n . The spectrum of the received signal is shown below:



The first three stages of the receiver (after the antenna) are shown below.



The A/D converter samples at a sampling rate of f_s . Assume that $f_s \gg f_c$, as shown above.

Assuming that filter #1 has been properly designed to remove the narrowband interference, design filter #2 as a digital IIR biquad to maximize the ratio of the signal power to the noise power (i.e. the signal-to-noise ratio) as much as possible by properly placing poles and zeros.

(e) Is the filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.

Bandpass. We want to pass as much of the bandpass transmission band and reject out-of-band noise as much as possible.

(f) Give formulas for the pole-zero locations of the biquad. 8 points.

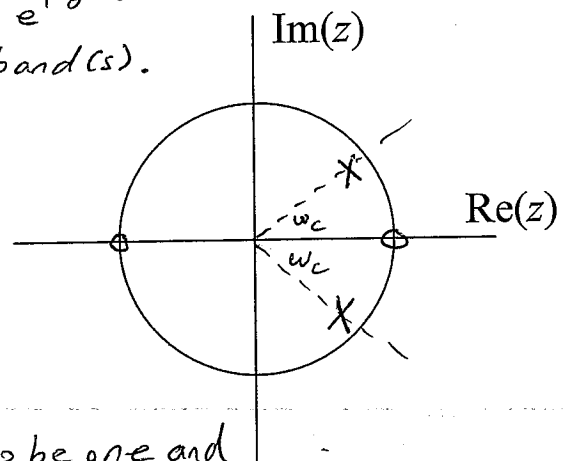
Poles near the unit circle indicate passband(s).

$$\omega_c = 2\pi \frac{f_c}{f_s}; \quad p_0 = 0.9 e^{-j\omega_c}; \quad p_1 = 0.9 e^{+j\omega_c}$$

Zeros on or near the unit circle indicate stopband(s).

$$z_0 = 1; \quad z_1 = -1$$

(g) Draw the poles and zeros on the pole-zero diagram on the right. 4 points.



(h) Compute the scaling constant (gain) for the filter's transfer function. 4 points.

$$H_2(z) = C_2 \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

Set the gain at the carrier frequency to be one and solve for C_2 :

$$H_2(z_c) = C_2 \frac{(1 - z_0 z_c)(1 - z_1 z_c)}{(1 - p_0 z_c)(1 - p_1 z_c)} = 1 \quad \text{with } z_c = e^{j\omega_c}$$

Problem 1.4. Phase. 20 points.

The following discrete-time linear time-invariant (LTI) system with input $x[n]$ and output $y[n]$ is governed by the difference equation

$$y[n] = b x[n] + 2 b x[n-1] + b x[n-2]$$

where b is a real-valued and positive constant. Either prove that the LTI system has linear phase, or give a counter-example.

Proof that the LTI system has linear phase.

1. Compute the transfer function by taking the z -transform of both sides of the equation:

$$Y(z) = b X(z) + 2 b z^{-1} X(z) + b z^{-2} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = b + 2 b z^{-1} + b z^{-2} \text{ for } z \neq 0$$

2. Compute the frequency response. Since the unit circle is in the region of convergence of the transfer function,

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = b + 2 b e^{-j\omega} + b e^{-2j\omega}$$

3. Isolate the magnitude and phase responses.

$$\begin{aligned} H_{\text{freq}}(\omega) &= e^{-j\omega} (b e^{j\omega} + 2b + b e^{-j\omega}) \\ &= e^{-j\omega} (2b \cos \omega + 2b) \end{aligned}$$

phase response with phase $-\omega$ magnitude response because $2b \cos \omega + 2b \geq 0$

4. Phase is $-\omega$, which is linear with slope of -1 . QED.

Problem 1.5. Potpourri. 20 points.

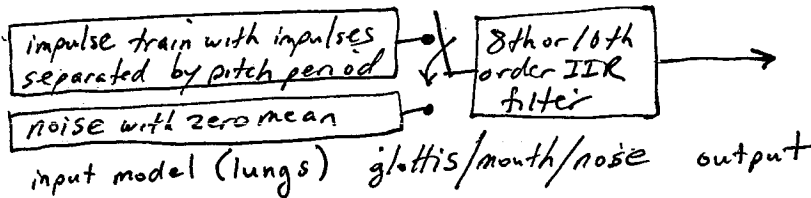
(a) Give at least one application of each of the following types of filters. 6 points.

- Lowpass — anti-aliasing filter (before sampling) in an A/D converter; anti-imaging filter (after interpolation) in a D/A converter
- Bandpass — transmit and receive filters to reject out-of-band noise and interferers (as in problem 1.3); audio filtering (remove DC)
- Bandstop — remove AM radio station in ADSL transmission band
- Highpass — enhance lines and texture in images (from in-class demo)
- Notch — remove narrowband interferer such as 60 Hz power line frequency (see problem 1.2)
- Allpass — phase correction after A/D conversion (which has an IIR filter in it)

to match human hearing; filtering of EKG signals

(b) Using digital signal processing, how would you change a sampled voice signal of a male speaker into a female speaker or a female speaker into a male speaker? 6 points.

Class lecture described how speech is compressed on cell phones.



Every 10-40 ms of speech is modeled as shown on left.
 Male voice: pitch \approx 100 Hz.
 Female voice: pitch \approx 200 Hz.

(c) There are four combinations below. For each combination, please indicate whether you would use a digital finite impulse response (FIR) filter or a digital infinite impulse response (IIR) filter and why. 8 points.

So, keep the same IIR filter model for the speech and change the pitch period of the input.

	Floating-point programmable digital signal processor	Fixed-point programmable digital signal processor
Bandpass filtering of digital audio (stereo) prior to conversion to analog and playback	same non-linear phase distortion on both audio channels can be cancelled by human auditory system IIR because (a)	FIR because (b)
Lowpass filtering of digital audio (single channel) prior to conversion to analog and transmission on AM radio channel	for single-channel audio, linear phase is essential FIR because (c)	FIR because of (b) and (c)

- (a) IIR filters generally have significantly lower implementation complexity than FIR filters to meet the same magnitude specification.
- (b) FIR filters are always stable, regardless of implementation technology.
- (c) Linear phase over all frequencies only possible with FIR filter with impulse response that is symmetric about midpoint.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 12, 2007

Course: EE 345S Evans

Name: _____
 Set *Solution*
 Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system.***
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Digital Filter Analysis
2	30		Upconversion
3	25		Digital Filter Design
4	20		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 25 points.

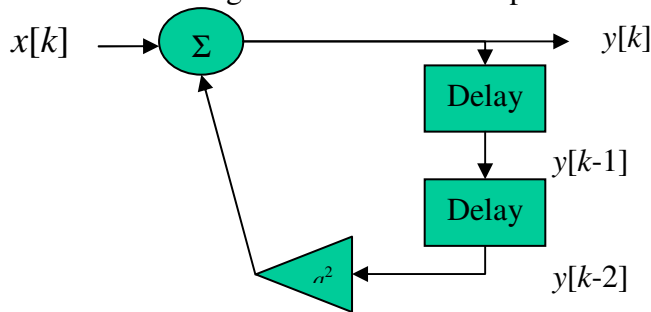
A causal discrete-time linear time-invariant filter with input $x[k]$ and output $y[k]$ is governed by the following difference equation

$$y[k] = a^2 y[k - 2] + (1 - a) x[k]$$

where a is a real-valued constant with $0 < a < 1$.

Note: The output is a combination of the current input and the output two samples ago.

- (a) Is this a finite impulse response filter or infinite impulse response filter? Why? 2 points.
The current output $y[k]$ depends on previous output $y[k-2]$. Hence, the filter is IIR.
- (b) Draw the block diagram for this filter. 4 points.



- (c) What are the initial conditions and what values should they be assigned? 4 points.
 $y[0] = a^2 y[-2] + (1 - a) x[0]$ Hence, the initial conditions are $y[-1]$ and $y[-2]$,
 $y[1] = a^2 y[-1] + (1 - a) x[1]$ i.e., the initial values of the memory locations for
 $y[2] = a^2 y[0] + (1 - a) x[2]$ $y[k - 1]$ and $y[k - 2]$. These initial conditions should
 be set to zero for the filter to be linear & time-invariant
- (d) Find the equation for the transfer function in the z -domain including the region of convergence. 5 points.

Take the z-transform of both sides of the difference equation:

$$Y(z) = a^2 z^{-2} Y(z) + (1 - a) X(z)$$

$$Y(z) - a^2 z^{-2} Y(z) = (1 - a) X(z)$$

$$(1 - a^2 z^{-2}) Y(z) = (1 - a) X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - a}{1 - a^2 z^{-2}} = \frac{1 - a}{(1 - az^{-1})(1 + az^{-1})} \quad \text{Hence, poles are located at } z = a \text{ and } z = -a.$$

Since the system is causal, the region of convergence is $|z| > a$.

- (e) Find the equation for the frequency response of the filter. 5 points.

System is stable because the two poles are located inside the unit circle since $0 < a < 1$.

Because the system is stable, we can convert the transfer function to a frequency response:

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = \frac{1 - a}{1 - a^2 e^{-j2\omega}}$$

- (f) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? What value of the parameter a would you use? 5 points.

Poles are at angles 0 rad/sample (low frequency) and π rad/sample (high frequency).

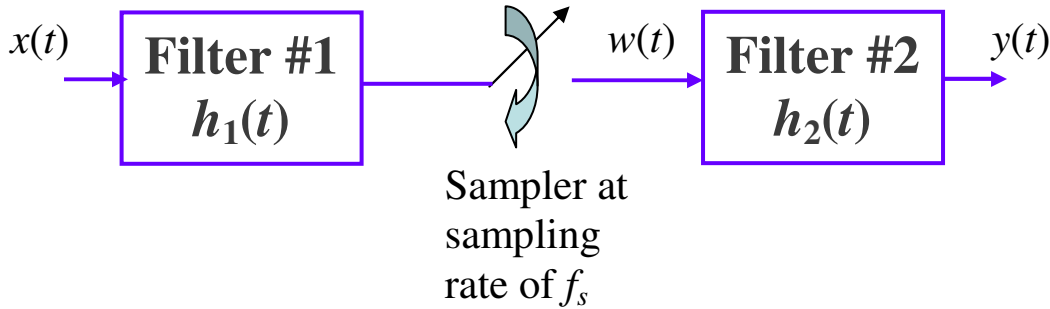
When $a \approx 0$, the filter is close to allpass.

When $a \approx 1$, the filter is bandstop. Poles close to the unit circle indicate the passband(s).

Problem 1.2 Upconversion. 30 points.

You're the owner of The Zone AM radio station (AM 1300 kHz), and you've just bought KLBJ (AM 590 kHz). As a temporary measure, you decide to broadcast the same content (speech/audio) over both stations.

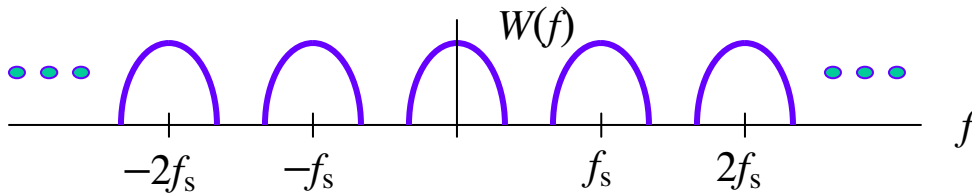
Carrier frequencies for AM radio stations are separated by 10 kHz. The speech/audio content is limited to a bandwidth of 5 kHz.



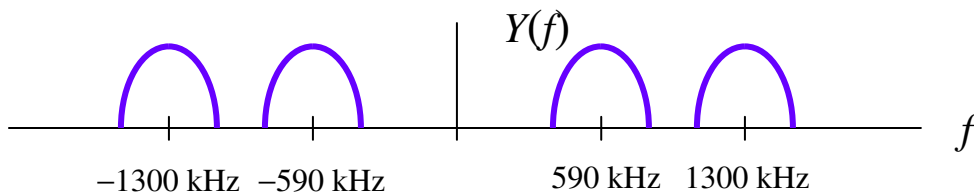
The output $y(t)$ should contain an AM radio signal at carrier frequency 590 kHz and an AM radio signal at carrier frequency 1300 kHz. The input $x(t) = 1 + k_a m(t)$, where $m(t)$ is the speech/audio signal to be broadcast. Since $m(t)$ could come from an audio CD, the bandwidth of $m(t)$ could be as high as 22 kHz. **Note: Filter #1 is anti-aliasing filter. Filter #2 is a two-passband bandpass filter.**

(a) Continuous-Time Analysis. 15 points.

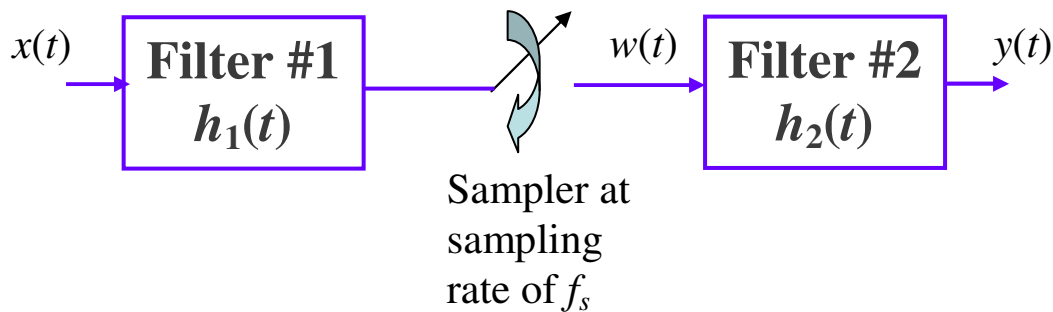
- 1) Specify a passband frequency, passband deviation, stopband frequency, and stopband attenuation for filter #1. **Speech/audio bandwidth for AM radio is limited to 5 kHz. Filter #1 enforces this requirement (see homework problems 2.3 and 3.3). Assuming the ideal passband response is 0 dB, $A_{\text{pass}} = -1$ dB and $A_{\text{stop}} = -90$ dB. The 90 dB of comes from the dynamic range of the audio CD. Also, $f_{\text{stop}} < 5$ kHz. We'll choose $f_{\text{pass}} = 4.3$ kHz and $f_{\text{stop}} = 4.8$ kHz. The transition region is roughly 10% of f_{pass} .**
- 2) Give the sampling rate f_s of the sampler. **We want to produce replicas of filter #1 output centered at 590 kHz and 1300 kHz. Also $f_s > 2f_{\text{max}}$ and $f_{\text{max}} = 4.8$ kHz. So, $f_s = 10$ kHz.**
- 3) Draw the spectrum of $w(t)$. **Each lobe below is $2f_{\text{max}}$ wide.**



- 4) Give the filter specifications to design filter #2. **Filter #2 passbands are 585.7–594.3 kHz and 1295.7–1304.3 kHz, and stopbands are 0–585 kHz, 595–1295 kHz, and greater than 1305 kHz. These bands have counterparts in negative frequencies.**
- 5) Draw the spectrum of $y(t)$. **Each lobe below is $2f_{\text{max}}$ wide.**



The block diagram for the system is repeated here for convenience:



(b) Discrete-Time Implementation. 15 points.

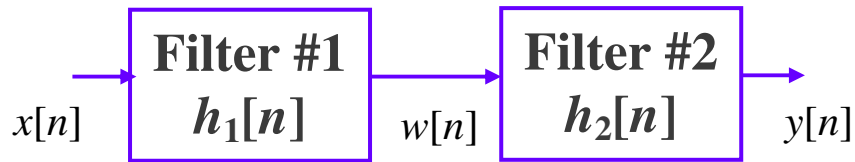
- 1) Give a second sampling rate to convert the continuous-time system to a discrete-time system. **There are two conditions on the second sampling rate, as seen in homework problem 3.2. First, we'll need to pick a second sampling rate f_{s2} for $x(t)$, $w(t)$, and $y(t)$ that minimizes aliasing. The maximum frequencies of interest for $x(t)$ and $y(t)$ are 22 kHz and 1305 kHz, respectively. In theory, $w(t)$ is not bandlimited. Second, we'll need to pick the second sampling rate to be an integer multiple of f_s . In summary,**

$$f_{s2} > 2 (1305 \text{ kHz}) \quad \text{and} \quad f_{s2} = k f_s, \text{ where } k \text{ is an integer}$$

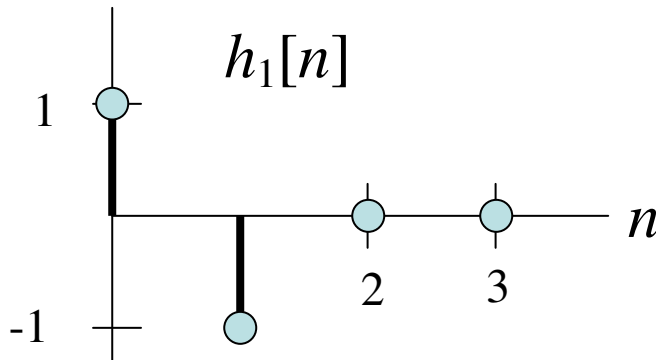
- 2) Would you use a finite impulse response (FIR) or infinite impulse response (IIR) filter for filter #1? Why? **In audio, phase is important. AM radio stations generally broadcast single-channel audio. (AM stereo had gains in popularity in the 1990s, but has been in decline due to digital radio.) Assuming single-channel transmission, filter #1 should have linear phase. Hence, filter #1 should be FIR.**
- 3) What filter design method would you use to design filter #1? Why? **I would use the Parks-McClellan (Remez exchange) algorithm to design the shortest linear phase FIR filter.**
- 4) Would you use a finite impulse response (FIR) or infinite impulse response (IIR) filter for filter #2? Why? **As per part (2), filter #2 should have linear phase, and hence be FIR. Also, filter #2 is a multiband bandpass filter. It is not clear how to use classical IIR filter design methods to design such a filter.**
- 5) What filter design method would you use to design filter #2? Why? **Through homework assignments, we have designed multiband FIR filters using the Parks-McClellan (Remez exchange) algorithm. The Kaiser window method is for lowpass FIR filters. The FIR Least Squares method could be used. I would use the Parks-McClellan (Remez exchange) algorithm to design the shortest multiband linear phase FIR filter.**

Problem 1.3 Digital Filter Design. 25 points.

Consider the following cascade of two causal discrete-time linear time invariant (LTI) filters with impulse responses $h_1[n]$ and $h_2[n]$, respectively:



Filter #1 has the following impulse response:



The (group) delay through filter #1 is $\frac{1}{2}$ sample. **Note: Filter #1 is a first-order difference filter.**

Design filter #2 so that it satisfies all three of the following conditions:

- a. Cascade of filter #1 and filter #2 has a bandpass magnitude response,
- b. Cascade of filter #1 and filter #2 has (group) delay that is an integer number of samples, and
- c. Filter #2 has minimum computational complexity.

Group delay is defined as the negative of the derivative (with respect to frequency) of the phase response. As discussed in lecture, a linear phase FIR filter with N coefficients has a group delay of $(N-1)/2$ samples for all frequencies. So, a first-order difference filter has a delay of $\frac{1}{2}$ samples.

As we saw in the mandrill (baboon) image processing demonstration, a cascade of a highpass filter (first-order differencer) and a lowpass filter (averaging filter) has a bandpass response, provided that there is overlap in their passbands.

A two-tap averaging filter has a group delay of $\frac{1}{2}$ samples. A cascade of a first-order difference filter and a two-tap averaging filter would have a group delay of 1 sample.

A two-tap averaging filter with coefficients equal to one would only require 1 addition per output sample. No multiplications required. This is indeed low computational complexity.

$$h_2[n] = \delta[n] + \delta[n-1]$$

Problem 1.4. Potpourri. 20 points.

Please determine whether the following claims are true or false and support each answer with a brief justification. If you give a true or false answer without any justification, then you will be awarded zero points for that answer.

- (a) The automatic order estimator for the Parks-McClellan (a.k.a. Remez Exchange) algorithm always gives the shortest length FIR filters to meet a piecewise constant magnitude response specification. 4 points. **False, for two different reasons. First, the Parks-McClellan algorithm designs the shortest length linear phase FIR filters with floating-point coefficients to meet a piecewise constant magnitude response specification. Second, the automatic order estimator is an important but nonetheless empirical formula developed by Jim Kaiser. It can be off the mark by as much as 10%. Sometimes, the order returned by the automatic order estimator does not meet the filter specifications.**
- (b) All linear phase finite impulse response (FIR) filters have even symmetry in their coefficients. Assume that the FIR coefficients are real-valued. 4 points. **False. It true that FIR filters that have even symmetry in their coefficients (about the mid-point) have linear phase. However, as mentioned in lecture, FIR filters with coefficients that have odd symmetry (about the mid-point) also have linear phase.**
- (c) If linear phase finite impulse response (FIR) floating-point filter coefficients were converted to signed 16-bit integers by multiplying by 32767 and rounding the results to the nearest integer, the resulting filter would still have linear phase. 4 points. **True. An FIR filter has linear phase if the coefficients have either odd symmetry or even symmetry about the mid-point. Multiplying the coefficients by a constant does not change the symmetry about the mid-point. In addition, $\text{round}(-x) = -\text{round}(x)$. Hence, rounding does not affect symmetry either.**
- (d) Floating-point programmable digital signal processors are only useful in prototyping systems to determine if a fixed-point version of the same system would be able to run in real time. 4 points. **False, due to the word “only”. It is true that floating-point programmable DSPs are useful in feasibility studies become committing the design time and resources to map a system into fixed-point arithmetic and data types. Beyond that, however, floating-point programmable DSPs are commonly used in low-volume products (e.g. sonar imaging systems and radar imaging systems) and in high-end audio products (e.g. pro-audio, car audio, and home entertainment systems).**
- (e) In the TMS320C6000 family of programmable digital signal processors, consider an equivalent fixed-point processor and floating-point processor, i.e. having the same clock speed, same on-chip memory sizes and types, etc. The fixed-point processor would have lower power consumption. 4 points. **This one could go either way. True. If the data types in a floating-point program were converted from 32-bit floats to 16-bit short integers, and the floating-point computations were converted to fixed-point computations, then the fixed-point processor would consume less power. The fixed-point processor would only need to load from on-chip memory half as often, and multiplication (addition) would take 2 cycles (1 cycle) instead of 4 cycles. Fixed-point multipliers and adders take far fewer gates than their floating-point counterparts, which saves on power consumption. False. If a floating-point program were run by emulating floating-point computations to the same level of precision on a fixed-point processor, then the fixed-point processor would actually consume more power.**

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 7, 2008

Course: EE 345S Evans

Name: _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system.***
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Digital Filter Analysis
2	25		Sinusoidal Generation
3	30		Digital Filter Design
4	20		Potpourri
<i>Total</i>	100		

Problem 1.1 *Digital Filter Analysis.* 25 points.

A causal discrete-time linear time-invariant filter with input $x[k]$ and output $y[k]$ is governed by the following equation

$$y[k] = x[k] + a x[k-1] + x[k-2]$$

where a is a real-valued constant with $1 \leq a \leq 2$.

(a) Is this a finite impulse response filter or infinite impulse response filter? Why? 2 points.

(b) Draw the block diagram for this filter. 4 points.

(c) What are the initial conditions and what values should they be assigned? 4 points.

(d) Find the equation for the transfer function in the z -domain including the region of convergence. 5 points.

(e) Find the equation for the frequency response of the filter. 5 points.

(f) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 5 points.

Problem 1.2 Sinusoidal Generation. 25 points.

Some programmable digital signal processors have a ROM table in on-chip memory that contains values of $\cos(\theta)$ at uniformly spaced values of θ .

Consider the array $c[n]$ of cosine values taken at one degree increments in θ and stored in ROM:

$$c[n] = \cos\left(\frac{\pi}{180}n\right) \text{ for } n = 0, 1, \dots, 359.$$

(a) If the array $c[n]$ were repeatedly sent through a digital-to-analog (D/A) converter with a sampling rate of 8000 Hz, what continuous-time frequency would be generated? 5 points.

(b) How would you most efficiently use the above ROM table $c[n]$ to compute $s[n]$ given below. 5 points.

$$s[n] = \sin\left(\frac{\pi}{180}n\right) \text{ for } n = 0, 1, \dots, 359?$$

(c) How would you most efficiently use the above ROM table $c[n]$ to compute $d[n]$ given below. 5 points.

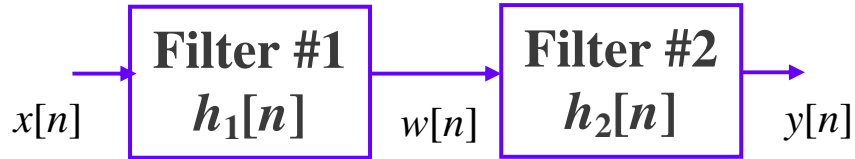
$$d[n] = \cos\left(\frac{\pi}{90}n\right) \text{ for } n = 0, 1, \dots, 179$$

(d) How would you most efficiently use the above ROM table $c[n]$ to compute $x[n]$ given below. 10 points.

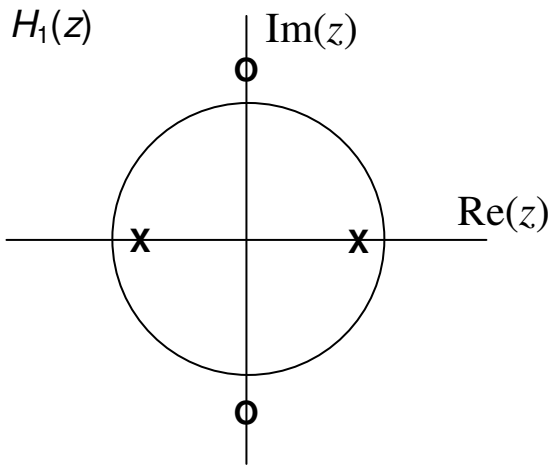
$$d[n] = \cos\left(\frac{\pi}{360}n\right) \text{ for } n = 0, 1, \dots, 719$$

Problem 1.3 Digital Filter Design. 30 points.

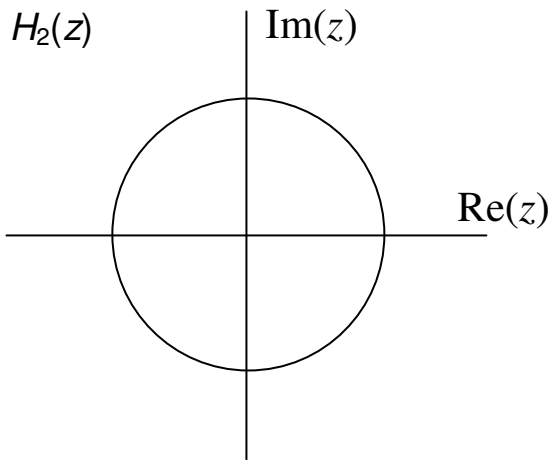
Consider the following cascade of two causal discrete-time linear time invariant (LTI) filters with impulse responses $h_1[n]$ and $h_2[n]$, respectively:



- (a) Poles (X) and zeros (O) for filter #1 are shown below. Assume that the poles have radii of 0.9, and the zeros have radii of 1.2. Is filter #1 a lowpass, highpass, bandpass, bandstop, allpass or notch filter? Why? 10 points.



- (b) Give the transfer function for $H_1(z)$. 5 points.
- (c) Design filter #2 by placing the minimum number of poles and zeros on the pole-zero diagram below so that the cascade of filter #1 and filter #2 is allpass. 10 points.



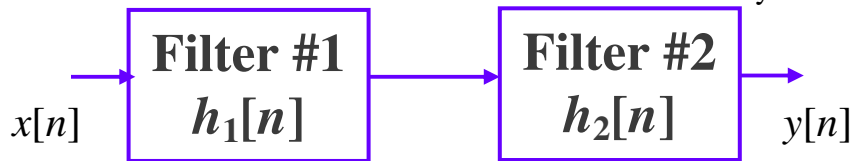
- (d) Give the transfer function $H_2(z)$. 5 points.

Problem 1.4. Potpourri. 20 points.

Please determine whether the following claims are true or false and support each answer with a brief justification. **If you give a true or false answer without any justification, then you will be awarded zero points for that answer.**

(a) Assume that a particular linear phase finite impulse response (FIR) filter design meets a magnitude specification and that no lower-order linear phase FIR filter exists to meet the specification. The linear phase FIR filter will always have the lowest implementation complexity on the TI TMS320C6713 programmable digital signal processor among all filters that meet the same specification. 5 points.

(b) Consider the cascade of two discrete-time linear time-invariant systems shown below.



If the order of the filters in cascade is switched, then the relationship between $x[n]$ and $y[n]$ will always be the same as in the original system. 5 points.

(c) Continuous-time analog signals conform nicely to the Nyquist Sampling Theorem because they are always ideally bandlimited. 5 points.

(d) If $\delta[n]$ were input to a discrete-time system and the output were also $\delta[n]$, then system could only be the identity system, i.e. the output is always equal to the input. 5 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 17, 2008

Course: EE 345S Evans

Name: Set Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. *Please disable all wireless connections on your computer system.*
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Digital Filter Analysis
2	25		Digital Filter Design
3	30		Software Defined Receiver
4	20		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 25 points.

A causal discrete-time linear time-invariant filter with input $x[k]$ and output $y[k]$ is governed by the following equation

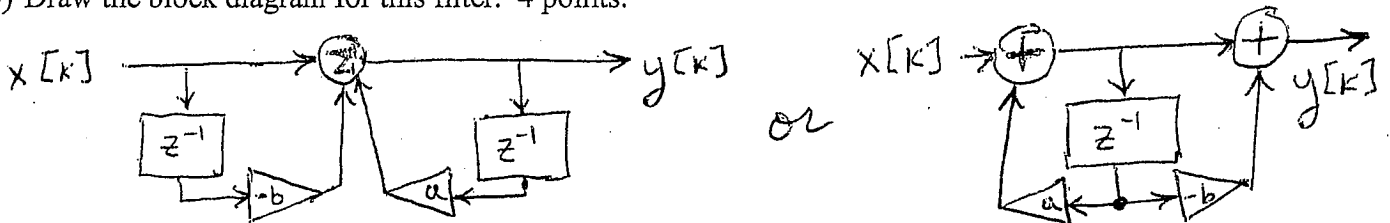
$$y[k] = a y[k-1] + x[k] - b x[k-1]$$

where $|a| < 1$ and $a \neq b$.

- (a) Is this a finite impulse response filter or infinite impulse response filter? Why? 2 points.

IIR filter. The current output value $y[k]$ depends on the previous output value $y[k-1]$.

- (b) Draw the block diagram for this filter. 4 points.



- (c) What are the initial conditions? what values should they be assigned and why? 4 points.

Let $k=0$. $y[0] = a y[-1] + x[0] - b x[-1]$.

Initial conditions $y[-1]$ and $x[-1]$ should be set to zero to

- (d) Find the equation for the transfer function in the z-domain including the region of guarantee convergence. 5 points.

Take the z-transform of both sides:

$$Y(z) = a z^{-1} Y(z) + X(z) - b z^{-1} X(z)$$

$$(1 - a z^{-1}) Y(z) = (1 - b z^{-1}) X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - b z^{-1}}{1 - a z^{-1}}$$

LTI properties.

Pole at $z = a$:

$$1 - a z^{-1} = 0$$

$$a z^{-1} = 1$$

$$z = a$$

For causal system,

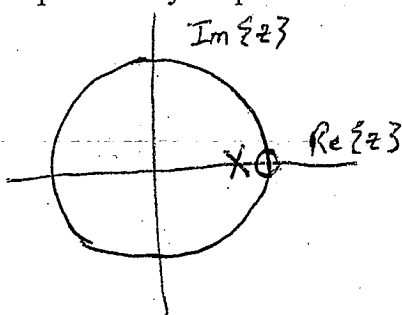
$$|z| > |a|$$

- (e) Find the equation for the frequency response of the filter. 5 points.

Because the region of convergence includes the unit circle, i.e. $|z| > |a|$ and $|a| < 1$,

$$H_{freq}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 - b e^{-j\omega}}{1 - a e^{-j\omega}}$$

- (f) If $a = 0.9$ and $b = 1.0$, would this filter be lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 5 points.

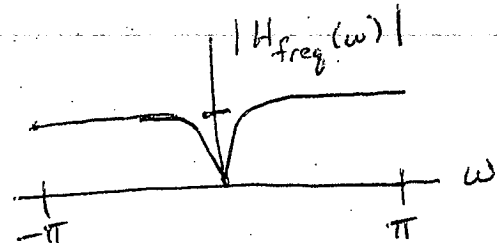


Pole at $z = 0.9$, i.e. at $\omega = 0$

Zero at $z = 1$, i.e. at $\omega = \pi$

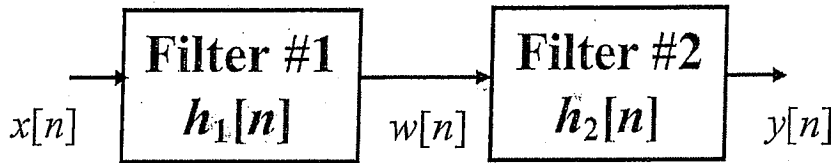
Notch filter.

DC removed.

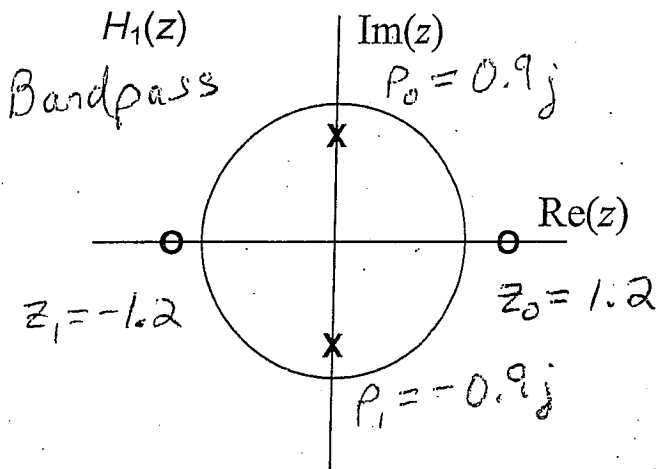


Problem 1.2 Digital Filter Design. 25 points.

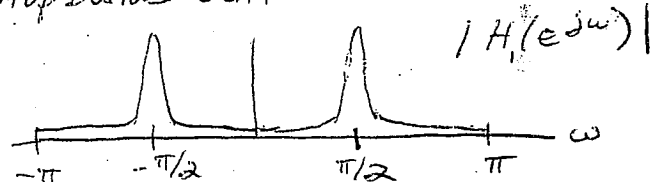
Consider the following cascade of two causal discrete-time linear time invariant (LTI) filters with impulse responses $h_1[n]$ and $h_2[n]$, respectively:



- (a) Poles (X) and zeros (O) for filter #1 are shown below. Assume that the poles have radii of 0.9, and the zeros have radii of 1.2. Is filter #1 a lowpass, highpass, bandpass, bandstop, allpass or notch filter? Why? 10 points.

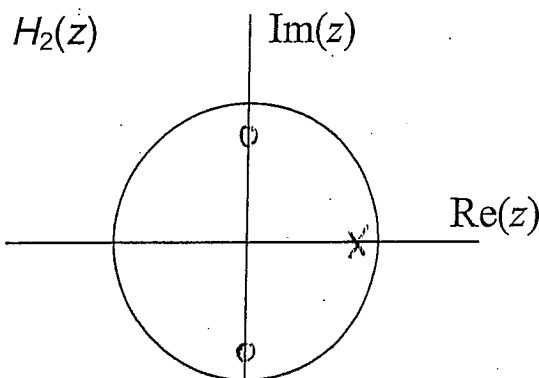


Poles near (but inside) the unit circle indicate passband(s).
 Zeros on or near the unit circle indicate stopband(s).
 So, we have passbands centered at $\omega = \frac{\pi}{2}$ and $\omega = -\frac{\pi}{2}$, and stopbands centered at $\omega = 0$ and $\omega = \pi$.



- (b) Give the transfer function for $H_1(z)$. 5 points.

- (c) Design filter #2 by placing the minimum number of poles and zeros on the pole-zero diagram below so that the *cascade* of filter #1 and filter #2 is *lowpass*. Give the values of the pole and zero locations. 10 points.



$$H_1(z) = C_1 \frac{(z-z_0)(z-z_1)}{(z-p_0)(z-p_1)}$$

Normalize at DC, i.e. $\omega = 0$
 or $z = e^{j\omega} = 1$ for $\omega = 0$,

$$H_1(z) \Big|_{z=1} = 1 \text{ and solve for } C_1$$

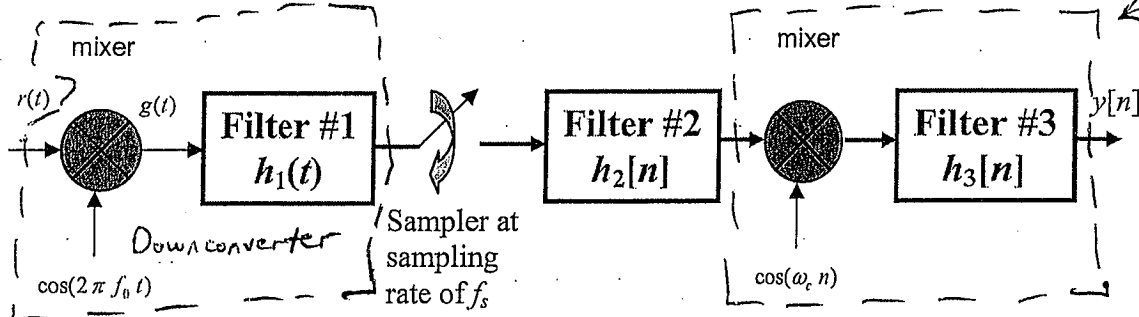
Pole at $z = 0.95$.
 Upper zero at $z = 0.9j$.
 Lower zero at $z = -0.9j$.

There are many solutions here. Filter #2 should be a lowpass filter. We'll need a pole to counteract the zero at $z = z_0$ and a zero to counteract the pole at $z = p_0$ (either in all-pass or notch configuration) or cancel it. Similarly,

Note: In practice, the last two blocks would be replaced by a digital FM demodulator operating at a carrier frequency of ω_c .

Problem 1.3 Software Defined Receiver. 30 points.

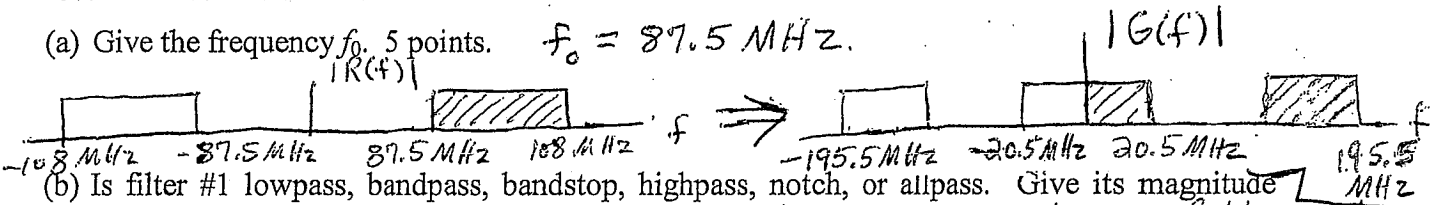
The digital FM radio below has two stages of downconversion, one in continuous time and one in discrete time. Only the discrete-time stage is dependent on the FM station selected by the user, and hence, it can be controlled by software.



The radio receives a bandpass waveform $r(t)$ containing all of the FM radio stations. The bandpass waveform $r(t)$ is downconverted to baseband and sampled. Then, the sampled signal is filtered by filter #2 to extract the bandpass waveform for the FM radio station selected by the user. The output of filter #2 is downconverted to baseband.

The FM band in the US goes from 87.5 MHz to 108.0 MHz, inclusive, and hence -108.0 MHz to -87.5 MHz is also used. Each FM channel has 200 kHz of transmission bandwidth.

(a) Give the frequency f_0 . 5 points. $f_0 = 87.5 \text{ MHz}$.



(b) Is filter #1 lowpass, bandpass, bandstop, highpass, notch, or allpass. Give its magnitude response specification. 5 points. Filter #1 is lowpass anti-aliasing filter and filter #1 is lowpass to extract low frequencies of $G(f)$. 10% rolloff. $f_{pass} = 20.5 \text{ MHz}$, $f_{stop} = 22.75 \text{ MHz}$. $A_{pass} = 1 \text{ dB}$. $A_{stop} = 40 \text{ dB}$

(c) Give the constraints on the sampling rate f_s . 5 points.

$$f_s > 2 f_{stop}$$

Let $w_{band} = 2\pi \frac{200 \text{ kHz}}{f_s}$ be the width of an FM channel after sampling.

(d) Is filter #2 lowpass, bandpass, bandstop, highpass, notch, or allpass. Give its magnitude response specification. 5 points.

Filter #2 is a bandpass filter that is centered at ω_c and has a width of w_{band} in its passband. Allow rolloff of $0.1 w_{band}$.

(e) Give a formula for the discrete-time frequency ω_c in terms of the sampling rate f_s and the user-selected FM radio channel f_{FM} . 5 points.

$$\omega_c = 2\pi \frac{f_{FM} - f_0}{f_s}$$

(f) Is filter #3 lowpass, bandpass, bandstop, highpass, notch, or allpass. Give its magnitude response specification. 5 points.

Lowpass filter. $w_{pass} = 0.9 w_{stop}$. $w_{stop} = \frac{1}{2} w_{band}$. $A_{pass} = 1 \text{ dB}$. $A_{stop} = 40 \text{ dB}$.

Problem 1.4. Potpourri. 20 points.

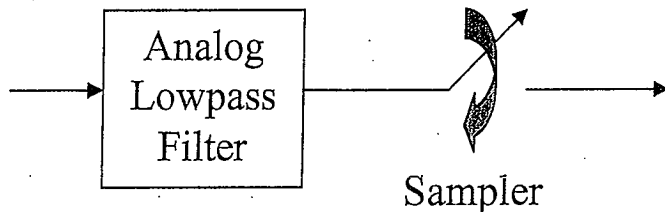
Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) Consider a filter design specification that consists of a constant magnitude response over each frequency band of interest and the allowable deviation from the ideal response in each band. The Parks-McClellan algorithm can always be used to find the shortest linear phase FIR filter with floating-point coefficients to meet the filter design specification. 5 points.

Comment: The approximation used to estimate the minimum order PM filter provides a starting point to search for the minimum filter order.

Solution: The Parks-McClellan algorithm is iterative and may fail to converge for large orders (> 400) as demonstrated in class. **FALSE**

- (b) Consider the cascade of a filter and sampling device below operating at room temperature:



FALSE

It is always possible to design the anti-aliasing causal lowpass filter so that its output is ideally bandlimited to f_{max} so that the sampling rate can be chosen according to $f_s > 2f_{max}$ in order to obey the Sampling Theorem. 5 points. *Practical sampling rates are on the order of 10 GHz.*

Solution #1: Significant thermal noise at frequencies up to 10^{15} Hz is in the input signal, analog filter, filter output, sampler, and sampler output.

Solution #2: In order for the analog lowpass filter to be ideal, i.e. have a rectangular passband, the filter's input response could not be causal.

- (c) Consider the design of a lowpass filter with a maximum passband frequency, passband ripple, minimum stopband frequency, and stopband attenuation. An IIR filter designed with the elliptic design algorithm to meet the filter specification will always have fewer multiplication-accumulation operations per output sample than an FIR filter designed to meet the same filter specification. 5 points.

FALSE

Solution: Consider the FIR filter with an impulse response of $\delta[n] + \delta[n-1]$, i.e. a two-tap averaging filter. An elliptic IIR filter designed to meet the same magnitude specification cannot have fewer MACs.

Solution #2: Consider $\omega_0 = \pi$. $x[n] = \cos(\pi n) \rightarrow z^{-2} \rightarrow y[n] = \cos(\pi(n-2)) = \cos(\pi n - 2\pi) = \cos(\pi n)$. **FALSE.**

- (d) If $\cos(\omega_0 n)$ were input to a discrete-time linear time-invariant system and the output were also $\cos(\omega_0 n)$, then system could only be the identity system, i.e. the output is always equal to the input. 5 points.

Solution #1: According to the Fundamental Theorem of Linear Systems, $x[n] = \cos(\omega_0 n) \Rightarrow y[n] = |H_{freq}(\omega_0)| \cos(\omega_0 n + \angle H_{freq}(\omega_0))$. We are only seeing that $H_{freq}(\omega_0) = 1$. We have no idea about other ω .

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 12, 2009

Course: EE 345S Evans

Name: Set Solution
Last, / First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. *Please disable all wireless connections on your computer system.*
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Digital Filter Analysis
2	30		Digital Filter Implementation
3	25		Simultaneous Broadcast
4	20		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 25 points.

A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following equation

$$y[n] = a^2 y[n-2] + x[n] + x[n-2]$$

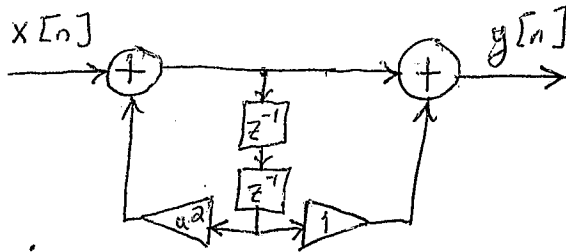
where $|a| < 1$.

- (a) Is this a finite impulse response filter or infinite impulse response filter? Why? 2 points.

Infinite impulse response filter. Filter output relies on previous output values.

- (b) Draw the block diagram for this filter. 4 points.

(Using any of the three direct form filter structures would have been fine here.)



Filter structure based on slide 6-

- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

By computing the first two output values,

$$y[0] = a^2 y[-2] + x[0] + x[-2]$$

$$y[1] = a^2 y[-1] + x[1] + x[-1]$$

initial conditions are

$y[-1], y[-2], x[-1]$ and $x[-2]$

- (d) Find the equation for the transfer function in the z-domain including the region of convergence. 5 points.

With initial conditions set to zero,

$$Y(z) = a^2 z^{-2} Y(z) + X(z) + z^{-2} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + z^{-2}}{1 - a^2 z^{-2}} \text{ for } |z| > |a|$$

The initial conditions should be set to zero so that the filter satisfies linearity and time-invariant properties.

- (e) Find the equation for the frequency response of the filter. 5 points.

Since the region of convergence includes the unit circle, the substitution $z = e^{j\omega}$ is valid:

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 + e^{-2j\omega}}{1 - a^2 e^{-2j\omega}}$$

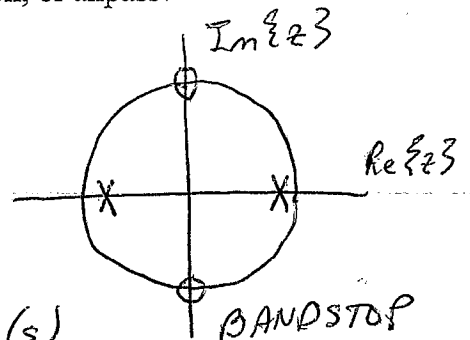
- (f) If $a = 0.9$, would this filter be lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 5 points.

The filter has poles at $z = a$ and $z = -a$, and zeros at $z = +j$ and $z = -j$.

The pole-zero diagram is shown for $a = 0.9$.

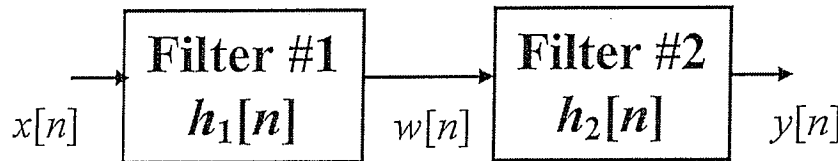
Poles near unit circle indicate pass band(s).

Zeros on or near unit circle indicate stop band(s).

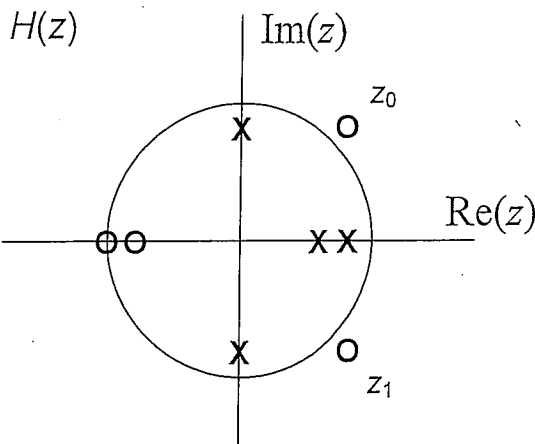


Problem 1.2 Digital Filter Implementation. 30 points.

Consider a cascade of two causal discrete-time linear time invariant (LTI) *biquad* filters with impulse responses $h_1[n]$ and $h_2[n]$, respectively: The impulse response of the cascade is $h[n]$.



Poles (X) and zeros (O) for the cascade of the filters are shown below.



Zero locations

$$z_0 = 1.05 e^{j\pi/4}$$

$$z_1 = 1.05 e^{-j\pi/4}$$

$$z_2 = -1$$

$$z_3 = -0.9$$

Pole locations

$$p_0 = 0.9 e^{j\pi/2}$$

$$p_1 = 0.8$$

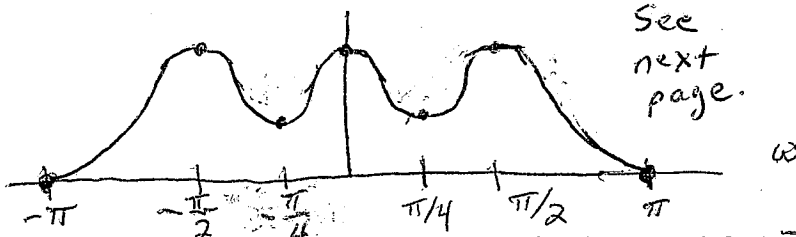
$$p_2 = 0.6$$

$$p_3 = 0.9 e^{-j\pi/2}$$

Poles near the unit circle indicate passband(s).

Zeros near or on the unit circle indicate stopband(s).

(a) Sketch the magnitude response of the cascade. Describe the frequency selectivity. 10 points.



See next page.

Passbands centered at $\omega = -\frac{\pi}{2}$, $\omega = 0$ and $\omega = \frac{\pi}{2}$.
Stopbands centered at $\omega = -\frac{\pi}{4}$ and $\omega = \frac{\pi}{4}$.

Additional stopbands at/near $\omega = -\pi$ and $\omega = \pi$.
Multiband filter.

(b) How would you compute the gain for the cascade? 5 points.

Set gain at $\omega = 0$ to be one, i.e. at $z = 1$:

$$H(1) = \frac{C(1-z_0)(1-z_1)(1-z_2)(1-z_3)}{(1-p_0)(1-p_1)(1-p_2)(1-p_3)} = 1$$

(c) Which poles and zeros would you choose for filter #1? Why? 5 points.

We want to put the pole pair with the lowest quality factor first, i.e. p_1 and p_2 . Then, we pick the closest zeros, i.e. z_0 and z_1 .

Lowpass near $\omega = 0$
Bandpass near $\omega = \frac{\pi}{2}$
and $\omega = -\frac{\pi}{2}$

(d) Is filter #1 a lowpass, highpass, bandpass, bandstop, allpass or notch filter?

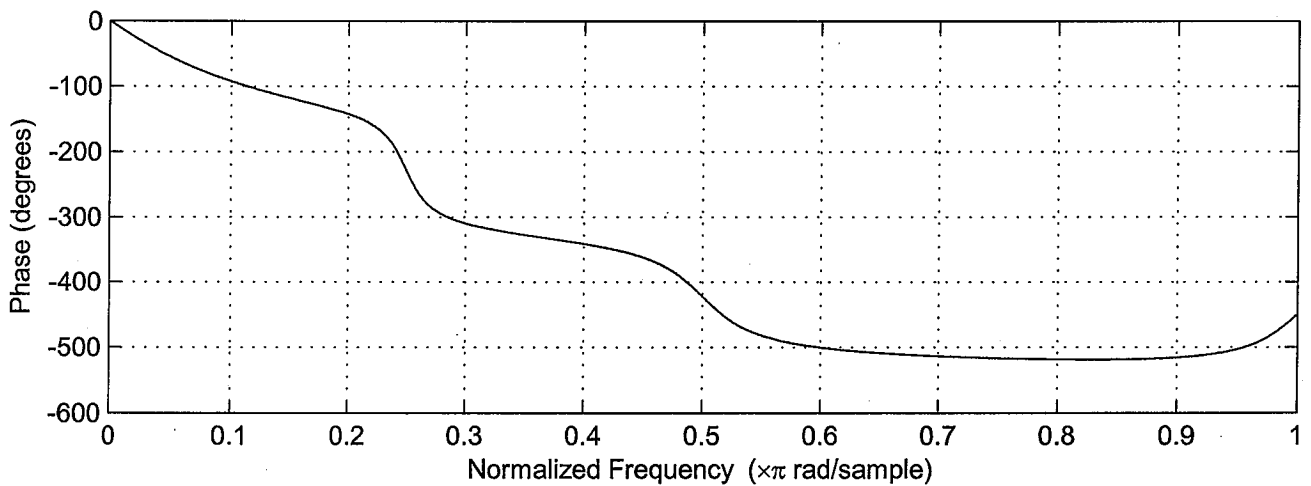
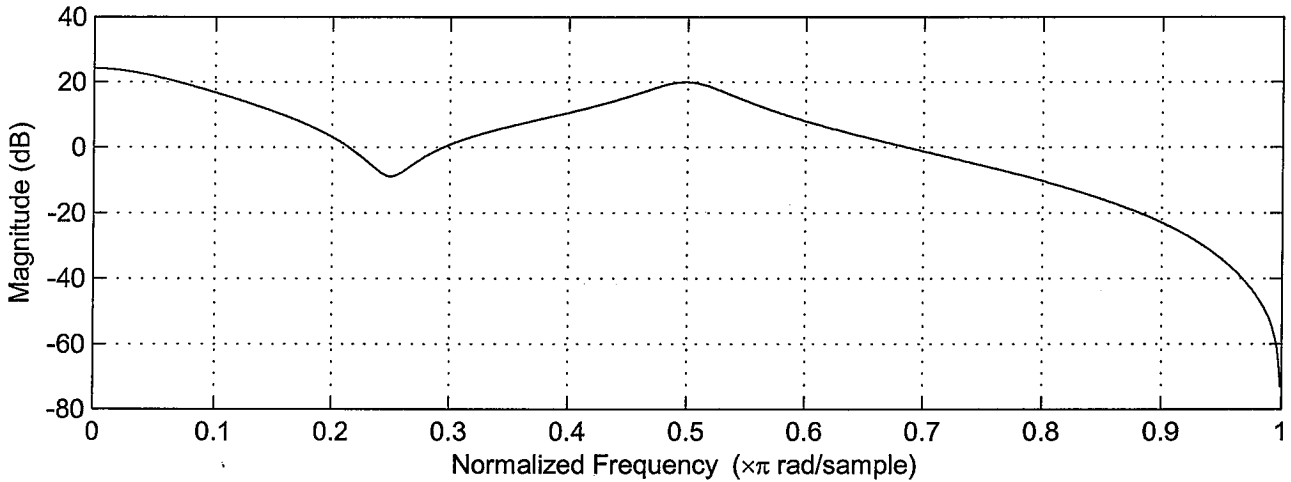
Why? 5 points.

Lowpass. Passband is at $\omega = 0$, stopbands at $\omega = \frac{\pi}{4}$ and $\omega = -\frac{\pi}{4}$.

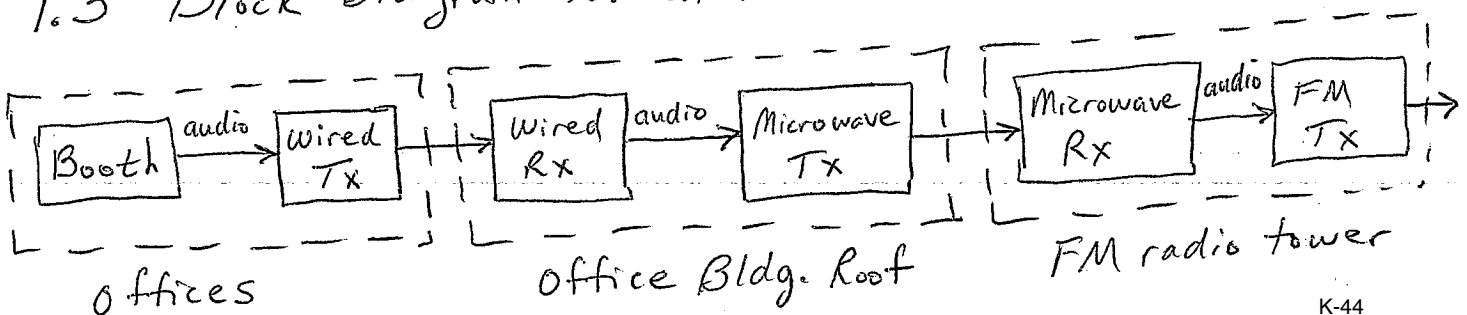
(e) Which poles and zeros would you choose for filter #2? Why? 5 points.

We'll choose the remaining poles p_0 and p_3 and the remaining zeros z_2 and z_3 .

1.2(a). Magnitude response



1.3 Block Diagram for an FM radio Station



Problem 1.3 Simultaneous Broadcast. 25 points.

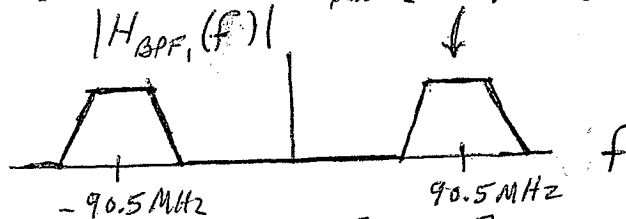
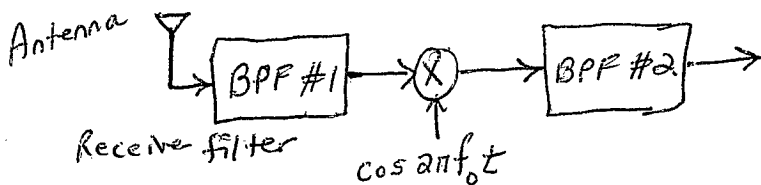
UT Austin runs radio station KUT 90.5 FM in Austin, Texas, and KUTX 90.1 FM in San Angelo, Texas.

Assume that KUT 90.5 FM "simultaneously" broadcasts the same content on KUTX 90.1 FM. This problem will ask you to explore three different ways to achieve "simultaneous" broadcast.

In the US, spacing between adjacent FM stations is 200 kHz. Assume that the FM radio transmission occupies 180 kHz centered at the station frequency.

Please see previous page.

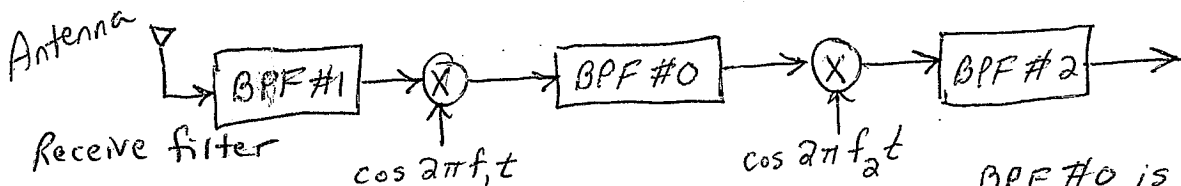
- (a) Draw a block diagram using mixers and filters to directly convert an FM radio transmission from a station frequency of 90.5 MHz to a station frequency on 90.1 MHz. Describe the way you would choose the parameters for each block; e.g., for filters, give the design specifications and the filter design method you would use. In your answer, the FM transmission should not be downconverted to baseband. 10 points.



$$f_0 = 90.5 \text{ MHz} - 90.1 \text{ MHz} = 0.4 \text{ MHz}$$

BPF #2 is same shape as BPF #1 with $f_c = 90.1 \text{ MHz}$.

- (b) Draw a block diagram using mixers and filters to convert an FM transmission at 90.5 MHz to an intermediate frequency to 10 MHz and from an intermediate frequency of 10 MHz to 90.1 MHz. Describe the way you would choose the parameters for each block; e.g., for filters, give the design specifications and the filter design method you would use. In your answer, the FM transmission should not be downconverted to baseband. 10 points.



$$f_1 = 90.5 \text{ MHz} - 10 \text{ MHz} = 80.5 \text{ MHz}$$

$$f_2 = 90.1 \text{ MHz} - 10 \text{ MHz} = 80.1 \text{ MHz}$$

BPF #1 and BPF #2 are same as in (a).
BPF #0 is same shape as BPF #1 in (a) with center frequency at 10 MHz.

- (c) Given that the distance from Austin, Texas, to San Angelo, Texas, is more than 200 miles, please suggest an alternate way to achieve simultaneous FM transmission of content on KUT and KUTX by using existing communication infrastructure but without having to receive and repeat in-air transmission. 5 points.

An assumption in this problem is that there is an FM transmitter for KUT 90.5 MHz in Austin and another FM transmitter for KUTX 90.1 MHz in San Angelo. We can communicate the audio (baseband) content being generated in the studios in Austin to San Angelo via streaming audio using cable/DSL modems or using multiple phone lines (≥ 6) with dedicated modems.

Problem 1.4. Potpourri. 20 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) Consider a filter design specification that consists of a constant magnitude response over each frequency band of interest and the allowable deviation from the ideal response in each band. Assume that the minimum order estimator for the Parks-McClellan filter design algorithm indicates that a 1,000th order finite impulse response (FIR) filter is required. The Parks-McClellan filter design algorithm should always be used to design the FIR filter. 5 points.

False. From in-class demos, the Parks-McClellan algorithm failed to converge for an FIR length of 250. One could use a Kaiser window method to design the FIR filter, which would have a much longer length than 1,000.

- (b) Assume that a discrete-time linear time-invariant (LTI) system has a transfer function in the z-transform domain given by $H(z)$. One can always find the frequency response of the LTI system $H_{\text{freq}}(\omega)$ by substituting $z = e^{j\omega}$ in $H(z)$. 5 points.

False. The z-transform of $u[n]$ is $\frac{1}{1-z^{-1}}$ for $|z| > 1$. The substitution of $z = e^{j\omega}$ would be invalid because the unit circle is not in the region of convergence. The discrete-time

- (c) Discrete-time finite impulse response (FIR) filters are always bounded-input bounded-output ^{Fourier} stable. 5 points.

True. Assume that the FIR filter is linear and time-invariant (init. cond. are zero). LTI system is BIBO stable if

transform of $u[n]$ is $\frac{1}{1-e^{-j\omega}} + \sum_{m=-\infty}^{\infty} \delta(\omega - 2\pi m)$

- (d) If $\delta[n]$ were input to a discrete-time linear time-invariant system and the output were also $\delta[n]$, then system could only be the identity system, i.e. the output is always equal to the input. 5 points.

This does not match $\frac{1}{1-e^{-j\omega}}$

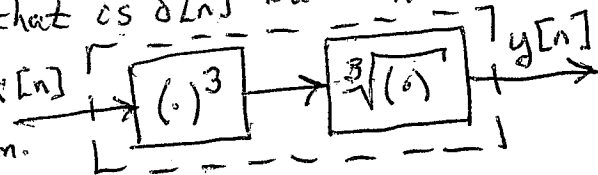
$$\sum_{n=0}^{\infty} |h[n]| < \infty$$

$$\sum_{n=0}^{N-1} |h[n]| < \infty$$

Finite sum of finite values \Rightarrow must be finite

This can be argued either way. True. The impulse response uniquely identifies the LTI system. Here, $h[n] = \delta[n]$. $H(z) = \frac{Y(z)}{X(z)} = 1 \Rightarrow Y(z) = X(z) \Rightarrow y[n] = x[n]$.

False. The following system has an impulse response that is $\delta[n]$ but is not the identity LTI system.



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 16, 2009

Course: EE 345S Evans

Name: Set, Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system.***
- Please turn off all cell phones and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Digital Filter Analysis
2	27		Sinusoidal Generation
3	30		ECG Filter Design
4	18		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 25 points.

A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following equation

$$y[n] = x[n] - 2x[n-1] + x[n-2]$$

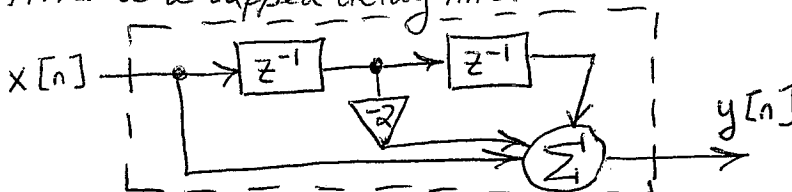
- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 2 points.

Finite impulse response filter. The impulse response is $h[n] = \delta[n] - 2\delta[n-1] + \delta[n-2]$, which has a finite

- (b) Draw the block diagram for this filter. 4 points.

non-zero extent of 3 samples.

The filter is a tapped delay line.



- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

Let $n=0$. $y[0] = x[0] - 2x[-1] + x[-2]$. Initial conditions are $x[-1]$ and $x[-2]$. The initial conditions should be set to zero to satisfy linear and time-invariant properties.

- (d) Find the equation for the transfer function of the filter in the z -domain including the region of convergence. 5 points.

$$Y(z) = X(z) - 2z^{-1}X(z) + z^{-2}X(z)$$

$$Y(z) = (1 - 2z^{-1} + z^{-2})X(z)$$

$$\frac{Y(z)}{X(z)} = 1 - 2z^{-1} + z^{-2} \quad \text{for } z \neq 0$$

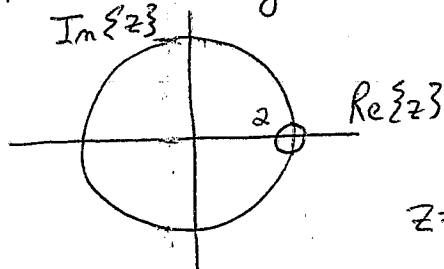
- (e) Find the equation for the frequency response of the filter. 5 points.

Since the region of convergence includes the unit circle,

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = 1 - 2e^{-j\omega} + e^{-j2\omega}$$

- (f) Would the frequency selectivity of the filter be best described as lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 5 points.

Pole-zero diagram Two zeros are located at $z=1$:



$$1 - 2z^{-1} + z^{-2} = 0$$

$$(1 - z^{-1})(1 - z^{-1}) = 0$$

$z=1 \Rightarrow \omega=0$. Stopband at $\omega=0$.

Please note that

$$u(nT_s) = \frac{1}{T_s} u(n) = \frac{1}{T_s} u[n].$$

Problem 1.2 Sinusoidal Generation. 27 points.

Consider generating a causal discrete-time cosine waveform $x[n]$ that has a fixed frequency of $\omega_0 = 2\pi N/L$, where N and L are relatively prime integers and $N < L/2$. A formula for $x[n]$ is

$$x[n] = \cos(\omega_0 n) u[n]$$

identity is used in the solution for part (a).

- (a) If $x[n]$ were input into a digital-to-analog converter operating at sampling rate f_s , give a formula for the frequency f_0 of the causal cosine waveform at the converter output in terms of N, L and f_s . 6 points.

Start with a causal continuous-time signal $\cos(2\pi f_0 t) u(t)$.

Sample at uniformly spaced samples: $t = nT_s = \frac{n}{f_s}$:

$$X[n] = \cos\left(2\pi \frac{f_0}{f_s} n\right) u[n] \Rightarrow \omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{N}{L} \Rightarrow f_0 = \frac{N}{L} f_s$$

- (b) When passing $x[n]$ to the input of the digital-to-analog converter on the C6713 DSK board used in lab, how many different continuous-time frequency values f_0 could the causal cosine waveform take? Why? 6 points.

i. If f_s were held constant, then f_0 could take any one of an infinite number of frequencies in the interval $(-\frac{1}{2}f_s, \frac{1}{2}f_s)$. This is not specific to the C6713 board.

ii. If $\frac{N}{L}$ were held constant, then f_0 could take on one of seven values because the D/A converter on the C6713 can take one of

- (c) How many entries would be in the lookup table to store $x[n]$? Why? 6 points.

Since N and L are relatively prime, $x[n]$ repeats itself every L samples after $n=0$.

$$\text{For example, } x[L] = \cos\left(2\pi \frac{N}{L} L\right) = \cos(0) = x[0]$$

(L samples of $x[n]$ contains N periods.)

- (d) Let $N=3$ and $L=70$. Consider using a lookup table for $x[n]$ to generate a higher frequency cosine waveform $y[k]$ by keeping every M th sample in the lookup table for $x[n]$ by using

$$y[k] = x[Mk] \text{ for } k=0, 1, \dots, L_y-1 \text{ where } L_y = L/M \text{ a.k.a. downsampling by } M$$

How many different values of M could be used without introducing aliasing? 9 points.

$$y[k] = x[Mk] = \cos\left(2\pi \frac{N}{L} (Mk)\right) u[Mk] = \cos\left(2\pi \frac{N}{\frac{L}{M}} k\right) u[Mk]$$

$$= \cos\left(2\pi \frac{N}{L_y} k\right) u[Mk]$$

i. M must be a positive integer

ii. $N < \frac{1}{2} L_y \Rightarrow N < \frac{L}{2M} \Rightarrow M < \frac{L}{2N} \Rightarrow M < 11.67$

iii. M must be a factor of L so that $y[k]$ is periodic.

$$L = 1 \cdot 2 \cdot 5 \cdot 7 \Rightarrow M \in \{1, 2, 5, 7, 10\}$$

Problem 1.3 ECG Filter Design. 30 points.

This problem asks you to design an infinite impulse response (IIR) filter for an ECG system by according to the following specification:

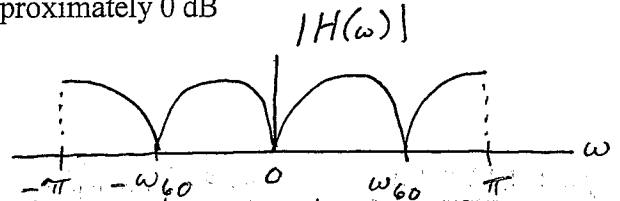
- Sampling rate f_s of 200 Hz
- Stopband attenuation at 0 Hz of at least -40 dB
- Passband from 10 Hz to 50 Hz with amplitude approximately 0 dB
- Stopband attenuation at 60 Hz of at least -40 dB to reduce interference at 60 Hz from the power line
- Passband from 70 Hz to 100 Hz with amplitude approximately 0 dB

Recall from problem 1.2 that $\omega_0 = 2\pi \frac{f_0}{f_s}$

$$\omega_{60} = 2\pi \frac{60}{200} = \frac{3}{5}\pi$$

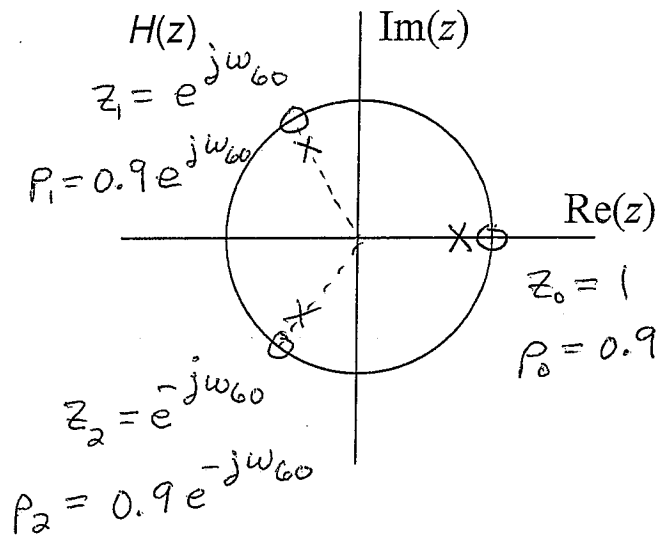
The IIR filter is to have an equal number of poles and zeros.

The IIR filter is to have a minimum number of poles.



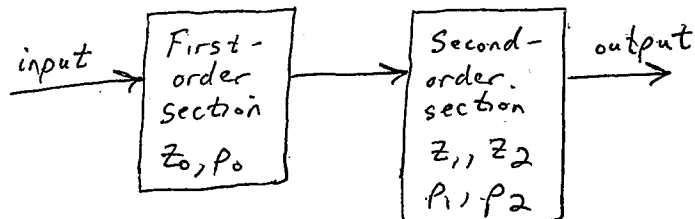
(a) Design the filter by manually placing poles and zeros. Give the pole and zero locations of the filter. 20 points.

Use a cascade of two notch filters.
 First notch filter is at 0 Hz.
 Second notch filter is at 60 Hz.



(b) Let the filter order be N . If N is even, then group the poles and zeros into $N/2$ biquads. If N is odd, then group the poles and zeros into one first-order section and $(N-1)/2$ biquads. Please specify each section in the order you would place it in a cascaded implementation, where the order goes from ECG filter input to ECG filter output. 10 points.

Filter sections should be placed in order of ascending quality factors from filter input to filter output.
 A real-valued pole has the lowest possible quality factor.



Problem 1.4. Potpourri. 18 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- ⚡ #1. The lookup table cannot handle frequency changes (other than select harmonics).
- (a) Consider generating a sinusoidal waveform on the TMS320C6713 digital signal processor Use either by storing one period of the sinusoid in a lookup table, executing a difference equation, or alternate using a C function call. One should always use the lookup table method because it has the lowest computational complexity. 6 points. FALSE.

To generate $\cos(\omega_0 n) u[n]$, where $\omega_0 = 2\pi \frac{N}{L}$,

Method	MACs/sample	ROM	RAM	Quality fip	Quality fxp
C call	30	22	1	Second Best	n/a
Diff. equ.	2	2	3	Worst	Second Best
Lookup table	0	L	0	Best	Best

#2. When L is large, the lookup table may not fit in on-chip memory. Switch to diff. equation method.

- (b) Consider implementing a finite impulse response (FIR) filter in hand-coded assembly language using single-precision floating-point data and arithmetic on the TMS320C6713 digital signal processor on the C6713 DSK board in lab. The longest FIR filter that could be implemented on an audio CD input signal in real time is 1000 coefficients. 6 points. FALSE.

$f_s = 44.1$ kHz for audio CD. Assume one channel.

Clock speed for C6713 processor on C6713 DSK is 225 MHz.

For each audio sample, we can spend $\frac{225 \text{ MHz}}{44.1 \text{ kHz}} = 5102$ clock cycles.

It takes $N+28$ clock cycles on C6700 to implement FIR filter with N coefficients $\Rightarrow N = 5074$.

However, this assumes that

- (c) Consider implementing a finite impulse response (FIR) filter solely in single-precision floating-point data and arithmetic. There are no conditions under which the filter would be linear and time-invariant. 6 points. FALSE.

A necessary but not sufficient condition is that all of the initial conditions are zero. In addition, none of the intermediate single precision floating-point addition or multiplication operations can exceed precision of single precision floating point format. This was the case for the Mandrill demonstration - input signal was 8 bits per pixel and filter coefficients were +1 and -1.

array and past/present inputs fit into on-chip memory. C6713 on the DSK has 1 Kword of 21 data memory. Longest possible filter is 512 coefficients.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 12, 2010

Course: EE 345S Evans

Name: _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system.***
- Please turn off all cell phones and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.**

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Digital Filter Analysis
2	30		Filter Design Tradeoffs
3	24		Downconversion
4	18		Potpourri
<i>Total</i>	100		

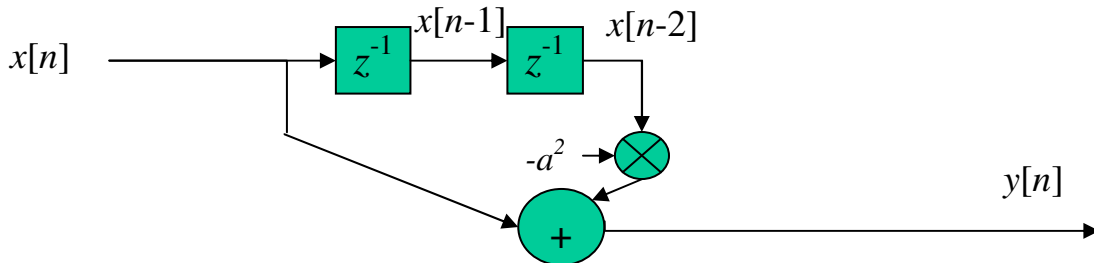
Problem 1.1 Digital Filter Analysis. 28 points.

A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following equation, where a is real-valued,

$$y[n] = x[n] - a^2 x[n-2]$$

(a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 2 points.
The impulse response can be computed by letting the input be discrete-time impulse, i.e. $x[n] = \delta[n]$. The response (output) is $h[n] = \delta[n] - a^2 \delta[n-2]$. The impulse response is finite in extent (3 samples in extent). Hence, the filter is a finite impulse response filter.

(b) Draw the block diagram for this filter. 4 points. **Adapting a tapped delay block diagram,**



(c) What are the initial conditions? What values should they be assigned and why? 4 points.

$y[0] = x[0] - a^2 x[-2]$ Hence, the initial conditions are $x[-1]$ and $x[-2]$,
 $y[1] = x[1] - a^2 x[-1]$ i.e., the initial values of the memory locations for
 $y[2] = x[2] - a^2 x[0]$ $x[n-1]$ and $x[n-2]$. These initial conditions should
be set to zero for the filter to be linear & time-invariant

(d) Find the equation for the transfer function of the filter in the z -domain including the region of convergence. 5 points. **Taking z -transform of both sides of difference equation gives**

$$Y(z) = X(z) - a^2 z^{-2} X(z), \text{ which gives } Y(z) = (1 - a^2 z^{-2}) X(z): H(z) = \frac{Y(z)}{X(z)} = 1 - a^2 z^{-2}$$

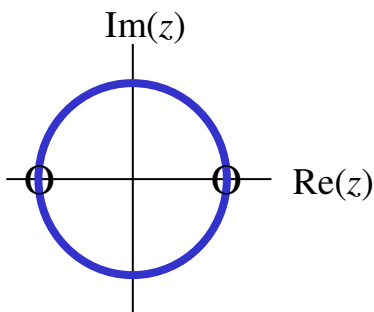
Two zeros at $z=a$ and $z=-a$, and two poles at origin. ROC is entire z plane except origin.

(e) Find the equation for the frequency response of the filter. 5 points. **Since the ROC includes the unit circle, we can convert the transfer function to a frequency response as follows:**

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = 1 - a^2 e^{-j2\omega}$$

(f) For this part, assume that $0.9 < a < 1.1$. Draw the pole-zero diagram. Would the frequency selectivity of the filter be best described as lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 8 points.

Zeros on or near the unit circle indicate the stopband. There are two zeros at $z = a$ and $z = -a$, which correspond to frequencies $\omega = 0$ rad/sample and $\omega = \pi$ rad/sample, respectively. This corresponds to a bandpass filter.



Problem 1.2 Filter Design Tradeoffs. 30 points.

Consider the following filter specification for a narrowband lowpass discrete-time filter:

- Sampling rate f_s of 1000 Hz
- Passband frequency f_{pass} of 10 Hz with passband ripple of 1 dB
- Stopband frequency f_{stop} of 40 Hz with stopband attenuation of 60 dB

Evaluate the following filter implementations on the C6700 digital signal processor in terms of linear phase, bounded-input bounded-output (BIBO) stability, and number of instruction cycles to compute one output value. Assume that the filter implementations below use only single-precision floating-point data format and arithmetic, and are written in the most efficient C6700 assembly language possible.

- (a) .Finite impulse response (FIR) filter of order 83 designed using the Parks-McClellan algorithm and implemented as a tapped delay line. *9 points.* **Problem implies that the Parks-McClellan algorithm has converged. After convergence, the Parks-McClellan algorithm always gives an FIR filter whose impulse response is even symmetric about the midpoint, which guarantees linear phase over all frequencies.**

Linear phase: In passband? **Yes, see above.**
 In stopband? **Yes, see above.**

BIBO stability: YES or NO. Why? **An FIR filter is always BIBO stable. Each output sample is a finite sum of weighted current and previous input samples. Each weight is finite in value, and each input sample is finite in value. A finite sum of finite values is bounded.**

Instruction cycles: **83+1+28 = 112 cycles, by means of Appendix N in course reader.**

- (b) Infinite impulse response (IIR) filter of order 4 designed using the elliptic algorithm and implemented as cascade of biquads. One pole pair has radius 0.992 and quality factor 3.9. The other has radius 0.9785 and quality factor of 0.8. *9 points.* **Homework problem 3.3 concerned the design of IIR filters using the elliptic design algorithm. The solution to homework problem 3.3 plotted magnitude and phase response of an elliptic design.**

Linear phase: In passband? **Approximate linear phase over some of passband**
 In stopband? **Approximate linear phase over some of stopband**

BIBO stability: YES or NO. Why? **Yes, the poles are inside the unit circle. The quality factors are quite low; hence, the poles are unlikely to become BIBO unstable when implemented.**

Instruction cycles: **2 (5 + 28) = 66 cycles by means of Appendix N in course reader. (An efficient implementation could overlap the implementation for biquad #2 after data reads are finished for biquad #1, which would save 11 instruction cycles.)**

(c) IIR filter with 2 poles and 62 zeros. Poles were manually placed to correspond to 10 Hz and have radii of 0.95. Zeros were designed using the Parks-McClellan algorithm with the above specifications. Implementation is a tapped delay line followed by all-pole biquad. *12 points.*

Linear phase: In passband? **Approximate linear phase over some of passband**
 In stopband? **Approximate linear phase over some of stopband**

BIBO stability: YES or NO. Why? **Yes, the poles are inside the unit circle. The quality factor is quite low; hence, the poles are unlikely to become BIBO unstable when implemented.**

Instruction cycles: **(62+1+28) + (2 + 28) = 121 cycles by means of Appendix N in course reader. (An efficient implementation can remove the second 28 cycles of overhead to give a total of 93 cycles.)**

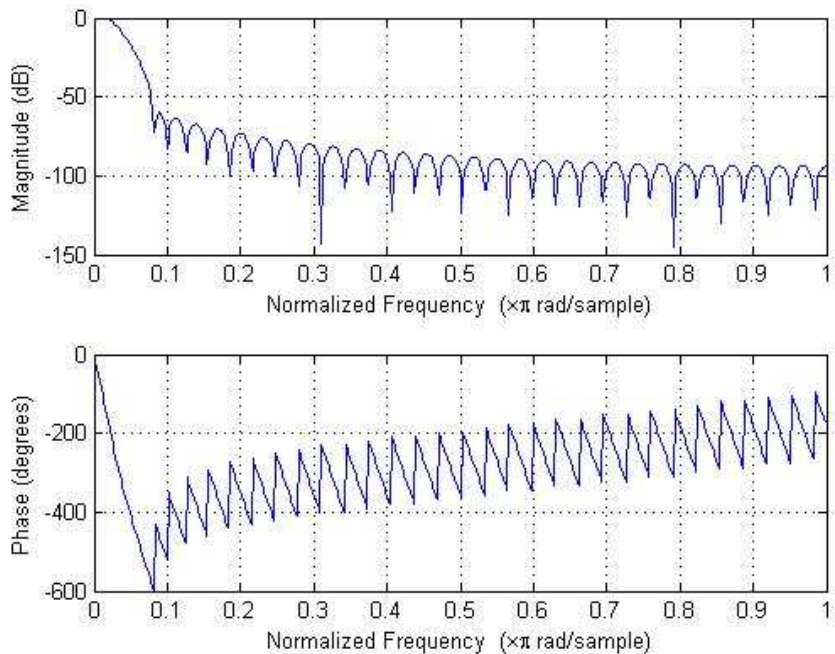
Pole locations: **Angle of first pole: $\omega_0 = 2 \pi f_0 / f_s = 2 \pi (10 \text{ Hz}) / (1000 \text{ Hz})$. Pole locations are at $0.95 \exp(j \omega_0)$ and $0.95 \exp(-j \omega_0)$.**

Matlab code to design the filter for part (c), which was not required for the test:

numerCoeffs = firpm(62, [0 .02 0.08 1], [1 1 0 0]) / 165;

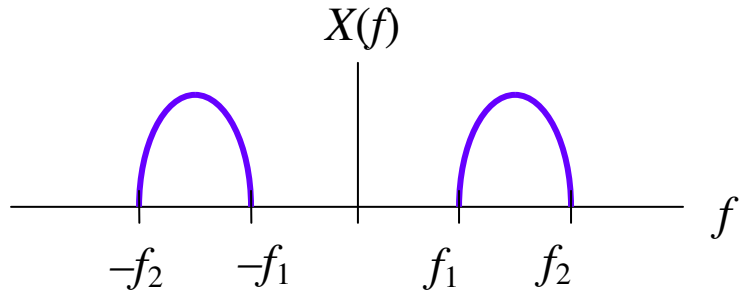
denomCoeffs = conv([1 -0.95*exp(j*2*pi*10/1000)], [1 -0.95*exp(-j*2*pi*10/1000)]);

freqz(numerCoeffs, denomCoeffs)



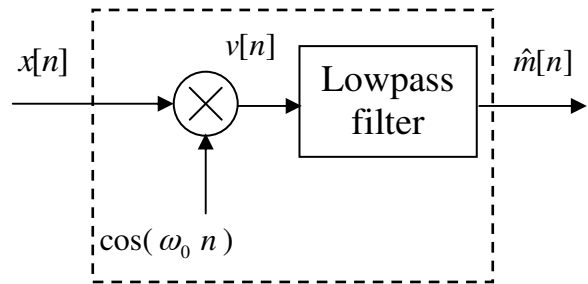
Problem 1.3 Downconversion. 24 points.

Consider the bandpass continuous-time analog signal $x(t)$. Its spectrum is shown on the right. The signal $x(t)$ was formed through upconversion. Our goal will be to recover the baseband message signal $m(t)$ by processing $x(t)$ in discrete time.



Let B be the bandpass bandwidth in Hz of $x(t)$ given by $B = f_2 - f_1$
 f_c be the carrier frequency in Hz given by $f_c = \frac{1}{2}(f_1 + f_2)$ where $f_c > 2B$.
 f_s be the sampling rate in Hz for sampling $x(t)$ to produce $x[n]$
 ω_{pass} be the passband frequency of a discrete-time filter in rad/sample
 ω_{stop} be the stopband frequency of a discrete-time filter in rad/sample

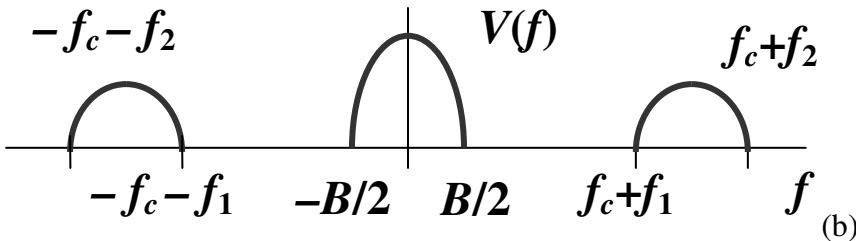
(a) Downconversion method #1. 12 points,
 Uses sinusoidal amplitude demodulation.
 Give formulas for ω_0 , ω_{pass} , ω_{stop} and f_s .



$$\omega_0 = 2\pi \frac{f_c}{f_s} \quad f_s > 2(2f_c + B/2)$$

$$\omega_{\text{pass}} = 2\pi \frac{B/2}{f_s} \quad \omega_{\text{stop}} = 2\pi \frac{f_c + f_1}{f_s}$$

Analyze in continuous-time first.

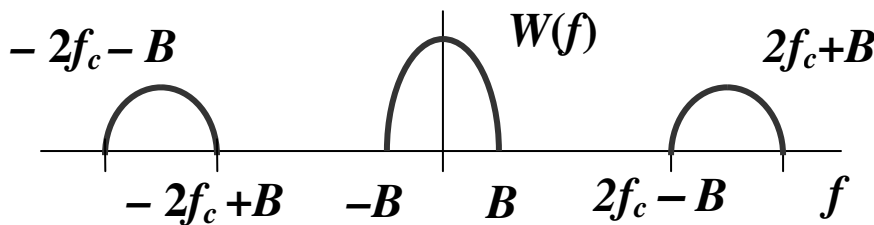
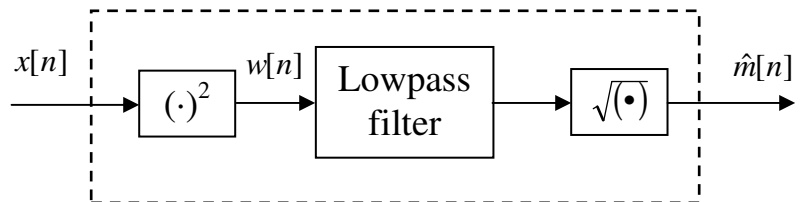


(c) Downconversion method #2. 12 points.

Uses a squaring device. Assume that output values of the lowpass filter are non-negative.
 Give formulas for ω_{pass} , ω_{stop} and f_s .

Analyze in continuous time first.

$$W(f) = X(f) * X(f)$$



$$\omega_{\text{pass}} = 2\pi \frac{B}{f_s}$$

$$\omega_{\text{stop}} = 2\pi \frac{2f_c - B}{f_s}$$

$$f_s > 2(2f_c + B)$$

Problem 1.4. Potpourri. 18 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) Consider an infinite impulse response (IIR) filter with four complex-valued poles (occurring in conjugate symmetric pairs) and no zeros. When implemented in handwritten assembly on the C6713 digital signal processor using only single-precision floating-point data format and arithmetic, a cascade of four first-order IIR sections would be more efficient in computation than implementing the filter as a cascade of two second-order IIR sections. *9 points*.

False. Assume input $x[n]$ is real-valued. Poles located at p_1, p_2, p_3 and p_4 . Outputs for the four first-order sections follow:

$$y_1[n] = x[n] + p_1 y_1[n-1]$$

$$y_2[n] = y_1[n] + p_2 y_2[n-1]$$

$$y_3[n] = y_2[n] + p_3 y_3[n-1]$$

$$y_4[n] = y_3[n] + p_4 y_4[n-1]$$

For the cascade of first-order sections, the final output value can be calculated as

$$y_4[n] = x[n] + p_1 y_1[n-1] + p_2 y_2[n-1] + p_3 y_3[n-1] + p_4 y_4[n-1]$$

Cascade of biquads has real-valued feedback coefficients. Its final output value is

$$v_2[n] = x[n] + b_1 v_1[n-1] + b_2 v_1[n-2] + b_3 v_2[n-1] + b_4 v_2[n-2]$$

Case #1: All poles are real-valued. Cascade of first-order sections requires the same execution time as a tapped delay line with five coefficients, or 33 instruction cycles, according to Appendix N in course reader. Same goes for the cascade of biquads.

Case #2: Poles are complex-valued and occur in conjugate symmetric pairs. Cascade of biquads still takes 33 instruction cycles. For cascade of first-order sections, the first section output $x[n] + p_1 y_1[n-1]$ is complex-valued. In subsequent sections, the complex-valued multiply-add operation will require four times the number of real-valued multiply-add operations. Cascade of biquads will hence require fewer cycles.

- (b) Consider implementing an infinite impulse response (IIR) filter solely in single-precision floating-point data format and arithmetic. There are no conditions under which the implemented filter would be linear and time-invariant. *9 points*.

False. Counterexample: $y[n] = x[n] + y[n-1]$ where $y[-1] = 0$ and $x[n] = \delta[n]$. The system passes the all-zero test. The system is linear and time-invariant only for a limited set of input signals.

True. Although a necessary condition for linear and time-invariance is that the initial conditions are zero, exact precision calculations in IIR filters require increasing precision as n increases in the worst case. (This is mentioned in lecture 6 slides when the block diagram for each of the three IIR direct form structures was discussed). Eventually, the increase in precision will exceed the precision of the single-precision floating-point data format. The clipping that results will cause linearity to be lost.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 15, 2010

Course: EE 445S Evans

Name: Set, Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Please disable all wireless connections on your computer system(s).**
- Please turn off all cell phones and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Digital Filter Analysis
2	30		Sinusoidal Generation
3	24		Upconversion
4	18		Potpourri
<i>Total</i>	100		

Problem 1.1 Digital Filter Analysis. 28 points.

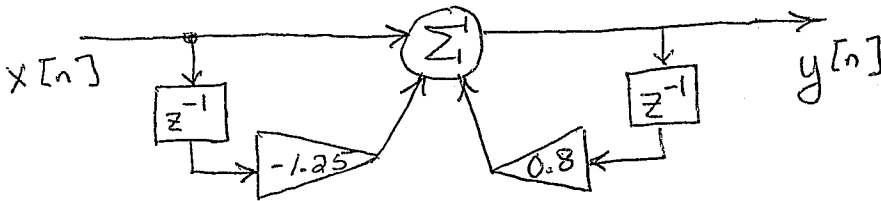
A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following difference equation:

$$y[n] - 0.8 y[n-1] = x[n] - 1.25 x[n-1]$$

- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 2 points.

Infinite impulse response filter.
Output $y[n]$ depends on previous output $y[n-1]$.

- (b) Draw the block diagram for this filter. 4 points.



- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

$$y[0] = 0.8 y[-1] + x[0] - 1.25 x[-1]$$

Initial conditions are $x[-1]$ and $y[-1]$, which must be zero to satisfy system properties of LTI and causality.

- (d) Find the equation for the transfer function of the filter in the z-domain including the region of convergence. 5 points.

$$\begin{aligned} Y(z) - 0.8 z^{-1} Y(z) &= X(z) - 1.25 z^{-1} X(z) && \text{ROC} \\ Y(z) (1 - 0.8 z^{-1}) &= X(z) (1 - 1.25 z^{-1}) && |z| > 0.8 \\ H(z) = \frac{Y(z)}{X(z)} &= \frac{1 - 1.25 z^{-1}}{1 - 0.8 z^{-1}} && \begin{array}{l} \text{zero at } z = 1.25 \\ \text{pole at } z = 0.8 \end{array} \end{aligned}$$

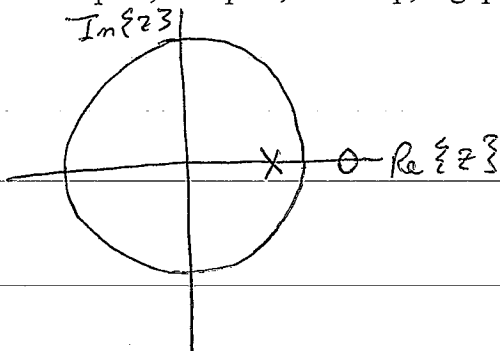
for causality.

- (e) Find the equation for the frequency response of the filter. 5 points.

The system is BIBO stable because the region of convergence includes the unit circle.

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 - 1.25 e^{-j\omega}}{1 - 0.8 e^{-j\omega}}$$

- (f) Draw the pole-zero diagram. Would the frequency selectivity of the filter be best described as lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 8 points.



$$\text{pole radius} = \frac{1}{\text{zero radius}}$$

This is an all-pass filter according to slides 6-8 and 6-20 as well as Appendix O in reader.

Problem 1.2 Sinusoidal Generation. 30 points.

Consider generating a causal discrete-time cosine waveform $y[n]$ that has a fixed frequency of $\omega_0 = 2\pi f_0 / f_s$, where f_0 is the continuous-time sinusoidal frequency and f_s is the sampling rate:

$$y[n] = \cos(\omega_0 n) u[n]$$

(a) What value of f_s must take to prevent aliasing? 6 points.

$f_s > 2 f_{max}$ where f_{max} is the maximum frequency of $\cos(2\pi f_0 t) u(t)$.

(b) We'll evaluate design tradeoffs on the C6000 family of digital signal processors. Assume that the most efficient assembly language implementation is used in all cases. 18 points.

f_{max} is slightly greater than f_0 .

Data Size	Operation	Throughput	Delay Slots
16-bit short	addition	1 cycle	0 cycles
16-bit by 16-bit short	multiplication	1 cycle	1 cycle
32-bit floating-point	addition	1 cycle	3 cycles
32-bit x 32-bit floating-point	multiplication	1 cycle	3 cycles
64-bit floating-point	addition	2 cycles	6 cycles
64-bit x 64-bit floating-point	multiplication	4 cycles	9 cycles

From slide 2-12, Horner's form would take $21 \cdot 10 = 210$ cycles for an 11th order polynomial calculation because we have to wait for each $(a_{11}x + a_{10})$ term to complete.

Please complete the following table:

Method	Memory for data and coefficients (in bytes)	Multiplication-add operations per output sample	C6000 cycles to finish multiplication-add operations to compute one output sample
C math library call Assuming GNU Scientific Library version 1.8	(23 words x 8 bytes/word) 184 bytes	30	$21 \cdot 10 + 19(2+4) + 15 = 339$ cycles
Difference equation using 32-bit floating-point data/arithmetic	(5 words x 4 bytes/word) 20 bytes	2	$2 \cdot 2 + 6 = 10$ cycles
Difference equation using arithmetic on 16-bit (short) data and coefficients	(5 words x 2 bytes/word) 10 bytes	2	$2 \cdot 1 + 2 = 4$ cycles

There are 10 terms. Assume remaining 19 MACs can be pipelined in C math library call.

← From slide 2-16 →

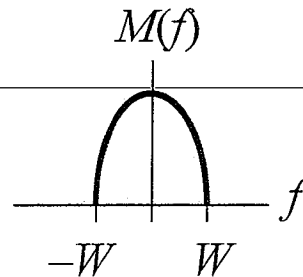
(c) Which method would you advocate using? Why? 6 points.

On the C6000, I would use a difference equation, which is 34x or 85x more efficient than a C math library call for the floating-point (32-bit) or fixed-point (16-bit) version, respectively.

In 16-bit format, one could compute two multiplies in parallel because there are two multiply units.

Problem 1.3 Upconversion. 24 points.

Consider a baseband continuous-time analog signal $m(t)$. Its spectrum is shown on the right. Our goal is to upconvert $m(t)$ into a bandpass signal $s(t)$. Our upconversion will be implemented in discrete time.



Let W : baseband bandwidth in Hz of $m(t)$

B : transmission bandwidth of $s(t)$

f_c : carrier frequency in Hz of $s(t)$ where $f_c > 3W$

f_s : sampling rate in Hz for sampling of $m(t)$ to give $m[n]$ and $s(t)$ to give $s[n]$

ω_{pass} : passband freq. of discrete-time filter in rad/sample

ω_{stop} : stopband freq. of discrete-time filter in rad/sample

(a) Upconversion method #1. 14 points,

Uses sinusoidal amplitude modulation.

Give formulas for the following parameters:

$$\omega_0 = 2\pi \frac{f_c}{f_s}$$

$$B = 2W$$

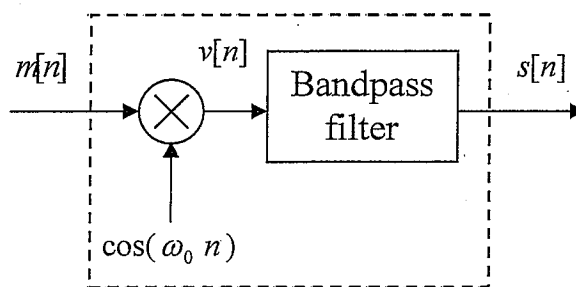
$$\omega_{stop1} = 0.9 \omega_{pass1}$$

$$\omega_{pass1} = 2\pi \frac{f_c - W}{f_s}$$

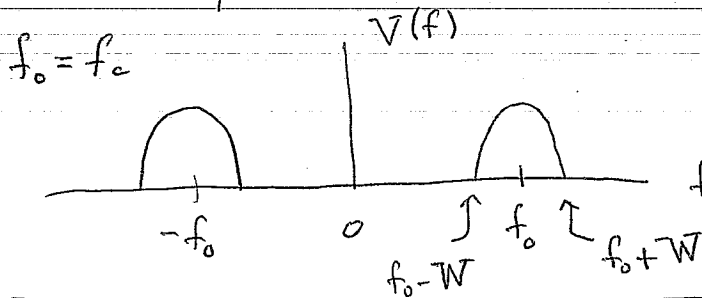
$$\omega_{pass2} = 2\pi \frac{f_c + W}{f_s}$$

$$\omega_{stop2} = 1.1 \omega_{pass2}$$

$$f_s > 2(1.1(f_c + W))$$



Work the problem in continuous-time first:



Software Receiver Design Exercise 5.10

(b) Upconversion method #2. 10 points.

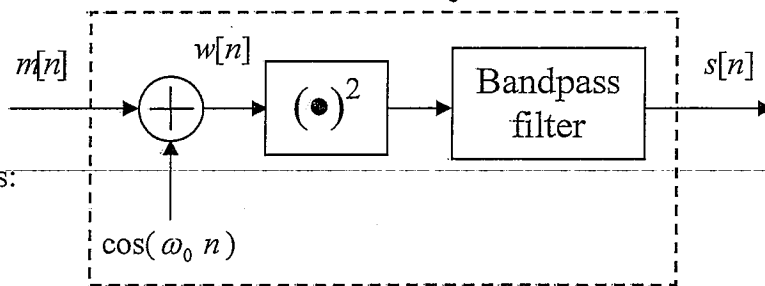
Uses a squaring device and the bandpass filter from part (a).

Give formulas for the following parameters:

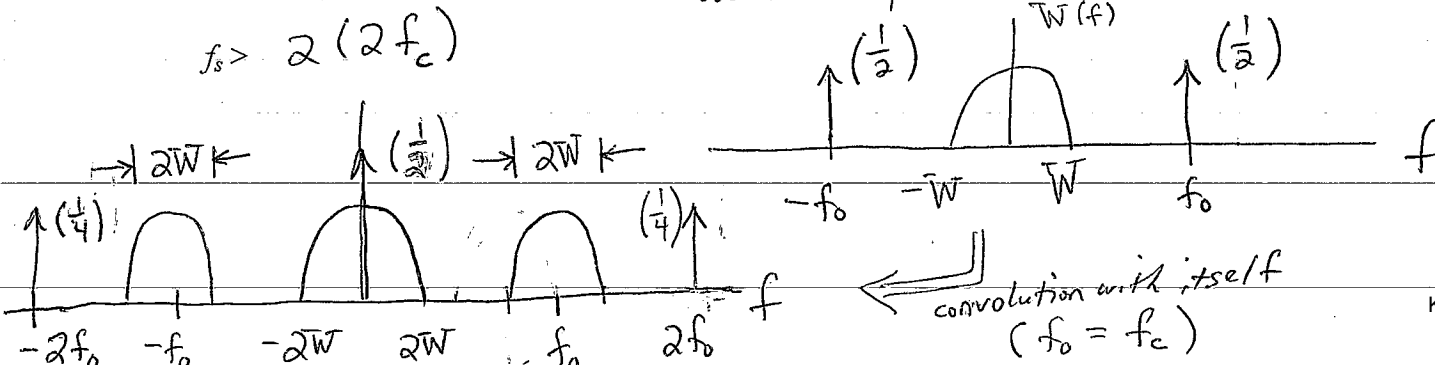
$$\omega_0 = 2\pi \frac{f_c}{f_s}$$

$$B = 2W$$

$$f_s > 2(2f_c)$$



Work the problem in continuous-time first:

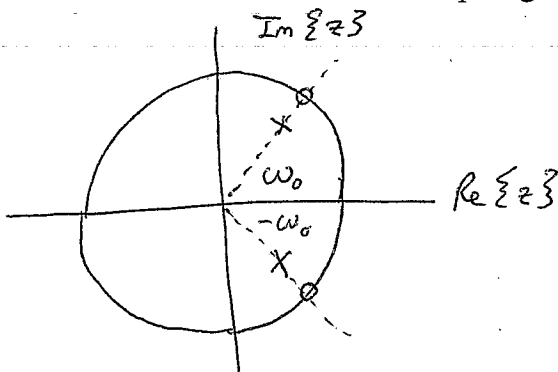


Problem 1.4. Potpourri. 18 points.

- (a) For a system design, you have determined that you need to design a linear phase discrete-time finite impulse response filter to meet piecewise magnitude constraints. The Parks-McClellan design algorithm fails to converge. What filter design method would you use and why? 9 points.

Use the Kaiser window method. It gives linear phase discrete-time FIR filter designs of shorter length than the FIR Least Squares design method.

- (b) For a system design, you have determined that you need a discrete-time biquad notch filter to remove a narrowband interferer at discrete-time frequency ω_0 . The actual discrete-time frequency will vary over time when the system is deployed in the field. Give the poles and zeros for the notch filter. Set the biquad gain to 1. 9 points.



Two poles:

$$p_0 = 0.9 e^{j\omega_0}$$

$$p_1 = 0.9 e^{-j\omega_0}$$

Two zeros:

$$z_0 = e^{j\omega_0}$$

$$z_1 = e^{-j\omega_0}$$

Biquad gain:

$$c = 1$$

$$H(z) = c \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 11, 2011

Course: EE 445S Evans

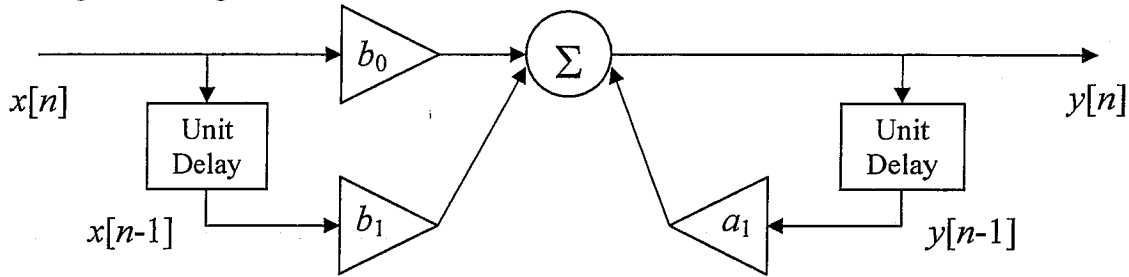
Name: Set, Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Please disable all wireless connections on your computer system(s).**
- Please turn off all cell phones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Discrete-Time Filter Analysis
2	18		Discrete-Time IIR Filtering
3	30		Analysis and Synthesis
4	24		Downconversion
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 28 points.

A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is described by the following block diagram:



- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 2 points.

Infinite impulse response filter due to feedback - the current output depends on the previous output

- (b) Give the difference equation for the filter. 4 points.

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

Initial conditions are the initial values in the unit delay blocks, i.e., the initial values of $x[n-1]$ and $y[n-1]$ when $n=0$.

Values must be zero to satisfy linear, time-invariant and causal properties.

- (d) Find the equation for the transfer function of the filter in the z-domain including the region of convergence. 5 points.

$$Y(z) = a_1 z^{-1} Y(z) + b_0 X(z) + b_1 z^{-1} X(z)$$

$$Y(z) - a_1 z^{-1} Y(z) = b_0 X(z) + b_1 z^{-1} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} ; \text{ROC is } |z| > |a_1|$$

- (e) Find the equation for the frequency response of the filter. 5 points.

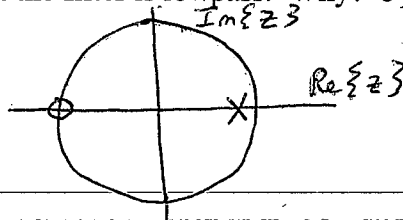
Since $|a_1| < 1$ because this is a filter, the region of convergence includes the unit circle:

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{b_0 + b_1 e^{-j\omega}}{1 - a_1 e^{-j\omega}}$$

- (f) Give values for b_0 , b_1 and a_1 so that the filter is lowpass. Why? 8 points.

Pole is at $z = a_1$

Zero is at $z = -\frac{b_1}{b_0}$



$$a_1 = 0.9$$

$$b_0 = 1$$

$$b_1 = 1$$

Problem 1.2 Discrete-Time IIR Filtering. 18 points.

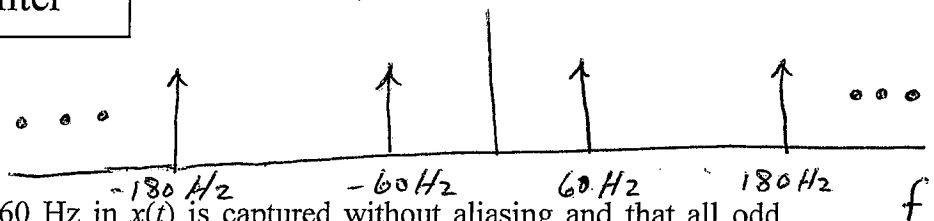
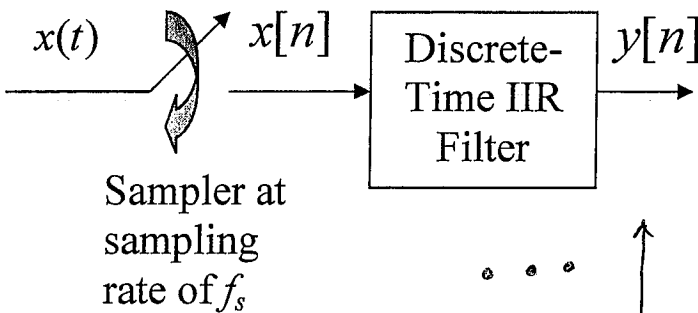
In the US, wall power is at a main frequency of 60 Hz.

Assume that odd harmonics (180 Hz, 300 Hz, etc.) are also present.

Assume that even harmonics (120 Hz, 240 Hz, etc.) are not present.

Design the following signal processing system to remove 60 Hz and its odd harmonics.

Intent is of this problem is to eliminate narrowband noise induced by power line main frequency and its odd harmonics. This is a common problem.



- (a) Pick a sampling rate f_s so that 60 Hz in $x(t)$ is captured without aliasing and that all odd harmonics of 60 Hz in $x(t)$ alias to 60 Hz. Justify your answer. 9 points.

Two constraints:

- $f_s > 2 f_{max}$ $f_s = 120 \text{ Hz}$

From homework #0, we know that a causal sinusoid has a small bandwidth. f_{max} is slightly higher than 60 Hz.

- Either $-180 + f_s = 60$ or $180 - f_s = 60$

This gives $f_s = 240 \text{ Hz}$ or $f_s = 120 \text{ Hz}$

We rule out $f_s = 120 \text{ Hz}$ because it violates first constraint

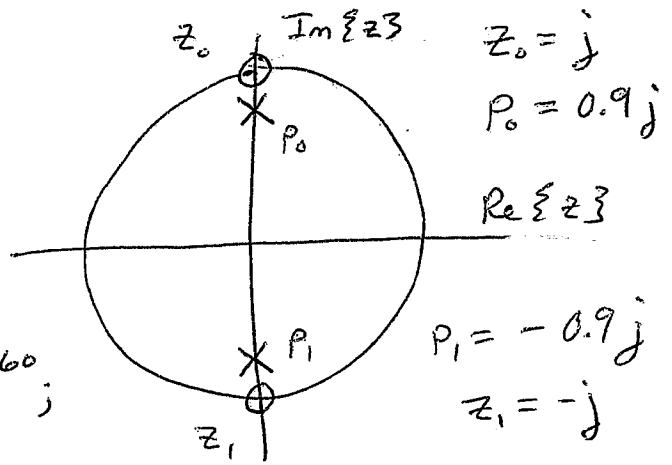
- (b) Using the sampling rate in (a), give the poles, zeros and gain of the discrete-time IIR biquad (filter) in the block diagram above to remove 60 Hz and hence all odd harmonics of 60 Hz. 9 points.

$$\omega_0 = 2\pi \frac{f_0}{f_s}$$

$$\omega_{60} = 2\pi \frac{60 \text{ Hz}}{240 \text{ Hz}} = \frac{\pi}{2}$$

$$p_0 = 0.9 e^{j\omega_{60}}; \quad p_1 = 0.9 e^{-j\omega_{60}};$$

$$z_0 = e^{j\omega_{60}}; \quad z_1 = e^{-j\omega_{60}}$$



$$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}; \quad \text{Set } H(1) = 1 \text{ and solve for } C$$

Zeros on or near unit circle indicate stopband(s)

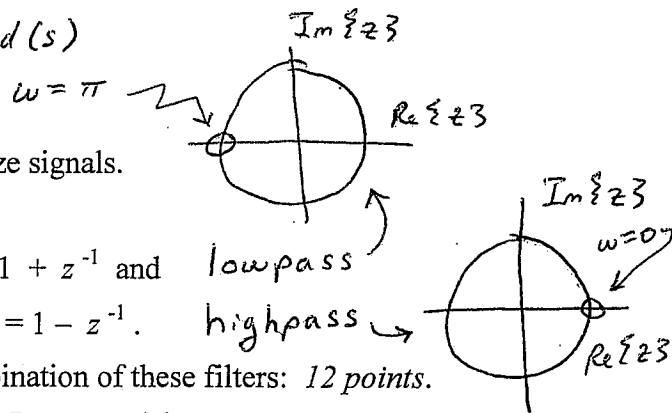
Problem 1.3. Analysis and Synthesis. 30 points.

One of the common uses of filters is to analyze and synthesize signals.

For analysis, we will be using the two following two filters:

a two-tap averaging filter with a transfer function of $H_1(z) = 1 + z^{-1}$ and

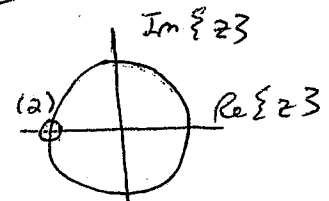
a first-order difference filter with a transfer function of $H_2(z) = 1 - z^{-1}$.



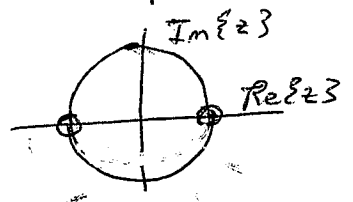
(a) What is the frequency selectivity of the following combination of these filters: 12 points.

i. $H_1(z) + H_2(z) = (1 + z^{-1}) + (1 - z^{-1}) = 2$ allpass

ii. $H_1(z) H_1(z) = (1 + z^{-1})(1 + z^{-1})$ lowpass
 $= 1 + 2z^{-1} + z^{-2}$



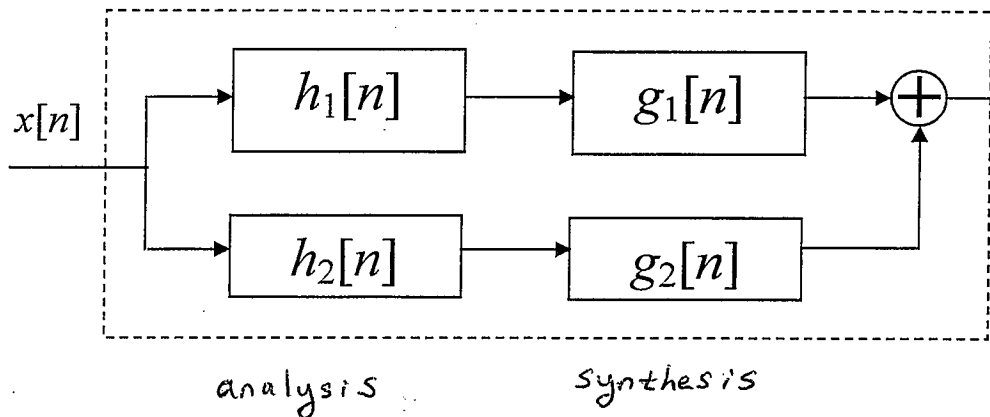
iii. $H_1(z) H_2(z) = (1 + z^{-1})(1 - z^{-1})$ bandpass
 $= 1 - z^{-2}$



iv. $H_2(z) H_2(z) = (1 - z^{-1})(1 - z^{-1})$ highpass
 $= 1 - 2z^{-1} + z^{-2}$

Two zeros at

(b) In the block diagram below, assume that $x[n]$ has 16 bits/sample. Design filters $G_1(z)$ and $G_2(z)$ to meet the following constraints: (i) filters $G_1(z)$ and $G_2(z)$ will be implemented in 32-bit floating point data and arithmetic; (ii) the input-output relationship between $x[n]$ and $y[n]$ is LTI; and (iii) the filtering of $x[n]$ to give $y[n]$ gives an all-pass response. Assume that $y[n]$ will be represented in a 32-bit floating-point format. 18 points.



Solution #2

$G_1(z) = G_2(z) = G(z)$
 and $G(z)$ is FIR and all-pass.
 Works because $H_1(z) + H_2(z)$ is all-pass.

$$Y(z) = (H_1(z)G_1(z) + H_2(z)G_2(z))X(z)$$

$$H(z) = H_1(z)G_1(z) + H_2(z)G_2(z)$$

In order for $G_1(z)$ and $G_2(z)$ to give LTI system, after implementation, they must be FIR filters. IIR filters need ∞ precision to satisfy LTI properties.

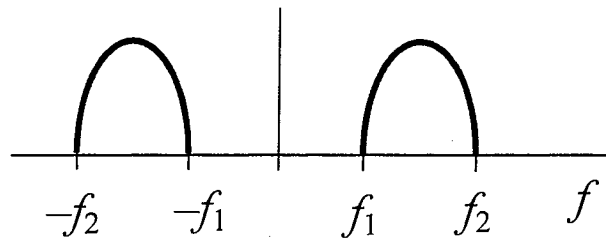
Solution #1

$$G_1(z) = -1 - z^{-1} = -H_1(z) \Rightarrow H(z) = -4z^{-1}$$

$$G_2(z) = 1 - z^{-1} = H_2(z)$$

Problem 1.4 Downconversion. 24 points.

Consider an upconverted continuous-time analog signal $s(t) = m(t) \cos(2\pi f_c t)$. Its spectrum is on the right. Our goal is to downconvert $s(t)$ into a baseband signal $m(t)$. Downconversion will be implemented in discrete time.



Let W : baseband bandwidth in Hz of $m(t)$

B : transmission bandwidth of $s(t)$ where $B = f_2 - f_1$

f_c : carrier frequency in Hz of $s(t)$ where $f_c = \frac{1}{2}(f_1 + f_2)$ and $f_c > 3W$

f_s : sampling rate in Hz for sampling of $m(t)$ to give $m[n]$ and $s(t)$ to give $s[n]$

ω_{pass} : passband frequency of discrete-time filter in rad/sample

ω_{stop} : stopband frequency of discrete-time filter in rad/sample

$$B = 2W$$

$$f_c > 9W$$

(a) Downconversion method #1. 12 points,

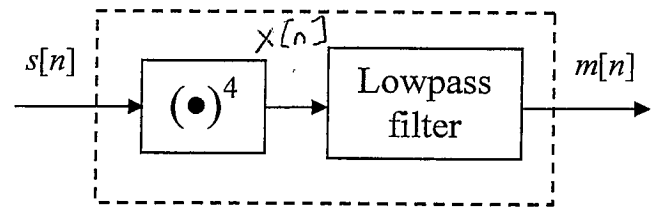
Uses a fourth-order static nonlinearity.

Give formulas for the following parameters:

$$\omega_{\text{pass}} = 2\pi \frac{2B}{f_s}$$

$$\omega_{\text{stop}} = 2\pi \frac{2f_c - 2B}{f_s}$$

$$f_s > 2(4f_c + 2B)$$



$$X(t) = s^4(t)$$

$$X(f) = (S(f) * S(f)) * (S(f) * S(f))$$

See below.

(b) Downconversion method #2. 12 points.

Uses an absolute value static nonlinearity.

The following Fourier series truncated to two terms might be helpful

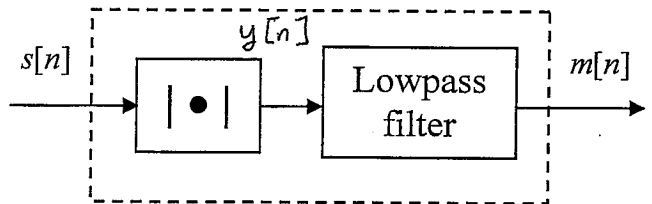
$|\cos(2\pi f_c t)| \approx a_0 + a_2 \cos(4\pi f_c t) \rightarrow$ Similar to squaring device, w/o carrier.
where a_0 and a_2 are real constants.

Give formulas for the following parameters:

$$\omega_{\text{pass}} = 2\pi \frac{B}{f_s}$$

$$\omega_{\text{stop}} = 2\pi \frac{2f_c - B}{f_s}$$

$$f_s > 2(2f_c + B)$$

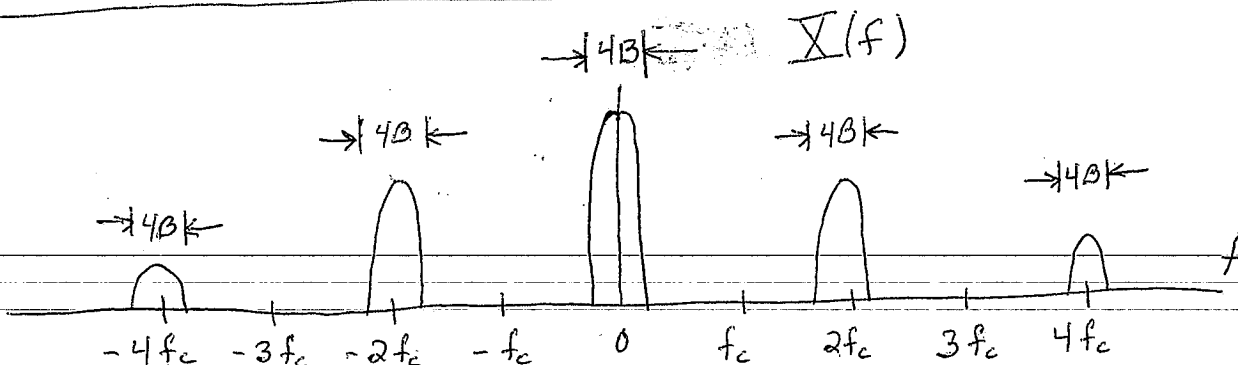


$$y(t) = |s(t)|$$

$$y(t) = |m(t) \cos(2\pi f_c t)|$$

$$y(t) = |m(t)| \cdot |\cos(2\pi f_c t)|$$

$$y(t) \approx |m(t)| \cdot (a_0 + a_2 \cos(4\pi f_c t))$$



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 14, 2011

Course: EE 445S Evans

Name: Set, Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. *Please disable all wireless connections on your computer system(s).*
- Please turn off all cell phones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Discrete-Time Filter Analysis
2	27		Need for Speed
3	27		I'd Like to Buy a Vowel
4	21		Potpourri
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 25 points.

A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following difference equation:

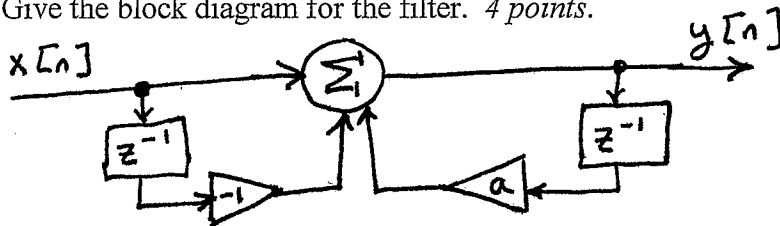
$$y[n] - a y[n-1] = x[n] - x[n-1]$$

where $0 < |a| < 0.9$.

- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 2 points.

Infinite impulse response filter due to feedback term $y[n-1]$.

- (b) Give the block diagram for the filter. 4 points.



- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

Let $n=0$: $y[0] = x[0] - x[-1] + a y[-1]$.

Initial conditions are $x[-1]$ and $y[-1]$. They must be set to zero to ensure system properties of linearity,

- (d) Find the equation for the transfer function of the filter in the z-domain including the region of convergence. 5 points.

Take the z-transform of both sides of the difference equation:

$$Y(z) - a z^{-1} Y(z) = X(z) - z^{-1} X(z) \Rightarrow \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{1 - a z^{-1}} = H(z)$$

Region of convergence is $|z| > |a|$ for causality.

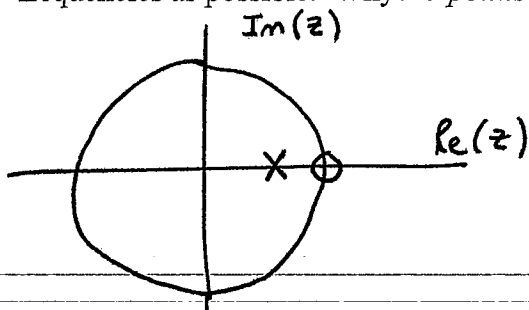
time-invariance and causality.

- (e) Find the equation for the frequency response of the filter. Justify your approach. 5 points.

Because $0 < |a| < 0.9$, the region of convergence $|z| > |a|$ includes the unit circle:

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 - e^{-j\omega}}{1 - a e^{-j\omega}}$$

- (f) Give a value for a so that the filter removes zero frequency and passes as many other frequencies as possible. Why? 5 points.



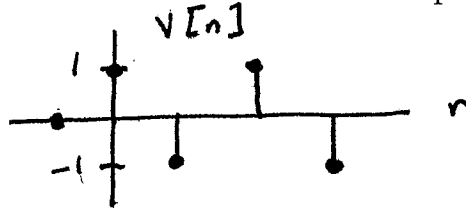
Regardless of the value of a where $0 < |a| < 0.9$, the filter has a zero at $z=1$, i.e. $\omega=0$. Placing the pole at $z=0.899$ would give a notch filter at DC.

Problem 1.2 Need for Speed. 27 points.

Consider the signal $v[n] = (-1)^n u[n]$ where $u[n]$ is the unit step function.

- (a) We can write $v[n]$ in the form $\cos(\omega_0 n) u[n]$. Give a value for the discrete-time frequency ω_0 in rad/sample. 6 points.

$\omega_0 = \pi$ rad/sample

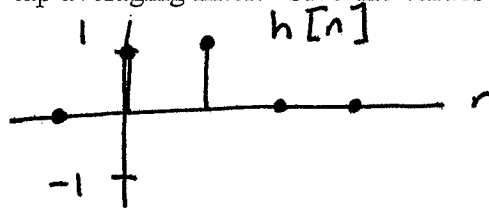


- (b) Let $h[n]$ be the impulse response of a lowpass filter, and $g[n]$ is an impulse response formed by $v[n] h[n]$.

- i. Let $h[n]$ be the impulse response of a two-tap averaging filter. Give the values of $g[n]$. 3 points.

$h[n] = \delta[n] + \delta[n-1]$

$g[n] = v[n] h[n]$
 $= \delta[n] - \delta[n-1]$

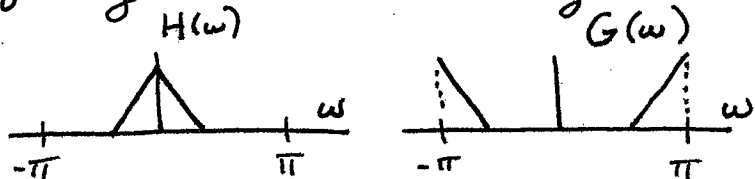


- ii. If $g[n]$ from part i were an impulse response of a linear time-invariant filter, what would its frequency selectivity be? Lowpass, highpass, bandpass, bandstop, allpass, or notch? 3 points.

Highpass. Impulse response $g[n] = \delta[n] - \delta[n-1]$ is impulse response for first-order difference filter, which has a highpass frequency response.

- iii. For a general lowpass $h[n]$, show that your answer in part ii holds in general. You may show this by using formulas, or by drawing pictures in the frequency domain. 6 points.

Multiplication of $h[n]$ by $v[n]$ causes shift in discrete-time frequency domain of $H(\omega)$ by π to the left and π to the right:



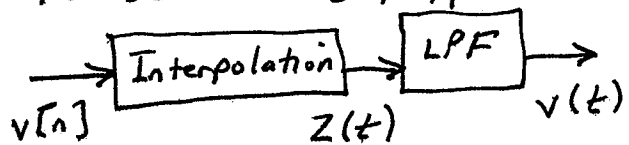
- (c) What is the resulting continuous-time analog signal that results when passing $v[n]$ through a digital-to-analog converter with sampling rate f_s ? 9 points.

For ideal D-to-A conversion:
 $\omega_0 = 2\pi \frac{f_0}{f_s} = \pi \Rightarrow f_0 = \frac{1}{2} f_s$

$v(t) = T_s \cos(\pi f_s t) u(t)$

Sampling $v(t)$ at $t = n T_s$ gives $v[n]$.

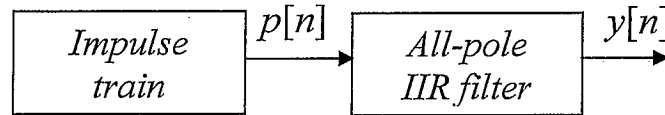
For standard D-to-A:



Depends on interpolation method and lowpass filter.

Problem 1.3. I'd Like to Buy a Vowel. 27 points.

The following discrete-time block diagram synthesizes a short segment of speech $y[n]$ for a vowel sound:

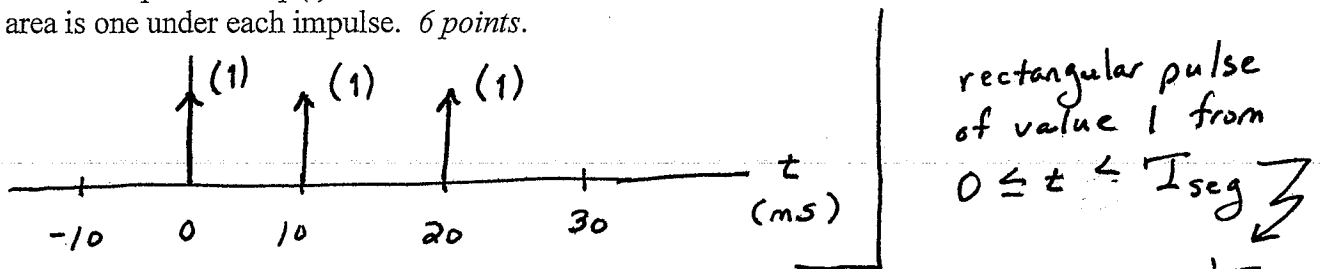


Assume the following:

- segment of speech lasts from 0 ms to 25 ms, inclusive. $T_{seg} = 25\text{ms}$
- impulses in the impulse train are separated by the speaker's pitch period
- pitch period is 1/(100 Hz) or 10 ms $T_p = 10\text{ms}$
- ~~sampling~~ rate f_s is 8000 Hz $T_s = \frac{1}{8000\text{Hz}} = 0.125\text{ms}$

sampling

(a) Plot the impulse train $p(t)$ in continuous time over the interval -15 ms to 35 ms. Assume that the area is one under each impulse. 6 points.



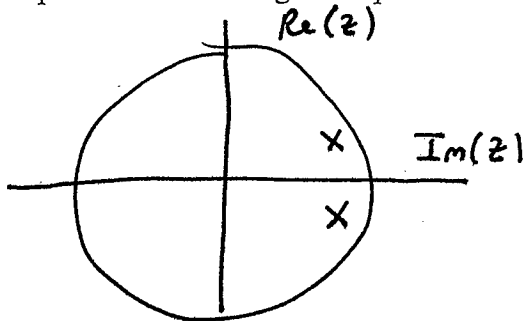
(b) Sketch the continuous-time Fourier transform for $p(t)$. 12 points.

$$p(t) = \delta(t) + \delta(t-10\text{ms}) + \delta(t-20\text{ms}) = \delta_{T_p}(t) \text{rect}\left(\frac{t - \frac{1}{2}T_{seg}}{T_{seg}}\right)$$

$$P(f) = \mathcal{F}\{\delta_{T_p}(t)\} * \mathcal{F}\left\{\text{rect}\left(\frac{t - \frac{1}{2}T_{seg}}{T_{seg}}\right)\right\}$$

Impulse train with impulses separated by T_p

(a) Design the first biquad for the all-pass IIR filter in discrete-time to pass 500 Hz. Give the pole locations and gain. 9 points.



$$\omega_{500} = 2\pi \frac{500\text{Hz}}{8000\text{Hz}} = \frac{\pi}{8}$$

Pole locations at

$$p_0 = 0.9 e^{j\omega_{500}}$$

$$p_1 = 0.9 e^{-j\omega_{500}}$$

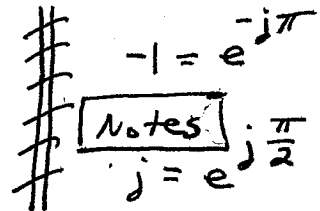
$$H(z) = C \frac{1}{(1-p_0 z^{-1})(1-p_1 z^{-1})}; \text{ Let } H(1) = 1 \text{ and solve for } C.$$

Problem 1.4 Potpourri. 21 points.

(a) The first-order difference, discrete-time, linear time-invariant filter has impulse response $h[n] = \delta[n] - \delta[n-1]$. 9 points.

i. Give a formula for the frequency response

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = 1 - e^{-j\omega}$$



ii. Show that the phase response is linear. (Note: The phase response may not necessarily go through the origin.)

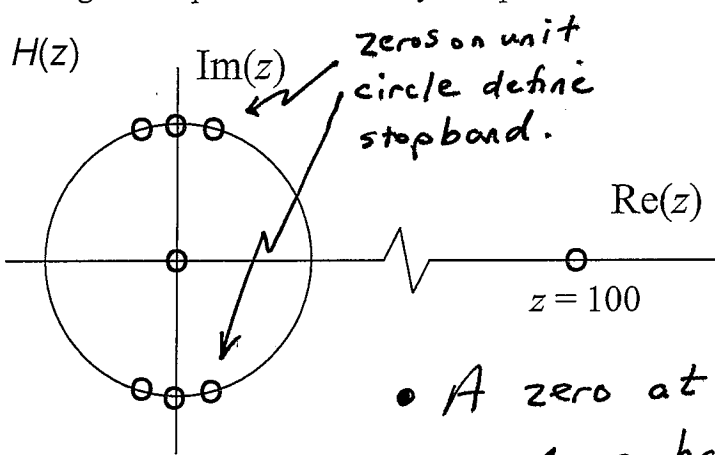
$$H_{\text{freq}}(\omega) = e^{-j\frac{\omega}{2}} \left(e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}} \right) = e^{-j\frac{\omega}{2}} \left(2j \sin \frac{\omega}{2} \right)$$

iii. Compute the group delay

$$D(\omega) = -\frac{d}{d\omega} \angle H_{\text{freq}}(\omega) = \frac{1}{2}$$

$$\angle H_{\text{freq}}(\omega) = \begin{cases} \frac{\pi}{2} - \frac{\omega}{2} & \text{for } \omega > 0 \\ -\frac{\pi}{2} - \frac{\omega}{2} & \text{for } \omega < 0 \end{cases}$$

(b) You've designed a discrete-time finite impulse response (FIR) filter to meet a magnitude specification by running a design program. Your FIR filter meets the original magnitude specifications with at least 0.2 dB to spare in all frequency bands of interest. The zeros are plotted below. What is the minimum order of the FIR filter that would meet the original magnitude specification? Why? 12 points.



- The initial FIR filter design has eight zeros and is an eighth-order filter.

- A zero at the origin in the z-plane has no effect on the magnitude response - it can be discarded.

- The zero at $z=100$ scales the magnitude response by a factor varying from 99 to 101. It can be replaced by a constant of 100. The maximum loss is by a factor of 1.01 in linear units, or 0.086 dB. This zero can be replaced by a constant and still keep filter within specification. The minimum order is 6.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 9, 2012

Course: EE 445S Evans

Name: _____

One,
Last,

Thing
First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Please disable all wireless connections on your computer system(s).**
- Please turn off all cell phones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Discrete-Time Filter Analysis
2	24		Music Therapy
3	27		Downconversion
4	24		Discrete-Time FIR Filter Design
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 25 points.

A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following difference equation:

$$y[n] = a y[n-1] + x[n] - b x[n-1]$$

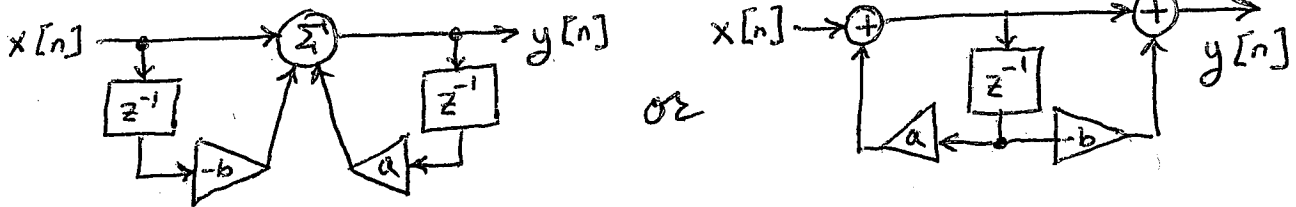
where $0 < |a| < 0.9$ and $|b| > 0.9$.

Note: A variation of this question appeared on the fall 2003 midterm #1.

- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 2 points.

IIR filter. The current value $y[n]$ depends on the previous output value $y[n-1]$.

- (b) Give the block diagram for the filter. 3 points.



- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

Let $n=0$. $y[0] = a y[-1] + x[0] - b x[-1]$. Initial conditions $y[-1]$ and $x[-1]$ should be zero to guarantee LTI properties.

- (d) Find the equation for the transfer function of the filter in the z-domain including the region of convergence. 5 points.

Take the z-transform of both sides:

$$Y(z) = a z^{-1} Y(z) + X(z) - b z^{-1} X(z)$$

$$(1 - a z^{-1}) Y(z) = (1 - b z^{-1}) X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - b z^{-1}}{1 - a z^{-1}}$$

Pole is at $z=a$:

$$1 - a z^{-1} = 0$$

$$a z^{-1} = 1$$

$$z = a$$

For a causal system,
 $|z| > |a|$

- (e) Find the equation for the frequency response of the filter. Justify your approach. 5 points.

Because the region of convergence includes the unit circle, i.e. $|z| > |a|$ and $|a| < 0.9$,

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 - b e^{-j\omega}}{1 - a e^{-j\omega}}$$

- (f) Assume that a and b are real-valued. Give the best values for a and b for the filter to have the following frequency selectivity:

$a = 0.89$, $b = -1$ 1. Lowpass. 3 points. When the pole at $z=a$ and zero at $z=b$ are separated in angle, pole indicates pass band and zero indicates stop band.

2. All-pass. 3 points. See Appendix O in reader.

$a = 0.8$

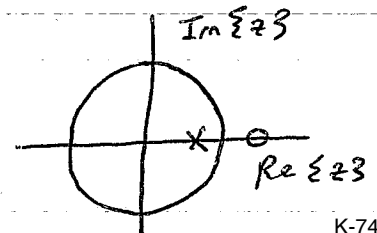
$b = 1.25$

For the real-valued case, $b = \frac{1}{a}$.

For $a = 0.8$, $b = 1.25$.

(See problem 1.1 on fall 2010 midterm #1.)

More generally, $|b| = \frac{1}{|a|}$ and $\angle b = \angle a$.



Problem 1.2 Music Therapy. 24 points.

People suffering from tinnitus, or ringing of the ears, hear a tone in their ears even when the environment is quiet. The tone is generally at a fixed frequency in Hz, denoted as f_0 .

A treatment for tinnitus is to listen to music in which the frequency f_0 has been removed.

Design the best discrete-time infinite impulse response biquad filter to remove frequency f_0 and pass all other frequencies as much as possible.

Assume that f_0 is 5512.5 Hz and the sampling rate f_s is 44100 Hz.

(a) Give the pole locations, zero locations, and gain. 18 points.

We want a notch filter to remove f_0 in Hz.

In discrete time, f_0 corresponds to $\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{5512.5 \text{ Hz}}{44100 \text{ Hz}}$

Pole locations:

$$p_0 = 0.9 e^{j\omega_0} \text{ and } p_1 = 0.9 e^{-j\omega_0}$$

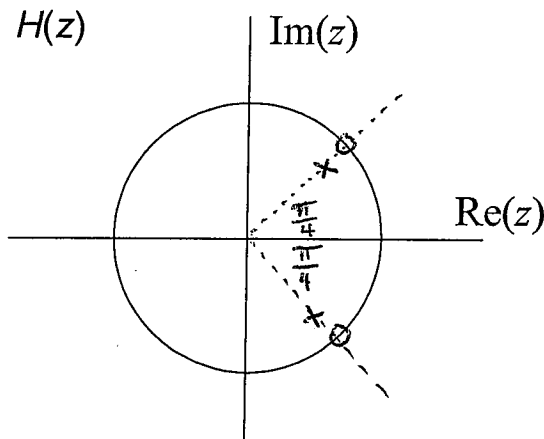
Zero locations:

$$z_0 = e^{j\omega_0} \text{ and } z_1 = e^{-j\omega_0}$$

$$H(z) = C \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)}$$

$$= \frac{\pi}{4}$$

(b) Draw the pole-zero diagram. 6 points.



Gain:

Simplest answer is $C = 1$.

Alternate answer is to set the DC gain to 1:

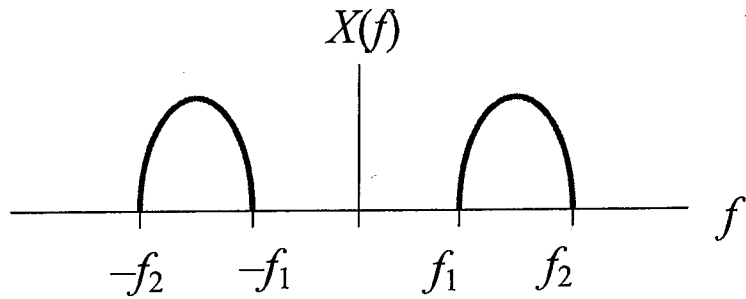
$$H(1) = 1$$

and solve for C .

Note: Tinnitus is a symptom of an underlying condition. The above treatment has been very successful in several published medical studies in relieving tinnitus. See "Music Therapy 'may help cut tinnitus noise levels'", Dec. 2009, BBC News: <http://news.bbc.co.uk/2/hi/health/8429715.stm>

Problem 1.3 Downconversion. 27 points.

Consider the bandpass continuous-time analog signal $x(t)$. Its spectrum is shown on the right. The signal $x(t)$ was formed through upconversion. Our goal will be to recover the baseband message signal $m(t)$ by processing $x(t)$ in discrete time.



Let B be the bandpass bandwidth in Hz of $x(t)$ given by $B = f_2 - f_1$
 f_c be the carrier frequency in Hz given by $f_c = \frac{1}{2}(f_1 + f_2)$ where $f_c > 2B$.
 f_s be the sampling rate in Hz for sampling $x(t)$ to produce $x[n]$
 ω_{pass} be the passband frequency of a discrete-time filter in rad/sample
 ω_{stop} be the stopband frequency of a discrete-time filter in rad/sample

Work out each part in continuous time first.

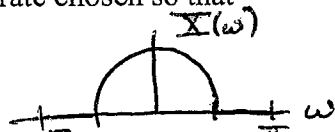
No anti-aliasing filter precedes the sampling device.

(a) Downconversion method #1. 12 points,

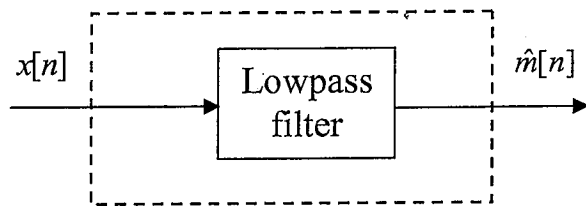
With the sampling rate chosen so that

$$f_s = 2B$$

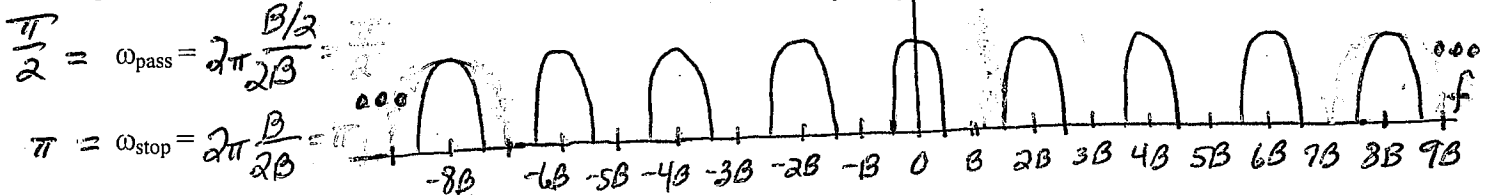
$$f_c = 4f_s = 8B$$



sketch the Fourier transform of $x[n]$ and give formulas for the following parameters:



Sampling $x(t)$ by f_s gives spectrum with replicas at offsets of $f_s = 2B$



$$\frac{\pi}{2} = \omega_{\text{pass}} = 2\pi \frac{B/2}{2B}$$

$$\pi = \omega_{\text{stop}} = 2\pi \frac{B}{2B}$$

(b) Downconversion method #2. 15 points.

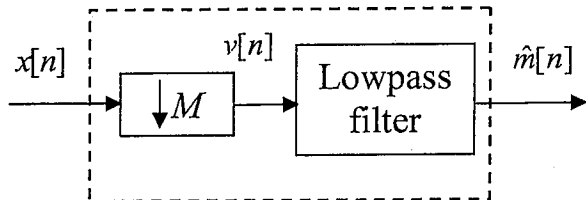
Downsampling factor M is an integer.

With

$$f_s > 2f_2$$

Note: $f_c > 2B$

sketch the Fourier transform of $x[n]$ and give formulas for the following parameters:



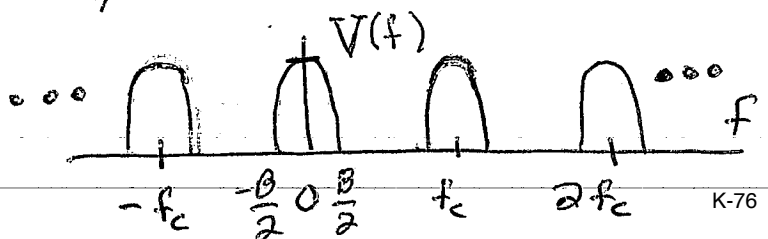
Downsampling by M reduces the input sampling rate by a factor of M . In continuous time, downsampling by M is sampling by $\frac{1}{M} f_s$. Spectrum of $X(\omega)$ is same as above. Spectrum of $v(t)$ is below:

$$f_s = M f_c$$

$$\omega_{\text{pass}} = 2\pi \frac{B/2}{f_s/M} = \pi \frac{B}{f_c}$$

$$\omega_{\text{stop}} = 2\pi \frac{B}{f_s/M} = 2\pi \frac{B}{f_c}$$

Note: With $f_s = M f_c$, $f_s > 2f_2$ only for $M \geq 3$.



Problem 1.4 Discrete-Time FIR Filter Design. 24 points.

One discrete-time filter design method determines the finite impulse response (FIR) filter coefficients by keeping the first L samples of the impulse response of an infinite impulse response (IIR) filter.

A key difficulty is in determining L .

(a) Consider the IIR impulse response $h[n] = (0.9)^n u[n]$. 9 points.

1. Compute the total energy in the IIR filter impulse response: $E_{total} = \sum_{n=0}^{\infty} |h[n]|^2$

$$E_{total} = \sum_{n=0}^{\infty} |0.9^n|^2 = \sum_{n=0}^{\infty} ((0.9)^n)^2 = \sum_{n=0}^{\infty} (0.81)^n$$

$$= \sum_{n=0}^{\infty} 0.81^n = \frac{1}{1-0.81} = \frac{1}{0.19} = 5.26316$$

2. Determine L so that the FIR filter impulse response contains 90% of the total energy of the IIR impulse response computed in part #1.

$$\sum_{n=0}^{L-1} 0.81^n \geq 0.9 E_{total}$$

$$\frac{1-0.81^L}{1-0.81} \geq \frac{0.9}{1-0.81} \Rightarrow 0.1 \geq 0.81^L$$

$$L = \frac{\ln 0.1}{\ln 0.81}$$

$$\sum_{n=0}^N a^n = \frac{1-a^{N+1}}{1-a}$$

$$\sum_{n=0}^{L-1} 0.81^n = \frac{1-0.81^L}{1-0.81}$$

$$L = 11$$

(b) Consider an IIR filter with N poles and M zeros. Determine L so that the computational complexity of the FIR filter is the same as the computational complexity in multiplication-accumulation operations of the IIR filter. Please specify the IIR filter structure you are assuming. 9 points.

- Direct form IIR filter structure takes $N+M+1$ multiplication-accumulation operations. Let $L = N+M+1$. (OR)
- Biquad cascade takes 5 multiplication-accumulation operations per cascade. For $N=M$ and N even, $L = 5 \frac{N}{2}$.

(c) Explain two important advantages for this filter design method. 6 points.

1. The FIR design method

1. always gives an BIBO stable filter when implemented
2. has the same implementation complexity as the original IIR filter when using (b).
3. gives a filter with a worst-case delay of $L-1$ samples

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 19, 2012

Course: EE 445S Evans

Name: _____ *Hat,* _____ *Cat In The*
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		Discrete-Time Filter Analysis
2	24		Discrete-Time Filter Implementation
3	27		System Identification
4	24		Upconversion
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 25 points.

A causal stable discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following transfer function:

$$H(z) = \frac{b}{1 - az^{-1}} = \frac{Y(z)}{X(z)}$$

for $|z| > |a|$. Here, a and b are real-valued, and b is not zero.

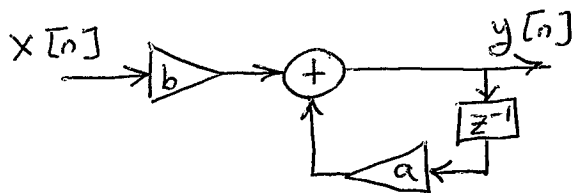
- (a) From the transfer function, derive the difference equation relating input $x[n]$ and output $y[n]$. 6 points.

$$\frac{Y(z)}{X(z)} = \frac{b}{1 - az^{-1}} \Rightarrow (1 - az^{-1})Y(z) = bX(z)$$

$$Y(z) - az^{-1}Y(z) = bX(z)$$

apply inverse z-transform: $y[n] - ay[n-1] = bx[n]$

- (b) Give the block diagram for the filter. 3 points.



$$y[n] = ay[n-1] + bx[n]$$

- (c) What are the initial conditions? What values should they be assigned and why? 4 points.

Let $n=0$: $y[0] = ay[-1] + bx[0]$

Initial condition is $y[-1]$. It must be set to zero to satisfy LTI system properties.

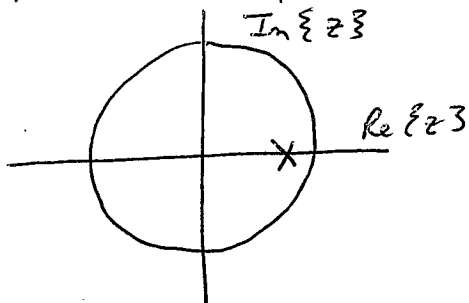
- (e) Find the equation for the frequency response of the filter. Justify your approach. 6 points.

Since the LTI system is stable, the region of convergence includes the unit circle:

$$H_{\text{freq}}(\omega) = H(z)|_{z=e^{j\omega}} = \frac{b}{1 - ae^{-j\omega}}$$

- (f) Give the best values of a and b for the filter to be lowpass with a DC response of 1. 6 points.

$H(z)$ has a pole at $z=a$. Passband is indicated



by angle of pole. Let $a=0.9$.

Response at DC ($\omega=0$):

$$H_{\text{freq}}(0) = \frac{b}{1 - a} = 1$$

$$b = 1 - a \Rightarrow b = 0.1$$

Problem 1.2 Discrete-Time Filter Implementation. 24 points.

Consider a causal stable second-order discrete-time filter with the following transfer function:

$$H(z) = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} = \frac{1 - (z_0 + z_1)z^{-1} + z_0 z_1 z^{-2}}{1 - (p_0 + p_1)z^{-1} + p_0 p_1 z^{-2}}$$

Zeros z_0 and z_1 are complex-valued and conjugate symmetric.

Poles p_0 and p_1 are complex-valued and conjugate symmetric.

Amplitudes of the input and output signals are real-valued.

Store complex numbers as a pair (real, imaginary).

(a) Biquad implementation. We expand the factored form into

$$H(z) = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

- i. Give formulas for the feedback coefficients in terms of the poles and zeros. 3 points.

$$a_1 = p_0 + p_1 \quad a_2 = -p_0 p_1$$

- ii. Which feedback coefficient suffers the largest loss of precision when feedback coefficients, zeros and poles are in 32-bit IEEE floating-point format? Why? 3 points.

In the worst case, a_1 loses 1 bit of accuracy due to addition and a_2 loses 23 bits of accuracy in mantissa due to

- iii. How many real-valued multiplication and addition operations does it take for the multiplication biquad to compute one output sample for each new input sample? 6 points.

$$y[n] = x[n] + b_1 x[n-1] + b_2 x[n-2] + a_1 y[n-1] + a_2 y[n-2]$$

4 real-valued multiplications and 4 real-valued additions

(b) Cascade of two first-order sections with transfer functions

$$H_1(z) = \frac{1 - z_0 z^{-1}}{1 - p_0 z^{-1}} \quad \text{and} \quad H_2(z) = \frac{1 - z_1 z^{-1}}{1 - p_1 z^{-1}}$$

- i. How many real-valued multiplications and real-valued additions are in one complex-valued multiplication? 3 points.

$$(a + jb)(c + jd) = (ac - bd) + j(bc + ad)$$

*4 real multiplies
2 real adds*

- ii. How many real-valued additions are in one complex-valued addition? 3 points.

$$(a + jb) + (c + jd) = (a + c) + j(b + d)$$

2 real adds

- iii. How many real-valued multiplications and additions would it take to implement a cascade of two first-order sections? 6 points.

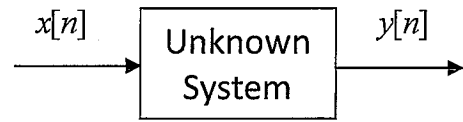
Each first-order section takes 2 complex MACs/sample.

Cascade takes 4 complex MACs/sample, or

16 real-valued MACs/sample. MAC = multiplication/accumulation

Problem 1.3 System Identification. 27 points.

Measuring the frequency response of an unknown linear time-invariant (LTI) is a common step in testing and calibration.



If $x[n]$ is all pass with unity gain, then $|Y(\omega)| = |H(\omega)|$

We perform the measurement by choosing an input $x[n]$ and observing the output $y[n]$.

Let $h[n]$ be the impulse response of the unknown discrete-time LTI system.

Let $X(\omega)$, $H(\omega)$ and $Y(\omega)$ be the discrete-time Fourier transforms of $x[n]$, $h[n]$ and $y[n]$

- (a) Show that using $x[n] = \delta[n]$, where $\delta[n]$ is the discrete-time impulse function, allows the measurement of $H(\omega)$ at all frequencies. 6 points.

$$Y(\omega) = H(\omega) X(\omega)$$

When $x[n] = \delta[n]$, $X(\omega) = 1$, and $Y(\omega) = H(\omega)$.

When $x[n] = \delta[n]$, we can observe all frequencies in $H(\omega)$.

- (b) Show that using $x[n] = \delta[n] + \delta[n-1]$ fails to measure $H(\omega)$ at all frequencies. 6 points.

$$X(\omega) = 1 + e^{-j\omega}, \text{ and at } \omega = \pi, X(\pi) = 0, \text{ and } Y(\pi) = 0.$$

We cannot observe the frequency response of $H(\omega)$ at $\omega = \pi$.

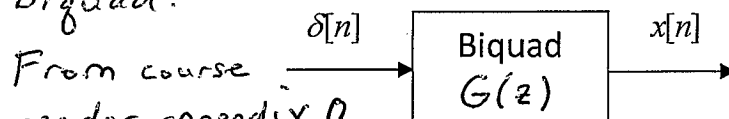
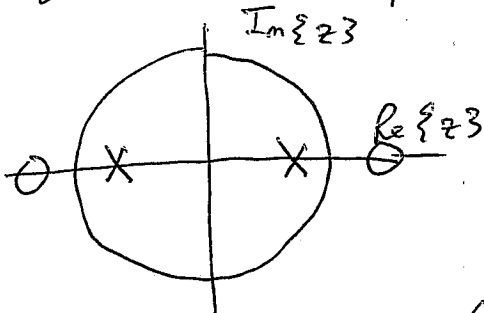
- (c) Determine a real, non-zero value of a_1 in $x[n] = \delta[n] + a_1 \delta[n-1]$ that will allow $H(\omega)$ to be measured at all frequencies. 6 points.

$X(\omega) = 1 + a_1 e^{-j\omega}$. When $a_1 = -1$, $X(0) = 0$, and we cannot observe the DC response of $H(\omega)$ at the output.

Any value for a_1 , except 1 and -1 will work.

- (d) Consider the causal discrete-time biquad filter below. Its impulse response will be used as the test signal $x[n]$ to measure $H(\omega)$ of the above unknown system at all frequencies. The biquad has poles at $z = 0.8$ and $z = -0.8$. Determine the zero locations and gain. 9 points.

We seek an all-pass biquad.



From course reader appendix D,

$$z_0 = \frac{1}{0.8} = 1.25$$

$$z_1 = \frac{1}{-0.8} = -1.25$$

Set DC gain to one.

$$G(z) \Big|_{z=1} = 1$$

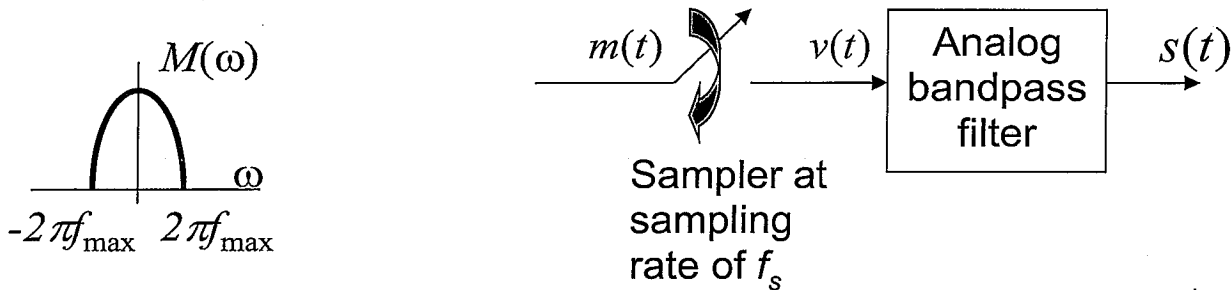
and solve for C. K-81

$$G(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

Problem 1.4 Upconversion. 24 points.

Upconversion shifts a baseband signal $m(t)$ in frequency to be centered at carrier frequency f_c .

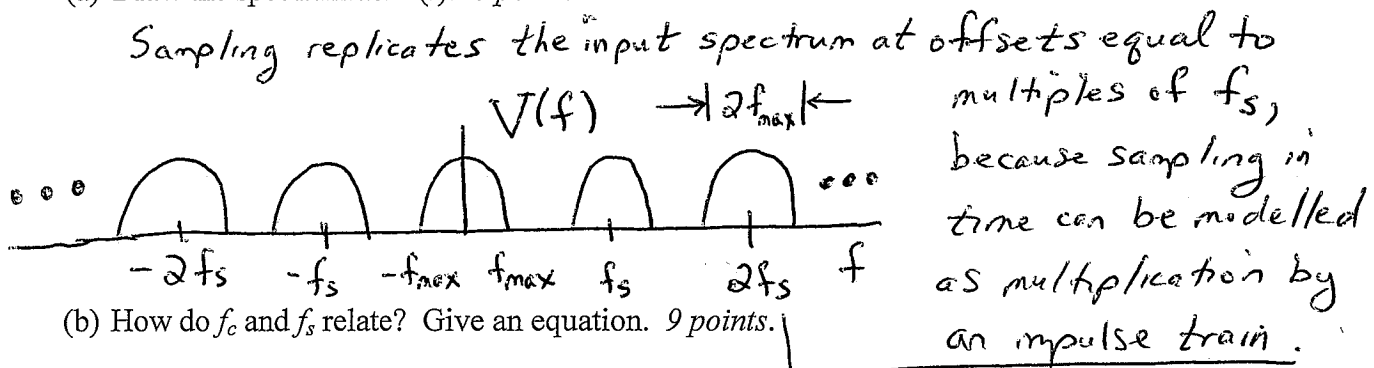
A conventional analog circuit for upconversion places a sampling device and an analog bandpass filter in cascade, as shown below.



This system is entirely in continuous time.

Assume that $f_s > 2f_{max}$.

(a) Draw the spectrum for $v(t)$. 6 points.



(b) How do f_c and f_s relate? Give an equation. 9 points.

$f_c = n f_s$ where n is a positive integer.

(c) Give a filter design specification for the analog bandpass filter. 9 points.

$$f_{stop_1} = 0.9 f_{pass_1} \quad \text{or} \quad f_{stop_1} = f_{pass_1} - (2 f_{max}) 0.1$$

$$f_{pass_1} = f_c - f_{max}$$

$$f_{pass_2} = f_c + f_{max}$$

$$f_{stop_2} = 1.1 f_{pass_2} \quad \text{or} \quad f_{stop_2} = f_{pass_2} + (2 f_{max}) 0.1$$

passband ripple of 1 dB
stopband attenuation of 40 dB

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 8, 2013

Course: EE 445S Evans

Name: Phineas and Ferb
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. *Please disable all wireless connections on your computer system(s).*
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Filter Implementation
3	24		Filter Design
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 28 points.

A causal stable discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following transfer function:

$$H(z) = 1 - z^{-3}$$

for $|z| \neq 0$.

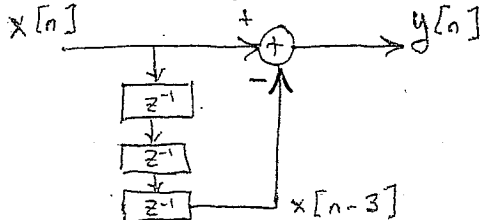
(a) From the transfer function, derive the difference equation relating input $x[n]$ and output $y[n]$.

6 points.

$$H(z) = \frac{Y(z)}{X(z)} = 1 - z^{-3}$$

$$Y(z) = (1 - z^{-3}) X(z) = X(z) - z^{-3} X(z) \Rightarrow y[n] = x[n] - x[n-3]$$

(b) Give the block diagram for the filter. 3 points.



(c) What are the initial conditions? What values should they be assigned and why? 4 points.

From the block diagram, there are three initial conditions $x[-1], x[-2], x[-3]$. They must be zero to satisfy linearity and time-invariant properties.

Alternate Solution For First Step	$y[0] = x[0] - x[-3]$	$y[2] = x[2] - x[-1]$
	$y[1] = x[1] - x[-2]$	$y[3] = x[3] - x[0]$

initial conditions are $x[-1], x[-2]$ and $x[-3]$.

(d) Find the equation for the frequency response of the filter. Justify your approach. 6 points.

The transfer function has a region of convergence of $z \neq 0$, which includes the unit circle: $H_{\text{freq}}(\omega) = H(z)|_{z=e^{j\omega}} = 1 - e^{-3j\omega}$

Alternate

Justification: LTI system is bounded-input bounded-output stable.

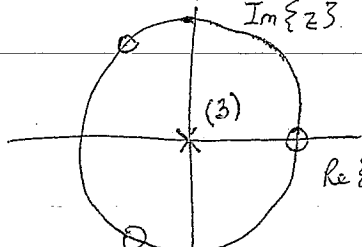
(e) What is the group delay through the filter? 3 points. $N=4$ coefficients.

Equal to midpoint of impulse response for linear phase FIR filter: $\frac{3}{2}$ samples

(f) Draw the pole-zero diagram. Is the filter lowpass, highpass, bandpass, bandstop, allpass or notch? 6 points. For $z \neq 0$,

$$H(z) = 1 - z^{-3} = 0 \Rightarrow 1 = z^{-3} \Rightarrow z^3 = 1$$

z is complex. Roots (zeros) of $z^3 - 1 = 0$ are $1, e^{j\frac{2}{3}\pi}$ and $e^{-j\frac{2}{3}\pi}$.



Notch filter.

Eliminates frequencies at $\omega=0$ and $\omega=\frac{2}{3}\pi$.

(Note: There are three poles at origin.)

$$H_{\text{freq}}(\omega) = e^{-j\frac{3}{2}\omega} (e^{j\frac{3}{2}\omega} - e^{-j\frac{3}{2}\omega})$$

Note: $j = e^{j\frac{\pi}{2}}$

$$= 2 \sin(\frac{3}{2}\omega) e^{-j\frac{3}{2}\omega}$$

$$gd(\omega) = -\frac{d}{d\omega} \angle H_{\text{freq}}(\omega)$$

$$= \frac{3}{2} = \frac{N-1}{2}$$

Problem 1.2 Discrete-Time Filter Implementation. 24 points.

Consider a causal fourth-order discrete-time infinite impulse response (IIR) filter with transfer function $H(z)$. A filter is a bounded-input bounded-output stable linear time-invariant system.

Input $x[n]$ and output $y[n]$ are real-valued.

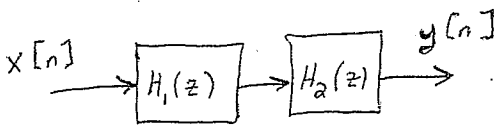
Cascade of biquads. We factor $H(z)$ into a product of two second-order sections (biquads)

$$H(z) = H_1(z) H_2(z)$$

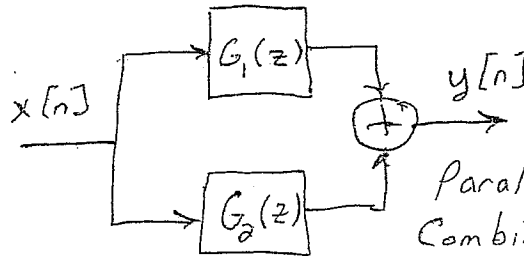
Parallel combination of biquads. We perform partial fraction decomposition on $H(z)$ to write it as a sum of two second-order sections (biquads)

$$H(z) = G_1(z) + G_2(z)$$

- (a) Draw the block diagrams for the *cascade of biquads* and the *parallel combination of biquads*. Each block in the block diagram would correspond to a biquad. 6 points.



Cascade of Biquads



Parallel Combination

- (b) Consider the implementation of the two filter structures on the TI TMS320C6700 DSP. Assuming that data values and filter coefficients are in 32-bit floating point.

- i. Compare the memory usage for the two structures. 3 points.

words	10 coefficients	10 coefficients
	12 current & previous inputs/outputs	12 current & previous inputs/outputs
	22 total (3 outputs of $H_1(z)$ can be shared)	23 total (input can be shared between $G_1(z)$ and $G_2(z)$)

- ii. Compare the execution cycles for the two structures. 6 points.

worst case	33 cycles + 33 cycles = 66 cycles	33 cycles + 33 cycles + 4 cycles = 70 cycles
overlapping	(4 + 5 + 28) cycles + 4 cycles = 41	(5 + 5 + 28) cycles + 4 cycles = 42 cycles

- (c) Consider the implementation of the two filter structures on a processor with two TI TMS 320C6700 DSP cores (CPUs). The cores share the same on-chip memory.

- i. Compare the memory usage for the two structures. 3 points. (1 store + 1 load inst.)

Same as (b)i. because of shared on-chip memory

- ii. Compare the execution cycles for the two structures. 6 points.

Implement cascade of biquads on a single core due to the high 10-cycle inter-core communication costs

Same as (b) ii.

Implement parallel combination by placing $G_1(z)$ on one core and $G_2(z)$ on other. They can run in parallel. Also, need 10 cycles to communicate result and 4 cycles for adding the two branches. 47 cycles total.

Problem 1.3 Filter Design. 24 points.

In North America, there is a narrowband WWVB timing signal being broadcast at 60 kHz.

The G.hnem powerline communication standard uses a sampling rate 800 kHz and operates in the 34.4 kHz to 478.1 kHz band.

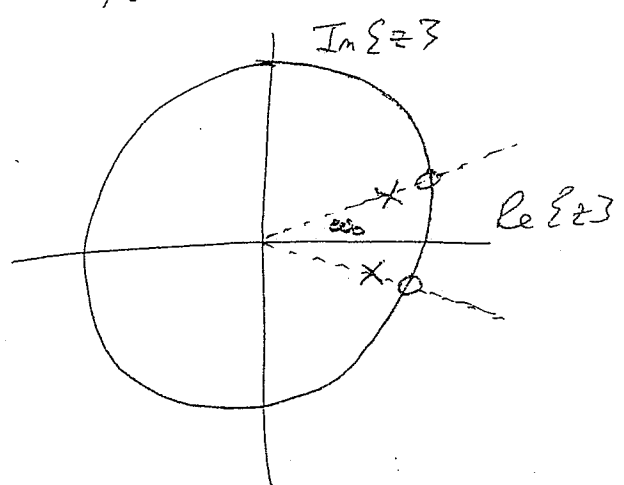
G.hnem receivers experience in-band interference from the WWVB signal.

- (a) Design a discrete-time second-order infinite impulse response (IIR) filter for a G.hnem transceiver to remove the 60 kHz WWVB interferer. Give poles, zeros and gain. 12 points.

Notch filter at $\omega_0 = 2\pi \frac{60 \text{ kHz}}{800 \text{ kHz}} = 2\pi \frac{3}{40} \text{ rad/sample}$.

Poles at $0.9e^{j\omega_0}$ and $0.9e^{-j\omega_0}$
 Zeros at $e^{j\omega_0}$ and $e^{-j\omega_0}$

$$H(z) = C_0 \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$



Set DC gain to be one:

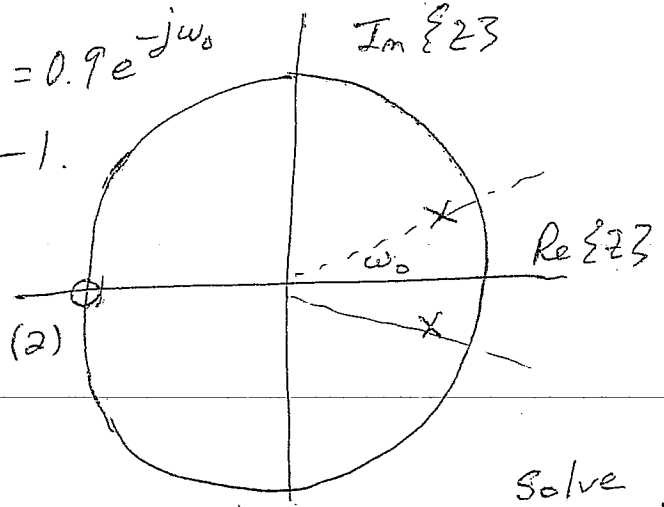
$$H(z) \Big|_{z=1} = 1 \text{ solve for } C_0.$$

- (b) Design a discrete-time second-order infinite impulse response (IIR) filter for a G.hnem transceiver to extract the 60 kHz WWVB signal for use in generating timestamps for power load profiles at the consumer's site. Give poles, zeros and gain. 12 points.

$\omega_0 = 2\pi \frac{60 \text{ kHz}}{800 \text{ kHz}} = 2\pi \frac{3}{40} \text{ rad/sample}$ Bandpass filter.

Poles at $p_0 = 0.9e^{j\omega_0}$ and $p_1 = 0.9e^{-j\omega_0}$
 Zeros at $z_0 = -1$ and $z_1 = -1$.

$$G(z) = C_1 \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$



Set DC gain to be one:

$$G(z) \Big|_{z=1} = 1 \text{ solve for } C_1.$$

(Alternatively, set gain at ω_0 to be one: $G(z) \Big|_{z=e^{j\omega_0}} = 1$ solve for C_1)

Problem 1.4. Potpourri. 24 points.

- (a) You want to design a linear phase finite impulse response (FIR) filter with 10,000 coefficients that meets a magnitude specification. Which FIR filter design method would you advocate using? 6 points. As demonstrated in lecture...

Parks-McClellan (Remez) filter design algorithm is iterative and fails to converge for filter orders ~ 400 . Least squares filter design would require inversion of a $10,000 \times 10,000$ matrix, and may not produce reliable results. Kaiser window design uses formulas for the coefficients, and can design FIR filters with 10,000 coefficients.

- (b) Consider a causal first-order IIR filter with non-zero feedback coefficient a_1 and input signal $x[n]$. Output signal is $y[n] = a_1 y[n-1] + x[n]$. Input data, output data and feedback coefficient are unsigned 16-bit integers. As n increases, does the number of bits needed to keep calculations from losing precision always increase without bound? If yes, show that it is true for all non-zero values of a_1 . If no, give a counter-example. 6 points.

No. Let $x[n] = \delta[n]$ and $a_1 = 1$.

$$y[n] = u[n].$$

(Alternate answer from lecture discussion:
Let $a_1 = \frac{1}{2}$. $y[n]$ needs 17 bits.)

Alternate answer: Use an IIR filter design method and keep the first 10,000 samples of the impulse response.

- (c) Given three reasons why 32-bit floating-point data and arithmetic is better suited for audio processing than 16-bit integer data and arithmetic? 6 points. "Comparing fixed and floating point DSPs" article in course reader after lecture 2 slides

- 24-bits of mantissa + sign (integer component) gives 144 dB of dynamic range vs. 96 dB of dynamic range for 16-bit integer
- 8-bit of exponent allows wide dynamic range and is very accurate near zero (quiet regions with low accuracy in 16-bit integer)
- Audio uses IIR filters, and 32-bit precision is needed to reduce accumulation of numeric error (truncation/rounding) via feedback

- (d) What three instruction set architecture features would accelerate finite impulse response (FIR) filtering? 6 points.

Instruction set architecture (CPU core)

- Fast multiplier (pipelined)
- Fast adder (pipelined)
- Separate program and data buses
- Multiple data buses and simultaneous load from all buses
- Modulo addressing for circular buffer
- Fast down counting

Other processor enhancements

- Direct memory access controllers for ping-pong buffering + frame based input/output + buffer management (on chip)
- Dual-ported on-chip memory for two reads in same cycle

Autoincrement or autodecrement addressing modes K-87

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 18, 2013

Course: EE 445S Evans

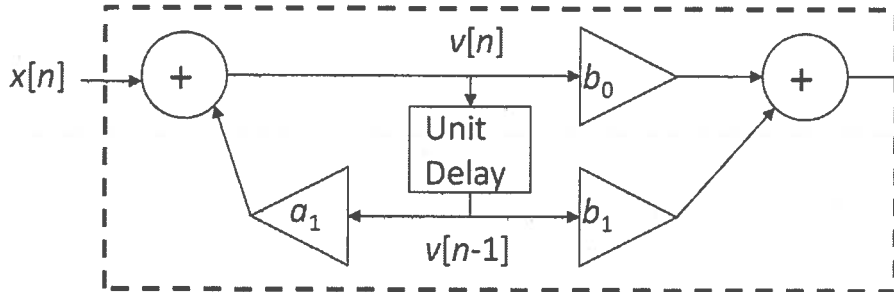
Name: Bollywood, Sally
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Please disable all wireless connections on your computer system(s).**
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	27 28		Discrete-Time Filter Analysis
2	24		Discrete-Time Filter Design
3	24		System Identification
4	24		Modulation and Demodulation
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 21 points.

A causal stable discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following block diagram:



First-Order
IIR
Section.

Derived from
slide 6-6 on
Discrete-Time

Biquad.

Constants a_1 , b_0 and b_1 are real-valued, and $|a_1| < 1$.

- (a) From the block diagram, derive the difference equation relating input $x[n]$ and output $y[n]$. Your final answer should not include $v[n]$. 6 points.

Working backwards from transfer function in part (c) below,

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \Rightarrow (1 - a_1 z^{-1}) Y(z) = (b_0 + b_1 z^{-1}) X(z)$$

Applying the inverse z-transform to both sides,

$$y[n] - a_1 y[n-1] = b_0 x[n] + b_1 x[n-1]$$

- (b) What are the initial condition(s)? What value(s) should they be assigned and why? 4 points.

$v[-1] = 0$ for the system to be causal, linear and time-invariant.

Equivalently, $x[-1] = 0$ and $y[-1] = 0$.

- (c) What is the transfer function in the z-domain? What is the region of convergence? 5 points.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{V(z)}{X(z)} \cdot \frac{Y(z)}{V(z)} = \frac{1}{1 - a_1 z^{-1}} \cdot (b_0 + b_1 z^{-1})$$

$$= \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \text{ for } |z| > |a_1|$$

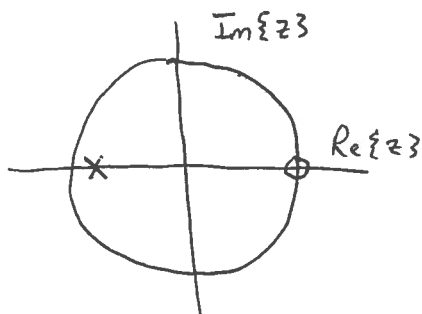
- (d) Find the equation for the frequency response of the filter. Justify your approach. 6 points.

Because $|a_1| < 1$, the region of convergence $|z| > |a_1|$

includes the
unit circle.

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{b_0 + b_1 e^{-j\omega}}{1 - a_1 e^{-j\omega}}$$

- (e) For $a_1 = -0.9$, $b_0 = 1$, and $b_1 = -1$, draw the pole-zero diagram. What is the best description of the frequency selectivity: lowpass, highpass, bandstop, bandpass, allpass or notch? 7 points.



Passband is centered at $\omega = \pi$
due to pole at $z = -0.9$.

Stopband is centered at $\omega = 0$
due to zero at $z = 1$.

Highpass filter.

Problem 1.2 Discrete-Time Filter Design. 24 points. *Configurable/programmable notch filter*

Consider a causal second-order discrete-time infinite impulse response (IIR) filter with transfer function $H(z)$.

The filter is a bounded-input bounded-output stable, linear, and time-invariant system.

Input $x[n]$ and output $y[n]$ are real-valued.

The feedback and feedforward coefficients are real-valued. \Rightarrow Poles are conjugate symmetric.

You will be asked to design and implement a notch filter: Zeros are conjugate symmetric.

f_0 is the frequency in Hz to be eliminated, and

f_s is the sampling rate in Hz where $f_s > 2f_0$

Assume that the gain of the biquad is 1. $\Rightarrow C = 1$

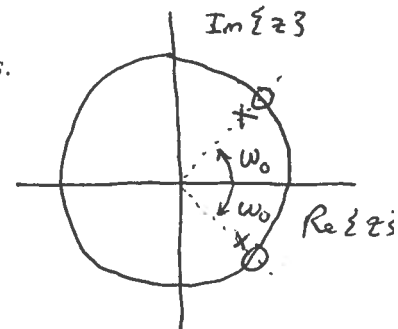
(a) Give a formula for the discrete-time frequency ω_0 in rad/sample to be eliminated. 3 points.

$$\omega_0 = 2\pi \frac{f_0}{f_s}$$

(b) Give formulas for the two poles and the two zeros as functions of ω_0 . 6 points.

Poles: $p_0 = 0.9 e^{j\omega_0}$ and $p_1 = 0.9 e^{-j\omega_0}$

Zeros: $z_0 = e^{j\omega_0}$ and $z_1 = e^{-j\omega_0}$



(c) Give formulas for the three feedforward and two feedback coefficients. Simplify the formulas to show that all of these coefficients are real-valued. 9 points.

$$H(z) = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} = \frac{1 - (z_0 + z_1)z^{-1} + z_0 z_1 z^{-2}}{1 - (p_0 + p_1)z^{-1} + p_0 p_1 z^{-2}}$$

Feed forward coefficients

$$b_0 = 1$$

$$b_1 = -(z_0 + z_1) = -(e^{j\omega_0} + e^{-j\omega_0}) = -2 \cos \omega_0$$

$$b_2 = z_0 z_1 = e^{j\omega_0} e^{-j\omega_0} = 1$$

Feedback coefficients

$$a_1 = p_0 + p_1 = 1.8 \cos \omega_0$$

$$a_2 = -p_0 p_1 = -0.81$$

(d) How many multiplication-accumulation operations are needed to compute one output sample given one input sample? 3 points.

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + x[n] + b_1 x[n-1] + x[n-2]$$

3 multiplications and 4 additions \Rightarrow 4 multiply-accumulates

(e) How many instruction cycles on the TI TMS3206748 digital signal processor used in lab will take to compute one output sample given one input sample? 3 points.

$N = 5$ coefficients

$N + 28 = 33$ instruction cycles from Appendix N in reader.

Alternate Solution:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{1-z^{-1}} = (1+z^{-1})(1-z^{-1}) = 1-z^{-2} \Rightarrow h[n] = \delta[n] - \delta[n-2]$$

Problem 1.3 System Identification. 24 points. Solution uses de-convolution.

Consider a causal discrete-time finite impulse response (FIR) filter with impulse response $h[n]$.

The filter is a bounded-input bounded-output stable, linear, and time-invariant system.

For input $x[n] = u[n]$, the output is $y[n] = \delta[n] + \delta[n-1]$. Let $h[n]$ have $M+1$ coefficients.

(a) Determine the impulse response $h[n]$. 18 points.

$$y[n] = x[n] * h[n] = \sum_{m=0}^M h[m] x[n-m]$$

0 $1 = y[0] = h[0]x[0] \Rightarrow 1 = h[0] \Rightarrow h[0] = 1$

1 $1 = y[1] = h[0]x[1] + h[1]x[0]$
 $1 = h[0] + h[1] \Rightarrow h[1] = 0$

2 $y[2] = h[0]x[2] + h[1]x[1] + h[2]x[0]$
 $0 = h[0] + h[1] + h[2] \Rightarrow h[2] = -h[0] \Rightarrow h[2] = -1$

3 $y[3] = h[0]x[3] + h[1]x[2] + h[2]x[1] + h[3]x[0]$
 $0 = h[0] + h[1] + h[2] + h[3] \Rightarrow h[3] = 0$

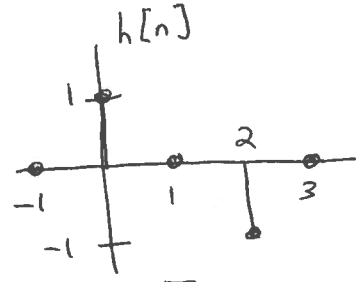
Check: $h[n] * u[n] \stackrel{?}{=} \delta[n] + \delta[n-1]$ **YES**

(b) Compute the group delay through the filter as a function of frequency. 6 points.

$$H_{\text{freq}}(\omega) = 1 - e^{-j2\omega}$$

$$= e^{-j\omega} (e^{j\omega} - e^{-j\omega})$$

$$= \underbrace{2 \sin(\omega)}_{\text{amplitude term}} \underbrace{j e^{-j\omega}}_{\text{phase term}}$$



Note: $j = e^{+j\frac{\pi}{2}}$

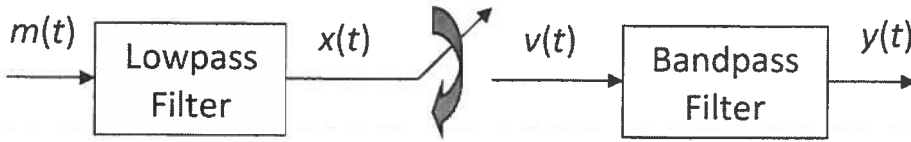
Group delay is 1 sample.

$$\text{Delay}(\omega) = -\frac{d}{d\omega} \angle H_{\text{freq}}(\omega) = 1$$

Except for two points of discontinuity, $\angle H_{\text{freq}}(\omega) = -\omega + \frac{\pi}{2}$

Problem 1.4. Modulation and Demodulation. 24 points.

A mixer can be used to realize sinusoidal amplitude modulation $y(t) = x(t) \cos(2\pi f_c t)$ for baseband signal $x(t)$:

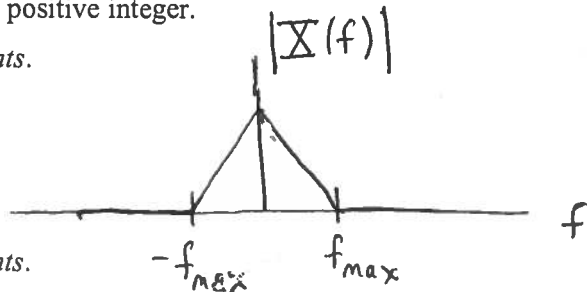


Sampler at sampling rate of f_s

Assume that $x(t)$ is an ideal ~~bandpass~~^{baseband} signal whose magnitude spectrum is zero for $|f| > f_{max}$.

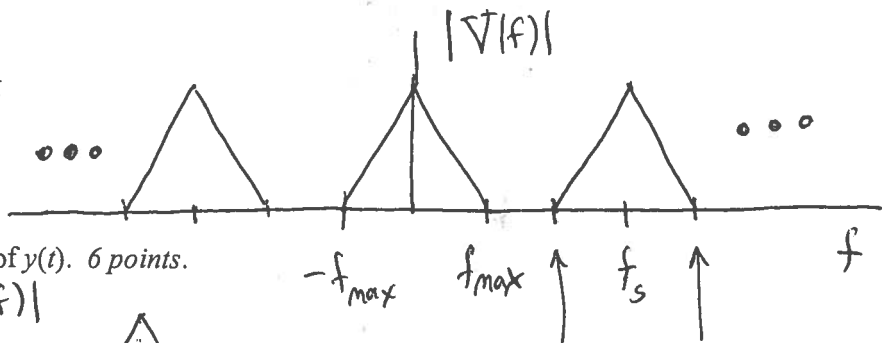
Assume that $f_s > 2f_{max}$ and $f_c = m f_s$ where m is a positive integer.

(a) Draw the magnitude spectrum of $x(t)$. 6 points.

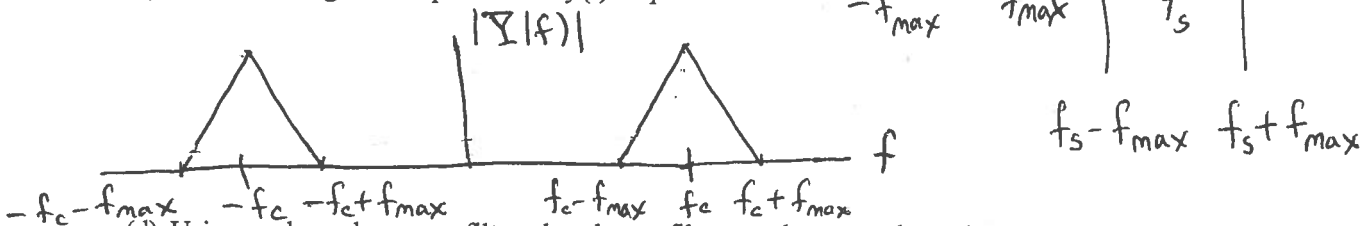


(b) Draw the magnitude spectrum of $v(t)$. 6 points.

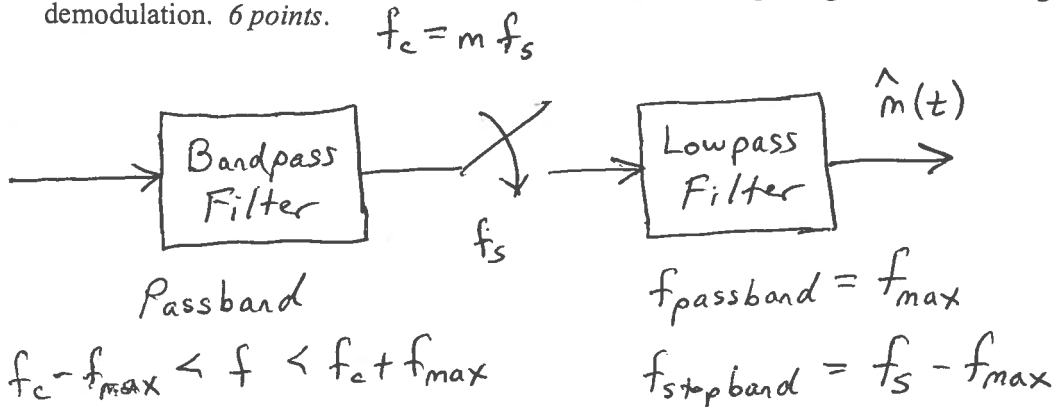
Spectrum of $X(f)$ is replicated at offsets in frequency equal to multiples of f_s



(c) Draw the magnitude spectrum of $y(t)$. 6 points.



(d) Using only a lowpass filter, bandpass filter, and a sampler, give a block diagram for demodulation. 6 points.



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 7, 2014

Course: EE 445S Evans

Name: _____ **Team , High 5** _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Discrete-Time Filter Analysis
2	24		Improving Signal Quality
3	24		Filter Bank Design
4	24		Potpourri
<i>Total</i>	100		

Discussion of the solutions is available at

<http://www.youtube.com/watch?v=HRX3x45CSIA&list=PLaJppqXMef2ZHIKM4vpwHIAWyRmw3TtSf>

Problem 1.1 Discrete-Time Filter Analysis. 28 points.

A causal stable discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following transfer function:

$$\frac{Y(z)}{X(z)} = H(z) = C \frac{(z-z_0)(z-z_1)}{(z-p_0)(z-p_1)} = C \frac{(1-z_0z^{-1})(1-z_1z^{-1})}{(1-p_0z^{-1})(1-p_1z^{-1})} = \frac{C - C(z_0+z_1)z^{-1} + Cz_0z_1z^{-2}}{1 - (p_0+p_1)z^{-1} + p_0p_1z^{-2}}$$

Constant C is real-valued and is not equal to zero. Zero locations are z_0 and z_1 . Pole locations are p_0 and p_1 where $|p_0| < 1$ and $|p_1| < 1$.

- (a) From the transfer function, give formulas for the feedforward coefficients and the feedback coefficients in terms of the pole locations, zero locations and constant C . 6 points.

$$\begin{aligned} b_0 &= C \\ b_1 &= -C(z_0+z_1) & a_1 &= p_0+p_1 \\ b_2 &= Cz_0z_1 & a_2 &= -p_0p_1 \end{aligned}$$

- (b) Give the difference equation relating input $x[n]$ and output $y[n]$ in terms of the feedforward and feedback coefficients. 6 points.

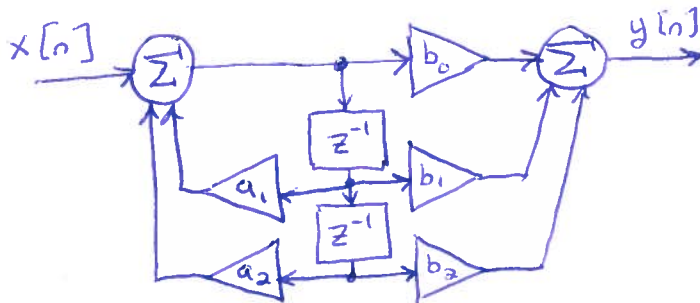
$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

- (c) What are the initial condition(s)? What value(s) should they be assigned and why? 4 points.

Initial conditions are $x[-1]$, $x[-2]$, $y[-1]$ and $y[-2]$.

They should be set to zero to ensure system properties of

- (d) Draw a block diagram for the filter. 6 points.

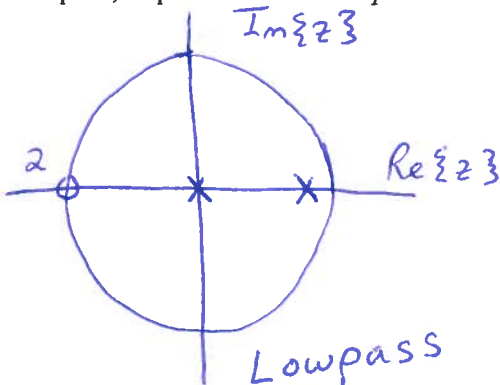


causality and linearity and time-invariance.

One can see the initial conditions by computing the first value of $y[n]$:

$$y[0] = a_1 y[-1] + a_2 y[-2] + b_0 x[0] + b_1 x[-1] + b_2 x[-2]$$

- (e) For zeros $z_0 = -1$ and $z_1 = -1$ and poles $p_0 = 0.9$ and $p_1 = 0$, draw the pole-zero diagram. What is the best description of the frequency selectivity of the filter: lowpass, highpass, bandstop, bandpass, allpass or notch? 6 points.



When poles and zeros are separated in angle, a pole near the unit circle indicates the passband (at $\omega = 0$ since $z = 0.9$) and a zero on or near the unit circle indicates the stopband (at $\omega = \pi$ since $z = -1$ or equivalently at $\omega = -\pi$). Pole at origin does not affect magnitude response.

Problem 1.2 Improving Signal Quality. 24 points.

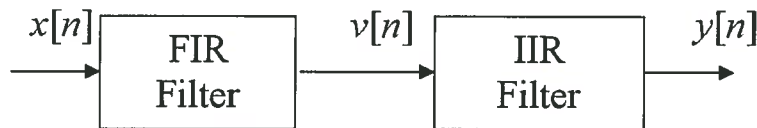
In smart grids, communication between customer power meters and the local utility can occur over the (outdoor) power line:

- Transmission band: 40-90 kHz
- Sampling rate: 400 kHz

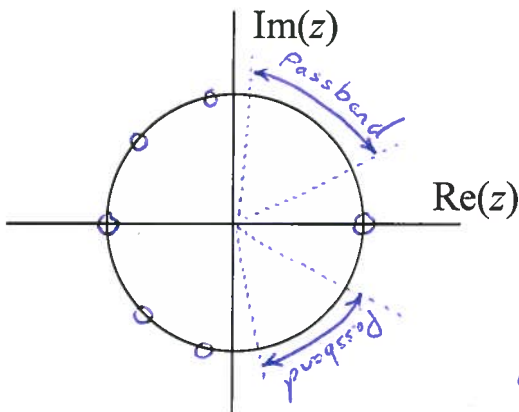
Consider the following sources of distortion:

- Additive noise
- Narrowband interferer at 50 kHz

Consider the following cascade of filters in the receiver to improve signal quality:



- (a) Design a sixth-order finite impulse response (FIR) filter to reduce out-of-band additive noise by manually placing zeros on the pole-zero diagram below. 9 points.



Sixth-order FIR filter means six zeros, and six trivial poles at $z=0$ which will be omitted here. Passband frequencies:
 $2\pi \frac{40 \text{ kHz}}{400 \text{ kHz}} < |\omega| < 2\pi \frac{90 \text{ kHz}}{400 \text{ kHz}}$
 or $\frac{\pi}{5} < |\omega| < \frac{9}{20}\pi$. Many solutions.

- (b) Design an infinite impulse response (IIR) filter **biquad** to remove the 50 kHz interferer.

- i. Give formula for discrete-time frequency ω_0 in rad/sample of the interferer. 3 points.

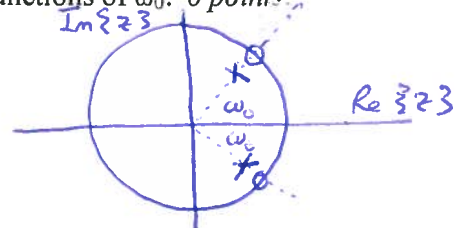
$$\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{50 \text{ kHz}}{400 \text{ kHz}} = \frac{\pi}{4} \text{ rad/sample}$$

- ii. Give formulas for the two poles and the two zeros as functions of ω_0 . 6 points.

notch filter

$$z_0 = e^{j\omega_0} \quad p_0 = 0.9 e^{j\omega_0}$$

$$z_1 = e^{-j\omega_0} \quad p_1 = 0.9 e^{-j\omega_0}$$



- (c) How many instruction cycles on the TI TMS3206748 digital signal processor used in lab will take to compute one output sample $y[n]$ given one input sample $x[n]$? 6 points.

According to Appendix N in the course reader, each filter would take $N+28$ cycles where N is the number of coefficients:

$$\underbrace{(7+28)}_{\text{FIR}} + \underbrace{(5+28)}_{\text{IIR}} = 68 \text{ cycles.}$$

For an FIR filter, the group delay is constant if the impulse response is either even symmetric (linear phase case) or odd symmetric (generalized linear phase case) about its midpoint of $\frac{N-1}{2}$ for N coefficients.

Problem 1.3 Filter Bank Design. 24 points.

Show on the right is a bank of N filters to decompose signal $x[n]$ into N frequency bands.

Each filter has a finite impulse response (FIR):

- Filter $h_0[n]$ is lowpass.
- Filter $h_{N-1}[n]$ is highpass.
- All other filters are bandpass.

Each FIR filter has N coefficients.

(a) Let filter $h_0[n]$ be an averaging filter. 6 points.

i. What is the null bandwidth? Why?

$h_0[n]$ unnormalized version is a rectangular pulse of length N samples.



For a continuous-time rectangular pulse, the null bandwidth is $\frac{1}{T}$ in Hz and $\frac{2\pi}{T}$ in rad/s for a pulse of length T sec.

ii. What is the group delay? Why?

Impulse response is even symmetric about its midpoint. Group delay is $\frac{N-1}{2}$ samples per the note at the top of the page.

Null bandwidth is $\frac{2\pi}{N}$, as mentioned in solutions for homework problem 2.1.

(b) Derive the filter $h_{N-1}[n]$ from $h_0[n]$ in part (a). 9 points.

i. With $g[n] = (-1)^n$, show that $h_{N-1}[n] = g[n] h_0[n]$ is highpass.

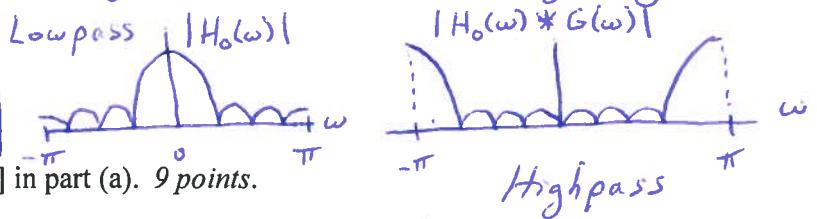
$$g[n] = (-1)^n = e^{j\pi n} = \cos(\pi n)$$

This product $g[n] h_0[n]$ is amplitude modulation by cosine.

The frequency response of $H_0(\omega)$ will shift left and right by π .

ii. What is the group delay? Why?

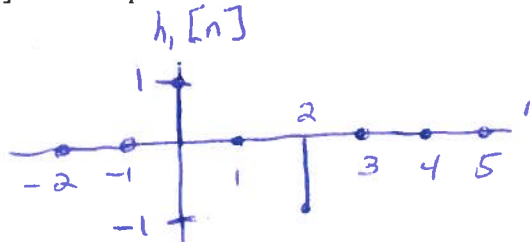
Impulse response is odd symmetric about its midpoint. Group delay is $\frac{N-1}{2}$ samples as per note.



(c) For the case $N=3$, derive $h_1[n]$ from $h_0[n]$ in part (a). 9 points.

i. Give $g_1[n]$ so that $h_1[n] = g_1[n] h_0[n]$ is a bandpass filter centered at $\pi/2$.

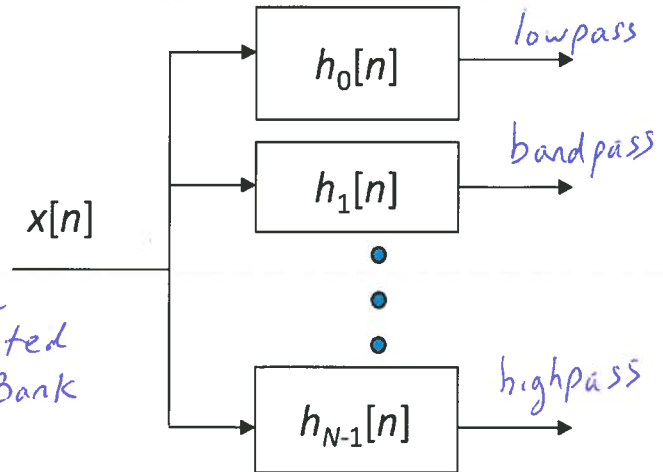
$$g_1[n] = \cos\left(\frac{\pi}{2}n\right)$$



ii. What is the group delay? Why?

Impulse response is odd symmetric about its midpoint.

Group delay is $\frac{N-1}{2}$ samples as per the note.



Cosine Modulated Filter Bank

Problem 1.4. Potpourri. 24 points.

(a) Assuming the use of an analog-to-digital converter at the front end of a signal processing system, what are the design tradeoffs in a signal processing system when increasing the sampling rate beyond twice the maximum frequency of interest with respect to

- i. Signal quality. 6 points. Signal quality will increase with increasing $f_s > 2f_0$. Aliasing will always occur in practice when sampling because thermal noise is present at all frequencies. Increasing f_s beyond $2f_0$ will decrease the amount of aliasing and increase the amount of noise (noise energy) between f_0 and $\frac{1}{2}f_s$. We can apply discrete-time filtering to control the latter. Also, the oversampling allows better tracking of the amplitude of the signal in time.
- ii. Implementation complexity. 6 points.

Increasing the sampling rate will increase the number of samples per second and hence increase the number of operations (add, multiply) per second.

Also, increasing the sampling rate will increase filter orders and hence both memory size (to store previous inputs and/or outputs) and math operations per second. Filter order is inversely proportional to the transition bandwidth:

(b) Due to certain digital signal processing operations, esp. in communication systems, signals can have a large DC offset. This is a particular problem when implementing a system in fixed-point (integer) data and arithmetic. How would you suggest removing the DC offset? 6 points.

Consider $y[n] = x[n] + \underbrace{C_0}_{\text{DC offset}}$

Answer #1: Use floating-point data and arithmetic.

DC offset will mostly affect the exponent.

Answer #2: Subtract average value of $y[n]$ from $y[n]$ periodically.

Answer #3: Use notch IIR filter at $\omega = 0$.

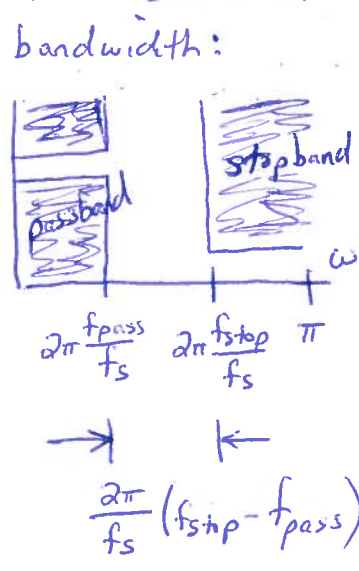
(c) You are asked to design a discrete-time bandpass filter to pass subwoofer frequencies (20-200 Hz) in a digital audio signal that has been sampled at 44.1 kHz. Would you advocate using a finite impulse response (FIR) filter or an infinite impulse response (IIR) filter? Why? 6 points.

We would like a good phase response. Either a linear phase FIR filter or an IIR filter with approximate linear phase in passband could work.

For stability in implementation, an FIR filter would be preferred.

Using $f_{\text{data}} = 44.1 \text{ kHz}$ for $f_{\text{stop}_1} = 2 \text{ Hz}$, $f_{\text{pass}_1} = 20 \text{ Hz}$, $f_{\text{pass}_2} = 200 \text{ Hz}$, $f_{\text{stop}_2} = 300 \text{ Hz}$, passband ripple of 1 dB, stopband attenuation of 60 dB, we need a 144th-order elliptic IIR filter or an 8883-order Kaiser FIR filter.

and hence both memory size (to store previous inputs and/or outputs) and math operations per second. Filter order is inversely proportional to the transition bandwidth:



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 17, 2014

Course: EE 445S Evans

Name: Code Lyoko
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Discrete-Time Filter Analysis
2	24		Upconversion
3	30		Filter Design
4	18		Potpourri
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 28 points.

A causal stable discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following equation in the discrete-time domain:

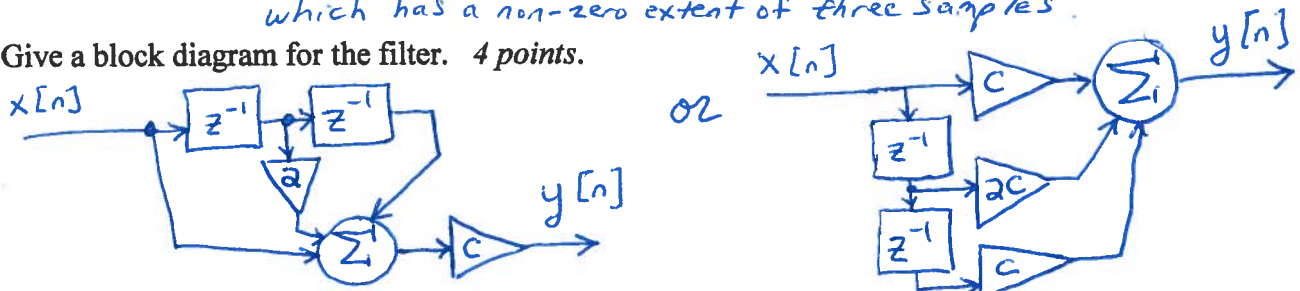
$$y[n] = C (x[n] + 2 x[n-1] + x[n-2])$$

Constant C is real-valued and is not equal to zero.

(a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 4 points.

FIR. Reason #1: Output value does not depend on previous output values.
Reason #2: Impulse response is $h[n] = C (\delta[n] + 2\delta[n-1] + \delta[n-2])$ which has a non-zero extent of three samples.

(b) Give a block diagram for the filter. 4 points.



(c) What are the initial condition(s)? What value(s) should they be assigned and why? 4 points.

Let $n=0$: $y[0] = C (x[0] + 2x[-1] + x[-2])$; Initial conditions

Let $n=1$: $y[1] = C (x[1] + 2x[0] + x[-1])$; $x[-1]$ and $x[-2]$. Set to zero to satisfy system properties: causality, linearity, time-invariance.

(d) Find the equation for the transfer function of the filter in the z -domain including the region of convergence. 4 points.

Take z -transform of both sides of the difference equation,

$$Y(z) = C (X(z) + 2z^{-1}X(z) + z^{-2}X(z))$$

$$H(z) = \frac{Y(z)}{X(z)} = C (1 + 2z^{-1} + z^{-2}) \text{ for all } z \text{ except } z=0.$$

(e) Find the equation for the frequency response of the filter. 4 points.

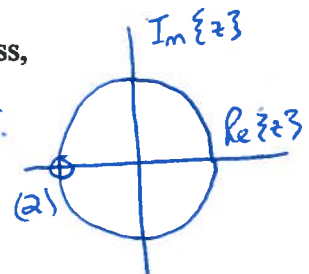
Because the region of convergence includes the unit circle,

$$H_{\text{freq}}(\omega) = H(z) \Big|_{z=e^{j\omega}} = C (1 + 2e^{-j\omega} + e^{-2j\omega})$$

(f) What is the best description of the frequency selectivity of the filter: lowpass, highpass, bandstop, bandpass, allpass or notch? Why? 4 points

Lowpass. Reason #1: Two zeros at $z=-1 \Rightarrow \omega = \pi, -\pi$. Zeros indicate center of stop band.

Reason #2: Try values of ω . $H_{\text{freq}}(\omega)$ is $4C @ \omega=0, 2jC @ \omega=\frac{\pi}{2}, 0 @ \omega=\pi$.



(g) Find a value for C that normalizes the magnitude response. 4 points.

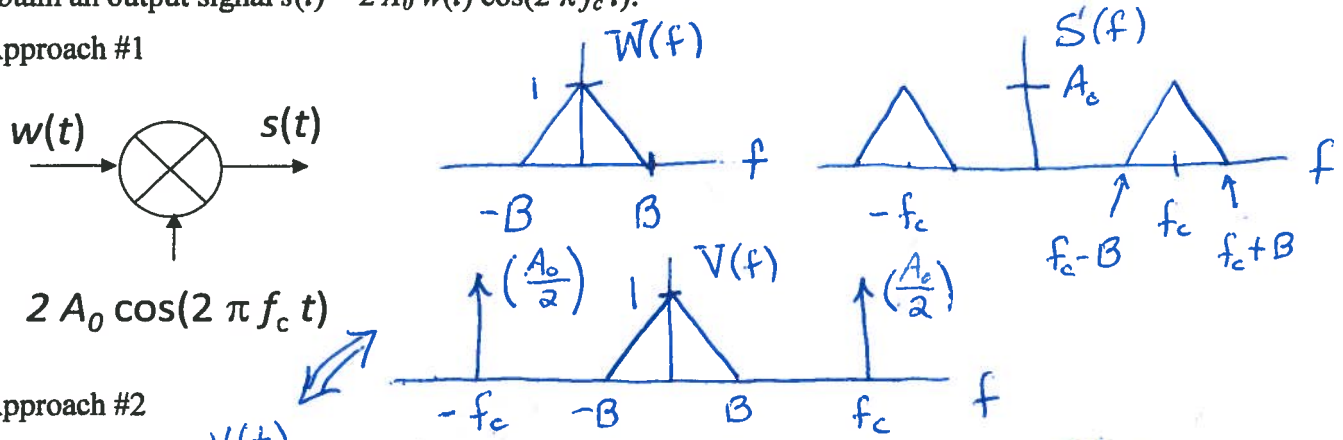
Peak value occurs at $\omega=0$ or $z=1$.

$$H_{\text{freq}}(0) = C (1 + 2 + 1) = 4C = 1 \Rightarrow C = \frac{1}{4}$$

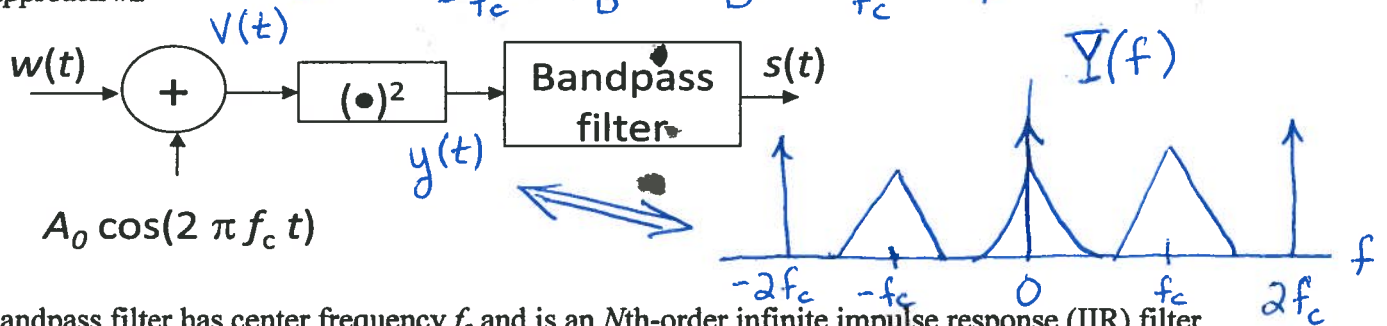
Problem 1.2 Upconversion. 24 points.

Here are two approaches to upconvert a baseband message signal $w(t)$ to a carrier frequency f_c to obtain an output signal $s(t) = 2 A_0 w(t) \cos(2 \pi f_c t)$:

Approach #1



Approach #2



Bandpass filter has center frequency f_c and is an N th-order infinite impulse response (IIR) filter.

Baseband message signal $w(t)$ has bandwidth B where $f_c > 2 B$. $f_c > 3 B$.

(a) For approach #1, please determine

- minimum sampling rate f_s needed for a discrete-time implementation. 3 points.
- multiplication-addition operations/second for the discrete-time implementation. 9 points.

$f_s > 2 f_{max}$ where $f_{max} = f_c + B$ as shown above.

Discrete-time implementation by sampling system at $t = nT_s = \frac{n}{f_s}$.

- Multiplication of $w[n]$ and $2 A_0 \cos(\omega_c n)$ where $\omega_c = 2\pi \frac{f_c}{f_s} = 2\pi \frac{N}{L}$
- Generation of $\cos(\omega_c n)$ using lookup table with L entries. 0 mults.
- Pre-compute constant $2 A_0$. $2 f_s$ multiplication-additions/s

(b) For approach #2, please determine

- minimum sampling rate f_s needed for a discrete-time implementation. 3 points.
- multiplication-addition operations/second for the discrete-time implementation. 9 points.

$f_s > 2 f_{max}$ where $f_{max} = 2 f_c$ as shown above.

Discrete-time implementation by sampling system at $t = nT_s = \frac{n}{f_s}$.

- Addition of $w[n]$ and $A_0 \cos(\omega_c n)$ plus multiplication for $A_0 \cos(\omega_c n)$.
- Generation of $\cos(\omega_c n)$ using lookup table with L entries. 0 mults.
- Squaring block takes 1 multiplication per sample
- Bandpass filter takes $(2N+1)$ multiplication-additions/sample

Total $(2N+3) f_s$ multiplication-additions/s.

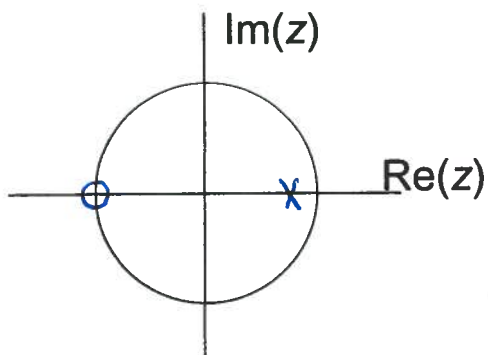
Problem 1.3 Filter Design. 30 points.

Consider design of discrete-time linear time-invariant filters by manually placing only real-valued poles and real-valued zeros.

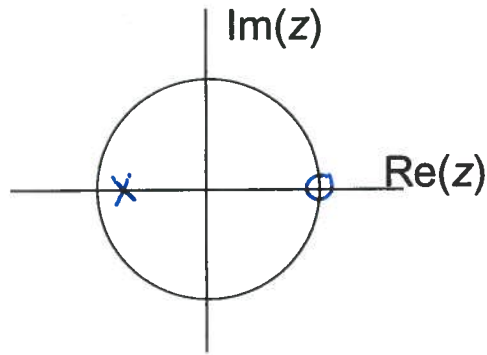
For each frequency selectivity below, indicate YES if at least one filter could be designed to give that selectivity, and NO if there isn't any filter that could be designed to give that selectivity.

If YES, please place real-valued pole(s) and zero(s) to achieve the frequency selectivity.

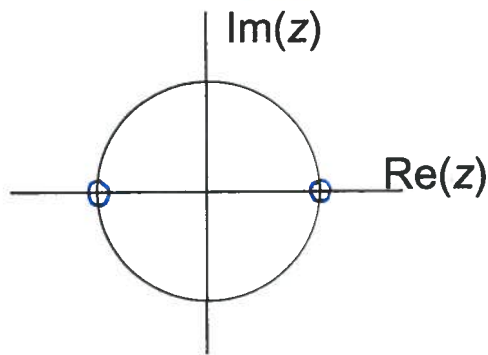
Lowpass: YES or NO



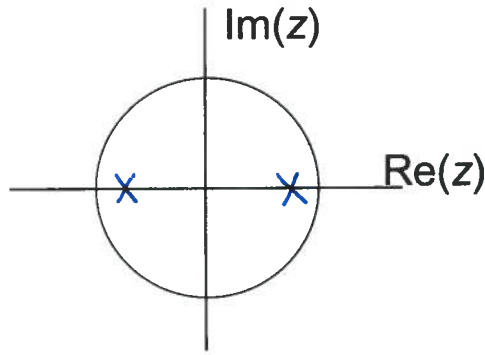
Highpass: YES or NO



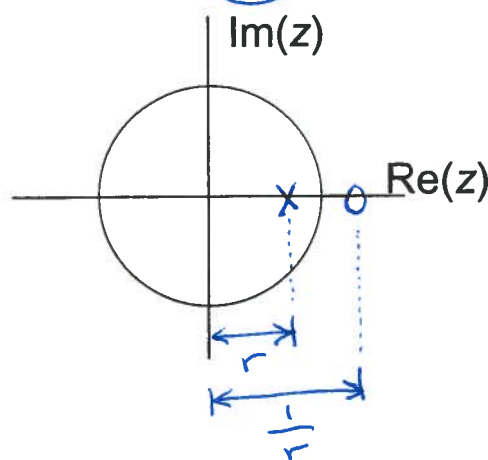
Bandpass: YES or NO



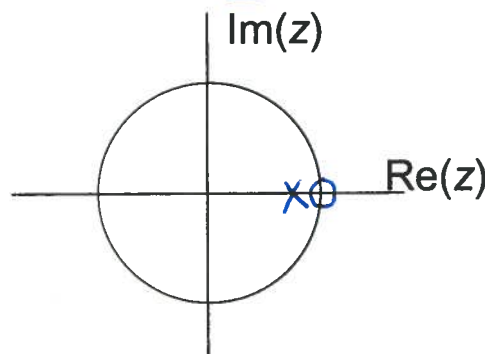
Bandstop: YES or NO



Allpass: YES or NO



Notch: YES or NO



Problem 1.4. Potpourri. 18 points.

Consider the design of a discrete-time linear time-invariant finite impulse response (FIR) filter by using the following steps: (1) design a discrete-time linear time-invariant infinite impulse response (IIR) filter to meet the design specification, and (2) truncate the impulse response of the IIR filter to a finite number of coefficients.

(a) How would you estimate the length of the FIR filter needed? 6 points.

Possible answer #1: FIR length = IIR coefficients = $2N+1$ for an N th-order IIR filter.

Possible answer #2: Estimate the FIR filter length by using a Kaiser window filter order estimator.

Possible answer #3: Keep enough of the IIR impulse response to contain 90% of total energy.

(b) If the FIR filter does not meet the design specification, how would you modify the design procedure to obtain an FIR filter of the same length that meets the design specification? 6 points.

Possible answer #1: Increase the IIR filter order.

Possible answer #2: Find the amount of stopband attenuation specification missed by FIR filter and add this to the specification entered for the IIR design and repeat steps (1) and (2).

(c) Claim: The FIR filter would always have linear phase. Either prove the claim to be true for all possible designs, or give a counterexample to show the claim is false. 6 points.

Counterexample: Let $h_{\text{IIR}}[n] = (0.9)^n u[n]$.

Keep first two samples:

$$h_{\text{FIR}}[n] = \delta[n] + 0.9 \delta[n-1]$$

Impulse response is not symmetric or anti-symmetric with respect to its midpoint.

Therefore, phase is not linear.

Claim is False.

Note: Truncating IIR impulse responses is a common method for modeling wireline/wired communication channels, which have IIR responses. We'll cover this in the Channel Impairments lecture.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 13, 2015

Course: EE 445S Evans

Name: _____ **Corneil & Bernie** _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Discrete-Time Filter Analysis
2	24		Discrete-Time Filter Design
3	24		Audio Effects System
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 28 points.

The Al-Alaoui Differentiator is a causal stable discrete-time linear time-invariant filter with a transfer function in the following form:

$$H(z) = C \frac{z-1}{z+\frac{1}{7}} = C \frac{1-z^{-1}}{1+\frac{1}{7}z^{-1}} = \frac{C - Cz^{-1}}{1 + \frac{1}{7}z^{-1}}$$

Constant C is real-valued and is not equal to zero.

- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 4 points.

IIR filter. There is a non-zero pole, and from part (b), the input-output difference equation requires the previous output value.

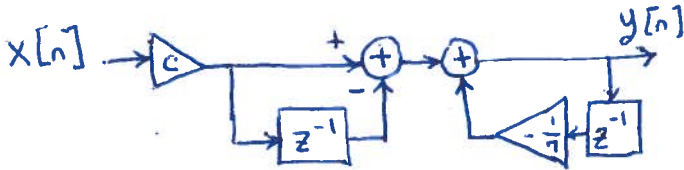
- (b) From the transfer function, give the difference equation governing the filter with input $x[n]$ and output $y[n]$. 4 points.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{C - Cz^{-1}}{1 + \frac{1}{7}z^{-1}} \Rightarrow (1 + \frac{1}{7}z^{-1})Y(z) = (C - Cz^{-1})X(z)$$

$$Y(z) = -\frac{1}{7}z^{-1}Y(z) + CX(z) - Cz^{-1}X(z)$$

$$y[n] = -\frac{1}{7}y[n-1] + Cx[n] - Cx[n-1]$$

- (c) Give a block diagram for the filter. 4 points.



- (d) What are the initial condition(s)? What value(s) should they be assigned and why? 4 points.

$$\text{Let } n=0: y[0] = -\frac{1}{7}y[-1] + Cx[0] - Cx[-1]$$

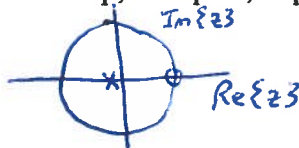
Initial conditions are $x[-1]$ and $y[-1]$. They should be set to zero

- (e) Find the equation for the frequency response of the filter. 4 points. to satisfy LTI properties.

Because the LTI system is stable, or equivalently because the region of convergence $|z| > \frac{1}{7}$ includes the unit circle,

$$H_{\text{freq}}(\omega) = H(z)|_{z=e^{j\omega}} = C \frac{1 - e^{-j\omega}}{1 + \frac{1}{7}e^{-j\omega}}$$

- (f) What is the best description of the frequency selectivity of the filter: lowpass, highpass, bandstop, bandpass, allpass or notch? Why? 4 points



Zero at $z=1$. Stopband centered at $\omega=0$.

Pole at $z=-\frac{1}{7}$. Little effect on magnitude response because it isn't close to unit circle.

Highpass

- (g) Find a numeric value for C that normalizes the magnitude response. 4 points.

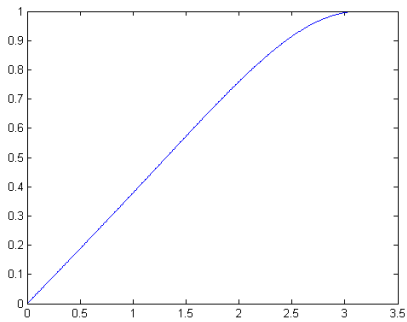
Since the filter is highpass, normalize the frequency response at $\omega=\pi$, which is $z=e^{j\omega}=-1$.

$$1 = H_{\text{freq}}(\pi) = C \frac{1 - (-1)}{1 + \frac{1}{7}(-1)} = C \frac{2}{\frac{6}{7}} \approx 1 \Rightarrow C = \frac{1}{2} \cdot \frac{6}{7} = \frac{3}{7}$$

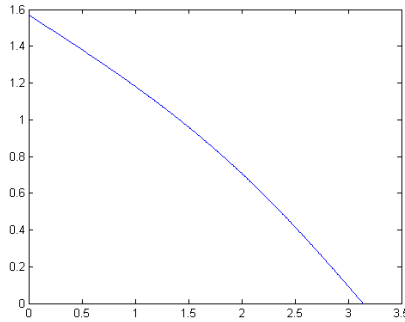
Problem 1.1 Al-Alaoui Differentiator (more information)

Using the following Matlab code,

```
C = 3/7;  
feedforwardCoeffs = C*[1 -1];  
feedbackCoeffs = [1 (1/7)];  
[H, w] = freqz(feedforwardCoeffs, feedbackCoeffs);  
figure(1); plot(w, abs(H));  
figure(2); plot(w, angle(H));
```



Magnitude Response (Linear Scale)



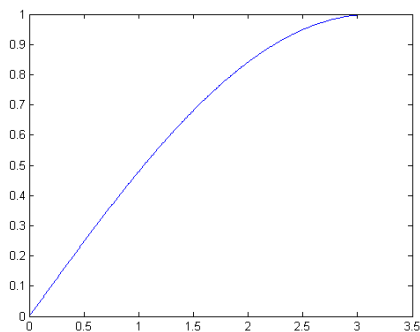
Phase Response (Radians)

The Al-Alaoui Differentiator was first reported in the following article:

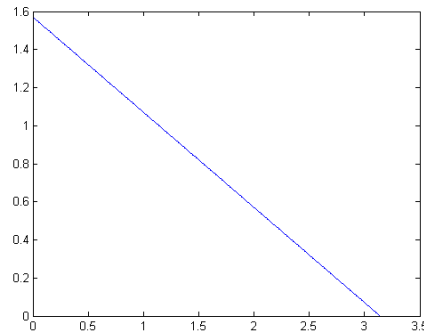
M. A. Al-Alaoui, "Novel Digital Integrator and Differentiator", *IEE Electronic Letters*, vol. 29, no. 4, Feb. 18, 1993.

Here are the magnitude and phase plots for the traditional differentiator:

```
C = 1/2;  
feedforwardCoeffs = C*[1 -1];  
[H, w] = freqz(feedforwardCoeffs);  
figure(1); plot(w, abs(H));  
figure(2); plot(w, angle(H));
```



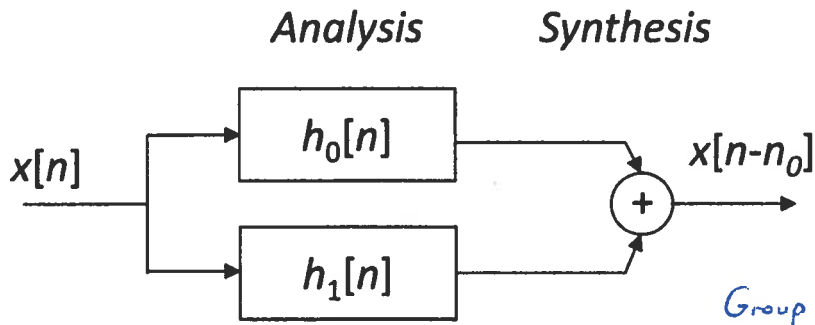
Magnitude Response (Linear Scale)



Phase Response (Radians)

Problem 1.2 Discrete-Time Filter Design. 24 points.

Consider the design of the two discrete-time filters shown below:



Group delay = n_0
 For a constant group delay, we need to use linear phase FIR filters.

Here, n_0 is a given constant integer delay in samples where $n_0 > 0$.

FIR filter #0 is lowpass with cutoff frequency of $\pi/2$ rad/sample.

FIR filter #1 is highpass with cutoff frequency of $\pi/2$ rad/sample.

At the cutoff/cuton frequency, which is in the transition band, the magnitude response is -6 dB.

The filters should have the highest stopband attenuation possible for their filter lengths.

(a) Give the filter specification and design method for FIR Filter #0 below. 9 points.

$$\omega_{\text{passband}} = \frac{\pi}{2} - \frac{\pi}{2n_0}$$

$$\omega_{\text{stopband}} = \frac{\pi}{2} + \frac{\pi}{2n_0}$$

$$\text{order} = 2n_0$$

design method = Parks-McClellan.

- Cutoff frequency is near midpoint of ω_{passband} and ω_{stopband} .
- We also want 10% roll off from passband to stopband

(b) Give the filter specification and design method for FIR Filter #1 below. 9 points.

$$\omega_{\text{stopband}} = \frac{\pi}{2} - \frac{\pi}{2n_0}$$

$$\omega_{\text{passband}} = \frac{\pi}{2} + \frac{\pi}{2n_0}$$

$$\text{order} = 2n_0$$

design method = Parks-McClellan

(c) How would you adjust the FIR filter coefficients to make sure that the output of the synthesis section is $x[n - n_0]$? 6 points.

$$\begin{aligned} \text{Synthesis output: } y[n] &= h_0[n] * x[n] + h_1[n] * x[n] \\ &= x[n - n_0] = (h_0[n] + h_1[n]) * x[n] \end{aligned}$$

Normalize energy in $h_0[n]$ to be $\frac{1}{2}$
 and $h_1[n]$ to be $\frac{1}{2}$.

Problem 1.3 Audio Effects System. 24 points.

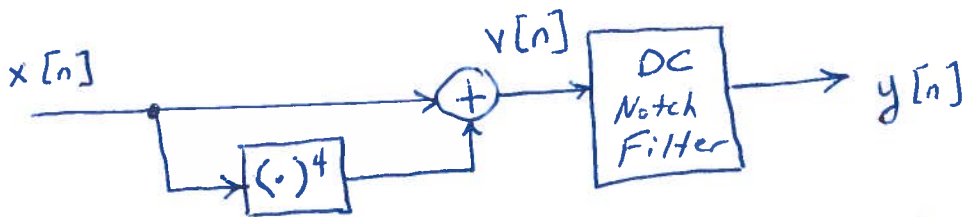
This problem asks you to design a discrete-time audio effects system that will

- Accept a sinusoidal signal representing a musical note as the input
- Output the input sinusoidal signal plus the same note from the next two highest octaves

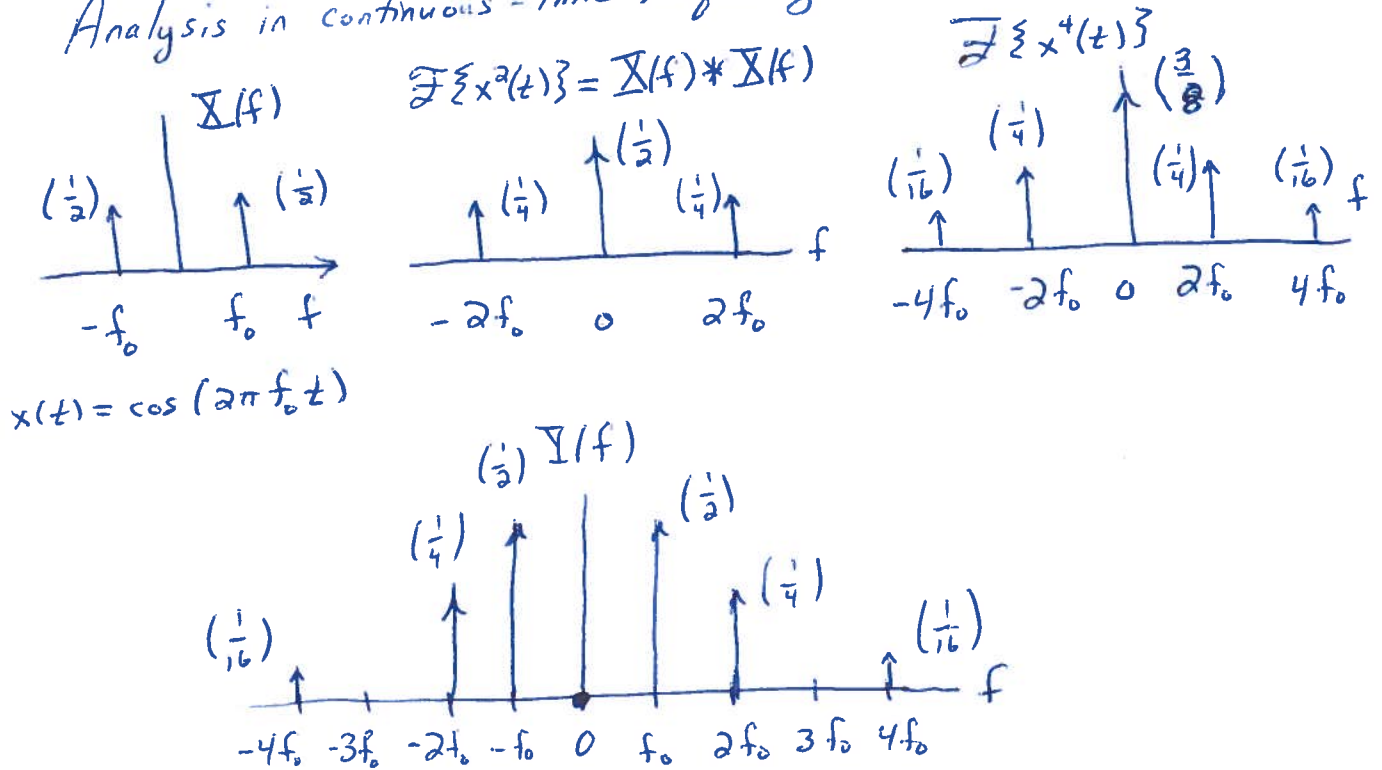
For example, if a 440 Hz note ('A' in the Western scale) is the input, then the output will be the 440 Hz note plus notes at 880 Hz and 1760 Hz.

Assume that the sampling rate f_s is 44.1 kHz.

(a) Draw a block diagram for your system. If your system generates a DC or zero-frequency component, please add a DC notch filter to your system. Sketch the Fourier transform of the output signal when the input is a sinusoidal signal. 18 points.



Analysis in continuous-time frequency domain



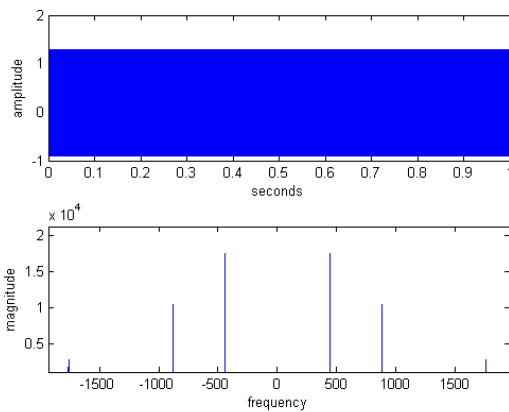
(b) How many multiplications per second does your audio effects system require? 6 points.

DC notch filter: $y[n] = 0.95y[n-1] + x[n] - x[n-1]$
 Fourth-order power block: square the square of input
 3 multiplications/sample
 $3 f_s$ samples/s

1.3 Audio Effects System

Here is the Matlab code to test the solution in part (a) for an input of a sinusoid at 440 Hz:

```
f0 = 440;  
fs = 44100;  
Ts = 1 / fs;  
time = 1;  
n = 1 : time*fs;  
w0 = 2*pi*f0/fs;  
x = cos(w0*n);  
v = x + x.^4;  
y = filter( [1 -1], [1 -0.95], v);  
plotspec(y, Ts);
```



The frequencies at f_0 , $2 f_0$ and $4 f_0$ are visible in the spectrum (bottom plot above).

One can playback the audio signal without and with DC removal. The Matlab command sound assumes that the vector of samples to be played is in the range of $[-1, 1]$ inclusive.

```
sound(v, fs);      %%% without DC removal
```

```
sound(y, fs);     %%% with DC removal
```

To avoid clipping of amplitude values outside of the range $[-1, 1]$, try

```
sound(v / max(abs(v)), fs);      %%% without DC removal
```

```
sound(y / max(abs(y)), fs);     %%% with DC removal
```

To hear the effect of DC offset, try

```
sound(v + 10, fs);
```

Problem 1.4. Potpourri. 24 points.

This problem will explore the design tradeoffs in working with a block of samples instead each sample individually.

- (a) What is the advantage of moving blocks of samples from off-chip to on-chip instead of moving one sample at a time? 6 points.

Increased throughput. Block transferred at one time - avoids overhead of reading each sample separately (bus arbitration, interrupt service routine).

- (b) What is the disadvantage of moving blocks of samples from off-chip to on-chip instead of moving one sample at a time? 6 points.

Increased latency for first sample to be processed. We have to read entire block of samples before first sample can be processed.

- (c) Name and describe the subsystem on the TI TM320C6748 digital signal processor used in the laboratory component that moves blocks of samples from off-chip to on-chip? 6 points.

Enhanced direct memory access (EDMA) controller. EDMA handles input/output off-chip/on-chip without using the CPU core. EDMA can support multiple channels at same time. EDMA can also reformat data.

- (d) When implementing convolution between a finite-length signal stored on chip and a signal streaming into the processor from off chip, we encounter an issue when moving from one block of streaming samples to the next. Briefly describe what this issue is (why it exists) and how to correct for it. 6 points.

$h[n]$ finite length



$x[n]$ infinite length broken into blocks



- When block #2 is read, block #1 is overwritten (normally).
- However, convolving $h[n]$ with block #2 requires the last samples of block #1 due to flip-and-slide.
- If $h[n]$ has N samples, we need to keep $N-1$ samples of the previous block.
- We can implement this by using a circular buffer of length $N-1 + \text{BlockSize}$, and each new block overwrites the oldest samples.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 16, 2015

Course: EE 445S Evans

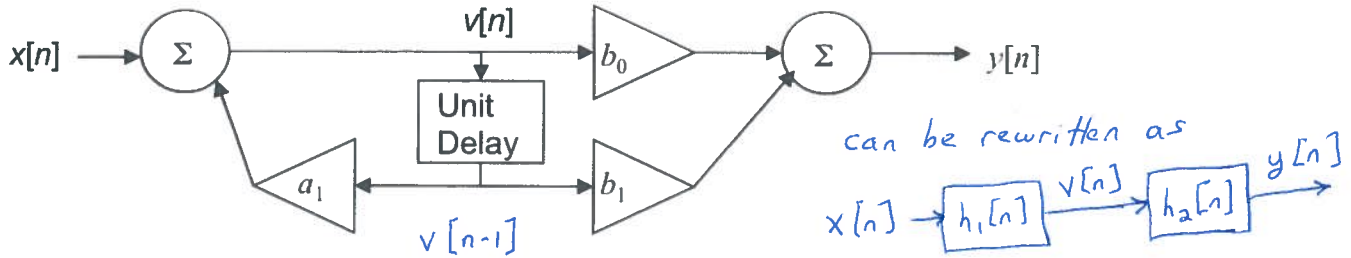
Name: Teen Titans Go!
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Discrete-Time Filter Analysis
2	27		Discrete-Time Filter Design
3	27		Modulation and Demodulation
4	18		Potpourri
<i>Total</i>	100		

Problem 1.1 Discrete-Time Filter Analysis. 28 points.

A causal stable discrete-time linear time-invariant filter with input signal $x[n]$ and output signal $y[n]$ is described by the following block diagram:



- (a) Is this a finite impulse response filter or an infinite impulse response filter? Why? 4 points.

IIR. There are feedback paths to the input. Also, in part (e), the transfer function in the z-domain has non-trivial (non-zero) poles.

- (b) From the block diagram, give the difference equation relating input signal $x[n]$ and the intermediate signal $v[n]$. 4 points.

$$v[n] = x[n] + a_1 v[n-1]$$

- (c) From the block diagram, give the difference equation relating the intermediate signal $v[n]$ and the output signal $y[n]$. 4 points.

$$y[n] = b_0 v[n] + b_1 v[n-1]$$

- (d) What are the initial condition(s)? What value(s) should they be assigned and why? 4 points.

Let $n=0$: $v[0] = x[0] + a_1 v[-1]$ Initial condition is $v[-1]$.
 $y[0] = b_0 v[0] + b_1 v[-1]$ $v[-1] = 0$ to ensure LTI property.

- (e) Based on your answer in parts (b), (c) and (d), derive the transfer function in the z-domain of the filter given in part (a). 4 points.

From (b), $V(z) = X(z) + a_1 z^{-1} V(z)$ From (c), $Y(z) = b_0 V(z) + b_1 z^{-1} V(z)$

$$H_1(z) = \frac{V(z)}{X(z)} = \frac{1}{1 - a_1 z^{-1}}$$

$$H_2(z) = \frac{Y(z)}{V(z)} = b_0 + b_1 z^{-1}$$

- (f) Based on your answer in (e), give the frequency response of the filter. 4 points.

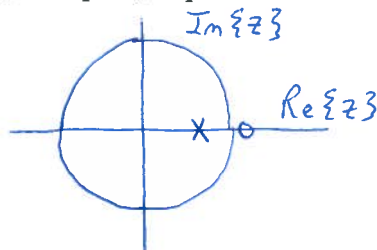
System is (BIBO) stable: $H_f(\omega) = H(e^{j\omega}) = \frac{b_0 + b_1 e^{-j\omega}}{1 - a_1 e^{-j\omega}}$ \downarrow ROC: $|z| > |a_1|$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{Y(z)}{V(z)} \cdot \frac{V(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}}$$

- (g) For $a_1 = 0.8$, $b_0 = 1$, and $b_1 = -1.05$, what is the best description of the frequency selectivity of the filter: lowpass, highpass, bandstop, bandpass, allpass or notch? Why? 4 points

zero at $z = -\frac{b_1}{b_0} = 1.05$

pole at $z = a_1 = 0.8$

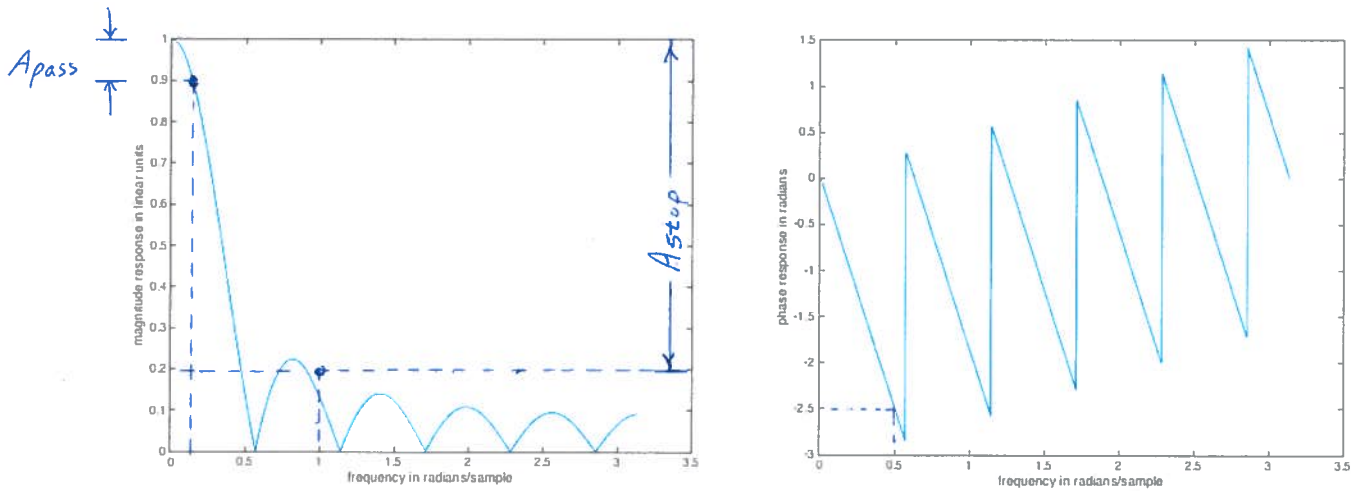


If notch, zero at $z = 1$.
 If allpass, zero at $z = \frac{1}{0.8} = 1.25$.
 Highpass filter.

Note: Not all FIR filters exhibit linear phase. Consider $h[n] = \delta[n] + 2\delta[n-1]$.

Problem 1.2 Discrete-Time Filter Design. 27 points.

While searching online, you find the following magnitude and phase responses vs. rad/sample for a finite impulse response (FIR) filter. Magnitude response is in linear units (not decibels).



(a) What is the order of the FIR filter? Why? 6 points.

Magnitude response has five zeros on the unit circle in positive frequencies. Five more are on the unit circle for negative frequencies. At least 10th order, because there could be zeros not on the unit circle. Phase response has a

(b) Is the phase response linear? Why or why not? 6 points.

Yes. The phase response has a constant slope of -5 except at frequencies that are zeroed out in mag. response and don't pass. For a linear phase FIR filter, the negated slope is the group delay, which is order/2.

(c) From the plots, infer the magnitude response specification for the filter. 12 points. Order is 10.

i. Passband frequency (in rad/sample)

$$\omega_{pass} = 0.2 \text{ rad/sample}$$

ii. Stopband frequency (in rad/sample)

$$\omega_{stop} = 1.0 \text{ rad/sample}$$

iii. Passband ripple (in dB)

$$A_{pass} = 20 \log_{10}(1) - 20 \log_{10}(0.9) = 0.9151 \text{ dB}$$

iv. Stopband attenuation (in dB)

$$A_{stop} = 20 \log_{10}(1) - 20 \log_{10}(0.2) = 13.98 \text{ dB}$$

Note: There are many possible answers here.

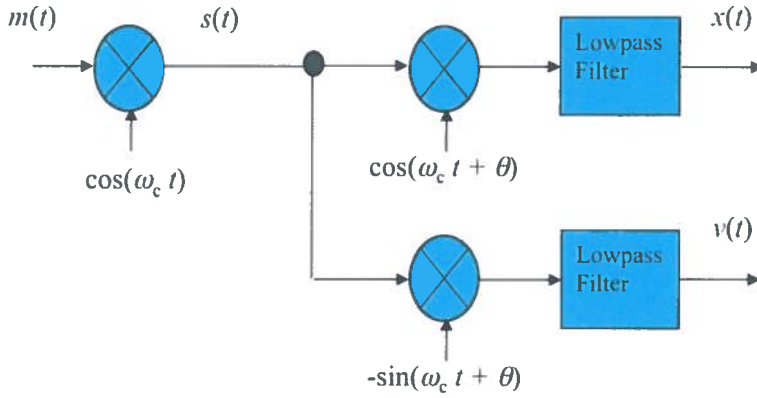
(d) Consider designing an infinite impulse response (IIR) filter to meet the magnitude response specification in part (c). Give an IIR filter design method that would give the fewest coefficients and still meet the magnitude response specification. 3 points.

Answer #1: Elliptic

Answer #2: Chebyshev Type II if we want to match the monotonic decreasing magnitude response in the passband and ripple in the stopband. Elliptic design would ripple in both bands.

Problem 1.3 Modulation and Demodulation. 27 points.

Consider sinusoidal amplitude modulation using the cosine, and sinusoidal amplitude demodulation using both the cosine and sine, as shown below. The phase offset between the receiver and transmitter is given by θ .



Trig formulas (JSK p. 404)

$$\cos(x)\cos(y) = \frac{1}{2}\cos(x-y) + \frac{1}{2}\cos(x+y)$$

$$\sin(x)\cos(y) = \frac{1}{2}\sin(x-y) + \frac{1}{2}\sin(x+y)$$

Note: This problem gives a preview of phase locked loops (PLLs).

Assume the lowpass filters are ideal and have a gain of 2. Please ignore the delay through the filters.

(a) When $\theta = 0$, derive formulas for $x(t)$ and $v(t)$ in terms of the message signal $m(t)$. 9 points.

$$\begin{aligned} x(t) &= \text{LPF} \{ m(t) \cos(\omega_c t) \cos(\omega_c t + \theta) \} \\ &= \text{LPF} \left\{ \frac{1}{2} m(t) (\cos(-\theta) + \cos(2\omega_c t + \theta)) \right\} = \cos(\theta) m(t) \end{aligned}$$

$$\begin{aligned} v(t) &= \text{LPF} \{ m(t) \cos(\omega_c t) (-\sin(\omega_c t + \theta)) \} \\ &= \text{LPF} \left\{ -\frac{1}{2} m(t) (\sin(\theta) + \sin(2\omega_c t + \theta)) \right\} = -\sin(\theta) m(t) \end{aligned}$$

When $\theta = 0$, $x(t) = m(t)$ and $v(t) = 0$.

(b) When $\theta = \pi/2$, derive formulas for $x(t)$ and $v(t)$ in terms of the message signal $m(t)$. 9 points.

When $\theta = \frac{\pi}{2}$, $x(t) = 0$ and $v(t) = -m(t)$.

Note: $\theta = \theta_{\text{transmitter}} + \theta_{\text{receiver}}$. We know θ_{receiver} , but we don't know $\theta_{\text{transmitter}}$ at the receiver.

(c) Describe an algorithm to adjust θ . 9 points.

We would like to find θ that maximizes power in $x(t)$.
or θ that minimizes power in $v(t)$.

Algorithm #1: Try several values for θ and keep the one that gives the largest [smallest] power in $x(t)$ [$v(t)$].

Algorithm #2: Pick an initial guess for θ and then adapt its value using steepest ascent (descent) to maximize (minimize) power in $x(t)$ ($v(t)$).

Algorithm #3: Use a Costas loop by multiplying $x(t)$ and $v(t)$. JSK p. 208.

Problem 1.4. Potpourri. 18 points.

(a) Describe design tradeoffs in signal quality vs. implementation complexity for the following two linear phase finite impulse response (FIR) filter designs. 9 points.

	Linear Phase FIR Filter Design		
	100 coefficients	1000 coefficients	Increase by Factor of
Multiplications/sample	100	1000	10x
Memory size in words	200	2000	10x
Group delay in samples	49.5	499.5	~10x

$\frac{N}{2}$
 $\frac{N-1}{2}$

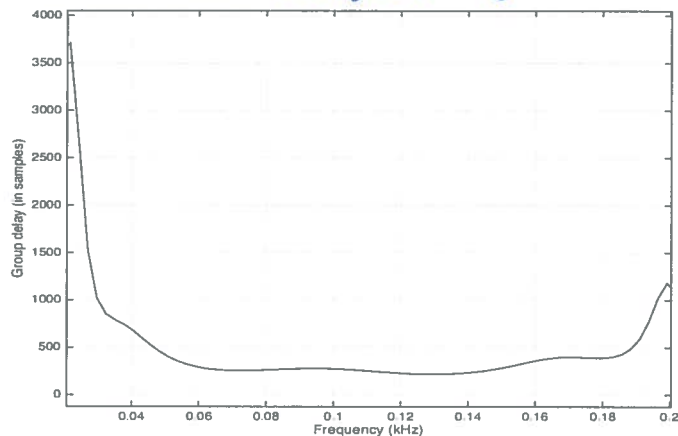
(b) Two real-time digital bandpass filter designs are being considered for an audio speaker.

- 9405th-order linear phase finite impulse response (FIR) filter
- 12th-order infinite impulse response (IIR) filter

The group delay for the IIR filter design over passband frequencies is plotted below.

Which filter design would you advocate using? Why? 9 points.

Note: Sub-wolfer design.



Group delay is 4702.5 samples for the FIR filter or 106.6 ms at an audio CD sampling rate (44.1 kHz). This is too long. The IIR filter has a group delay of 1000-3700 samples for frequencies between 20 Hz and 30 Hz, and a group delay of 250-500 samples for most of the pass band. 500 samples would mean 11.3 ms of delay at an audio CD sampling rate. Good.

Phase distortion is an important consideration in the design of an audio system. The human auditory system will in general be able to cancel out the same mild distortion in multi-channel audio. This FIR filter has linear phase over all frequencies, whereas the IIR filter has approximate linear phase over most of the passband.

Implementation complexity includes 9406 multiplications/sample (FIR filter) and 25 multiplications/sample (IIR filter).

Advocate using an IIR filter.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 11, 2016

Course: EE 445S Evans

Name: The Little Prince
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

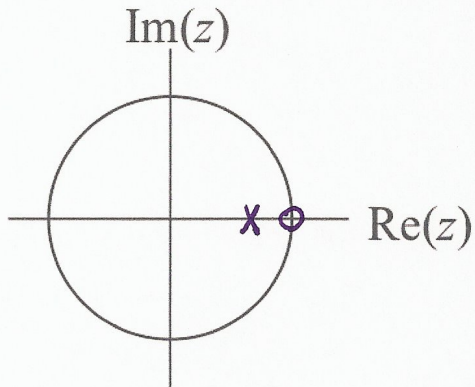
<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Design & Analysis
2	24		BIBO Stability
3	24		Upsampling
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Filter Design & Analysis. 28 points.

Design and analyze a first-order discrete-time infinite impulse response filter to remove DC, which is a discrete-time frequency of 0 rad/sample.

(a) Place the pole and zero on the pole-zero diagram below. 4 points

HW 3.1(c)



IIR notch filter to remove DC and keep as much of the other frequencies as possible.

(b) Give numeric values of the pole and zero in part (a). Why did you choose these values? 4 points

HW 3.1(c)

Pole is at $p_0 = 0.9$. Zero is at $z_0 = 1$.

A zero at $z_0 = 1$ will zero out the frequency at $\omega_0 = 0$ rad/sample because $z_0 = e^{j\omega_0} = e^{j0} = 1$. A pole at the same angle will pass as

(c) Give a formula for the discrete-time frequency response and draw the magnitude response. 6 points.

$$H(z) = C \frac{1 - z_0 z^{-1}}{1 - p_0 z^{-1}}$$

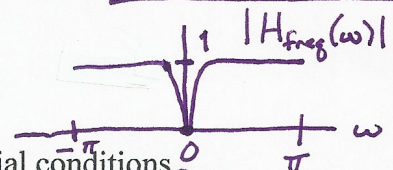
With $|p_0| < 1$, system is bounded-input bounded-output stable.

HW 2.1

other frequencies as possible.

$$H_{freq}(\omega) = C \frac{1 - z_0 e^{-j\omega}}{1 - p_0 e^{-j\omega}}$$

Normalize frequency response at $\omega = \pi$: $C = \frac{1.9}{2}$.



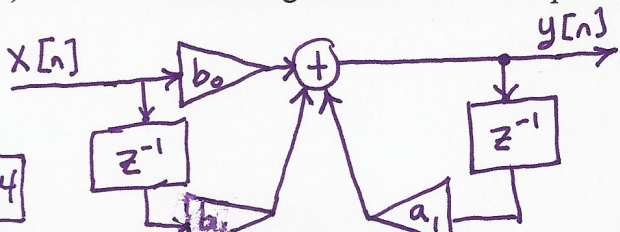
(d) Give the difference equation relating output $y[n]$ and input $x[n]$ including the initial conditions. 4 points

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \Rightarrow (1 - a_1 z^{-1})Y(z) = (b_0 + b_1 z^{-1})X(z)$$

$$(1 - a_1 z^{-1})Y(z) = (b_0 + b_1 z^{-1})X(z)$$

HW 0.4

(e) Draw the block diagram for the filter. 4 points.



HW 0.4

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

quality; and (3) DC removal prevents clipping of amplitude when going through D/A conversion.

with $x[-1] = 0$ & $y[-1] = 0$ and $a_1 = p_0$, $b_0 = C$, $b_1 = -C z_0$

HW 1.3

(f) Why is removing the DC offset (average value) important in speech and audio systems? 6 points.

When speech/audio is sampled, signal has little or no DC value. Humans cannot hear DC - normal hearing is 20Hz to 20kHz. Content from 0Hz to 20Hz would be noise/interference. (1) Removing DC offset improves signal-to-noise ratio; (2) For playback, speech/audio goes through D/A converter and needs to be converted to an integer format - DC offset uses many bits. DC removal improves

Problem 1.2 Stability. 24 points.

For a discrete-time linear time-invariant system with impulse response $h[n]$, the system is bounded-input bounded-output (BIBO) stable if and only if

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

HW 2.1

- (a) Using the above definition, prove that a discrete-time finite impulse response (FIR) filter is always BIBO stable. 12 points.

An FIR filter has a finite number of non-zero coefficients.

$$\sum_{n=0}^{N-1} |h[n]| \leq NB < \infty$$

where $|h[n]| \leq B < \infty$ and N is number of FIR coefficients.

In other words, a finite sum of finite values must be finite.

HW 0.2

HW 0.4

HW 1.1

HW 2.1

- (b) Give an example of a BIBO unstable linear time-invariant system, an application that uses the BIBO unstable system, and how the application uses the BIBO unstable system. 12 points.

Example #1: Sinusoidal generator (problem 1.4 on this midterm).

Sinusoidal amplitude modulation multiplies a baseband signal by a sinusoidal signal to shift the center frequency of the baseband signal to the (carrier) frequency of the sinusoid.

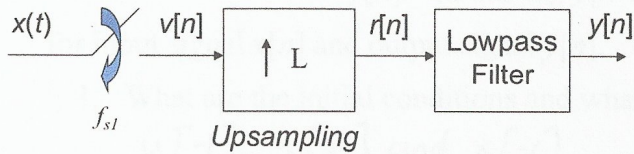
Example #2: Discrete-time integrator - $y[n] = y[n-1] + x[n]$

with $y[-1] = 0$. Can be used to keep a running sum of temperature or signal power (i.e. signal amplitude squared).

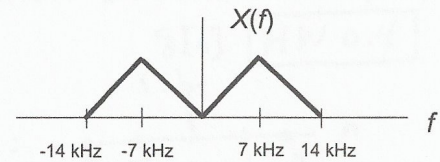
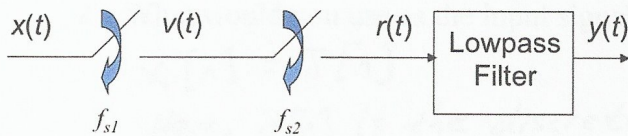
Running sum of signal power is total signal energy. Can also be used to create audio effects.

Problem 1.3 Upsampling. 24 points.

Consider the following block diagram to sample an audio signal $x(t)$ at $f_{s1} = 32$ kHz to produce a discrete-time signal $v[n]$ and then change the sampling rate to $f_{s2} = 96$ kHz via the discrete-time operation of upsampling by 3 (i.e. $L = 3$) followed by a discrete-time lowpass filter to produce $y[n]$.



The continuous-time equivalent to the above block diagram is

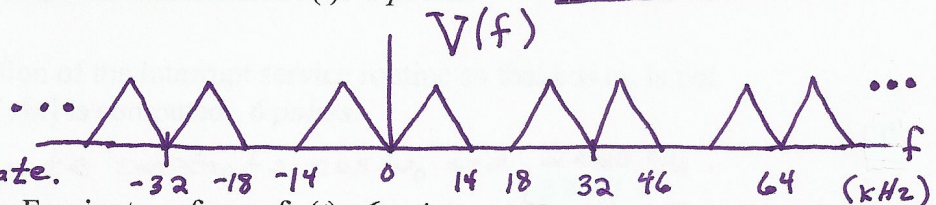


Using $X(f)$ given above to the right, which is the continuous-time Fourier transform of $x(t)$, please complete the following analysis:

- (a) Draw $V(f)$, which is the continuous-time Fourier transform of $v(t)$. 6 points.

HW 0.3 HW 2.2

Sampling causes a periodic replication of the input spectrum at offsets that are multiples of the sampling rate.



- (b) Draw $R(f)$, which is the continuous-time Fourier transform of $r(t)$. 6 points.

$R(f)$ will contain $V(f)$ plus versions of $V(f)$ shifted left and right by multiples of 96 kHz. $R(f)$ will be equal to $V(f)$ except that the amplitude will be scaled by $\frac{1}{3}$.

HW 0.3

- (c) Give the passband and stopband frequencies in Hz for the continuous-time lowpass filter to use to recover $x(t)$ from $r(t)$. 4 points.

$f_{pass} = 14 \text{ kHz}$

$f_{stop} = 16 \text{ kHz} \leftarrow \text{or } 15.4 \text{ kHz to hit } 10\% \text{ rolloff exactly.}$

HW 0.2
HW 2.2

- (d) Give the passband and stopband frequencies in rad/sample for the discrete-time lowpass filter in the upper block diagram to recover $v[n]$ from $r[n]$. 4 points.

$\omega_{pass} = 2\pi \frac{14 \text{ kHz}}{96 \text{ kHz}} = 2\pi \frac{f_{pass}}{f_{s2}}$

$\omega_{stop} = 2\pi \frac{16 \text{ kHz}}{96 \text{ kHz}} = 2\pi \frac{f_{stop}}{f_{s2}}$

HW 3.1

- (e) For the discrete-time lowpass filter, would you advocate to use a finite impulse response (FIR) filter or an infinite impulse response (IIR) filter? Why? 4 points.

HW 3.3(d)

For audio processing, both linear phase and low group delay (~ 0 ms) matter.
 FIR filter: linear phase over all frequencies but high group delay.
 IIR filter: near linear phase in passbands and low group delay.
 Advocate IIR filter design. See Fall 2015 Midterm #1 Problem 1.4(b).

Problem 1.4. Potpourri. 24 points.

(a) In lab #2, you implemented a cosine generator on the digital signal processing board in lab using a causal linear time-invariant filter with the difference equation

$$y[n] = (2 \cos \omega_0) y[n-1] - y[n-2] + x[n] - (\cos \omega_0) x[n-1]$$

for input signal $x[n]$ and output signal $y[n]$.

1. What are the initial conditions and what should their values be? 3 points.

HW 0.4

$y[-1], y[-2]$ and $x[-1]$.

All should be set to zero to satisfy linear time-invariant properties.

2. What would you use as the input signal $x[n]$? 3 points.

HW 0.4

$x[n] = \delta[n]$

Here, $\delta[n]$ is the discrete-time impulse:



3. Give the output signal $y[n]$ that is a solution to the above difference equation for the input signal $x[n]$ given in part 2 above? 3 points.

HW 0.4

$y[n] = \cos(\omega_0 n) u[n]$

4. Describe an efficient implementation of the interrupt service routine so that $\cos \omega_0$ is not computed every time a sample of $y[n]$ is computed. 6 points.

Approach #1: Pre-compute constants $\cos \omega_0$ and $2 \cos \omega_0$.

Approach #2: Compute $\cos \omega_0$ and $2 \cos \omega_0$ on first ISR call only.

Approach #3: Use a DMA approach to compute a buffer of values.

Approach #4: Use a lookup table to pre-compute 1 period of cosine.

(b) In lab #3, you implemented discrete-time infinite impulse response (IIR) filters in 32-bit IEEE floating-point data and coefficients on the digital signal processor board. What distortion could the **continuous-time output signal** of the digital-to-analog converter have if you were to implement the discrete-time IIR filter as a cascade of biquads but not implement the gain for each biquad? 9 points.

HW 3.3

The IIR filter output is in 32-bit IEEE floating point, which needs to be converted to a 16-bit short integer (signed) to send to the digital-to-analog (D/A) converter.

The conversion to 16-bit short integer assumes that the floating-point number is in the range of $[-1, 1]$, e.g.

$$\text{short16value} = (\text{short int})(\text{float32value} * 32768);$$

Case #1: Biquad gain product < 1 . Floating point values will go out of range $[-1, 1]$ and clipping will occur.

Case #2: Biquad gain product > 1 . Floating point values will be small compared to 1 in absolute value. Loss of quality/range.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 14, 2016

Course: EE 445S Evans

Name: _____ **Spirou et Fantasio** _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Sampling
3	27		Filter Design
4	21		Potpourri
<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

A discrete-time linear time-invariant (LTI) filter is described by the following transfer function

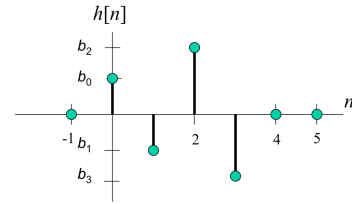
$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}$$

where b_0, b_1, b_2 and b_3 are the filter coefficients.

(a) Give a formula in discrete time for the impulse response $h[n]$.

Plot $h[n]$. 3 points. [See Lecture slide 5-9; Homework 1.1 & 2.1]

$$h[n] = b_0 \delta[n] + b_1 \delta[n - 1] + b_2 \delta[n - 2] + b_3 \delta[n - 3]$$



(b) Is the filter a finite impulse response (FIR) or infinite impulse response (IIR) filter? Why? 3 points.

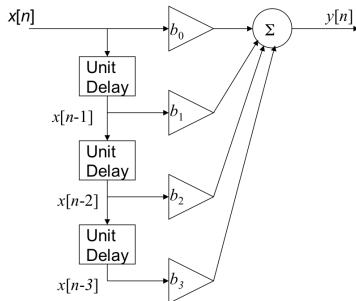
FIR filter. There are no poles in the transfer function other than artificial poles at the origin. Also, from the answer to part (c) below, the output only depends on current and previous input values— there is no feedback. [See Lectures 5&6; Homeworks 1,2&3; Lab 3]

(c) Give a formula in discrete time for the output $y[n]$ in terms of the input $x[n]$ including the initial conditions. 3 points. [See Lecture slides 3-9, 5-4, 5-6 & 5-11; Homeworks 1&2; Lab 3]

$$y[n] = b_0 x[n] + b_1 x[n - 1] + b_2 x[n - 2] + b_3 x[n - 3]$$

Initial conditions must be zero to satisfy LTI properties: $x[-1] = x[-2] = x[-3] = 0$

(d) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



[See Lecture slides 3-15 & 5-4]

(e) Give a formula for the discrete-time frequency response of the filter. 3 points. [See Lecture slide 5-11, 5-12, 5-13, 5-14, 5-17 & 5-18; Homework 2.1]

Substituting $z = \exp(j \omega)$ into $H(z)$ is valid because the region of convergence of $H(z)$ is $z \neq 0$ which includes the unit circle. Another justification is that all FIR filters are BIBO stable.

$$H_{freq}(\omega) = H(e^{j\omega}) = b_0 + b_1 e^{-j\omega} + b_2 e^{-2j\omega} + b_3 e^{-3j\omega}$$

(f) Give all possible conditions on the coefficients for the filter to have constant group delay. 6 points. [See Lecture slides 5-15, 5-17 & 5-18; Homework 1.3, 2.1, 2.3 & 3.2]

1. Filter coefficients are even symmetric w/r to their midpoint, i.e. $b_0 = b_3$ and $b_1 = b_2$
2. Filter coefficients are odd symmetric w/r to their midpoint, i.e. $b_0 = -b_3$ and $b_1 = -b_2$

(g) Using only values of +1 and -1, give values for the filter coefficients for a lowpass magnitude response. 4 points [See Lecture slides 3-8, 3-14, 5-15, 5-17 & 5-18; Homework 2.1(a)]

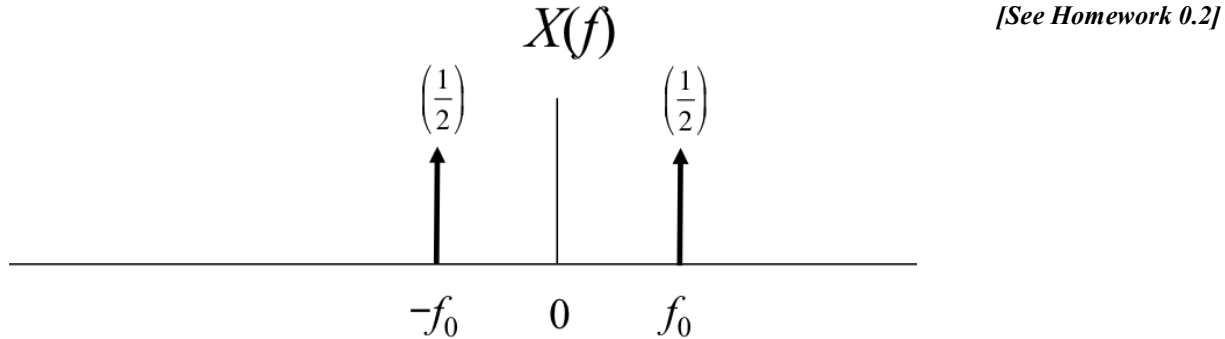
1. Filter coefficients are all +1. Gives an averaging filter scaled by 4. Lowpass. –OR–
2. Filter coefficients are all -1. Gives an averaging filter scaled by -4. Lowpass.

Problem 1.2 Sampling. 24 points.

Consider a two-sided continuous-time cosine signal with frequency f_0 in Hz given by

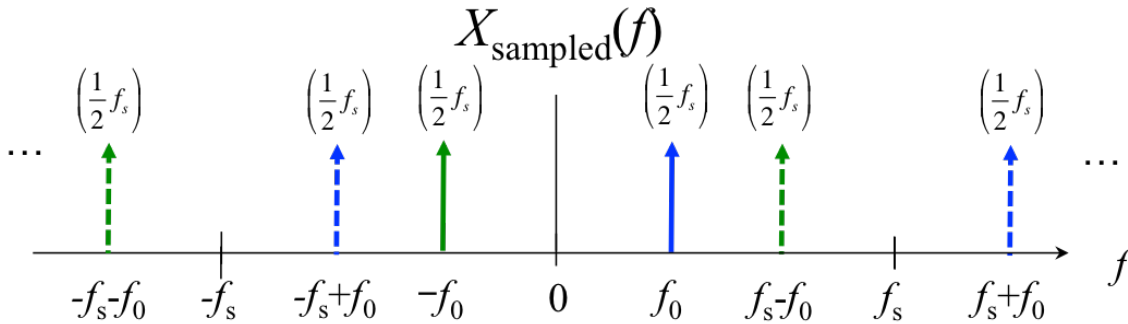
$$x(t) = \cos(2\pi f_0 t)$$

- (a) Plot the continuous-time Fourier transform of $x(t)$. 6 points. $X(f) = \frac{1}{2}\delta(f + f_0) + \frac{1}{2}\delta(f - f_0)$



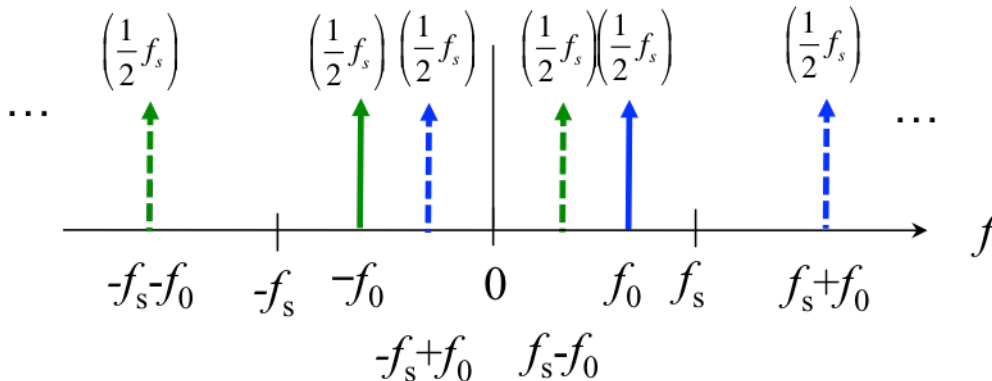
- (b) Plot the continuous-time Fourier transform of the result of sampling $x(t)$ at a sampling rate of f_s assuming that the Sampling Theorem has been satisfied, i.e. $f_s > 2f_0$. 6 points.

$$X_{\text{sampled}}(f) = f_s (\dots + X(f + f_s) + X(f) + X(f - f_s) + \dots)$$



[See Lecture slide 4-5; Lecture 4 demos; Homework 0.3]

- (c) Plot the continuous-time Fourier transform of $x(t)$ sampled at a sampling rate of f_s assuming that $f_0 < f_s < 2f_0$, which would not satisfy the Sampling Theorem. 6 points. Let $f_s = 1.5f_0$ below.



[See Lecture slides 4-4, 4-5 & 4-7; Lecture 4 demos; Homework 0.3]

- (d) In part (c), give a formula for the continuous-time frequency that would result after trying to reconstruct $x(t)$ from its sampled version. 6 points.

Sampling theorem says $f_s > 2f_{\text{max}}$ or equivalently $f_{\text{max}} < \frac{1}{2}f_s$. Apparent frequency is $f_s - f_0$.

Alternate response: Reconstruction applies a lowpass filter that passes frequencies from $-\frac{1}{2}f_s$ to $\frac{1}{2}f_s$ which means that the frequency that would result is $f_s - f_0$.

Problem 1.3 Filter Design. 27 points.

People suffering from tinnitus, or ringing of the ears, hear a tone in their ears even when the environment is quiet. The tone is generally at a fixed frequency in Hz, denoted as f_c .

Filtering music to remove as much as possible an octave of frequencies from f_1 to f_2 that contains f_c as its center frequency can provide relief of tinnitus symptoms.

This problem will ask you to design a sixth-order discrete-time infinite impulse response (IIR) filter to remove the octave of frequencies. The **sampling rate** is f_s where $f_s > 4 f_2$.

[See Lecture 6 in-class discussion on tinnitus; Homework 1.2(c) solution for discussion of octaves]

- (a) Give formulas for f_1 and f_2 in terms of f_c given that $f_c = \frac{1}{2} (f_1 + f_2)$. 6 points.

To cover an octave of frequencies, $f_2 = 2 f_1$.

Coupled with $f_c = \frac{1}{2} (f_1 + f_2)$, we have $f_1 = (2/3) f_c$ and $f_2 = (4/3) f_c$

- (b) Give formulas for discrete-time frequencies ω_1 , ω_c and ω_2 that correspond to continuous-time frequencies f_1, f_c and f_2 , respectively. 3 points. [See Lecture slide 1-10; Homework 0.4]

$$\omega_1 = 2\pi \frac{f_1}{f_s} \text{ and } \omega_c = 2\pi \frac{f_c}{f_s} \text{ and } \omega_2 = 2\pi \frac{f_2}{f_s}$$

- (c) Give formulas in terms of ω_1 , ω_c and ω_2 for the pole and zero locations for the sixth-order discrete-time IIR filter. Assume that the gain is one. 12 points. [See Lecture 6-6, 6-7 & 6-8 slides; Lab 3]

A sixth-order discrete-time IIR filter has six poles and six zeros. Here, the gain C is 1. Zeros have to be real-valued or occur in conjugate symmetric pairs. Same with the poles.

Filter should attenuate frequencies between ω_1 and ω_2 as well as between $-\omega_2$ and $-\omega_1$.

Bandstop filter. Zeros on or near the unit circle indicate the stopband.

Poles inside and near the unit circle indicate the passband(s).

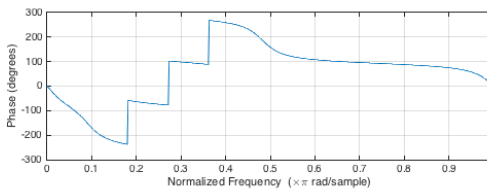
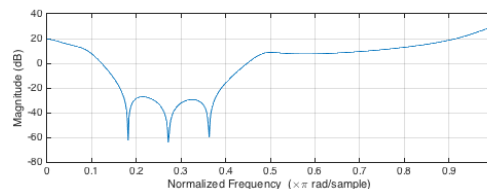
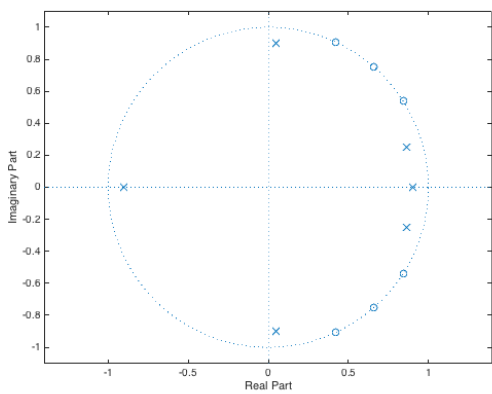
Zeros would be at frequencies ω_1 , ω_c and ω_2 as well as their negative values.

Because $f_s > 4 f_2$, ω_2 will be between 0 and $\pi/2$.

Zeros: $e^{j\omega_2}, e^{j\omega_c}, e^{j\omega_1}, e^{-j\omega_1}, e^{-j\omega_c}, e^{-j\omega_2}$

Poles: $re^{j(\frac{1}{2})\omega_1}, r, re^{-j(\frac{1}{2})\omega_1}, re^{-j(\frac{4}{3})\omega_2}, -r$ and $re^{j(\frac{4}{3})\omega_2}$ where $r = 0.9$

- (d) Draw the pole-zero diagram. 6 points. Example plotted for $f_c = 6000$ Hz and $f_s = 44100$ Hz.



[Lecture 5-13, 5-14, 6-6, 6-7, 6-19, 6-20 & 6-21 slides; Lecture 6 demos; Homework 3.1&3.3; Lab 3]

Problem 1.4. Potpourri. 21 points.

(a) Oversampling generally gives a higher signal quality but at a higher implementation complexity. If we increase the sampling rate by a factor of K , analyze the increase in implementation complexity for finite impulse response (FIR) filtering in terms of

1. Multiplication operations per second. 3 points. [See Lecture slide 5-4; Reader Handout N; Lab 3]

FIR filter of N coefficients requires N multiplication operations to compute one output sample given a new input sample. Filter runs at the sampling rate f_s and hence computes $N f_s$ multiplications/s. If the sampling rate is increased by a factor of K , then the multiplication operations will increase by a factor of K .

2. Memory reads per second. 3 points. [See Lecture slides 5-4 & 5-24; Reader Handout N; Lab 3]

FIR filter must read N coefficients and N current/previous input values in computing one output sample. Filter runs at the sampling rate f_s and hence reads $2 N f_s$ words/s. If the sampling rate is increased by a factor of K , then the memory reads per second will increase by a factor of K .

(b) In lab #2, you implemented a cosine generator on the digital signal processing board in lab using a lookup table to store one period of values for [See Lecture slides 1-10 to 1-16; Homework 0.4; Lab 2]

$$x[n] = \cos(\omega_0 n)$$

1. Assuming the sampling theorem has been satisfied, i.e. $f_s > 2 f_0$, give the range of values that ω_0 can take. Please be sure to include negative, zero and positive frequencies. 3 points.

$$\omega_0 = 2\pi \frac{f_0}{f_s} \text{ and } f_0 < \frac{1}{2} f_s \text{ which means that } -\pi < \omega_0 < \pi$$

Note: The discrete-time frequency domain is periodic with period 2π .

2. What is the discrete-time period in samples for $x[n]$? 3 points.

$$\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{N}{L} \text{ where } N \text{ and } L \text{ are integers with common factors removed.}$$

Discrete-time periodicity is L . See [Discrete Time Periodicity handout from Lecture 1](#).

3. Describe a way to use a smaller lookup table to save memory. 3 points.

If the discrete-time period L is even, the cosine could be computed over half of the period and the half can be determined through symmetry.

If the discrete-time period L is a multiple of four, cosine could be computed over one-fourth of the period and the rest of the samples can be determined through symmetry.

If the discrete-time period L is a multiple of eight, cosine could be computed over one-eighth of the period and the rest of the samples can be determined through symmetry.

(c) If discrete-time signal $\cos(\omega_0 n)$ is input to a squaring block, what discrete-time frequencies will appear on the output? 6 points. [See Lecture slides 3-7 & 3-8; Homework 1.3]

Output will have a component of 0 frequency (DC) and a component at a frequency of $2\omega_0$:

$$\cos^2(\omega_0 n) = \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 n)$$

If $2\omega_0 > \pi$, then the component at $2\omega_0$ will alias. Consider $\cos(\pi n) = (-1)^n$. Output of the squaring block is 1, which has discrete-time frequency components of 0, 2π , etc.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 10, 2017

Course: EE 445S Evans

Name: Busters, Myth
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Filter Implementation
3	27		Filter Design
4	21		Potpourri
<i>Total</i>	100		

Midterm #1

Spring 2017

See Lecture Slides 3-15, 5-4 to 5-20, 6-6, 6-7
HW 0.4, 1.1, 2.1, 2.2 & Lab #3

Problem 1.1 Filter Analysis. 28 points.

An finite impulse filter (FIR) design technique is to truncate the impulse response of an infinite impulse response (IIR) filter.

Consider the following causal IIR filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = a y[n-1] + x[n]$$

where a is real-valued such that $0.8 < a < 1$ and $y[-1] = 0$.

- (a) Compute the first three values of the ~~input~~ impulse response $h[n]$ for the IIR filter in terms of a . Plot $h[n]$ in terms of a . 3 points.

Let $x[n] = \delta[n]$.

$$h[0] = a h[-1] + \delta[0] = 1$$

impulse

$$h[1] = a h[0] + \delta[1] = a$$

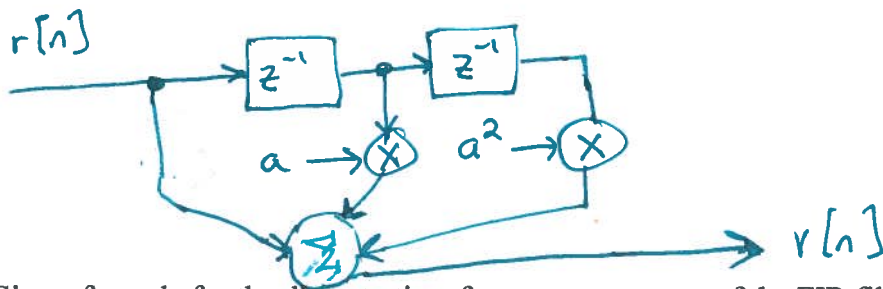
$$h[2] = a h[1] + \delta[2] = a^2$$

For the remaining parts, use the first three values of $h[n]$ computed in part (a) as the impulse response of a causal, linear time-invariant FIR filter.

- (b) For the FIR filter, give a formula in discrete time for the output $v[n]$ in terms of the input $r[n]$ including the initial conditions. 3 points.

$$v[n] = r[n] + a r[n-1] + a^2 r[n-2] \text{ with } r[-1] = 0 \text{ and } r[-2] = 0$$

- (c) Draw the block diagram of the FIR filter relating input $r[n]$ and output $v[n]$. 6 points.



- (d) Give a formula for the discrete-time frequency response of the FIR filter. 4 points.

Take z-transform: $V(z) = R(z) + a z^{-1} R(z) + a^2 z^{-2} R(z)$

$$\frac{V(z)}{R(z)} = 1 + a z^{-1} + a^2 z^{-2} \text{ for all } z \neq 0. \text{ Since region of}$$

- (e) What is the frequency selectivity of the FIR filter: lowpass, highpass, bandpass, bandstop, allpass, convergence notch? 6 points

Since a is real-valued,

Zeros: $z^2 + a z + a^2 = 0$

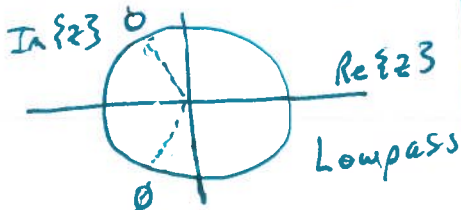
$$z = \frac{1}{2}(-a \pm \sqrt{a^2 - 4a^2}) = \frac{1}{2}(-a \pm j\sqrt{3}a)$$

includes unit circle,

$$H_{\text{freq}}(\omega) = H(z)|_{z=e^{j\omega}} = 1 + a e^{-j\omega} + a^2 e^{-j2\omega}$$

- (f) Does the FIR filter have linear phase? If yes, then give the conditions on the coefficients for the filter to have linear phase. If no, then show that the coefficients cannot meet the conditions for linear phase. 6 points

Since $0.8 < a < 1$,



For a linear phase FIR filter, the impulse response must be symmetric or anti-symmetric about the midpoint.

Symmetric: $a^2 \stackrel{?}{=} 1$. Not possible because $0.8 < a < 1$
Anti-symmetric: $a^2 \stackrel{?}{=} -1$ and $a \stackrel{?}{=} 0$.

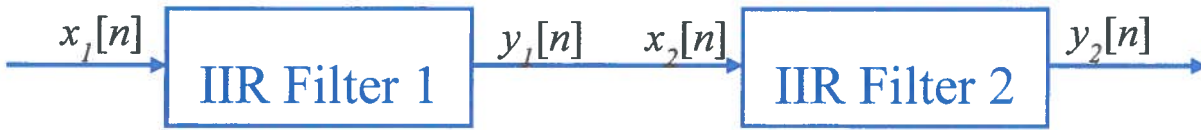
Midterm #1

Spring 2017

See Lecture Slides 1-13, 1-16, 6-2 to 6-11
HW 0.4, 1.1, 2.1, 3.3 & Lab #3

Problem 1.2 Filter Implementation. 24 points.

Consider the following cascade of first-order infinite impulse response (IIR) filters:



The input-output relationships in the time domain follow:

$$y_1[n] = a_1 y_1[n-1] + b_0 x_1[n] + b_1 x_1[n-1]$$

$$x_2[n] = y_1[n]$$

$$y_2[n] = c_1 y_2[n-1] + d_0 x_2[n] + d_1 x_2[n-1]$$

Feedback coefficients a_1 and c_1 , and feedforward coefficients b_0, b_1, d_0 , and d_1 , are real-valued.

Initial conditions are zero: $y_1[-1] = 0, x_1[-1] = 0, y_2[-1] = 0$, and $x_2[-1] = 0$.

Input data, coefficients and output data are stored in the same word size (B bits).

(a) Give the overall second-order transfer function, $H(z)$, of the cascade in the z -domain. 6 points.

$$H(z) = H_1(z)H_2(z) = \left(\frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \right) \left(\frac{d_0 + d_1 z^{-1}}{1 - c_1 z^{-1}} \right) = \frac{b_0 d_0 + (b_0 d_1 + b_1 d_0) z^{-1} + b_1 d_1 z^{-2}}{1 - (a_1 + c_1) z^{-1} + a_1 c_1 z^{-2}}$$

(b) For the overall second-order (biquad) transfer function $H(z)$ of the cascade,

1. Give the feedforward coefficients. 3 points.

$$b_0 d_0, \quad b_0 d_1 + b_1 d_0, \quad b_1 d_1$$

2. Give the feedback coefficients. 3 points.

$$-(a_1 + c_1), \quad a_1 c_1$$

3. Give the worst-case loss of precision in bits in each feedback coefficient in part 2. 6 points.

1 bit due to carry, B bits due to multiplication of two

(c) Compare the cascade of first-order IIR filters vs. the biquad in parts (a) and (b). 6 points. B -bit numbers

	# Multiplications per output sample	Data Storage (words)
Cascade	6	6 filter coefficients 4 previous input/output values
Biquad	5	5 filter coefficients 4 previous input/output values

and rounding/truncating back to B bits

Note: 6 is also a valid answer if 1 is included among the feedback coefficients in (b)2.

Midterm #1
Spring 2017

See Lecture Slides 1-3, 1-5, 1-10, 5-3
Lecture Slides 6-2 to 6-11; 6-19 to 6-24
HW 1.3, 2.2, 3.1, 3.3 & Lab #3

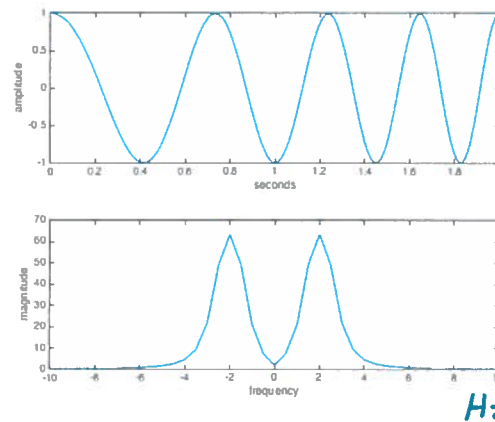
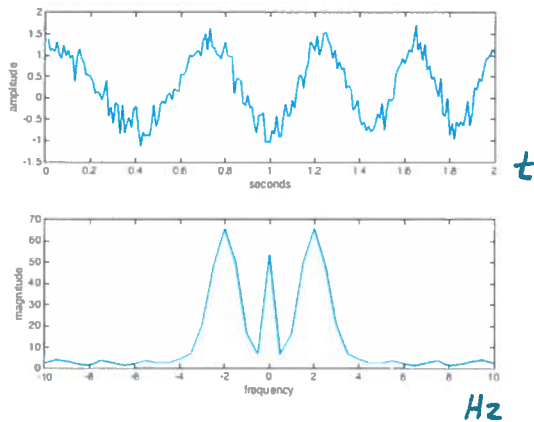
Problem 1.3 Filter Design. 27 points.

A sinusoidal signal of interest has a principal frequency that can vary over time in the range 1-3 Hz.

Using a sampling rate of $f_s = 20$ Hz, a sinusoidal signal was acquired for 2s and shown below on the left in the upper plot. The lower plot is the magnitude of the signal's frequency content.

The acquired signal has interference and other impairments that reduce the signal quality.

The signal shown below on the right is the sinusoidal signal without the impairments.



real-valued
amplitude in
time-domain.

impairments
at DC (0 Hz)
and high
frequencies (4-10 Hz).

Design a second-order infinite impulse response (IIR) filter to filter the acquired signal above on the left to give the sinusoidal signal above on the right

(a) Give the poles and zeros of the second-order IIR filter. 15 points.

Passband: 1-3 Hz with center frequency $f_c = 2$ Hz.

Stopbands: Around 0 Hz and 4-10 Hz

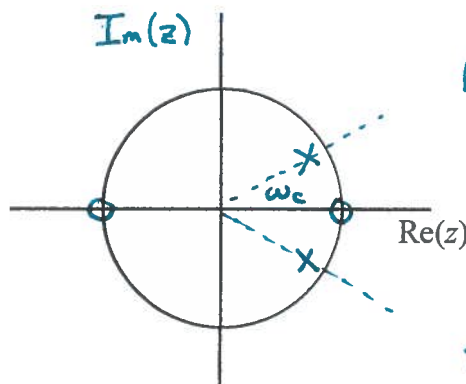
Poles: $p_0 = 0.9e^{j\omega_c}$; $p_1 = 0.9e^{-j\omega_c}$

$$\omega_c = 2\pi \frac{f_c}{f_s} = 2\pi \frac{2 \text{ Hz}}{20 \text{ Hz}} = \frac{\pi}{5} \text{ rad/sample.}$$

Zeros: $z_0 = -1$; $z_1 = 1$

i.e. $e^{j\pi} = -1$ and $e^{j0} = 1$.

(b) Draw the pole-zero diagram for the second-order IIR filter. 6 points.



Bandpass filter
with passband 1-3 Hz.

Second-order IIR
filter means exactly
two non-trivial poles.

(c) Normalize the filter's passband magnitude response to 1 in linear units. 6 points.

$$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

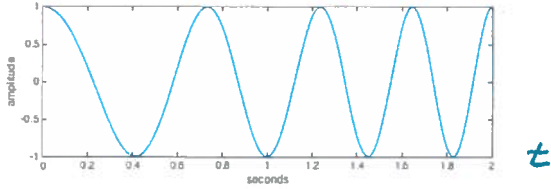
At $z = e^{j\omega_c}$, set $H(z) = 1$ and solve for C.

Midterm #1
Spring 2017

See Lecture Slides 1-6, 1-10, 1-15, 3-7
HW 0.1, 0.3, 0.4, 1.2, 1.3, 2.2 & Lab #2

Problem 1.4. Potpourri. 21 points.

(a) An example of a sinusoidal signal whose principal frequency is varying with time is shown on the right.



1. Describe a time-domain only method to determine the principal frequency over time. 3 points.

Time between two zero crossings is half of the period, i.e. $\frac{1}{2}T_0$.
Estimate sinusoidal frequency: $f_0 = \frac{1}{T_0}$.

2. Describe a method that uses frequency-domain information to determine the principal frequency over time. 3 points.

Spectrogram. A spectrogram takes the Fourier transform over different blocks of time so that principal (largest) frequency

(b) Consider generating an A major chord by playing the notes A, C# and E at the same time where the note frequencies are $f_A = 440$ Hz, $f_{C\#} = 550$ Hz and $f_E = 660$ Hz, respectively:

$$x(t) = \cos(2\pi f_A t) + \cos(2\pi f_{C\#} t) + \cos(2\pi f_E t)$$

components can be isolated in time.

1. Determine the corresponding discrete-time frequencies ω_A , $\omega_{C\#}$ and ω_E for a sampling rate of $f_s = 44100$ Hz. 3 points.

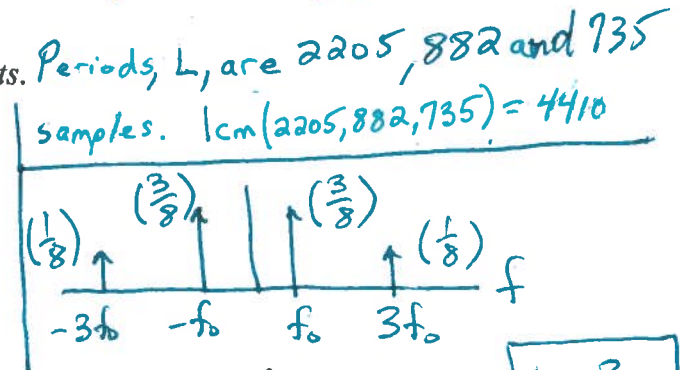
$$\omega_A = 2\pi \frac{f_A}{f_s} = 2\pi \frac{440 \text{ Hz}}{44100 \text{ Hz}}; \omega_{C\#} = 2\pi \frac{550 \text{ Hz}}{44100 \text{ Hz}}; \omega_E = 2\pi \frac{660 \text{ Hz}}{44100 \text{ Hz}}$$

2. What is the smallest discrete-time period in samples for $x[n]$? 3 points.

$$\omega_A = 2\pi \frac{22}{2205}; \omega_{C\#} = 2\pi \frac{11}{882}; \omega_E = 2\pi \frac{11}{735} = 2\pi \frac{N}{L}$$

3. Describe an efficient algorithm to generate $x[n]$. 3 points. Periods, L , are 2205, 882 and 735 samples. $\text{lcm}(2205, 882, 735) = 4410$

Compute $x[n]$ for the smallest discrete-time period in (2) above and store the amplitude values in a lookup table.

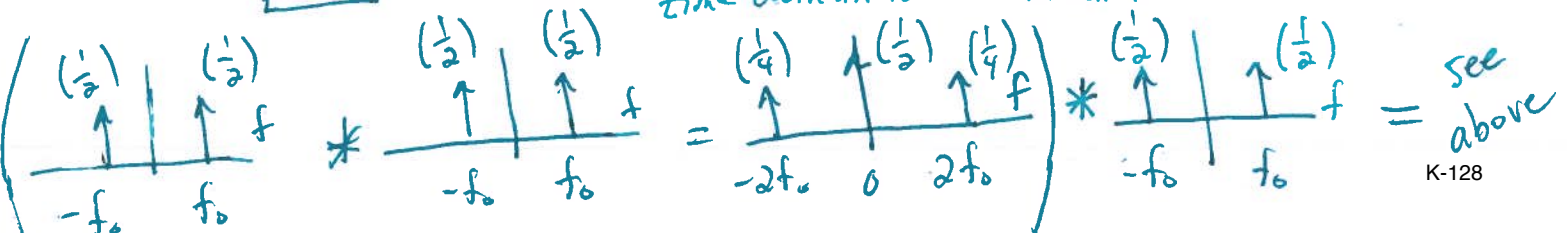


(c) If discrete-time signal $x[n] = \cos(\omega_0 n)$ is input to block that outputs $y[n] = x^3[n]$, what discrete-time frequencies will appear on the output? 6 points.

$$\omega_0, 3\omega_0$$



$y[n] = x^3[n] = \cos^3(\omega_0 n)$ multiplication in the time domain is convolution in the Fourier domain.



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 20, 2017

Course: EE 445S Evans

Name: _____ **Chase,** **Annabeth**
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Mixers
3	24		Filter Design
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

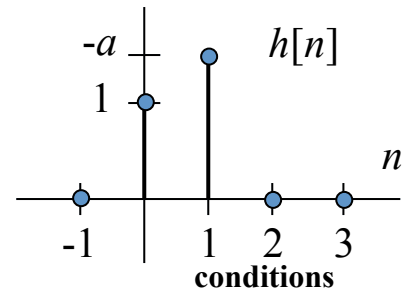
Consider the following causal finite impulse response (FIR) linear time-invariant (LTI) filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = x[n] - a x[n-1]$$

- (a) Give a formula for the impulse response $h[n]$. Plot $h[n]$. 3 points.

Input an impulse signal: $x[n] = \delta[n]$.

Impulse response is $h[n] = \delta[n] - a \delta[n-1]$.



- (b) What are the initial conditions? What are their values? 3 points.

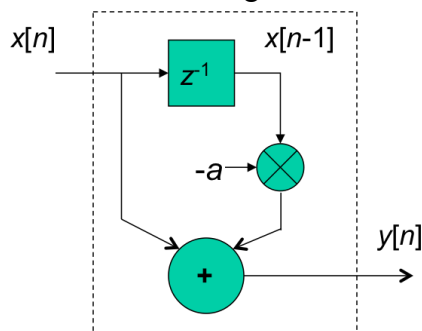
Compute output samples at $n = 0, 1$, etc., to reveal the initial

$y[0] = x[0] - a x[-1]$ which gives an initial condition of $x[-1]$.

$y[1] = x[1] - a x[0]$ which does not reveal any additional initial conditions.

System must be at rest at $n = 0$ for LTI properties to hold. Hence, $x[-1] = 0$.

- (c) Draw the block diagram of the FIR filter relating input $x[n]$ and output $y[n]$. 6 points.



- (d) Give a formula for the discrete-time frequency response of the FIR filter. 4 points.

With $h[n] = \delta[n] - a \delta[n-1]$, $H(z) = 1 - a z^{-1}$ for $z \neq 0$. Since the region of convergence $z \neq 0$ includes the unit circle, we can compute $H_{freq}(\omega) = H(e^{j\omega}) = 1 - a e^{-j\omega}$.

- (e) Does the FIR filter have linear phase? If yes, then give the conditions on the coefficient a for the filter to have linear phase. If no, then show that the coefficients cannot meet the conditions for linear phase. 6 points

In order for an FIR filter to have linear phase, its impulse response would need to be either even or odd symmetric about the midpoint. This would mean $a = -1$ or $a = 1$, respectively. Although not expected as an answer, $a = 0$ would give a zero phase response.

- (f) If parameter a were real-valued, what are all of the possible frequency selectivities that the FIR filter could provide: lowpass, highpass, bandpass, bandstop, allpass, notch? 6 points

Transfer function $H(z) = 1 - a z^{-1}$ has a zero at $z = a$. Location of zero indicates stopband.

We can rewrite $H(z) = \frac{z-a}{z}$, which has an artificial pole at $z = 0$.

Allpass response when $a \approx 0$. Highpass response $a > 0.5$. Lowpass response $a < -0.5$.

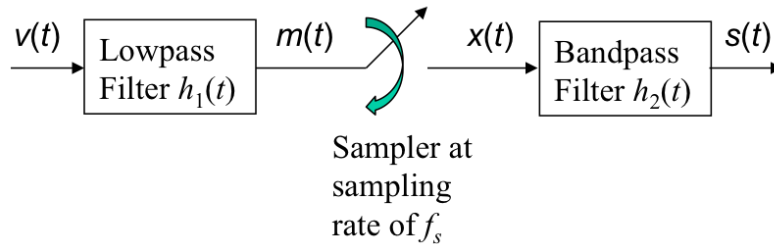
Note: Lowpass averaging filter when $a = -1$. Highpass first-order difference when $a = 1$.

Problem 1.2 Mixers. 24 points.

Mixing provides an efficient implementation in analog continuous-time circuits for sinusoidal amplitude modulation of the form

$$s(t) = m(t) \cos(2 \pi f_c t)$$

where $m(t)$ is the baseband message signal with bandwidth W , and f_c is the carrier frequency such that $f_c > W$



We discussed mixers for about 30 minutes during the Q&A session for homework #3 on Friday, Oct. 13th.

(a) Give the passband and stopband frequencies for the lowpass filter. 3 points.

$$f_{\text{pass}} = W.$$

f_{stop} could be $1.1W$ or $\frac{1}{2}f_s$. Using $1.1W$ would give 10% rolloff from passband to stopband.

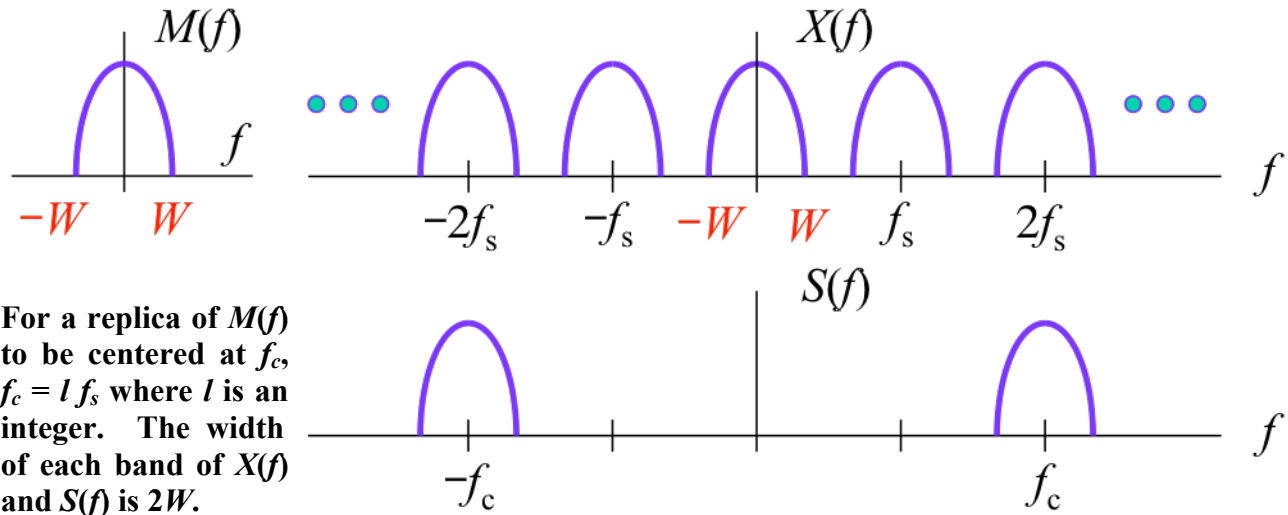
(b) Give the passband and stopband frequencies for the bandpass filter. 3 points

$$f_{\text{pass1}} = f_c - W \text{ and } f_{\text{pass2}} = f_c + W. \text{ Width of passband is } 2W. \text{ 10\% of that would be } 0.2W.$$

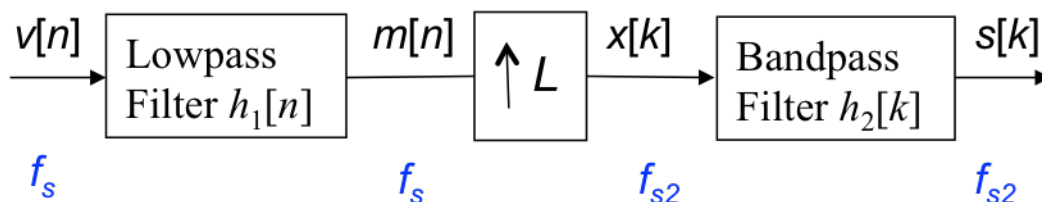
$$f_{\text{stop1}} = f_{\text{pass1}} - 0.2W \text{ and } f_{\text{stop2}} = f_{\text{pass2}} + 0.2W \text{ to give 10\% rolloff from passband to stopband.}$$

(c) Draw the spectrum for $m(t)$, $x(t)$, and $s(t)$. You do not need to draw the spectrum for $v(t)$. 9 points.

Baseband signal $m(t)$ has bandwidth of W ; $X(f)$ will have replicas of $M(f)$ due to sampling.



(d) In order to simulate the mixer in discrete-time, e.g. in MATLAB, we use discrete-time filters for the lowpass and highpass filters and replace the sampling block with an upsampling block.



- i. Give the constraints on the sampling rate to convert the mixer to discrete time. *6 points.*

In addition to f_s for the sampler in the mixer, we now have a second sampling rate f_{s2} for converting the mixer to discrete time. The highest frequency of interest in the mixer is $f_c + W$. Hence, $f_{s2} > 2(f_c + W)$. The sampling rate f_s will be used for signals $v(t)$ and $m(t)$ as well as the lowpass filter. The upsampler by L will convert an input sampling rate of f_s to an output sampling rate of f_{s2} where $L = f_{s2} / f_s$. Hence, $f_{s2} = L f_s$.

- ii. Determine the upsampling factor. *3 points.* $L = f_{s2} / f_s$

```
% MATLAB code for midterm #1
% problem 1.2(d) for Fall 2017
%
% Programmer: Prof. Brian L. Evans
% The University of Texas at Austin
% Date: October 23, 2017
%
% Mixer
% s(t) = m(t) cos(2 pi fc t)
%
% v(t)      m(t)          x(t)      s(t)
% ---> LPF ---> Sampler ---> BPF --->
%
% System Parameters
% In Continuous Time
W = 300;      % Message bandwidth Hz
fs = 1000;   % Sampling rate Hz
k = 15;      % kth replica at fc
fc = k*fs;   % Carrier frequency Hz
% In Discrete Time
L = 4*k;     % Upsampling factor
fs2 = L*fs;  % For x(t) and s(t)
% fs sampling rate for v(t) and m(t)
% fs2 > 2 (fc + W) and fs2 = L fs
% Since fc > W, fs2 >= 4 fc

% Time reference for v(t) and m(t)
Ts = 1/fs;
Tmax = 1;
t1 = Ts : Ts : Tmax;
N1 = length(t1);
n1 = 1 : N1;

% Generate v(t) as chirp to have all
% discrete-time frequencies by
% increasing instantaneous frequency
% from 0 to pi.
fstep = (fs/2) / Tmax;
wstep = pi*fstep*(Ts^2);
v = cos(wstep*n1.^2);
```

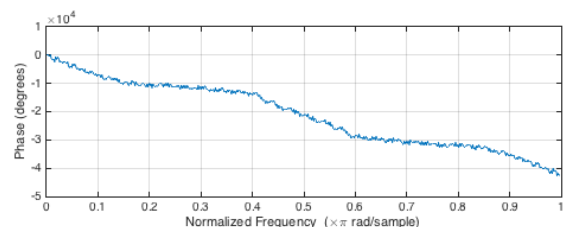
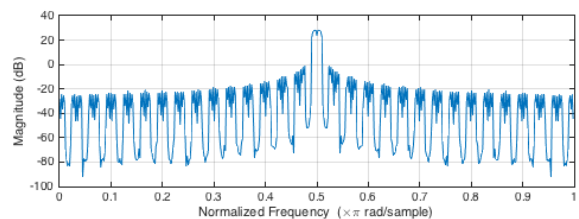
```
% Lowpass filter
% Truncated sinc pulse
fpass = W;
npass = -100 : 100;
hlpf = sinc(2*(fpass/fs)*npass);
hlpf = hlpf / sum(abs(hlpf).^2);

% Message signal
m = filter(hlpf, 1, v);

% Upsample by L
N2 = L*N1;
x = zeros(1, N2);
x(1:L:N2) = m;

% Bandpass filter centered with
% passband fc - W to fc + W
% 1. Design lowpass prototype
% 2. Modulate by cos(2 pi fc t)
fpass = W;
npass = -4*L : 4*L;
hprot = sinc(2*(fpass/fs2)*npass);
wc = 2*pi*fc/fs2;
hbpf = hprot .* cos(wc*npass);
hbpf = hbpf / sum(abs(hbpf).^2);

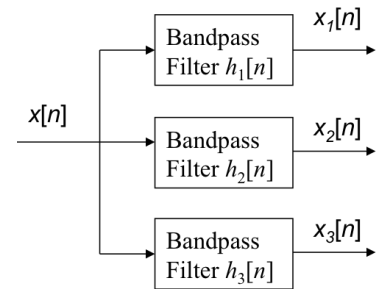
s = filter(hbpf, 1, x);
freqz(s);
```



Problem 1.3 Filter Design. 24 points.

Some audio systems split an audio signal in three frequency bands for playback over sub-wolfer, wolfer, and tweeter speaker elements.

The block diagram on the right performs the split in discrete time:



$x_1[n]$ contains sub-wolfer frequencies 20-200 Hz.

$X_2[n]$ contains wolfer frequencies 200-2,000 Hz.

$X_3[n]$ contains tweeter frequencies 2,000-20,000 Hz.

Assume that the sampling rate is 48000 Hz.

Each bandpass filter should have group delay of less than 10 ms.

(a) Give passband ripple and stopband attenuation values for these filters. 6 points.

$A_{\text{pass}} = 1$ dB and $A_{\text{stop}} = 80$ dB based on homework problems 2.3 and 3.3 on filter design for audio applications. (We'll explore these settings in lecture 8 on quantization. For example, in an A/D converter with B bits, the stopband attenuation would be $6B + 2$ dB.)

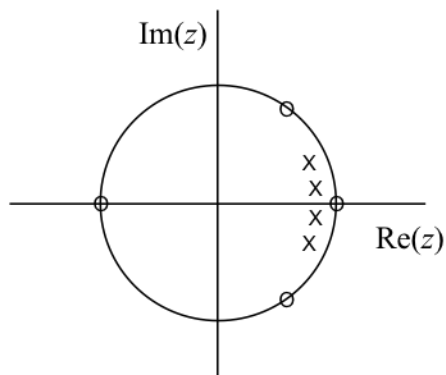
(b) Give passband and stopband discrete-time frequencies to design the bandpass filter $h_2[n]$. 6 points.

Seek 10% rolloff from stopband1 to passband1, and from passband2 to stopband2. Convert continuous-time frequency f_0 in Hz to discrete-time frequency $\omega_0 = 2\pi f_0 / f_s$.

Answer #1: $f_{\text{stop1}} = 20$ Hz, $f_{\text{pass1}} = 200$ Hz, $f_{\text{pass2}} = 2000$ Hz, $f_{\text{stop2}} = 2180$ Hz. This would give $\omega_{\text{stop1}} = 0.000833\pi$, $\omega_{\text{pass1}} = 0.00833\pi$, $\omega_{\text{pass2}} = 0.0833\pi$, $\omega_{\text{stop2}} = 0.0908\pi$, all in rad/sample.

Answer #2: $f_{\text{cutoff1}} = 200$ Hz and $f_{\text{cutoff2}} = 2000$ Hz, which could mean $f_{\text{stop1}} = 110$ Hz, $f_{\text{pass1}} = 290$ Hz, $f_{\text{pass2}} = 1910$ Hz, $f_{\text{stop2}} = 2090$ Hz to give slightly more than 10% rolloffs. This would give $\omega_{\text{stop1}} = 0.00458\pi$, $\omega_{\text{pass1}} = 0.0121\pi$, $\omega_{\text{pass2}} = 0.0796\pi$, $\omega_{\text{stop2}} = 0.0871\pi$, all in rad/sample.

(c) Draw the pole-zero diagram for a fourth-order infinite impulse response (IIR) bandpass filter $h_2[n]$. 6 points.



Poles near the unit circle indicate the passband(s).

Zeros on/near the unit circle indicate stopband(s).

Poles occur in conjugate symmetric pairs or are real-valued; same goes for zeros.

Try to keep zeros and poles away from each other.

Many possible answers.

Answer: Four poles at passband frequencies and their negatives. $p_0 = r \exp(j \omega_{\text{pass1}})$; $p_1 = r \exp(-j \omega_{\text{pass1}})$;

$p_2 = r \exp(j \omega_{\text{pass2}})$; $p_3 = r \exp(-j \omega_{\text{pass2}})$. Use $r = 0.9$.

Put zero at $z = 1$ to enforce stopband1, and other three zeros to enforce stopband2. **SEE MATLAB CODE BELOW.**

(d) For a linear phase finite impulse response (FIR) filter, indicate the maximum length that would still meet the group delay constraint. The maximum length would apply to all three filters. 6 points.

A group delay of 10ms means $(0.01\text{s})(48000 \text{ samples/s}) = 480$ samples.

A linear phase FIR filter with N coefficients has a group delay of $(N-1)/2$ samples.

So, $N = 961$ samples. (For lower group delay, use discrete-time IIR filters.)
MATLAB Code and Plots for 1.3(c)

```

% Four poles
wpass1 = 0.0121*pi;
wpass2 = 0.0796*pi;
r = 0.9;
p0 = r * exp(j*wpass1);
p1 = r * exp(-j*wpass1);
p2 = r * exp(j*wpass2);
p3 = r * exp(-j*wpass2);
denom1 = [1 -(p0+p1) p0*p1];
denom2 = [1 -(p2+p3) p2*p3];
denom = conv(denom1, denom2);

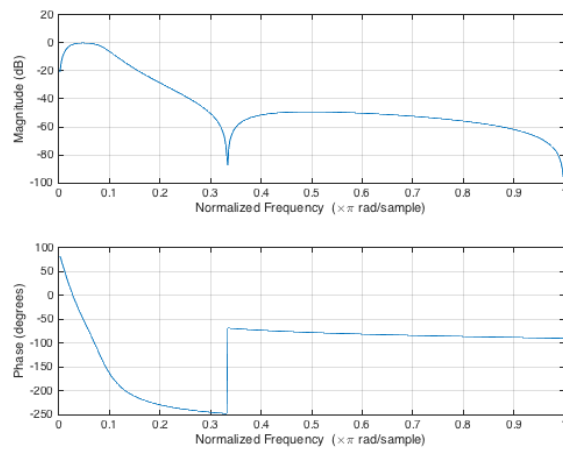
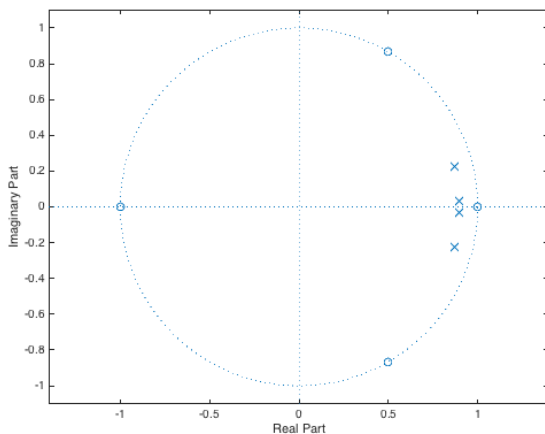
% Four zeros
zeroAngle = pi/3;
z0 = exp(j*zeroAngle);
z1 = exp(-j*zeroAngle);
numer1 = [1 -(z0+z1) z0*z1];
z2 = 1;      %% at 0 rad/sample
z3 = -1;     %% at pi rad/sample
numer2 = [1 -(z2+z3) z2*z3];
numer = conv(numer1, numer2);

%%% Normalize response in middle
%%% of the passband
w0 = (wpass1 + wpass2)/ 2;
Hresp = freqz(numer, denom, [w0 w0]);
C = 1 / abs(Hresp(1));

%%% Plot pole-zero diagram
figure; zplane(C*numer, denom);

%%% Plot frequency response
figure; freqz(C*numer, denom);

```



Group delay is ~15 samples in the passband.

Problem 1.4. Potpourri. 24 points.

- (a) Consider a linear time-invariant (LTI) system that has bounded-input bounded-output stability. To measure its frequency response, one could input a discrete-time unit impulse $\delta[n]$ for $-\infty < n < \infty$, find the output signal $h[n]$, and take the discrete-time Fourier transform of $h[n]$. In practice, we cannot go back to $n = -\infty$ or wait until $n = \infty$. Give a practical method using a finite-length discrete-time input signal to estimate the frequency response of the LTI system. *12 points.*

For an unknown BIBO stable LTI system, we seek to find its frequency response $H_{freq}(\omega)$.

In the frequency domain, $Y_{freq}(\omega) = H_{freq}(\omega) X_{freq}(\omega)$.

We seek to compute $H_{freq}(\omega) = Y_{freq}(\omega) / X_{freq}(\omega)$.

Since the discrete-time frequency domain is periodic with period 2π , we need a finite-length signal $x[n]$ whose frequency response $X_{freq}(\omega)$ has all frequencies from $-\pi$ to π in it and does not equal zero at any frequency value so as to avoid a division by zero error.

A two-sided discrete-time cosine signal $\cos(\omega_0 n)$ has frequency components at ω_0 and $-\omega_0$.

Use a chirp signal that linearly sweeps all frequencies from 0 to π .

- (b) Consider the following method to compute a cosine value by using a Taylor series at $\theta = 0$:

$$\cos(\theta) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} \theta^{2n} = 1 - \frac{1}{2} \theta^2 + \frac{1}{24} \theta^4 - \dots$$

Suppose that 10 non-zero terms were kept in the series expansion (i.e. $n = 0, 1, 2, \dots, 9$).

- i. How would you minimize the number of multiplications? *6 points.*

Factor the polynomial into Horner's form (lecture slide 1-12) to minimize the number of multiplications:

$$a_{18} x^{18} + a_{16} x^{16} + a_{14} x^{14} + \dots + a_0 = (\dots (((a_{18} x^2 + a_{16}) x^2 + a_{14}) x^2 \dots) x^2 + a_0$$

Number of multiplications reduces from 90 to 9.

Number of additions remains the same at 9.

- ii. Please complete the last row of entries for the new method. *6 points.*

Method	Multiplication-Add Operations	ROM (words)	RAM (words)	Quality in floating point
C math library call	30	22	1	Second best
Difference equation	2	2	3	Worst
Lookup table	0	L	0	Best
Taylor series	9	10	1	Second best

L is the smallest discrete-time period for the cosine signal.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 9, 2018

Course: EE 445S Evans

Name: _____

	Set,	Solution
	Last,	First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Mixers
3	24		Filter Design
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

Consider the following causal linear time-invariant (LTI) discrete-time filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = a_1 y[n-1] + x[n] + b_1 x[n-1] + b_2 x[n-2]$$

for $n \geq 0$, where coefficients a_1 , b_1 and b_2 are real-valued constants.

(a) What are the initial conditions and their values? Why? 3 points.

Let $n = 0$: $y[0] = a_1 y[-1] + x[0] + b_1 x[-1] + b_2 x[-2]$.

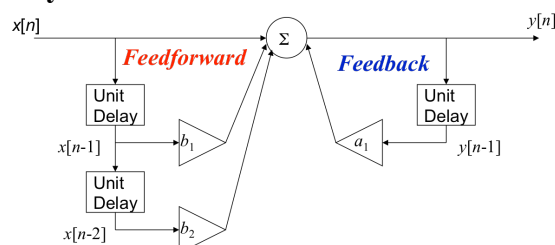
Let $n = 1$: $y[1] = a_1 y[0] + x[1] + b_1 x[0] + b_2 x[-1]$.

From the above, the initial conditions are $y[-1]$, $x[-1]$ and $x[-2]$.

The initial conditions have to be zero to guarantee linearity and time-invariant properties.

(b) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.

Any of the three IIR direct forms could be used here, e.g. from [Lecture Slide 6-9](#)



(c) Derive a formula for the transfer function in the z -domain. 4 points.

Take the z -transform of both sides of the difference equation using the knowledge that all of the initial conditions are zero ([Lecture Slide 6-8](#)):

$$Y(z) = a_1 z^{-1} Y(z) + X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z)$$

$$(1 - a_1 z^{-1}) Y(z) = (1 + b_1 z^{-1} + b_2 z^{-2}) X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1}} \text{ for } |z| > |a_1|$$

(d) Give the conditions on a_1 , b_1 and b_2 for the filter to be bounded-input bounded-output (BIBO) stable. 4 points.

Pole must be inside the unit circle ([Lecture Slides 6-12 & 6-13](#)): $|a_1| < 1$

A zero only plays a role in BIBO stability if it cancels a pole that leads to BIBO instability.

If either zero equals a_1 , the filter is in theory BIBO stable for any value of a_1 although an implementation of the original difference equation may not be BIBO stable.

(e) Give a formula for the discrete-time frequency response of the filter. 4 points.

When the LTI system is BIBO stable according to part (d),

$$H_{freq}(\omega) = H(z) \Big|_{z=e^{j\omega}} = \frac{1 + b_1 e^{-j\omega} + b_2 e^{-2j\omega}}{1 - a_1 e^{-j\omega}}$$

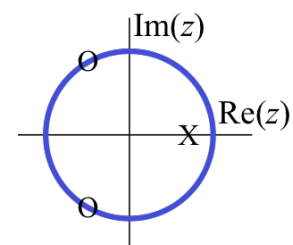
(f) Give numeric values for coefficients a_1 , b_1 and b_2 to design a lowpass filter that also eliminates frequencies at $\omega = 2\pi/3$ and $\omega = -2\pi/3$ in the stopband. Draw the pole-zero diagram. 7 points

Zeros are at $z_0 = e^{j2\pi/3}$ and $z_1 = e^{-j2\pi/3}$

$$(1 - z_0 z^{-1})(1 - z_1 z^{-1}) = 1 - (z_0 + z_1)z^{-1} + z_0 z_1 z^{-2}$$

So, $b_1 = -(z_0 + z_1) = -2 \cos\left(\frac{2\pi}{3}\right) = 1$ and $b_2 = z_0 z_1 = 1$

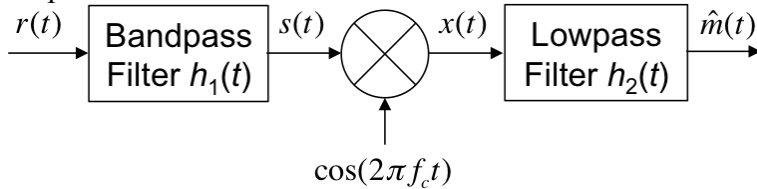
Passband is centered at $\omega = 0$. Pole is at $a_1 = 0.9 e^{j0} = 0.9$



Trivial pole at $z = 0$ not shown

Problem 1.2 Mixers. 24 points.

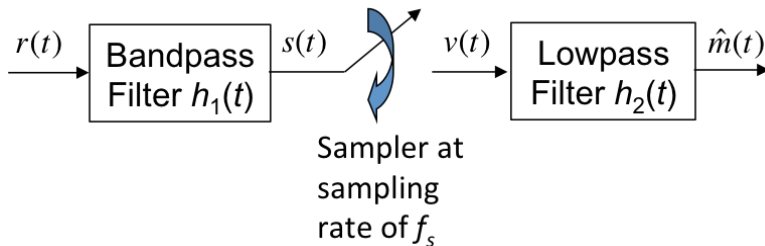
Mixing provides an efficient implementation in analog continuous-time circuits for sinusoidal amplitude **demodulation** of the form



This is a sequel to Problem 1.2 on Fall 2017 Midterm #1.

We had discussed connections between downsampling and downconversion in class on Feb. 23rd when going over solutions for homework #2 and I had followed the discussion with an announcement on Canvas on Feb. 25th (see page 5 below).

Here, $r(t)$ is the received bandpass signal of the form $r(t) = m(t) \cos(2\pi f_c t)$ where $m(t)$ is the baseband message signal with bandwidth W , and f_c is the carrier frequency such that $f_c > W$



(a) Give the passband and stopband frequencies for the bandpass filter. 3 points.

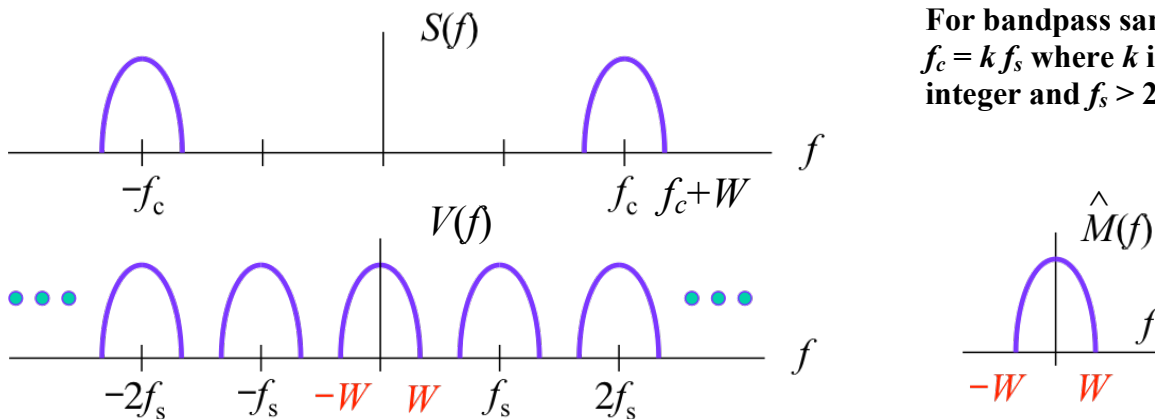
The bandpass filter passes the transmission band frequencies $[f_c - W, f_c + W]$ and attenuates out-of-band frequencies as much as possible to improve the signal-to-noise ratio. Passband frequencies are $f_c - W$ and $f_c + W$. Stopband frequencies are $f_c - 1.2W$ and $f_c + 1.2W$ to allow each transition bandwidth to be 10% of the passband bandwidth.

(b) Give the passband and stopband frequencies for the lowpass filter. 3 points

$f_{pass} = W$ and $f_{stop} = 1.1W$ to allow transition bandwidth be 10% of the passband bandwidth.

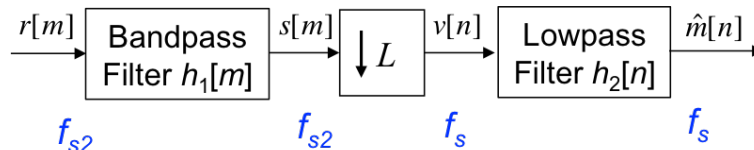
(c) Draw the spectrums for $s(t)$, $v(t)$ and $\hat{m}(t)$. You do not need to draw the spectrum for $r(t)$. 9 points.

$s(t)$ is a bandpass signal. Sampling replicates $S(f)$ at offsets of integer multiples of sampling rate f_s . Each band in $V(f)$ and $S(f)$ is $2W$ wide. $\hat{m}(t)$ is the estimated baseband signal.



For bandpass sampling, $f_c = k f_s$ where k is an integer and $f_s > 2W$.

(d) In order to simulate the mixer in discrete-time, e.g. in MATLAB, we use discrete-time filters for the lowpass and highpass filters and replace the sampling block with a downsampling block.



- i. Give the constraints on the sampling rate to convert the mixer to discrete time. *6 points.*

We have two sampling rates at play. Sampling rate f_s is used for the sampler in the continuous-time circuit for downconversion using mixing. Sampling rate f_{s2} is used to convert the overall system to discrete time.

(1) Nyquist-Shannon Sampling Theorem gives $f_{s2} > 2(f_c + W)$ and

(2) The downsampler by L will convert the sampling rate for the input signal of f_{s2} to a lower sampling rate on the output of f_s where $L = f_{s2} / f_s$. Hence, $f_{s2} = L f_s$.

Sampling rate f_{s2} is used for signals $r(t)$ and $s(t)$ and the bandpass filter. Sampling rate f_s is used for signals $v(t)$ and $\hat{m}(t)$ and the lowpass filter.

- ii. Determine the downsampling factor. *3 points.* $L = f_{s2} / f_s$

<pre> % MATLAB code for midterm #1 % problem 1.2(d) for Spring 2018 % % Programmer: Prof. Brian L. Evans % The University of Texas at Austin % Date: March 18, 2018 % % Sinusoidal amplitude modulation % r(t) = m(t) cos(2 pi fc t) % % % r(t) s(t) v(t) m(t) % ---> BPF ---> Sampler ---> LPF ---> % fs2 fs2 fs fs % System Parameters % Continuous Time W = 300; % Message bandwidth Hz fs = 1000; % Sampling rate in Hz k = 15; % kth replica at fc fc = k*fs; % Carrier frequency Hz % Discrete Time L = 4*k; % Downsampling factor fs2 = L*fs; % Sampling rate #2 % fs2 > 2 (fc + W) and fs2 = L fs % With W < fc, fs2 = L fc, L >= 4 % Simulation Parameters % Bandpass signal amplitudes have % near-zero mean, and downsampling % by keeping every Lth sample gives % values close to zero. Floating % point calculations lose accuracy % when values are below eps, where % eps is the largest positive value % for which 1.0 + eps = 1.0. In my % MATLAB, eps is 2.2204 x 10^(-16). SimGain = 3.69*10^11; % Time ref for m(t), r(t) & s(t) Tmax = 10; Ts2 = 1/fs2; t2 = Ts2 : Ts2 : Tmax; N2 = length(t2); n2 = 1 : N2; </pre>	<pre> % Baseband message w/ frequencies 0 to W fpass = W; npass = -100*L : 100*L; hprot = sinc(2*(fpass/fs2)*npass); hprot = hprot / sum(abs(hprot).^2); noise = randn(1, N2); m = filter(hprot, 1, noise); % r[n2] = m[n2] cos(wc n2) wc = 2*pi*fc/fs2; r = m .* cos(wc*n2); % DOWNCONVERSION BY DOWNSAMPLING % Design bandpass filter with % passband from fc - W to fc + W % 1. Design lowpass prototype % 2. Modulate by cos(2 pi fc t) fpass = W; npass = -10*L : 10*L; hprot = sinc(2*(fpass/fs2)*npass); wc = 2*pi*fc/fs2; hbpf = hprot .* cos(wc*npass); hbpf = hbpf / sum(abs(hbpf).^2); % Design lowpass filter with passband % frequency using truncated sinc pulse fpass = W; npass = -100 : 100; hlpf = sinc(2*(fpass/fs)*npass); hlpf = hlpf / sum(abs(hlpf).^2); % Signal processing steps r = SimGain * r; s = filter(hbpf, 1, r); % Bandpass v = s(1:L:end); % Downsample mhat = filter(hlpf, 1, v); % Lowpass % Compare message signals plotspec(m, Ts2); title('Original message signal'); Ts = 1/fs; figure; plotspec(mhat, Ts); title('Estimated message signal'); PowerInOrgMessage = sum(abs(m.^2)) PowerInEstMessage = sum(abs(mhat.^2)) SimGain </pre>
---	---

The comments in the MATLAB code for the **Simulation Parameters** indicate that a very wide range of amplitude values is needed for downconversion using downsampling, e.g. 11 orders of magnitude in the above simulation. This is reflected in the **SimGain** parameter. The reason is that the downsampling operation samples many amplitude values that are very close to zero. This also means that the approach is very sensitive to additive noise and interference that are in the transmission band.

Beginning of the class Canvas announcement sent on Feb. 25, 2018:

On Friday in lecture, a student asked about the connection between downconversion and downsampling and the connection between upconversion and upsampling. Downsampling can be used as a method for downconversion, and upsampling can be used as a method for upconversion. Downsampling and upsampling change the sampling rate through discrete-time methods, and were introduced on homework #2.

On Friday, I worked out how to perform upconversion and downconversion in continuous time using sampling. The sinusoidal [sinusoidal] amplitude modulation approach to upconversion involves a message signal being input into a lowpass filter followed by multiplication by $\cos(2\pi f_c t)$ followed by bandpass filtering to produce a bandpass transmission. In the sampling approach, the multiplication by $\cos(2\pi f_c t)$ is replaced by sampling at f_s where $f_c = m f_s$ where m is a positive integer and $f_s > 2B$ where B is the baseband bandwidth. The sampling approach to upconversion uses aliasing to its advantage.

Similarly, the sinusoidal [sinusoidal] amplitude demodulation approach for downconversion a receives a bandpass signal, then applies bandpass filtering, multiplication by $\cos(2\pi f_c t)$ and lowpass filtering to give an estimate of the message signal. In the sampling approach, multiplication by $\cos(2\pi f_c t)$ is replaced by sampling at f_s where $f_c = m f_s$ where m is a positive integer and $f_s > 2B$ where B is the baseband bandwidth. The sampling approach to downconversion, which is also known as bandpass sampling, uses aliasing to its advantage.

Lecture slides 4-12 and 4-13 also describe the continuous-time analysis. Problem 1.4 on Midterm #1 in Fall 2013 and Problem 1.4 on Midterm #1 in Fall 2012 describe upconversion using sampling, and Problem 1.4(d) on Midterm #1 in Fall 2013 describes downconversion through sampling.

For a discrete-time approach, we'll use f_{s2} as the overall sampling rate. In this case, we will be oversampling by choosing $f_{s2} > 2 f_{max}$ where $f_{max} = f_c + B$ where B is the bandwidth in Hz of the baseband message signal in continuous time.

For the upsampling approach to upconversion, the discrete-time baseband message signal would be upsampled and then bandpass filtered in discrete time to keep the replica that is centered at $\omega_c = 2\pi f_c / f_{s2}$. One possible upsampling factor is $L = f_{s2} / f_c$ where L is an integer. Please see problem 1.2(d) on the Spring 2017 Midterm #1 Exam.

For the downsampling approach to downconversion, the received discrete-time bandpass signal centered at discrete-time frequency $\omega_c = 2\pi f_c / f_{s2}$ would be downsampled and then lowpass filtered in discrete time to extract the discrete-time baseband signal. One possible downsampling factor is $M = f_{s2} / f_c$ where M is an integer. Please see Problem 1.3 on Midterm #1 in Spring 2012.

End of the class Canvas announcement sent on Feb. 25, 2018.

Please note that the solution for Problem 1.3(b) on Spring 2012 Midterm is related to Problem 1.2(d) above with the following specific settings:

$$B = 2W$$

$$f_{s2} = M f_c$$

Due to bandpass sampling in Problem 1.2(d) above, $f_c = k f_s$ where k is an integer, so

$$f_{s2} = M f_c = M (k f_s) = (M k) f_s$$

$$L = M k$$

Problem 1.3 Filter Design. 24 points.

Every time that a particular tone at continuous-time frequency f_0 in Hz is detected, a particular audio effects system plays the tone at frequency f_0 and a tone at frequency $3f_0$.

Assume that $20 \text{ Hz} < f_0 < 5000 \text{ Hz}$ and that the sampling rate is $f_s > 6f_0$.

The audio effects system will be running continuously. When frequency f_0 is not present, the audio effects system could generate very low volume sounds.

- (a) Design a **second-order discrete-time** linear time-invariant (LTI) infinite impulse response (IIR) filter to detect frequency f_0 by giving formulas for the locations of the two poles and two zeros of the filter. Normalize the gain at continuous-time frequency f_0 to be 1. 9 points.

Bandpass filter centered at $\omega_0 = 2\pi \frac{f_0}{f_s}$ where $\omega_0 \in (-\frac{\pi}{3}, \frac{\pi}{3})$ since $f_s > 6f_0$.

Poles are located at $p_0 = 0.9e^{j\omega_0}$ and $p_1 = 0.9e^{-j\omega_0}$ to ensure BIBO stability.

Place both zeros at $z = -1$ as in part (b) when the poles are close to $z = 1$ because $\omega_0 \approx 0$, or place the zeros at $z = 1$ and $z = -1$ for a more selective bandpass filter, or

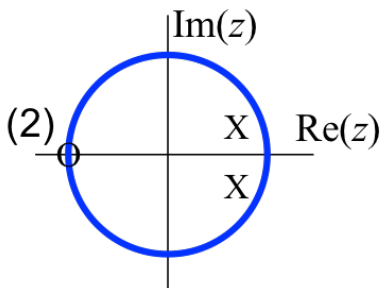
use the zeros for the bandpass resonator at $z = 0$ and $z = \cos(\omega_0)$ from [homework 2.1\(d\)](#).

The transfer function for a second-order IIR filter (biquad) from [Lecture Slide 6-6](#) is

$$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} \text{ for } |z| > 0.9$$

To normalize the gain at frequency ω_0 , we substitute $z = e^{j\omega_0}$ into $H(z)$ and solve for C.

- (b) Draw the pole-zero diagram for the poles and zeros given in part (a). 6 points.



```
f0 = 440; fs = 8*f0;
z0 = -1; z1 = -1; % Try z0 = 1
numer = [1 -(z0+z1) z0*z1];
r = 0.90; w0 = 2*pi*f0/fs;
p0 = r*exp(j*w0); p1 = r*exp(-j*w0);
denom = [1 -(p0+p1) p0*p1];
%%% Normalize response at f0 to 1
z = exp(j*w0); zv = [1 z^(-1) z^(-2)];
C = (denom * zv') / (numer * zv');
figure; zplane(C*numer, denom);
figure; freqz(C*numer, denom);
```

- (c) When the discrete-time IIR filter outputs a tone at continuous-time frequency f_0 , what additional signal processing step(s) would you apply to the filter output to generate tones at continuous-time frequencies f_0 and $3f_0$? 9 points. **Let the IIR filter output be $y[n]$ and $y[n] \approx \cos(\omega_0 n)$.**

Solution #1: Modulate $y[n]$ by $\cos(2\omega_0 n)$ to produce discrete-time frequencies ω_0 and $3\omega_0$.

Solution #2: Input $y[n]$ into a cubing block to give $y^3[n]$. That is, $y^3[n] = y[n] y[n] y[n]$. In the frequency domain, $Y(e^{j\omega})$ is a pair of Dirac deltas located at $-\omega_0$ and ω_0 with area π .

$\frac{1}{2\pi} Y(e^{j\omega}) * Y(e^{j\omega})$ gives Dirac deltas at $-2\omega_0, 0$, and $2\omega_0$. $\frac{1}{2\pi} (\frac{1}{2\pi} Y(e^{j\omega}) * Y(e^{j\omega})) * Y(e^{j\omega})$

gives Dirac deltas at $-3\omega_0, -\omega_0, \omega_0$, and $3\omega_0$. Or, $\cos^3(\omega_0 n) = \frac{3}{4} \cos(\omega_0 n) + \frac{1}{4} \cos(3\omega_0 n)$.

Solution #3: Sinusoidal demodulation by $\cos(\omega_0 n)$ and sinusoidal modulation by $\cos(3\omega_0 n)$.

Problem 1.4. Potpourri. 24 points.

(a) Consider a discrete-time infinite impulse response (IIR) filter that is causal, linear time-invariant (LTI), and bounded-input bounded-output (BIBO) stable and that is defined in terms of its poles, zeros and gain. When implementing the filter in 32-bit IEEE floating-point arithmetic and data:

i. Describe how an implementation could cause the filter to become BIBO unstable. *6 points.*

Assume that the poles, zeros and gain are represented in 32-bit IEEE floating-point, which uses 24 bits of mantissa + sign and 8 bits for the exponent.

When expanding the factored form of the transfer function in the z -domain to an unfactored form using 32-bit IEEE floating-point arithmetic, one would compute

$$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} = C \frac{1 - (z_0 + z_1)z^{-1} + z_0 z_1 z^{-2}}{1 - (p_0 + p_1)z^{-1} + p_0 p_1 z^{-2}}$$

The feedback coefficient $-(p_0 + p_1)$ would lose one bit of accuracy in the worst case due to a carry bit, and the feedback coefficient $p_0 p_1$ would lose 24 bits of accuracy in the mantissa in the worst case. Using cascade of biquads helps reduce loss of accuracy.

Converting an M th-order IIR filter into a single section would cause a loss of $24(M-1)$ bits of accuracy in the worst case for the feedback coefficient in front of the z^{-M} term.

Due to the loss of accuracy in the feedback coefficients, factoring the denominator to find the new location of poles may reveal that some of the poles have moved onto or outside the unit circle. Please see [Lecture Slides 6-20 and 6-23](#).

ii. Describe how an implementation could cause the loss of LTI properties. *6 points.*

By setting one or more of the initial conditions to a value other than zero.

iii. Give an example of a particular causal, LTI, BIBO stable discrete-time IIR filter for which its causal, LTI and BIBO stable properties are preserved when the filter is implemented in 32-bit IEEE floating-point arithmetic and data. *6 points.*

Let's consider a first-order, causal, LTI, BIBO stable, discrete-time IIR filter: $y[n] = 0.5 y[n-1] + 0.5 x[n]$. For LTI, $y[-1] = 0.0$. If $x[0] = 2.0$ and $x[n] = 1.0$ for $n > 1$, then $y[n] = 1.0$ for $n \geq 0$.

```
N = 1000;
x = ones(1, N);
x(1) = 2;
a = [1 -0.5];
b = [0.5];
y = filter(b, a, x);
sum(abs(y - 1.0)^2)
```

(b) For a finite impulse response (FIR) filter with N coefficients, what is the increase in the number of multiplication-addition operations if the input signal, FIR coefficients and output signal were complex-valued instead of real-valued? *6 points.*

1 complex multiplication $(a + jb)(c + jd) = (ac - bd) + j(bc + ad)$ would take 4 real-valued multiplications and 2 real-valued additions.

1 complex addition $(a + jb) + (c + jd) = (a + c) + j(b + d)$ would take 2 real-valued additions.

1 complex multiplication-addition would take 4 real-valued multiplication-additions.

An FIR filter with N coefficients would take N multiplication-additions to compute one output value.

A complex-valued FIR filter would take 4 times as many real-valued multiplication-additions to compute one output value.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 **Solutions**

Date: October 19, 2018

Course: EE 445S Evans

Name: _____ *Liv* & *Maddie* _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

	<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
<i>Liv</i>	1	28		Filter Analysis
<i>Maddie</i>	2	27		Minimum Phase FIR Filters
<i>Liv</i>	3	24		Bluetooth Receiver
<i>Maddie</i>	4	21		Potpourri
	<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

Consider the following causal linear time-invariant (LTI) discrete-time filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = x[n] + a x[n-2] + a^2 x[n-4]$$

for $n \geq 0$, where a is a real-valued coefficient where $0 < a < 1$.

- (a) Is this a finite impulse response (FIR) or infinite impulse response (IIR) filter? Why? 3 points.

FIR filter. Any of the following reasons would provide sufficient justification:

1. The impulse response extends for 5 samples from $n = 0$ to $n = 4$, which is finite in duration.
2. The output $y[n]$ does not depend on previous output values; i.e., there is no feedback.
3. In the transfer function in the z -domain in part (d), the only poles are trivial pole at $z = 0$.

- (b) What are the initial conditions and their values? Why? 6 points.

Let $n=0$: $y[0] = x[0] + a x[-2] + a^2 x[-4]$. $x[0]$ is first input value and not an initial condition.

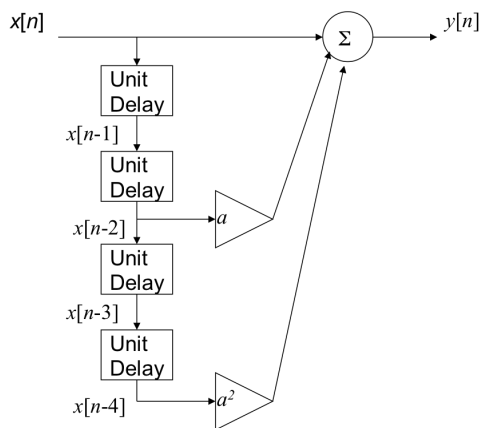
Let $n=1$: $y[1] = x[1] + a x[-1] + a^2 x[-3]$.

Let $n=2$: $y[2] = x[2] + a x[0] + a^2 x[-2]$. The initial conditions are $x[-1]$, $x[-2]$, $x[-3]$, and $x[-4]$.

The initial conditions have to be zero for linearity and time-invariant properties to hold.

Note: A causal system does not depend on future input values or future output values.

- (c) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



Note: The four initial conditions are visible here as the initial condition for each unit delay block.

- (d) Derive a formula for the transfer function in the z -domain. 4 points.

Lecture slide 5-11

Z-transform both sides of difference equation, knowing that all initial conditions are zero:

$$Y(z) = X(z) + a z^{-2} X(z) + a^2 z^{-4} X(z) \text{ which means } H(z) = \frac{Y(z)}{X(z)} = 1 + a z^{-2} + a^2 z^{-4} \text{ for } z \neq 0$$

- (e) Give a formula for the discrete-time frequency response of the filter. 3 points.

We can convert the transfer function $H(z)$ into the discrete-time frequency domain by substituting $z = \exp(j\omega)$ because FIR LTI systems are always Bounded-Input Bounded-Output stable, or equivalently, because the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = 1 + a e^{-2j\omega} + a^2 e^{-4j\omega}$$

- (f) Does the filter have linear phase over all frequencies? Why or why not? 6 points.

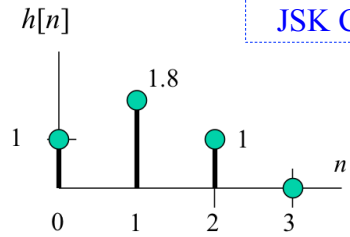
An FIR has linear phase if its impulse response is even or odd symmetric about its midpoint.

The impulse response has values 1, 0, a, 0, a². Because $0 < a < 1$, neither odd symmetry nor even symmetry about the midpoint ($n = 2$) is possible.

Problem 1.2 Minimum Phase FIR Filters. 27 points.

Minimum phase finite impulse response (FIR) filters have shorter length and lower group delay compared with linear phase FIR filters that meet the same magnitude specification.

We will use an algorithm by Herrman and Schüssler to convert an odd-length linear phase FIR filter to a minimum phase FIR filter.



Consider a linear phase FIR filter has the causal three-point impulse response $h[n]$ given above. Filter order M is 2. The filter is lowpass with stopband attenuation $A_{\text{stop}} = 0.2$ in linear units (-13.9794 dB).

(a) What is the group delay in samples through the linear phase FIR filter $h[n]$? 3 points.

HW 3.2

Group delay is defined as $-\frac{d}{d\omega} \angle H_{\text{freq}}(\omega)$ where $H_{\text{freq}}(\omega)$ is the frequency response.

For an N -point FIR filter with even or odd symmetry about the midpoint of its impulse response, the group delay is a constant value of $\frac{N-1}{2} = \frac{M}{2} = 1$ sample.

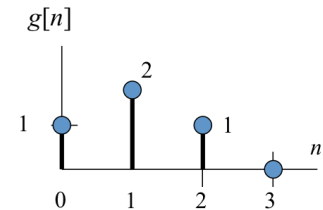
This corresponds to the index of the midpoint in the impulse response.

Lecture slides 5-12
5-15 5-17 & 5-18

(b) Please implement the steps in the Herrman and Schüssler algorithm below.

i. Compute the impulse response for a linear phase FIR filter $g[n] = A_{\text{stop}} \delta[n - (M/2)] + h[n]$ where A_{stop} is in linear units. Plot $g[n]$. 6 points.

$$g[n] = 0.2 \delta[n - 1] + h[n] = \delta[n] + 2 \delta[n - 1] + \delta[n - 2]$$



ii. Compute $G(z)$. 6 points.

$$G(z) = 1 + 2z^{-1} + z^{-2}$$

$$G(z) = (1 + z^{-1})^2 \text{ which has a double zero at } z = -1.$$

Lecture slides 5-5
5-17 5-18 & 5-19

iii. Form the minimum phase FIR filter $V(z)$ by keeping any zeros of $G(z)$ inside the unit circle and keeping one zero in each pair of repeated zeros on the unit circle in $G(z)$. 6 points.

$$V(z) = C(1 + z^{-1}) \text{ where } C \text{ is the gain.}$$

HW 3.2

iv. Compute the gain for $V(z)$ so that its response at $\omega = 0$ is the same as the response of $H(z)$ at $\omega = 0$. 6 points.

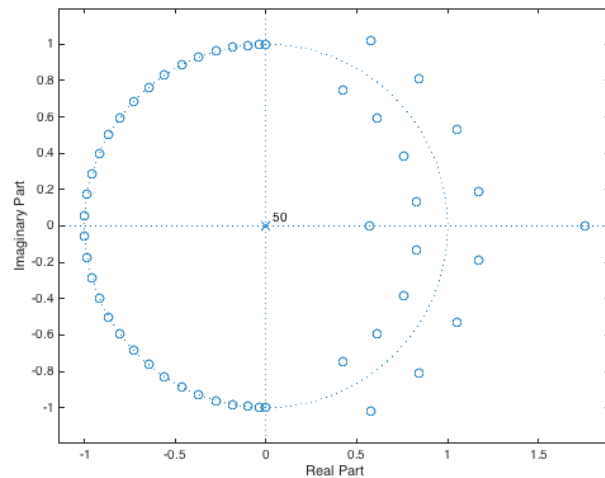
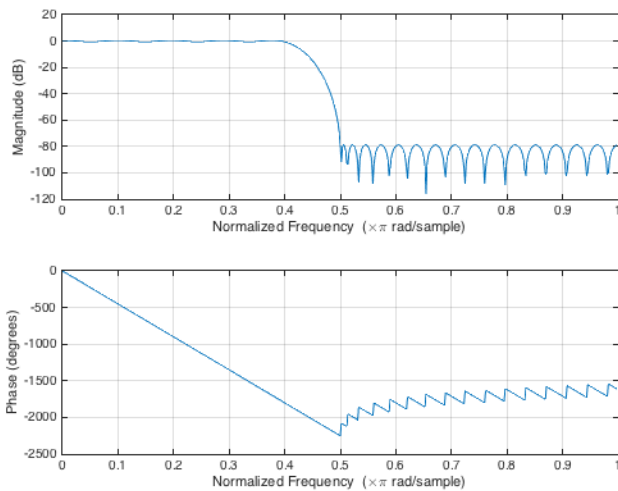
In the z -domain, discrete-time frequency ω is $z = e^{j\omega}$.

$$\text{When } \omega = 0, z = e^{j0} = 1.$$

$$V(1) = 2C \text{ and } H(1) = 3.8.$$

$$\text{This gives } 2C = 3.8 \text{ or } C = 1.9.$$

Note: The Herrman-Schüssler algorithm is based on observing the structure of zeros in a linear phase finite impulse response (FIR) filter. Here are the pole-zero and frequency response plots for a 50th-order linear phase FIR filter $h[n]$ using fdatool in MATLAB with default design parameters:



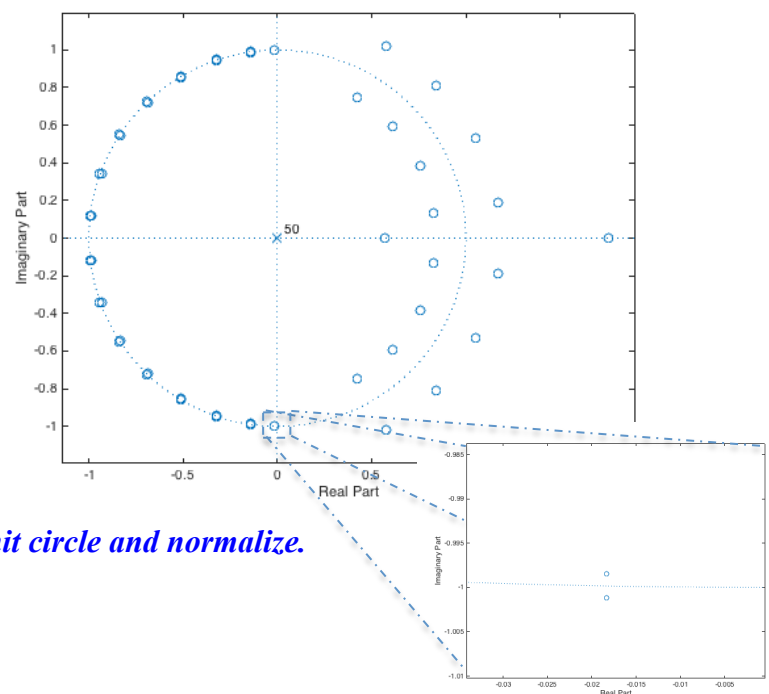
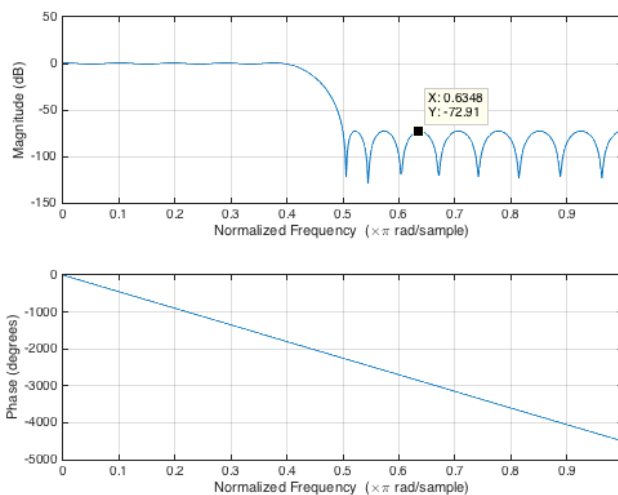
Filter design parameters are $f_{pass} = 9600$ Hz, $A_{pass} = 1$ dB, $f_{stop} = 12000$ Hz, $A_{stop} = 80$ dB, and $f_s = 48000$ Hz. In linear units, A_{stop} is 10^{-4} . The filter order is $M = 50$.

In zooming into the stopband of the freqz plot on the left and using the data cursor tool in MATLAB, the actual A_{stop} is -78.8 dB, which is 1.1482×10^{-4} in linear units.

In the pole-zero plot, the zero locations in the passband frequencies occur in reciprocal pairs at the same angle; i.e., the radius of one zero is the reciprocal of the radius of the other. The zeros for stopband frequencies are on the unit circle.

If we were to augment $h[n]$ using the Herrman-Schüssler algorithm, we would add enough offset to all frequencies to lift the stopband amplitude function (not magnitude function) to be non-negative: $g[n] = A_{stop} \delta[n - 25] + h[n]$. Phase response (below) becomes a line without any discontinuities.

The freqz and zplane plots for $g[n]$ are given below. Its A_{stop} is -72.9 dB, as highlighted. Zeros over stopband frequencies are in reciprocal pairs—they are so close in value that they appear in the same location in the zplane plot. The zplane plot for one of reciprocal pairs of zeros is enlarged.



To form $V(z)$, we would keep the zeros inside the unit circle and normalize.

Software Receiver Design Application

Lectures 0 1 4 5 & 6

Labs 2 & 3

HW 0.2 0.4 1.1(d) 2.1(d) 1.3 3.1 3.2 & 3.3

JSK Sec. 2.1 to 2.9

JSK Ch. 3 4 5 6 & 7

Problem 1.3 Bluetooth Receiver. 24 points.

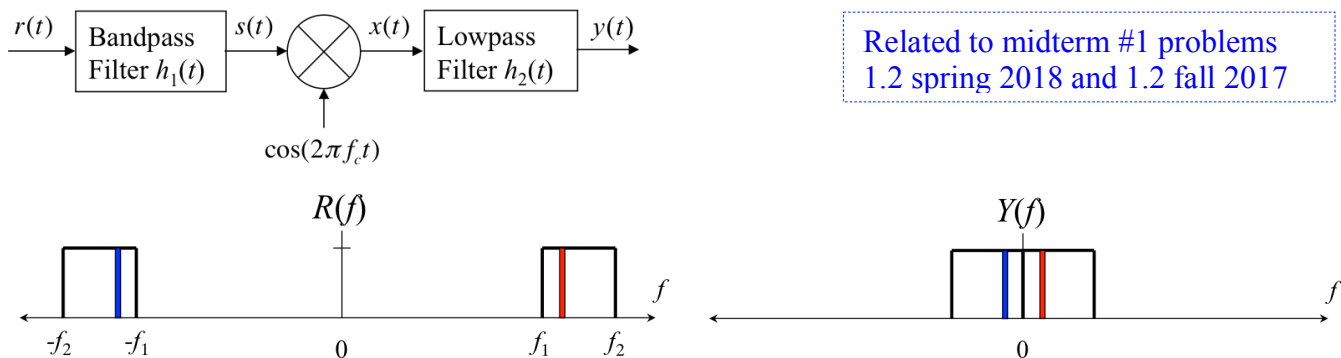
Bluetooth operates in the 2400-2499 MHz unlicensed band.

At any given time, Bluetooth will transmit on one of 79 channels, and each channel is 1 MHz wide.

Channel k begins at $(2402 + k)$ MHz where $k = 0, 1, \dots, 78$.

The Bluetooth receiver below has an analog/RF front end and a digital baseband receiver.

(a) **Analog/RF front end** block diagram is given below, where $r(t)$ is the received RF signal. In the plot for $R(f)$, one of the 1 MHz channels is shaded, and its counterpart in negative frequencies is also shaded. The spectrum of the analog/RF front end output signal, $y(t)$, is also shown. 6 points.



Related to midterm #1 problems
1.2 spring 2018 and 1.2 fall 2017

What is the carrier frequency f_c ?

$f_c = f_1$. Analog/RF front end performs sinusoidal amplitude demodulation to shift the positive frequency band in $R(f)$ to the left by f_1 and negative frequency band in $R(f)$ to the right by f_1 .

Note: This type of fixed analog/RF front end design would enable digital baseband processing to be programmed in software. The baseband performs the final demodulation stage for channel k . Bluetooth transmitters/receivers perform ~1600 hops/s in a pre-defined pattern in an attempt to avoid interference, and will stop using a channel that has strong noise/interference on it.

(b) Design a **second-order discrete-time** linear time-invariant (LTI) infinite impulse response (IIR) filter to extract channel k from $y[n]$ where $y(t)$ is sampled at a sampling rate of f_s to obtain $y[n]$ where $f_s \geq 200$ MHz. Give formulas for, and plot, the poles and zeros. 18 points.

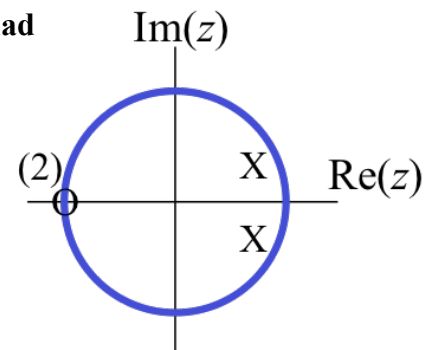
Due to the analog/RF front end in part (a), channel k in $y(t)$ resides between $(k+2)$ MHz and $(k+3)$ MHz. The center frequency is at $(k+2.5)$ MHz. Let $f_s = 200$ MHz.

To extract channel k , i.e. to receive channel k , we have been asked to use a second-order bandpass IIR filter. The center frequency would be $\omega_k = 2\pi \frac{(k+2.5) \text{ MHz}}{200 \text{ MHz}} = 2\pi \frac{k+2.5}{200}$.

A second-order filter has two poles and any number of zeros. A biquad would have two poles and two zeros.

Poles are at $p_0 = 0.9 e^{j\omega_k}$ and $p_1 = 0.9 e^{-j\omega_k}$

For small values of k , poles are close to 0 rad/sample. To avoid strong interaction between zeros and poles, place two zeros at $z = -1$.



Problem 1.4. Potpourri. 21 points.

(a) Compare the implementation complexity of an infinite impulse response (IIR) filter with N poles and N zeros for the different filter structures below. A biquad is a second-order IIR filter.

- i. Complete the table below while providing justification for each entry. Your answers would be in terms of N . 12 points.

Filter Structure	Number of coefficients	Number of input and output values to store	Multiplications per output sample	Additions per output sample
Direct Form	$2N+1$	$2N+2$	$2N+1$	$2N$
Cascade of Biquads (N even)	$5\left(\frac{N}{2}\right) = 2.5N$	$6\left(\frac{N}{2}\right) = 3N$	$5\left(\frac{N}{2}\right) = 2.5N$	$4\left(\frac{N}{2}\right) = 2N$
Cascade of Biquads (N odd)	$5\left(\frac{N-1}{2}\right) + 3$	$6\left(\frac{N-1}{2}\right) + 4$	$5\left(\frac{N-1}{2}\right) + 3$	$4\left(\frac{N-1}{2}\right) + 2$

Direct form calculation of the current output sample $y[n]$ would use the following:
 $y[n] = \sum_{m=1}^N a_m y[n-m] + \sum_{k=0}^N b_k x[n-k]$.

Direct form with two tapped delay lines stores $2N+2$ values, i.e. $N+1$ current/previous inputs and $N+1$ current/previous outputs (see lecture slide 6-9). Direct form with one tapped delay line stores $N+1$ intermediate values (see lecture slide 6-11).

Biquad i : $y_i[n] = a_1 y_i[n-1] + a_2 y_i[n-2] + b_0 x_i[n] + b_1 x_i[n-1] + b_2 x_i[n-2]$, which takes 5 multiplications and 4 additions. We would need to store the current input and two previous inputs, and the current output and two previous outputs.

When N is even, there are $N/2$ biquads in cascade.

When N is odd, there are $(N-1)/2$ biquads and a first-order IIR filter in cascade. A first-order IIR filter output is $y_i[n] = a_1 y_i[n-1] + b_0 x_i[n] + b_1 x_i[n-1]$.

- ii. What is the percentage increase in implementation complexity for the cascade of biquads vs. direct form when $N = 10$? 3 points.

Direct Form: 21 coefficients, 22 values to store, 21 multiplications, 20 additions

Cascade: 25 coefficients, 30 values to store, 25 multiplications, 20 additions.

19% more coefficients/multiplications. 36% more values to store. Same in additions.

(b) For infinite impulse response (IIR) filter orders greater than two, what is the primary advantage of using a cascade of biquads vs. a direct form filter structure? 6 points.

Using a cascade of biquads helps to ensure that a bounded-input bounded-output (BIBO) IIR filter remains BIBO stable after implementation. Classical IIR filter designs compute the pole and zero locations for a BIBO stable filter using closed-form formulas with high accuracy. When converting the factored form of the transfer function to unfactored form for direct form implementation, quantization error from the addition and multiplication operators can cause lead to error in the feedback and feedforward coefficient values. Refactoring the perturbed feedback coefficient values can yield poles on or outside of the unit circle, which means that the implementation is BIBO unstable.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Solutions 4.0*

Date: March 13, 2019

Course: EE 445S Evans

Name: _____ **Games,** **Brain**
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Sampling & Aliasing
3	24		Filter Design
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

Consider the following causal linear time-invariant (LTI) discrete-time filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = a x[n] + b x[n-1] - b x[n-2] - a x[n-3]$$

for $n \geq 0$, where a and b are real-valued positive coefficients.

- (a) Is this a finite impulse response (FIR) or infinite impulse response (IIR) filter? Why? 3 points.

FIR filter. Any of the following reasons would provide sufficient justification:

1. The impulse response extends for 4 samples from $n = 0$ to $n = 3$, which is finite in duration.
2. The output $y[n]$ does not depend on previous output values; i.e., there is no feedback.
3. In the transfer function in the z -domain in part (d), the only poles are trivial poles at $z = 0$.

- (b) What are the initial conditions and their values? Why? 6 points.

Let $n=0$: $y[0] = a x[0] + b x[-1] - b x[-2] - a x[-3]$.

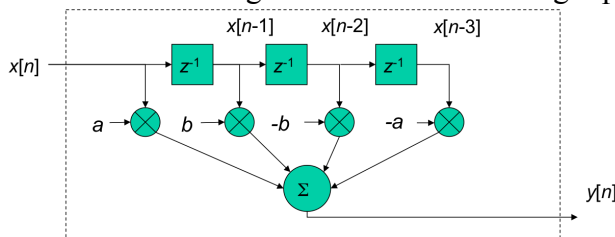
Let $n=1$: $y[1] = a x[1] + b x[0] - b x[-1] - a x[-2]$.

Let $n=2$: $y[2] = a x[2] + b x[1] - b x[0] - a x[-1]$. Etc.

Initial conditions are $x[-1]$, $x[-2]$, $x[-3]$ and must be zero for linearity and time-invariant properties to hold. Note that $x[0]$ is the first input value and not an initial condition.

Note: A causal system does not depend on future input values or future output values.

- (c) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



Note: The three initial conditions are visible here as the initial condition for each unit delay block.

Lecture slides 3-15 & 5-4

- (d) Derive a formula for the transfer function in the z -domain and the region of convergence. 4 points.

Z-transform both sides of difference equation, knowing that all initial conditions are zero:

$$Y(z) = a X(z) + b z^{-1} X(z) - b z^{-2} X(z) - a z^{-3} X(z) \text{ which means}$$

$$H(z) = \frac{Y(z)}{X(z)} = a + b z^{-1} - b z^{-2} - a z^{-3} \text{ for } z \neq 0$$

Lecture slide 5-11

- (e) Give a formula for the discrete-time frequency response of the filter. 3 points.

We can convert the transfer function $H(z)$ into the discrete-time frequency domain by substituting $z = \exp(j\omega)$ because FIR LTI systems are always Bounded-Input Bounded-Output stable, or equivalently, because the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = a + b e^{-j\omega} - b e^{-2j\omega} - a e^{-3j\omega}$$

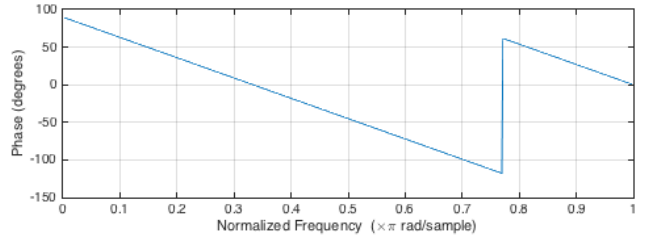
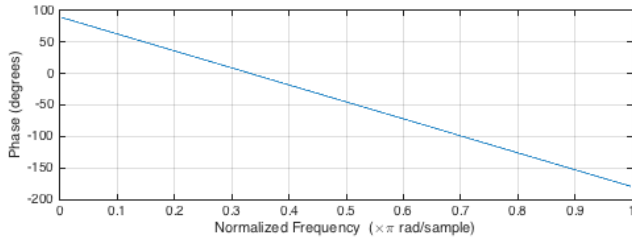
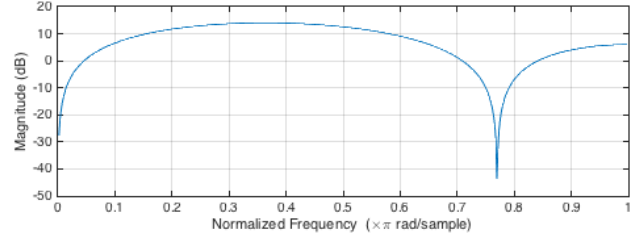
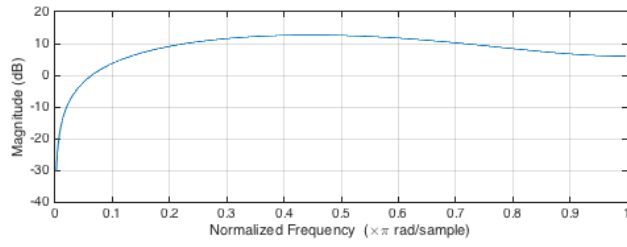
- (f) Give a formula for the phase response vs. discrete-time frequency and the group delay vs. discrete-time frequency. Does the filter have linear phase over all frequencies? Why or why not? 6 points.

$$H_{freq}(\omega) = e^{-j\frac{3}{2}\omega} \left(a e^{j\frac{3}{2}\omega} + b e^{j\frac{\omega}{2}} - b e^{-j\frac{\omega}{2}} - a e^{-j\frac{3}{2}\omega} \right) = 2 \left(a \sin\left(\frac{3}{2}\omega\right) + b \sin\left(\frac{\omega}{2}\right) \right) j e^{-j\frac{3}{2}\omega}$$

With $j = e^{j\frac{\pi}{2}}$, $\angle H_{freq}(\omega) = \frac{\pi}{2} - \frac{3}{2}\omega$ except for phase jumps (discontinuities) of π at frequencies that are zeroed out, which is generalized linear phase. $GD(\omega) = -\frac{d}{d\omega} \angle H_{freq}(\omega) = \frac{3}{2}$ samples.

Problem 1.1 Supplemental information not expected for students to have provided in their answers.

Matlab plots using `freqz([a b -b -a])` for $a = 1$ and $b = 2$ (left) and $a = 2$ and $b = 1$ (right)



Problem 1.2 Sampling and Aliasing. 24 points.

Lecture 1 & 4	Lab #2
HW 0.1 0.2 0.3	
Fall 2016 Midterm 1 Prob 2	

A frequency of 46 kHz is higher than the normal audible range of 20 Hz to 20 kHz for a human being.

Consider a continuous-time signal $x(t) = \cos(2 \pi f_0 t)$ where $f_0 = 46$ kHz.

Sample the signal using a sampling rate of $f_s = 48$ kHz.

- (a) Derive a formula for the discrete-time signal $x[n]$ that results from sampling $x(t)$. 3 points.

Sampling in the time domain can be modeled as an instantaneous closing and opening of a switch. Each time that the switch is closed, the input is gated to the output. In practice, this could be implemented by a pass transistor with a sampling clock feeding the gate terminal.

$$x[n] = x(t)|_{t=nT_s} = \cos(2 \pi f_0 (n T_s)) = \cos\left(2 \pi f_0 \left(\frac{n}{f_s}\right)\right) = \cos\left(2\pi \left(\frac{f_0}{f_s}\right) n\right)$$

The discrete-time frequency corresponding to continuous-time frequency f_0 is $\omega_0 = 2\pi \frac{f_0}{f_s}$

- (b) Using only analysis of $x[n]$ in the discrete-time domain, determine the discrete-time frequency to which the continuous-time frequency of f_0 will alias. 6 points.

$$x[n] = \cos\left(2\pi \left(\frac{f_0}{f_s}\right) n\right) = \cos\left(2\pi \left(\frac{46 \text{ kHz}}{48 \text{ kHz}}\right) n\right) = \cos\left(2\pi \left(\frac{23}{24}\right) n\right)$$

March 11th Lecture

We can subtract an offset in the argument of $2 \pi n$ without changing $x[n]$:

$$\cos\left(2\pi \left(\frac{23}{24}\right) n - 2\pi n\right) = \cos\left(2\pi \left(\frac{23}{24} - 1\right) n\right) = \cos\left(2\pi \left(-\frac{1}{24}\right) n\right) = \cos\left(2\pi \left(\frac{1}{24}\right) n\right)$$

Continuous-time frequency of f_0 will alias to a discrete-time frequency of $2\pi \frac{1}{24}$ rad/sample.

- (c) What is the equivalent continuous-time frequency for the aliased discrete-time frequency in (b)? 6 points.

With $\omega_1 = 2\pi \frac{f_1}{f_s}$ and $f_s = 48$ kHz, $f_1 = 2$ kHz.

- (d) Using only analysis in the continuous-time frequency domain of sampling applied to $x(t)$, determine the continuous-time frequency to which the continuous-time frequency f_0 will alias. The answer should be the same as part (c). 6 points.

In the time domain, we model instantaneous gating of input to output every T_s seconds as a multiplication of the input signal by an impulse train with impulses every T_s seconds. The output spectrum is the convolution of the input spectrum and an impulse train with impulses separated by f_s with area f_s . In the frequency domain, sampling creates replicas of the input spectrum at offsets of integer multiples of f_s . The Fourier transform of $\cos(2 \pi f_0 t)$ is $\frac{1}{2} \delta(f + f_0) + \frac{1}{2} \delta(f - f_0)$. Replicas are shown as dashed impulses below. Reconstructed frequencies are from $-\frac{1}{2} f_s$ to $\frac{1}{2} f_s$ and hence the aliased continuous-time frequency is 2 kHz.



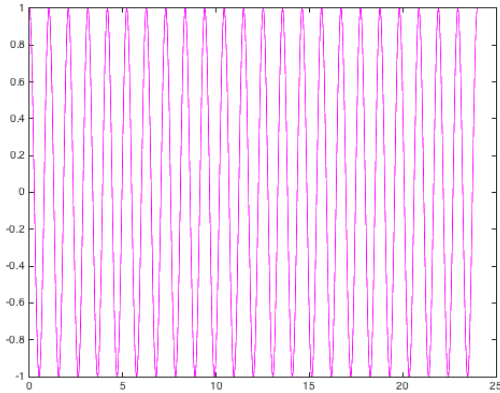
- (e) Is the aliased frequency audible? 3 points.

Yes, the aliased frequency of 2 kHz is in the audible range of 20 Hz to 20 kHz.

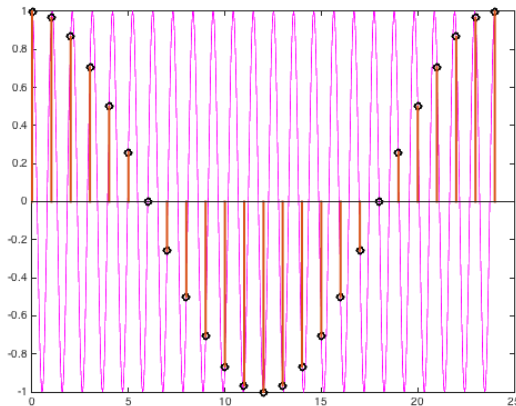
Problem 1.2 Supplemental information not expected for students to have provided in their answers.

Matlab code to show aliasing in the time domain

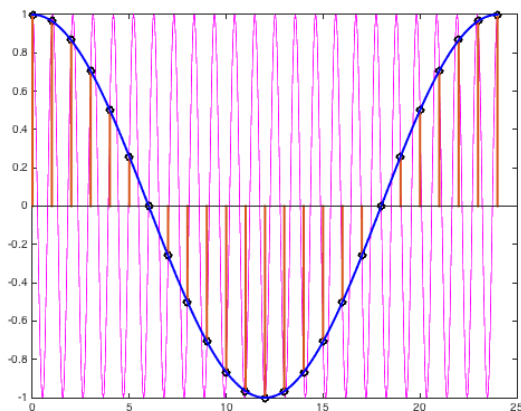
Plot $x(t) = \cos(2 \pi f_0 t)$



Plot samples $x(n T_s)$ superimposed on $x(t) = \cos(2 \pi f_0 t)$



Plot $x_1(t) = \cos(2 \pi f_1 t)$ and $x(n T_s)$ superimposed on $x(t) = \cos(2 \pi f_0 t)$



```

%% Part 1: Define Signals
wHat = 2*pi*(1/24);
nmax = 24;
n = 0:nmax;
x1 = cos(wHat*n);
x = cos(2*pi*(23/24)*n);

fs = 1;      %% fs=1 to align DT and CT
f1 = 2/48;  %% Actual fs goes in denom
w1Hat = 2*pi*f1/fs;
period = round(fs/f1);
f0 = 46/48; %% Actual fs goes in denom
w0Hat = 2*pi*f0/fs;
Ts = 1/fs;
tmax = (nmax/period)*(1/f1);
t = 0 : (Ts/100) : tmax;
x1cont = cos(2*pi*f1*t);
xcont = cos(2*pi*f0*t);

%% Part 2: Generate Plots
figure;
plot(t, xcont, 'm-', 'LineWidth', 1);

figure;
plot(t, xcont, 'm-', 'LineWidth', 1);
hold;
stem(n, x1, 'Linewidth', 2,
'MarkerEdgeColor', 'black');
stem(n, x, 'Linewidth', 2,
'MarkerEdgeColor', 'black');

figure;
plot(t, xcont, 'm-', 'LineWidth', 1);
hold;
stem(n, x1, 'Linewidth', 2,
'MarkerEdgeColor', 'black');
stem(n, x, 'Linewidth', 2,
'MarkerEdgeColor', 'black');
plot(t, x1cont, 'b-', 'LineWidth', 2);

```

Problem 1.3 Filter Design. 24 points.

An electrocardiogram (ECG) device records the heart's electrical potential versus time for monitoring heart health and diagnosing heart disorders. [1]

Use a sampling rate f_s of 240 Hz for the continuous-time ECG signal for a monitoring application. [1]

Design a third-order discrete-time infinite impulse response (IIR) filter to remove baseline wander noise below 0.5 Hz and powerline interference at 60 Hz in an ECG signal. [1]

Baseline wander noise is induced by electrode changes due to perspiration, movement and respiration.

The third-order discrete-time IIR filter will be a cascade of a first-order and a second-order section.

- (a) Design a first-order discrete-time IIR filter to remove DC (0 Hz) but pass as many of the other frequencies as possible with a gain of one in linear units. Please give the pole, zero, and gain. *6 points.*

Pole $p_0 = 0.95$ and zero $z_0 = 1$.

$$H_0(z) = C_0 \frac{1 - z_0 z^{-1}}{1 - p_0 z^{-1}} = C_0 \frac{z - z_0}{z - p_0}$$

Set $H_0(z) = 1$ at $z = \exp(j\pi) = -1$ to give $C_0 = 0.975$.

Fall 2014 Midterm 1 Prob 3 notch

Spring 2015 Midterm 1 Prob 3(b)

Spring 2016 Midterm 1 Prob 1

Lecture Slide 6-7 Filter Demos

- (b) Design a second-order discrete-time IIR filter to remove 60 Hz but pass as many of the other frequencies as possible with a gain of one in linear units. Please give the two poles, two zeros, and gain. *6 points.*

$$\omega_{60} = 2\pi \frac{f_{60}}{f_s} = 2\pi \frac{60 \text{ Hz}}{240 \text{ Hz}} = \frac{\pi}{2}$$

Poles and zeros are at the same angle ω_{60}

Poles at $p_1 = 0.9 e^{j\omega_{60}} = j0.9$ and $p_2 = 0.9 e^{-j\omega_{60}} = -j0.9$

Zeros at $z_1 = e^{j\omega_{60}} = j$ and $z_2 = e^{-j\omega_{60}} = -j$

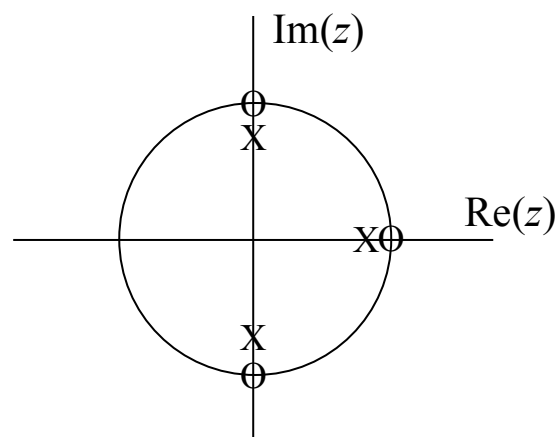
$$H_1(z) = C_1 \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})}$$

Normalize $H_1(z) = 1$ at $z = \exp(j0) = 1$ to give $C_0 = 0.905$.

- (c) Plot the poles and zeros for the third-order discrete-time IIR filter on the right. The circle on the right has a radius of 1. *6 points.*

- (d) What is the response of the discrete-time IIR filter to continuous-time frequencies in the ECG signal that are odd harmonics of 60 Hz, i.e. 180 Hz, 300 Hz, etc.? Why? *6 points.*

When sampled at the sampling rate of 240 Hz, continuous-time frequencies that are odd harmonics of 60 Hz will alias to the discrete-time frequency ω_{60} , and hence will be zeroed out by the discrete-time IIR filter.

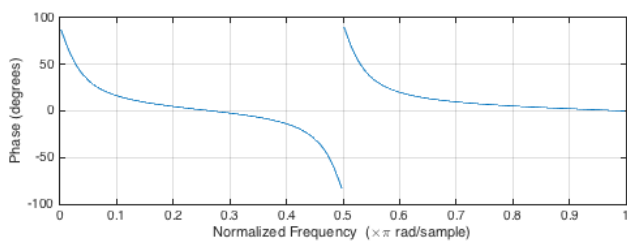
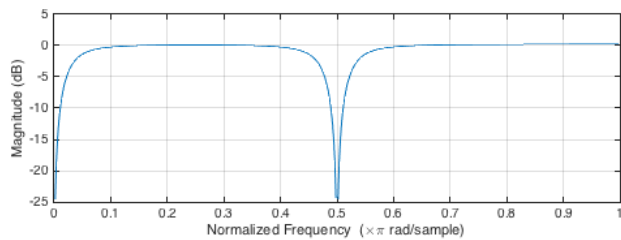


[1] Yong Lian and Jianghong Yu, "A Low Power Linear Phase Digital FIR Filter for Wearable ECG Device", *Proc. IEEE Int. Conf. on Engineering in Medicine and Biology Society*, pp. 7357-7360, 2005.

Problem 1.3 Supplemental information not expected for students to have provided in their answers.

Matlab code to specify and analyze the discrete-time IIR filter.

```
%% Zeros
z0 = 1;
z1 = j;
z2 = -j;
%% Poles
p0 = 0.9;
p1 = 0.9j;
p2 = -0.9j;
%% Gains for each stage
C0 = 0.975;
C1 = 0.905;
%% Expand factors to coefficients
zeros = [z0 z1 z2];
poles = [p0 p1 p2];
feedforwardCoeffs = C0*C1*poly(zeros);
feedbackCoeffs = poly(poles);
%% Filter frequency response
freqz( feedforwardCoeffs, feedbackCoeffs );
```



Problem 1.4. Potpourri. 24 points.

(a) A discrete-time signal with sampling rate of f_s of 8000 Hz has the following “UX” spectrogram. The spectrogram was computed using 1000 samples per block and an overlap of 900 samples.

i. Describe frequency components vs. time. *6 points.*

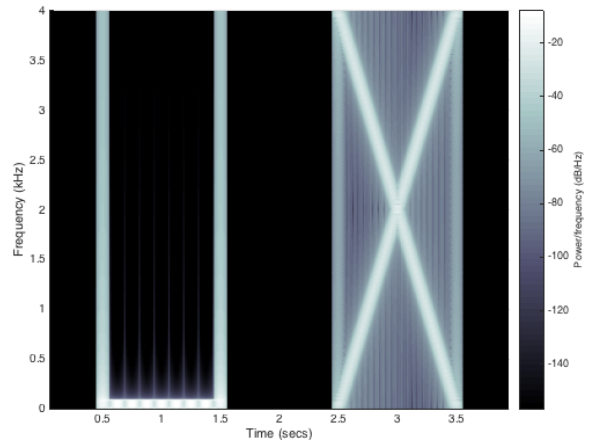
By using the intensity scale shown to the right of the spectrogram plot:

$t = 0.5s$: all frequencies present

$0.5s < t < 1.5s$: Low frequencies 0 to 0.1 kHz continuously present (in white) plus six less intense short bursts of frequencies 0 to 1 kHz equally spaced in time (short rect. pulses)

$t = 1.5s$: all frequencies present

$2.5s < t < 3.5s$: chirp increasing from 0 to $\frac{1}{2}f_s$ plus a chirp decreasing from $\frac{1}{2}f_s$ to 0



ii. What would the signal sound like when played as audio signal? *6 points.*

$0.5s < t < 1.5s$: Bass tones 20-100 Hz plus lower intensity 0-1 kHz freq. repeated 6 times

$2.5s < t < 3.5s$: Note increasing 0 to 4 kHz, and note decreasing 4 to 0 kHz, with time

(b) Consider an unknown causal, time-varying, nonlinear, discrete-time system with input $x[n]$ and output $y[n]$. We will model the system as a discrete-time linear time-invariant (LTI) finite impulse response (FIR) filter. Find the FIR coefficients.

i. Give a formula for a finite-length input signal other than an impulse that contains all frequencies. *3 points.*

Fall 2017 Midterm 1 Prob 4(a)

The discrete-time frequency domain has period 2π .

HW 1.2

Input signal $x[n]$ of N samples should contain all discrete-time frequencies from $-\pi$ to π . Use a chirp signal that linearly sweeps all frequencies from 0 to π :

$$x[n] = \cos\left(2\pi\left(\frac{n}{4N}\right)n\right) = \cos\left(\frac{2\pi}{4N}n^2\right) \quad \text{for } n = 0, 1, \dots, N-1$$

ii. Using your answer in part i, derive a time-domain algorithm to estimate the FIR filter coefficients. Your algorithm should also be able to determine how many FIR filter coefficients are meaningful. *9 points.*

In-class discussion Feb. 20th & 25th

Fall 2013 Midterm 1 Prob 3(a)

We base the algorithm on convolution:

$$y[n] = h[n] * x[n] = \sum_{k=0}^{K-1} h[k] x[n-k]$$

For each output value $y[n]$, we'll have one equation and one unknown $h[n]$:

$$y[0] = h[0] x[0] \quad \text{solve for } h[0] \text{ which works as long as } x[0] \text{ is not zero.}$$

$$y[1] = h[0] x[1] + h[1] x[0] \quad \text{solve for } h[1] \text{ which works as long as } x[0] \text{ is not zero}$$

$$\text{until } |h[n]| < 10^{-5} \text{ or } n = N$$

Alternate criterion to $|h[n]| < 10^{-5}$: $\sum_{k=0}^n |h[k]|^2 \geq 0.9R$ where $R = \frac{\sum_{m=0}^{N-1} |y[m]|^2}{\sum_{m=0}^{N-1} |x[m]|^2}$

The alternate criterion will be able to handle some of the FIR coefficients values being close to zero in absolute value without stopping the update of the

Problem 1.4 Supplemental information not expected for students to have provided in their answers

1.4(a) Matlab code to generate the spectrogram.

```

fs = 8000;
Ts = 1 / fs;
tmax = 4;
utSignal = zeros(1, tmax*fs);
tlsec = 0 : Ts : (1 - Ts);
%% Spectrogram parameters
Nfft = 1000;
Noverlap = 900;
%% Generate low frequency groups
f0 = fs / Nfft;
lowfcosines = zeros(1, length(tlsec));
for n = 1 : 10
    f1 = n*f0;
    lowfcosines = lowfcosines + cos(2*pi*f1*tlsec);
end
%% Create chirp signals
fstart = 0;
fend = fs/2;
fstep = fend - fstart;
phi = pi*fstep*(tlsec.^2);
upchirp = cos(2*pi*fstart*tlsec + phi);
downchirp = cos(2*pi*fend*tlsec - phi);
%% Draw U into spectrogram
utSignal(0.5*fs+1:1.5*fs) = lowfcosines;
%% Draw X into spectrogram
utSignal(2.5*fs+1:3.5*fs) = upchirp + downchirp;
%% Plot the spectrogram
spectrogram(utSignal, hamming(Nfft), Noverlap, Nfft, fs, 'yaxis');
colormap bone;

```

1.4(a)ii Matlab code to play the signal in the spectrogram in problem 1.4(a) as an audio signal

```
soundsc(utSignal, fs);
```

1.4(b)i Matlab code to generate chirp signal $x[n]$ of N samples in length. All frequencies are present in $x[n]$.

```

N = 10000;
n = 0 : N-1;
x = cos(((2*pi)/(4*N))*(n.^2));

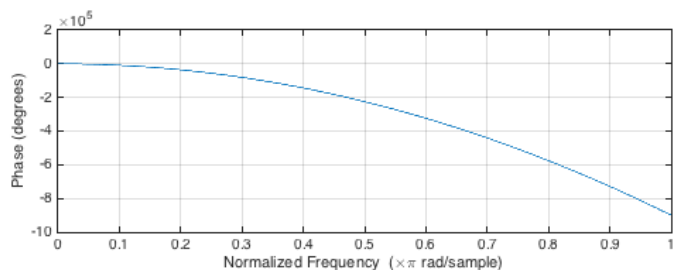
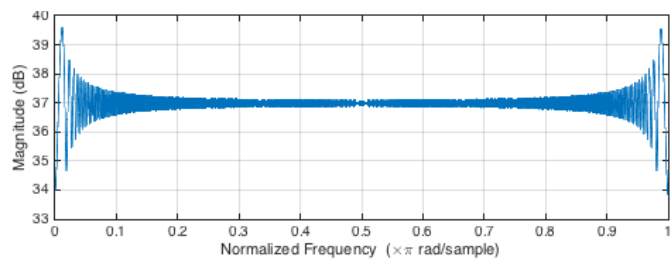
```

```
%% Plot frequency content in x
freqz(x, 1, N);
```

1.4(b)ii Although not asked, here are two frequency-domain algorithms.

Algorithm #1: Computer $H(z) = Y(z) / X(z)$, take inverse transform to find $h[n]$, and truncate $h[n]$ to keep 90% of energy or N coefficients, whichever is smaller.

Algorithm #2: Similar approach to Algorithm #1 using $H_{freq}(\omega) = Y_{freq}(\omega) / X_{freq}(\omega)$.



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Solutions 7.0*

Date: October 16, 2019

Course: EE 445S Evans

Name: _____ **Zappa,** _____ **Frank**
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

	<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
<i>Ahmet</i>	1	28		Filter Analysis
<i>Dweezil</i>	2	24		Pre-Distortion Filter Design
<i>Moon</i>	3	24		Acoustic Noise Reduction
<i>Diva</i>	4	24		Potpourri
	<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

Lectures 3 5 & 6	Lab #3
HW 0.4, 1.1 2.1 2.2, 2.3 & 3.2	
Midterm 1: Spring 2010 Prob 1, Fall 2016 Prob 1, Fall 2018 Prob 1	
JSK Ch. 7	

Consider the following causal linear time-invariant (LTI) discrete-time filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = x[n] - x[n-2]$$

for $n \geq 0$.

- (a) Is this a finite impulse response (FIR) or infinite impulse response (IIR) filter? Why? 3 points.

FIR filter. Any of the following reasons would provide sufficient justification:

1. The impulse response extends for 3 samples from $n = 0$ to $n = 2$, which is finite in duration.
2. The output $y[n]$ does not depend on previous output values; i.e., there is no feedback.
3. In the transfer function in the z -domain in part (d), the only poles are trivial poles at $z = 0$.

- (b) What are the initial conditions and their values? Why? 6 points.

Let $n=0$: $y[0] = x[0] - x[-2]$

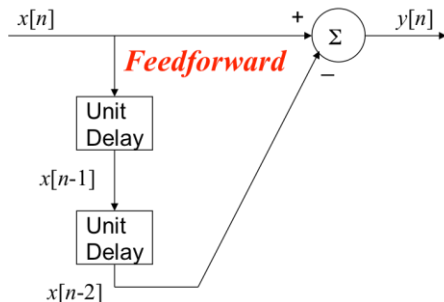
Let $n=1$: $y[1] = x[1] - x[-1]$

Let $n=2$: $y[2] = x[2] - x[0]$ etc.

Note: A causal system does not depend on future input values, or current/future output values.

Initial conditions are $x[-1]$ and $x[-2]$ and must be zero for linearity and time-invariant properties to hold. Note that $x[0]$ is the first input value and not an initial condition.

- (c) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



Note: The two initial conditions are visible here as the initial condition for each of the two unit delay blocks.

Note: Arrows are important because they indicate the order of calculations.

Lecture slides 3-15 & 5-4

- (d) Derive a formula for the transfer function in the z -domain and the region of convergence. 4 points.

Z-transform both sides of difference equation, knowing that all initial conditions are zero:

$$Y(z) = X(z) - z^{-2} X(z) \text{ which means that } H(z) = \frac{Y(z)}{X(z)} = 1 - z^{-2} \text{ for } z \neq 0$$

Lecture slide 5-11

- (e) Give a formula for the discrete-time frequency response of the filter. 3 points.

We can convert the transfer function $H(z)$ into the discrete-time frequency domain by substituting $z = \exp(j\omega)$ because FIR LTI systems are always Bounded-Input Bounded-Output stable, or equivalently, because the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = 1 - e^{-2j\omega}$$

- (f) Give a formula for the phase response vs. discrete-time frequency and the group delay vs. discrete-time frequency. Does the filter have linear phase over all frequencies? Why or why not? 6 points.

$$H_{freq}(\omega) = 1 - e^{-2j\omega} = e^{-j\omega} (e^{j\omega} - e^{-j\omega}) = 2 \sin(\omega) j e^{-j\omega} = 2 \sin(\omega) e^{j(-\omega + \frac{\pi}{2})}$$

$\angle H_{freq}(\omega) = -\omega + \frac{\pi}{2}$ except for phase jumps (discontinuities) of π at frequencies that are zeroed out, which is generalized linear phase. $GD(\omega) = -\frac{d}{d\omega} \angle H_{freq}(\omega) = 1$ sample.

Lecture slides 5-15 & 5-18

Midterm 1: Spring 2013 Prob 1(f), Fall 2013 1 Prob 3(b), Spring 2019 Prob 1(f)

The following expands on the solution for problem 1.1(f).

Additional explanation concerning phase discontinuities

We take the expression for the frequency response in amplitude and phase form

$$H_{freq}(\omega) = 2 \sin(\omega) e^{j(-\omega + \frac{\pi}{2})}$$

and convert it into magnitude and phase response. The amplitude form $2 \sin(\omega)$ is non-positive for $\omega \in (-\pi, 0]$ and non-negative for $\omega \in (0, \pi]$:

$$H_{freq}(\omega) = \begin{cases} (-2 \sin(\omega)) (-e^{j(-\omega + \frac{\pi}{2})}) & \omega \in (-\pi, 0] \\ 2 \sin(\omega) e^{j(-\omega + \frac{\pi}{2})} & \omega \in (0, \pi] \end{cases}$$

In the phase form, we can replace $-1 = e^{-j\pi}$ to give

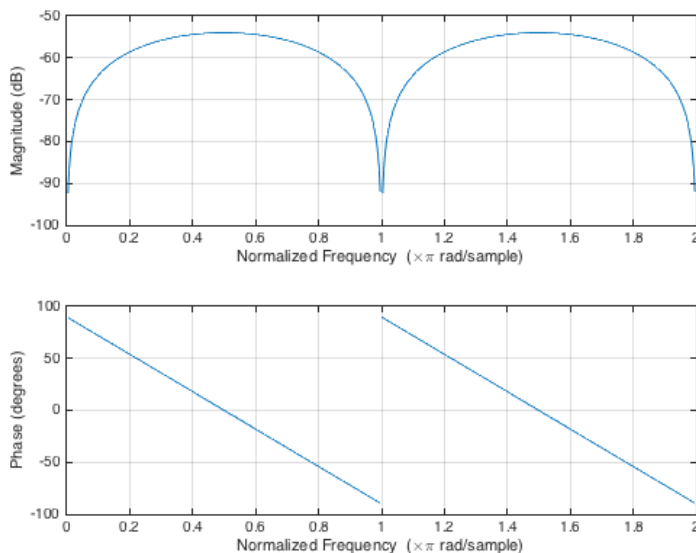
$$H_{freq}(\omega) = \begin{cases} (-2 \sin(\omega)) e^{j(-\omega - \frac{\pi}{2})} & \omega \in (-\pi, 0] \\ 2 \sin(\omega) e^{j(-\omega + \frac{\pi}{2})} & \omega \in (0, \pi] \end{cases}$$

Discrete-time frequency responses are periodic in ω with period 2π . Hence, there is a discontinuity in the phase response at integer multiples of π .

MATLAB code to plot magnitude and phase responses

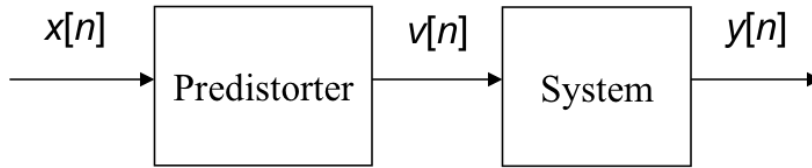
We can see the discontinuity in the phase response that occurs at integer multiples of π rad/sample by plotting the phase response from $[0, 2\pi]$ rad/sample:

`freqz([1 0 -1], 1000, 'whole')`



Problem 1.2 Predisortion Filter Design. 24 points.

A predisorter is used to compensate for distortion introduced by a system:



Lectures 5 & 6	Lab #3
HW 3.1 and 3.3	
JSK Ch. 7	
F06 Midterm 1.4	

The predisorter applies distortion to $x[n]$ that is the opposite of that particular distortion in the system, so that the distortion introduced by the predisorter cancels the distortion introduced by the system.

In this problem, both the predisorter and the system are

- Linear and time-invariant (LTI)
- Bounded-input bounded-output (BIBO) stable

Note: If the predisorter were placed after the system, then it would be called an equalizer. See Lecture Slides 5-3 & 6-6.

Each predisorter will be a first-order infinite impulse response (IIR) filter.

The goal in each part is to design a predisorter by placing its pole so that the cascade is all-pass.

All-pass cascade would mean $|G_{freq}(\omega)H_{freq}(\omega)| = g$ where g is a positive constant. A pole-zero pair in an all-pass configuration would have same angles and reciprocal magnitudes, or would have the same value and cancel out. Both filters are causal.

(a) The system has a transfer function $H(z) = 1 - 0.5z^{-1}$. 8 points.

- Give the transfer function of the predisorter, $G(z)$.
- Plot the pole and zero for the product $G(z)H(z)$ on the right.

$H(z)$ has a zero at $z = 0.5$. A pole at $z = 2$ for $G(z)$ would give a BIBO unstable system. $G(z)$ would have a pole at the same location $z = 0.5$.

$G(z) = \frac{1}{1-0.5z^{-1}}$ and $G(z)H(z) = 1$ which doesn't have poles or zeros.

(b) The system has a transfer function $H(z) = 1 - z^{-1}$. 8 points.

- Give the transfer function of the predisorter, $G(z)$.
- Plot the pole and zero for the product $G(z)H(z)$ on the right.

$H(z)$ has a zero at $z = 1$. A pole at $z = 1$ for $G(z)$ would give a BIBO unstable system. Place the pole at the same angle as the zero but at a radius of 0.9 to give a notch configuration.

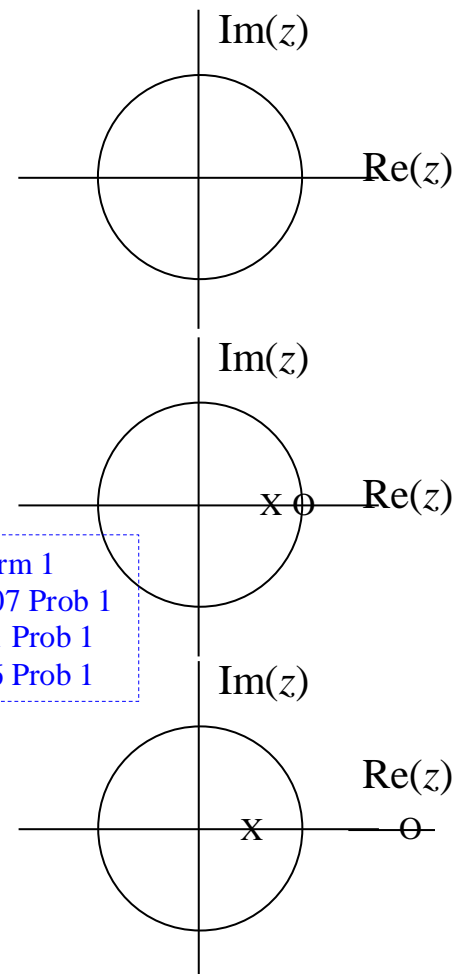
$G(z) = \frac{1}{1-0.9z^{-1}}$ and hence $G(z)H(z) = \frac{1-z^{-1}}{1-0.9z^{-1}}$

(c) The system has a transfer function $H(z) = 1 - 2z^{-1}$. 8 points.

- Give the transfer function of the predisorter, $G(z)$.
- Plot the pole and zero for the product $G(z)H(z)$ on the right.

$H(z)$ has a zero at $z = 2$. $G(z)$ would have a pole at $z = 0.5$.

$G(z) = \frac{1}{1-0.5z^{-1}}$ and hence $G(z)H(z) = \frac{1-2z^{-1}}{1-0.5z^{-1}}$

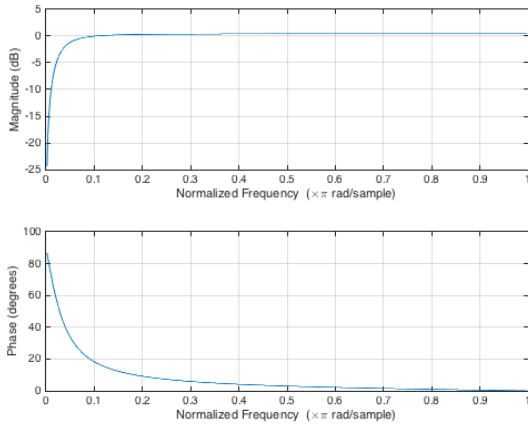


Midterm 1
Spring 2007 Prob 1
Fall 2011 Prob 1
Fall 2016 Prob 1

MATLAB results for the solutions for problem 1.2.

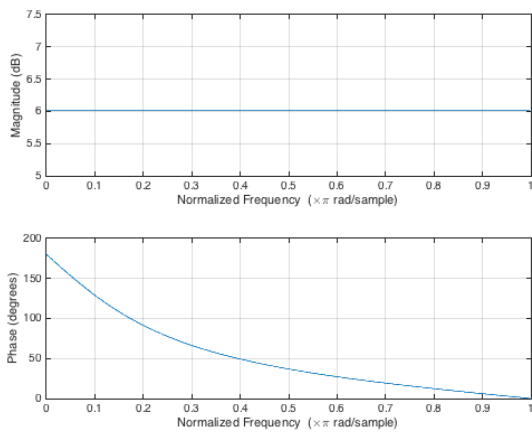
(b) We had seen a DC notch filter during lecture 6 at the five-second mark in the animation [“IIR filter with one pole and one zero”](#) that relates the time, z, and frequency domains (see demos)

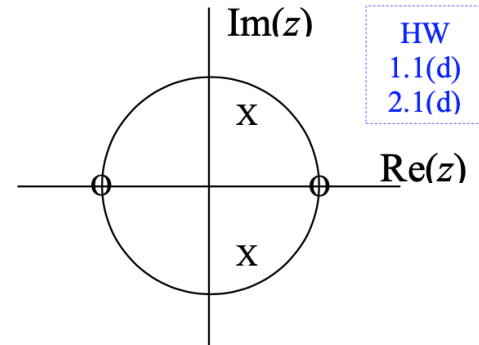
freqz([1 -1], [1 -0.9])



(c) This is an all-pass configuration as discussed on lecture slide 6-6.

freqz([1 -2], [1 -0.5])





Problem 1.3 Acoustic Noise Reduction. 24 points.

A car's audio system allows connection with a phone for hands-free use.

Inside the car, the primary sources of acoustic noise are from the engine and air conditioner.

Design discrete-time infinite impulse response (IIR) filters to be applied in cascade to the output of the microphone in the car's audio system to reduce acoustic noise when the phone is in use.

Assume a sampling rate of $f_s = 8$ kHz.

- (a) **Filter #1.** An air conditioner emits acoustic noise uniformly between 0 Hz and 4000 Hz. Primary speech frequencies are from 80 Hz to 3000 Hz. Give formulas for the two poles, two zeros, and gain of a discrete-time second-order IIR filter to reduce the air conditioning noise and improve the signal-to-noise ratio of the speech signal, and plot the poles and zeros on the right. 9 points.

Use bandpass filter w/ passbands 80 to 3000 Hz & -80 to -3000 Hz. Place pole at center of each passband and zeros at 0 & 4000 Hz.

$$\omega_c = 2\pi \frac{(80\text{Hz}+3000\text{Hz})/2}{8000 \text{ Hz}} = 2\pi \frac{77}{200} \text{ and } p_0 = 0.8e^{j\omega_c}, p_1 = 0.8e^{-j\omega_c}, z_0 = 1, \text{ and } z_1 = -1.$$

Solve for gain C by setting $H(z) = 1$ at $z = e^{j\omega_c}$ where $H(z) = C \frac{(1-z_0z^{-1})(1-z_1z^{-1})}{(1-p_0z^{-1})(1-p_1z^{-1})}$

- (b) **Filter #2.** The engine emits acoustic noise with two principal frequencies: the engine's rotational speed and its third harmonic. The engine's rotation speed in Hz, $f_{\text{Eng}}(t)$, varies over time.

- i. The current rotational speed of the engine in revolutions per minute (RPM) is $\Omega_{\text{RPM}}(t)$. Give formulas for $f_{\text{Eng}}(t)$ and its third harmonic $3f_{\text{Eng}}(t)$, in Hz, in terms of $\Omega_{\text{RPM}}(t)$. 3 points

Since 1 RPM = (1/60) Hz, we have: $f_{\text{Eng}}(t) = \frac{1}{60} \Omega_{\text{RPM}}(t)$ and $3f_{\text{Eng}}(t) = \frac{1}{20} \Omega_{\text{RPM}}(t)$

- ii. What is the highest rotational speed (in RPM) of the engine before aliasing of the third harmonic $3f_{\text{Eng}}(t)$ occurs? 3 points.

Aliasing will occur when $2f_{\text{max}} \geq f_s$. Third harmonic will not alias if $6f_{\text{Eng}}(t) \geq f_s$:

$$\frac{1}{10} \Omega_{\text{RPM}}(t) \geq 8000 \text{ Hz. Highest engine rotational speed before aliasing occurs is 80,000 RPM.}$$

- iii. Design a fourth-order discrete-time IIR filter to remove both principal frequencies, $f_{\text{Eng}}(t)$ and $3f_{\text{Eng}}(t)$, of the engine noise assuming $\Omega_{\text{RPM}}(t) = 2400$ RPM. Please specify the four poles, four zeros, and gain, and plot the poles and zeros below. 9 points

For notch at $f_{\text{Eng}}(t)$, $\omega_1 = 2\pi \frac{f_{\text{Eng}}(t)}{f_s} = 2\pi \frac{2400\text{RPM}}{8000 \text{ Hz}} \times \frac{1 \text{ Hz}}{60\text{RPM}} = \frac{1}{100} \pi$

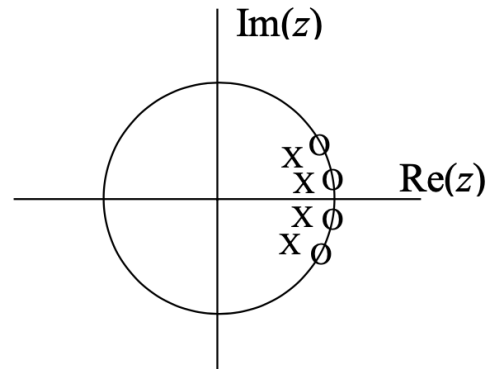
For notch at $3f_{\text{Eng}}(t)$, $\omega_3 = 3\omega_1 = \frac{3}{100} \pi$

Zeros at $z_0 = e^{j\frac{\pi}{100}}$, $z_1 = e^{-j\frac{\pi}{100}}$, $z_2 = e^{j\frac{3\pi}{100}}$, and $z_3 = e^{-j\frac{3\pi}{100}}$.

Poles at $re^{j\frac{\pi}{100}}$, $re^{-j\frac{\pi}{100}}$, $re^{j\frac{3\pi}{100}}$, and $re^{-j\frac{3\pi}{100}}$ where $r = 0.9$.

To solve for gain C , normalize $H(z)$ at $z = e^{j\pi} = -1$ and $H(-1) = 1$ where

$$H(z) = C \frac{(1 - z_0z^{-1})(1 - z_1z^{-1})(1 - z_2z^{-1})(1 - z_3z^{-1})}{(1 - p_0z^{-1})(1 - p_1z^{-1})(1 - p_2z^{-1})(1 - p_3z^{-1})}$$



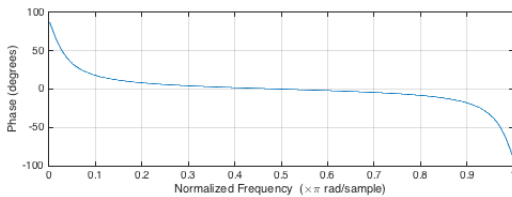
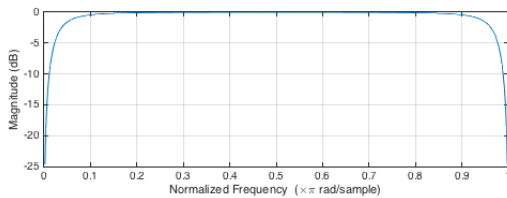
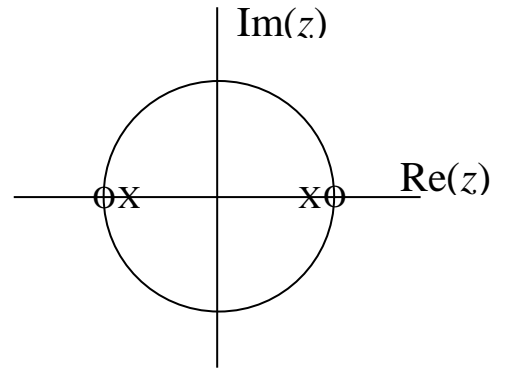
Problem 1.3. Alternate student solution for part (a).

We can use a double notch with real-valued poles and zeros if they are designed carefully.

$$p_0 = 0.9, p_1 = -0.9, z_0 = 1, \text{ and } z_1 = -1$$

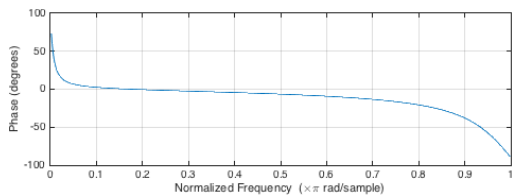
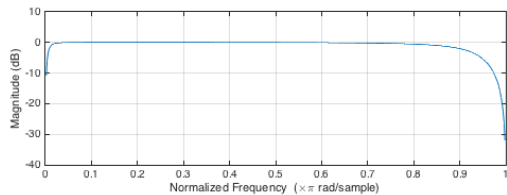
Find C by normalizing $H(z)$ at $z = e^{j\pi/2} = j$ which gives $C = 0.905$.

`freqz(0.905*[1 0 -1], [1 0 -0.81])`



Passband is from 400 to 3600 Hz. We can move the pole at $z = 0.9$ closer to the unit circle to narrow the stopband at 0 rad/sample, and move the pole at $z = -0.9$ further away from the unit circle to widen the stopband at π rad/sample. The following has a passband from 80 to 2920 Hz:

```
z0 = 1; z1 = -1;
numer = [1 -(z0+z1) z0*z1];
p0 = 0.98; p1 = -0.78;
denom = [1 -(p0+p1) p0*p1];
z = j;
C = abs(denom * [1 z^(-1) z^(-2)]') / (numer * [1 z^(-1) z^(-2)]');
freqz(C*numer, denom);
```



Note: Having real-valued poles has a distinct advantage in implementation because the quality factor for each real-valued pole is at the minimum value of 0.5. The time response due to a real-valued pole does not have an oscillating component and only has exponential decay. See Lab #3 Part 3 and Lecture Slides 6-16 and 6-17.

Problem 1.4. Potpourri. 24 points.

(a) Let $x(t) = \cos(2\pi f_0 t)$ be a continuous-time signal for $-\infty < t < \infty$. 6 points.

- i. From the block diagram below, derive a formula for $y(t)$ and write it as a sum of cosines.

$$x(t) \longrightarrow \boxed{(\cdot)^2} \longrightarrow y(t) \quad y(t) = x^2(t) = \cos^2(2\pi f_0 t) = \frac{1}{2} + \frac{1}{2} \cos(2\pi(2f_0)t)$$

- ii. Let $f_0 = 3000$ Hz. What negative, zero, and positive frequencies are present in $y(t)$?

Frequencies present are $-2f_0$, 0, and $2f_0$, i.e. -6000 Hz, 0 Hz, and 6000 Hz.

(b) Let $x(t) = \cos(2\pi f_0 t)$ be a continuous-time signal for $-\infty < t < \infty$. 12 points.

- i. Derive a formula for the discrete-time signal $x[n]$ obtained from sampling $x(t)$ at a sampling rate of f_s .

$$x[n] = x(t)|_{t=nT_s} = \cos(2\pi f_0(nT_s)) = \cos\left(2\pi \frac{f_0}{f_s} n\right)$$

F16 Midterm 1.2

- ii. Give a formula for the discrete-time frequency ω_0 of $x[n]$ in terms of f_0 and f_s .

$$\omega_0 = 2\pi \frac{f_0}{f_s}$$

- iii. From the block diagram below, derive a formula for $y[n]$ and write it as a sum of cosines.

$$x[n] \longrightarrow \boxed{(\cdot)^2} \longrightarrow y[n] \quad y[n] = x^2[n] = \cos^2(\omega_0 n) = \frac{1}{2} + \frac{1}{2} \cos(2\pi(2\omega_0)n)$$

- iv. Let $f_0 = 3000$ Hz and $f_s = 8000$ Hz. What negative, zero and positive discrete-time frequencies are present in $y[n]$ between $-\pi$ rad/sample and π rad/sample?

From part (a) ii, frequencies present are -6000 Hz, 0 Hz, and 6000 Hz. The component at -6000 Hz will alias to 2000 Hz, and the component at 6000 Hz will alias to -2000 Hz.

The discrete-time frequencies are $-\frac{\pi}{2}$, 0, and $\frac{\pi}{2}$ in units of rad/sample.

(c) Consider a 10th order infinite impulse response (IIR) filter with 10 complex-valued poles in conjugate pairs (i.e. $\alpha \pm j\beta$) and 10 complex-valued zeros in conjugate pairs. None of the poles is real-valued. None of the zeros is real-valued. 6 points. **Assume input signal is real-valued.**

- i. If the filter were implemented as a cascade of biquads (i.e. second-order sections), how many real-valued multiplications would be needed?

HW 3.3

Each biquad has a conjugate symmetric pair of poles and a conjugate symmetric pair of zeros; hence, it has five real-valued coefficients in its difference equation.

5 biquads x 5 real multiplications/biquad = 25 real multiplications

- ii. If the filter were implemented as a cascade of first-order sections, how many real-valued multiplications would be needed?

A first-order section has three complex-valued coefficients. Pole is located at $z = a_1$.

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

Each complex-valued multiplication takes four real-valued multiplications.

$$(a + jb)(c + jd) = (ac - bd) + j(bc + ad)$$

10 first-order sections x 12 real multiplications/section = 120 real multiplications

The next page has a more detailed answer that would find 95 real multiplications.

We can refine the answer to problem 1.4(c)2 to take into account that the input signal to the cascade of 10 first-order filters is real-valued, the output of the cascade is real-valued, and gain for each section is real-valued. This is a more detailed answer than expected on a midterm exam.

The difference equation for one of the first-order sections is

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

Since the first-order section has complex-valued pole p_0 , complex-valued zero z_0 , and real-valued gain b_0 , the feedback coefficient $a_1 = p_0$ is complex-valued. The zero location is at $-b_1/b_0$, so b_1 is complex-valued since b_0 is real-valued.

Multiplying a real number and a complex-number takes two real-multiplications.

For the first first-order section, the input $x[n]$ is real-valued and the output $y[n]$ is complex-valued. Hence, the first first-order section will require 4 real multiplications for $a_1 y[n-1]$, 1 real multiplication for $b_0 x[n]$, and 2 real multiplications for $b_1 x[n-1]$. This takes a total of 7 real multiplications.

The next 8 remaining first-order sections will each take $4 + 2 + 4 = 10$ real multiplications, for a total of 80 real multiplications.

In the last first-order section, $y[n]$ will be real-valued and the section will take 8 real multiplications.

The cascade of 10 first-order sections will take 95 real multiplications.

Additional explanation for problem 1.4(c). Not expected for students to include in their answers.

First-Order IIR Filter LTI \rightarrow Initial Conditions are zero

$$y[n] = b_0 x[n] + b_1 x[n-1] + a_1 y[n-1] \Rightarrow H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}}$$

Pole is at $z = a_1$ Zero is at $z = -\frac{b_1}{b_0}$

Second-Order IIR Filter (Biquad)

Input $x[n]$ and output $y[n]$ are real-valued, then the coefficients $\{b_0, b_1, b_2, a_1, a_2\}$ would be real-valued.

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + a_1 y[n-1] + a_2 y[n-2]$$

5 coefficients \rightarrow 5 real-valued multiplications
 6 words of memory \rightarrow 4 real-valued additions

All poles are either in conjugate symmetric pairs or are real-valued.
 Same goes for the zeros.

Quadratic formula $ax^2 + bx + c = 0$

$$\text{roots} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Cascade of Biquads

Poles: $\alpha \pm j\beta$
 Zeros: $\gamma \pm j\lambda$

If $a, b,$ and c are real-valued, then $\alpha = -\frac{b}{2a}$ is real-valued and $\beta = \frac{\sqrt{b^2 - 4ac}}{2a}$ is real if $b^2 - 4ac \geq 0$ or imaginary otherwise. Roots are at $\alpha \pm j\beta$ which gives two real roots or conjugate symmetric roots.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Solution Set 2.0*

Date: March 11, 2020

Course: EE 445S Evans

Name: _____
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed. ***Please disable all connections from your calculator to other electronic devices.***
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		Filter Analysis
2	24		Sampling
3	24		Audio Filter Design
4	24		Potpourri
<i>Total</i>	100		

Problem 1.1 Filter Analysis. 28 points.

Consider the following causal linear time-invariant (LTI) discrete-time filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = a x[n] + b x[n-1] + c x[n-2]$$

for $n \geq 0$. Coefficients a , b and c are real-valued. In addition, $a \neq 0$ and $c \neq 0$.

- (a) Is this a finite impulse response (FIR) or infinite impulse response (IIR) filter? Why? 3 points.

FIR filter. Any of the following reasons would provide sufficient justification:

1. The impulse response extends for 3 samples from $n = 0$ to $n = 2$, which is finite in duration.
2. The output $y[n]$ does not depend on previous output values; i.e., there is no feedback.
3. In the transfer function in the z -domain in part (d), the only poles are trivial poles at $z = 0$.

- (b) What are the initial conditions and their values? Why? 6 points.

Let $n=0$: $y[0] = a x[0] + b x[-1] + c x[-2]$

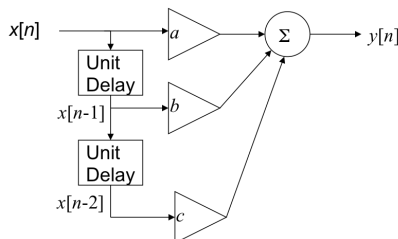
Let $n=1$: $y[1] = a x[1] + b x[0] + c x[-1]$

Let $n=2$: $y[2] = a x[2] + b x[1] + c x[0]$ and so forth.

Initial conditions are $x[-1]$ and $x[-2]$ and must be zero for linearity and time-invariant properties to hold. Note that $x[0]$ is the first input value and not an initial condition.

Note: A causal system does not depend on future input values, or current/future output values.

- (c) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



Note: The two initial conditions are visible here as the initial condition for each of the two unit delay blocks.

Lecture slides
3-15, 5-4 & 6-5

Note: Arrows are important because they indicate the order of calculations.

- (d) Derive a formula for the transfer function in the z -domain and the region of convergence. 4 points

Z-transform both sides of difference equation, knowing that all initial conditions are zero:

$$Y(z) = a X(z) + b z^{-1} X(z) + c z^{-2} X(z) \text{ and } H(z) = \frac{Y(z)}{X(z)} = a + b z^{-1} + c z^{-2} \text{ for } z \neq 0$$

- (e) Give a formula for the discrete-time frequency response of the filter. Justify the steps. 3 points.

We can convert the transfer function $H(z)$ into the discrete-time frequency domain by substituting $z = \exp(j\omega)$ because FIR LTI systems are always Bounded-Input Bounded-Output stable, or equivalently, because the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z) \Big|_{z=e^{j\omega}} = a + b e^{-j\omega} + c e^{-2j\omega}$$

- (f) Determine formulas for the relationships among the filter coefficients to make the filter have generalized linear phase over all frequencies. Give a numeric value for each coefficient to achieve generalized linear phase, and indicate what the frequency selectivity is. **Hint: Generalized linear phase means that the impulse response is odd symmetric about its midpoint.** 6 points.

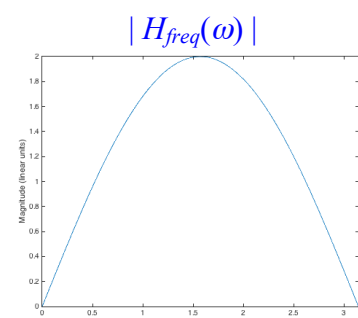
For this question: $h[n] = a \delta[n] + b \delta[n-1] + c \delta[n-2]$.

Odd symmetry $a = -c$ and $b = 0$. Values: $a = 1, b = 0, c = -1$.

$H(z) = 1 - z^{-2} = (1 - z^{-1})(1 + z^{-1})$. Two poles at $z = 0$.

Zeros at $\omega = 0$ rad/sample ($z = 1$) and $\omega = \pi$ rad/sample ($z = -1$) indicate stopbands. Bandpass frequency selectivity (see right).

Note the first-order differentiator has odd symmetry: $h[n] = \delta[n] - \delta[n-1]$

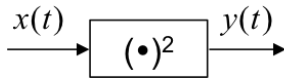


Problem 1.2. Sampling. 24 points.

For each problem below, determine the frequency (or frequencies) present in $x(t)$ and $y(t)$ as well as the single sampling rate you would use for the entire system to prevent aliasing.

Please note that $T_c = 1 / f_c$ and $T_0 = 1 / f_0$ in the following. *Each problem is worth 6 points.*

- (a) Let $x(t) = \cos(2\pi f_c t)$ be a continuous-time signal for $-\infty < t < \infty$.



Midterm #1:
Fall 2019 Prob 1.4(a)

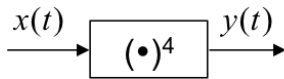
$x(t)$ has frequencies $-f_c$ and $+f_c$. $y(t) = x^2(t) = \frac{1}{2} + \frac{1}{2} \cos(2\pi (2f_c) t)$.

Or one could determine $y(t) = x(t)x(t)$ by computing $Y(f) = X(f)X(f)$ and inverse transform.

$y(t)$ has frequencies $-2f_c$, 0 , and $+2f_c$. Here, $f_{\max} = 2f_c$. Sampling Theorem : $f_s > 2f_{\max}$.

Note: Because the component at $2f_c$ in $y(t)$ is a cosine, one could use $f_s \geq 2f_{\max}$.

- (b) Let $x(t) = \cos(2\pi f_c t)$ be a continuous-time signal for $-\infty < t < \infty$.



$x(t)$ has frequencies $-f_c$ and $+f_c$.

Cascade of two squaring blocks. First squaring block gives frequencies $-2f_c$, 0 , and $+2f_c$.

Second squaring block: $-4f_c$, $-2f_c$, 0 , $+2f_c$, $+4f_c$. Here, $f_{\max} = 4f_c$. Choose $f_s > 2f_{\max}$

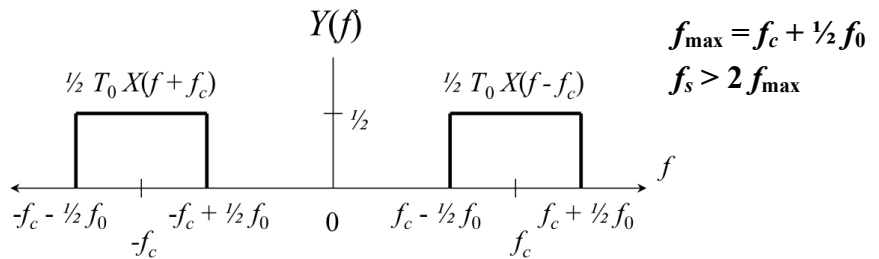
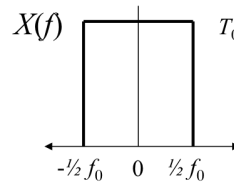
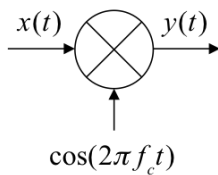
- (c) Let

$$x(t) = \text{sinc}\left(\frac{t}{T_0}\right)$$

be a continuous-time signal for $-\infty < t < \infty$
whose continuous-time Fourier transform is

$$X(f) = T_0 \text{rect}\left(\frac{f}{f_0}\right)$$

Here, $f_c > f_0$



$$f_{\max} = f_c + \frac{1}{2} f_0$$

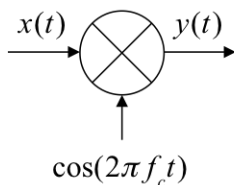
$$f_s > 2f_{\max}$$

- (d) Let

$$x(t) = \cos(2\pi f_c t) \text{sinc}\left(\frac{t}{T_0}\right)$$

be a continuous-time signal for $-\infty < t < \infty$

where $f_c > f_0$



$x(t)$ is $y(t)$ from part (c)

Frequencies in $X(f)$ are the same as those in $Y(f)$ in part (c)

For part (d), $Y(f)$ is the spectrum $X(f)$ shifted left by f_c and right by f_c (and each copy is scaled by $\frac{1}{2}$)

$$f_{\max} = 2f_c + \frac{1}{2} f_0$$

$$f_s > 2f_{\max}$$

Problem 1.3 Audio Filter Design. 24 points.

This problem asks you to evaluate tradeoffs in two designs for a filter for a tweeter/treble speaker:

- Speaker plays frequencies from roughly 2,000 Hz to 20,000 Hz.
- A discrete-time highpass filter will be placed in the speaker before the digital-to-analog (D/A) converter
- The D/A converter operates at a sampling rate of 48,000 Hz.

Highpass filter design specifications:

- Stopband frequency of 1800 Hz and passband frequency of 2000 Hz
- Stopband attenuation of 80 dB and passband tolerance of 1 dB
- Sampling rate of 48,000 Hz.

Proposed filter design #1: Finite Impulse Response (FIR) Filter.

Parks-McClellan (Equiripple) design. Order = 660. Meets specifications.

Proposed filter design #2: Infinite Impulse Response (IIR) Filter.

Elliptic (Equiripple) design. Order = 10. Meets specifications

- (a) Assuming the FIR filter is in direct form and the IIR filter is in a cascade of biquads (second-order sections), compute the number of multiplications per sample required by each. *6 points.*

FIR: $y[n] = h_0 x[n] + h_1 x[n - 1] + \dots + h_{660} x[n - 660]$

Needs 661 multiplications per output sample.

Order is 660: $H(z) = h_0 + h_1 z^{-1} + \dots + h_{660} z^{-660}$

IIR: Cascade of 5 biquads to get 10th order filter.

Biquad has 3 feedback and 2 feedforward coefficients,

Total: 5 x 5 = 25 multiplications per output sample.

- (b) For the IIR filter design, the group delay for frequencies greater than 4,000 Hz is less than 7 samples. What is the group delay for the FIR filter in the same range? *6 points*

The Parks-McClellan FIR filter design would have linear phase.

Its impulse response is even symmetric about its midpoint:

$$GD(\omega) = -\frac{d}{d\omega} \angle H(\omega) = \frac{Order}{2} = \frac{N-1}{2} = 330 \text{ samples}$$

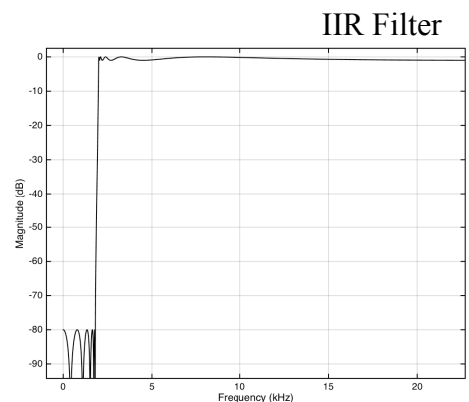
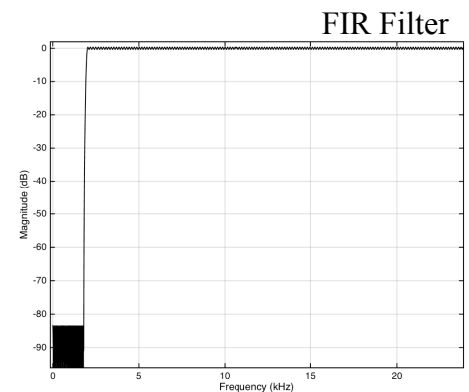
- (c) For the IIR filter design, the largest group delay of 64–500 samples occurred over the range of 2000 Hz to 2200 Hz. Is there a way you would recommend to alter the filter specifications so that the group delay would be less than 64 throughout the entire passband? *6 points*

Max group delay (GD) over passband determined by using Matlab filter design analysis tool:

1. Shift stopband freq. to 1600 Hz and passband freq. to 1800 Hz. Max GD = 67.5 samples
2. Shift stopband freq. to 500 Hz but keep passband freq. at 2000 Hz. Max GD = 53 samples
3. Reduce stopband attenuation to 12dB. Not very selective. Max GD = 38.9 samples
4. Reduce filter order to 4 (i.e. reduce stopband freq. to 450 Hz). Max GD = 34 samples

- (d) Which proposed filter design would you advocate using? *6 points.*

For the real-time application, use an IIR filter because it has 26 times lower complexity. In audio, both linear phase and group delay are important. The IIR filter has near linear phase over the passband. With adjustments in part (c), the IIR group delay is 5 times lower.



HW 1.3 2.3 3.2

Problem 1.4. Potpourri. 24 points.

- (a) You'd like to design a low-complexity lowpass finite impulse response (FIR) filter with an integer group delay. The two-tap averaging filter is a low-complexity lowpass FIR filter, but it has a group delay of $\frac{1}{2}$ sample. Design two different low-complexity lowpass FIR filters with integer group delays based on the two-tap averaging filter. *6 points.*

For a linear phase FIR filter of N coefficients, the group delay is $(N - 1) / 2$ samples. To have an integer group delay, N would need to be odd. For computational complexity, an FIR filter takes N multiplications per output sample. The lowest complexity possible is $N = 3$.

Linear phase means the impulse response is either even or odd symmetric about its midpoint.

For a three-coefficient lowpass FIR filter, we could use a

- i. three-coefficient averaging filter whose coefficients are $\{1, 1, 1\}$ or $\{1/3, 1/3, 1/3\}$.**
 - ii. create a three-coefficient lowpass filter by cascading two two-tap averaging filters. The resulting impulse response would be $\{1, 2, 1\}$ by convolving $\{1, 1\}$ with itself.**
- (b) Your system only has the ability to generate half of the carrier frequency you need for a communication system. What signal processing operations would you add to generate the carrier frequency? Draw a block diagram for your approach. *6 points.*

Continuous-time: Put sinusoidal signal $\cos(2\pi(1/2 f_c)t)$ through a squaring block which would produce frequencies $-f_c, 0, +f_c$. Apply a DC notch or bandpass filter to remove 0 Hz.

Discrete-time: Use a similar approach as above in continuous-time. Or use upsampling by 2.

- (c) We can use partial fractions decomposition to convert a transfer function into a parallel implementation. Consider a second-order system with conjugate symmetric poles p_0 and p_1 and conjugate symmetric zeros z_0 and z_1 . We can rewrite the second-order system as a sum of two first-order sections assuming that the poles are not equal:

$$H(z) = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} = \frac{1 - c_0 z^{-1}}{1 - p_0 z^{-1}} + \frac{1 - c_1 z^{-1}}{1 - p_1 z^{-1}}$$

Please note that constants c_0 and c_1 are complex-valued.

- i. How many real-valued multiplications per output sample are needed for the second-order system? *3 points.*

$$H(z) = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})} = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

HW 3.3

Lecture Slide 6-5

F19 Midterm #1 1.4(a)

Coefficients a_1, a_2, b_1, b_2 are real-valued because poles/zeros are conjugate symmetric.

Biquad will need 4 real multiplications per output sample.

- ii. How many real-valued multiplication operations per output sample are needed for the parallel combination of the two first-order sections? *3 points.*

Each first-order section requires 2 complex multiplications per output sample. It takes 4 real multiplications for a complex multiplications. $2 \times 2 \times 4 = 16$ real multiplications.

- iii. Assuming that the two first-order sections can be executed in parallel, which realization requires fewer real-valued multiplications per output sample to compute? *3 points.*

Each first-order section needs 8 real multiplications. The biquad requires fewer.

- iv. Repeat part iii assuming that poles p_0 and p_1 , zeros z_0 and z_1 , and constants c_0 and c_1 are real-valued. *3 points.* **Each first-order section would need 2 real multiplications, which are fewer than for the biquad.**

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Solutions 5.0*

Date: October 14, 2020

Course: EE 445S Evans

Name: _____ **With an “E”,** _____ **Anne**
Last, First

Please sign your name below to certify that you did not receive any help, directly or indirectly, on this test from another human other your instructor, Prof. Brian L. Evans, and to certify that you did not provide help, directly or indirectly, to another student taking this exam.

(please sign here) _____

- **Take-home exam** is scheduled for Wednesday, Oct. 14, 2020, 10:30am to 11:59pm.
 - The exam will be available on the course Canvas page at 10:30am on Oct. 14, 2020.
 - Your solutions can be on notebook paper, or on the test and your own paper, or whatever. This means that you won't have to print the test to complete the test.
 - Please upload your solution as a single PDF file to the course Canvas page by 11:59pm on Oct 14, 2020.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification.
- **Internet access.** Yes, you may fully access the Internet when answering exam questions provided that you comply with the other instructions on this page.
- **Academic integrity.** You shall not receive help directly or indirectly on this test from another human except your instructor, Prof. Evans. You shall not provide help, directly or indirectly, to another student taking this exam.
- **Send questions to Prof. Evans.** You may send questions or concerns about this midterm exam during the test to Prof. Evans via Canvas or by e-mail at bevans@ece.utexas.edu.
- **Contact by Prof. Evans.** Prof. Evans might contact all students in the class during the exam through Canvas announcements. Please periodically monitor those announcements.

<i>Character</i>	<i>Problem</i>	<i>Point Value</i>	<i>Your Score</i>	<i>Topic</i>
Diana Barry	1	28		Filter Analysis
Matthew Cuthbert	2	24		Filter Design
Anne Shirley **	3	24		Analysis of Filter Designs
Gilbert Blythe	4	24		Mixer
	<i>Total</i>	100		

**** Also known as “Anne of Green Gables”**

Problem 1.1 Filter Analysis. 28 points.

Consider the following averaging filter with N coefficients. It is a causal, linear time-invariant (LTI), finite impulse response (FIR), discrete-time filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = \frac{1}{N}x[n] + \frac{1}{N}x[n-1] + \dots + \frac{1}{N}x[n-(N-1)]$$

for $n \geq 0$. For example, when $N = 2$, $y[n] = (\frac{1}{2})x[n] + (\frac{1}{2})x[n-1] = (x[n] + x[n-1]) / 2$ which is the average of the current input sample $x[n]$ and previous input sample $x[n-1]$.

Please answer the following questions for an averaging filter for N coefficients.

- (a) What are the initial conditions and their values? Why? 6 points.

Look at the first output value $y[0]$ to see what the initial conditions are.

$$y[0] = \frac{1}{N}x[0] + \frac{1}{N}x[-1] + \dots + \frac{1}{N}x[-(N-1)]$$

Initial conditions are $x[-1], x[-2], \dots, x[-(N-1)]$ and they must be zero as a necessary condition for the system to be linear and time-invariant.

Also, the initial conditions correspond to setting the memory location in each delay element in the block diagram in part (b) to zero.

Note: A causal system does not depend on future input values, or current/future output values.

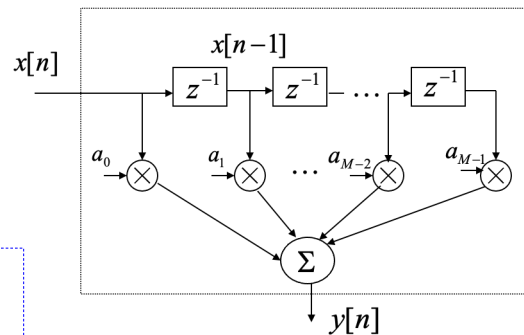
- (b) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.

Block diagram for an FIR filter from lecture slide 3-20:

The values of the FIR coefficients are

$$a_i = \frac{1}{N} \quad \text{for } i = 0, 1, \dots, N-1$$

Lecture slides 3-21, 5-5 & 6-5



Note: Arrows are important because they indicate the order of calculations.

- (c) Derive a formula for the transfer function in the z -domain and the region of convergence. 4 points.

Take z -transform of both sides of the difference equation with initial conditions being zero:

$$Y(z) = \frac{1}{N}X(z) + \frac{1}{N}z^{-1}X(z) + \dots + \frac{1}{N}z^{-(N-1)}X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{N} + \frac{1}{N}z^{-1} + \dots + \frac{1}{N}z^{-(N-1)} = \frac{1}{N}(1 + z^{-1} + \dots + z^{-(N-1)}) \quad \text{for } z \neq 0$$

Lecture slide 5-12

- (d) Give a formula for the discrete-time frequency response of the filter. Justify the steps. 3 points.

Since region of convergence $z \neq 0$ includes the unit circle, we substitute $z = e^{j\omega}$ in part (c):

$$H_{freq}(\omega) = H(e^{j\omega}) = \frac{1}{N}(1 + e^{-j\omega} + \dots + e^{-j(N-1)\omega})$$

Lecture slide 5-12

- (e) What is the group delay of the filter? 3 points.

Answer #1: Since the impulse response $\{\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\}$ is even symmetric about its midpoint at $n = \frac{N-1}{2}$, the group delay is $\frac{N-1}{2}$ samples.

Lecture slide 5-18

HW 1.2(a) & 2.1(a)

Answer #2: We factor the frequency response into amplitude $A(\omega)$ and phase $\theta(\omega)$ form:

$$H_{freq}(\omega) = \frac{1}{N}e^{-j(\frac{N-1}{2})\omega} \left(e^{j(\frac{N-1}{2})\omega} + e^{j(\frac{N-3}{2})\omega} + \dots + e^{-j(\frac{N-3}{2})\omega} + e^{-j(\frac{N-1}{2})\omega} \right)$$

Lectures 3 5 & 6

Lab #3

HW 0.1 1.1

1.3 2.1 2.2

2.3 & 3.2

Midterm 1

Problem 1

Spring 2010

Fall 2016

Fall 2018

JSK Ch. 7

$$H_{freq}(\omega) = \frac{2}{N} e^{-j\left(\frac{N-1}{2}\right)\omega} \left(\cos\left(\frac{N-1}{2}\right) + \cos\left(\frac{N-3}{2}\right) + \dots \right)$$

$$H_{freq}(\omega) = \underbrace{\frac{2}{N} \left(\cos\left(\frac{N-1}{2}\right) + \cos\left(\frac{N-3}{2}\right) + \dots \right)}_{A(\omega)} e^{j\frac{\left(-\frac{N-1}{2}\right)\omega}{\theta(\omega)}} = |A(\omega)| e^{j\theta(\omega)}$$

Discontinuities in the phase occur at frequencies eliminated by the filter, i.e. when $|A(\omega)|$ is zero. When $A(\omega)$ is negative, the phase $\theta(\omega)$ will shift by π , which won't affect the derivative:

$$GD(\omega) = -\frac{d}{d\omega} \theta(\omega) = \frac{N-1}{2}$$

Midterm 1 Problem 1.4(a)

HW 1.2(a)

FIR filters with linear phase must have even or odd symmetry w/r midpoint of impulse response.

(f) The averaging filter, for $N \geq 2$, is not only a lowpass filter, but can also be used to filter out the discrete-time frequency $2\pi/N$ rad/sample and its harmonics up to and including π rad/sample. By denoting f_s as the sampling rate,

i. What continuous-time frequency f_0 corresponds to discrete-time frequency $2\pi/N$? 3 points.

$$\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{1}{N} \text{ which gives } f_0 = \frac{f_s}{N}$$

HW 1.2(a)

ii. What continuous-time frequency f_1 corresponds to discrete-time frequency π ? 3 points.

$$\omega_1 = 2\pi \frac{f_1}{f_s} = \pi \text{ which gives } f_1 = \frac{1}{2} f_s$$

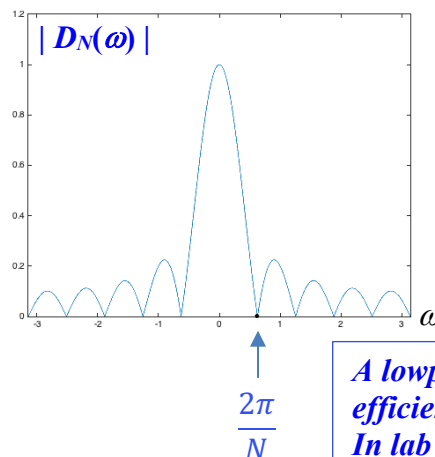
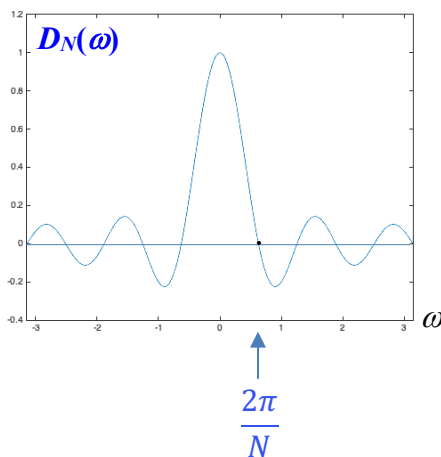
Lecture Slide 4-6

Epilog. When $N = 1$, the impulse response is a discrete-time impulse $\delta[n]$ whose frequency response is all-pass. The z-transform and discrete-time Fourier transform are a constant value of 1.

When $N \geq 2$, the impulse response is a rectangular pulse of N samples in duration. In continuous-time, the Fourier transform of a rectangular pulse of length T seconds is a sinc with its first zero at $2\pi/T$ rad/s (homework 0.1). In discrete-time, the frequency domain is periodic with period 2π . The discrete-time Fourier transform of a rectangular pulse of N samples is a periodic sinc:

$$H_{freq}(\omega) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\omega n} = \sum_{n=0}^{N-1} \left(\frac{1}{N}\right) (e^{-j\omega})^n = \frac{1}{N} \left(\frac{1 - e^{-jN\omega}}{1 - e^{-j\omega}} \right) = \frac{\sin\left(\frac{N}{2}\omega\right)}{N \sin\left(\frac{\omega}{2}\right)} e^{-j\left(\frac{N-1}{2}\right)\omega}$$

Below, we plot the amplitude function $D_N(\omega)$ on the left and its magnitude on the right for $N = 10$:



```
w = -pi : 0.001 : pi;
N = 10;
d = sin(N*w/2) ./ (N * sin(w/2));
figure;
plot(w, d)
ylim( [-0.4, 1.2] );
xlim( [-pi, pi] );
figure;
plot(w, abs(d));
ylim( [0, 1.2] );
xlim( [-pi, pi] );
```

A lowpass averaging filter can be made more efficient by using all ones for the coefficients. In lab 7, we'll learn about FIR comb filters which have even lower run-time complexity to eliminate a frequency and its harmonics.

Problem 1.2. Filter Design. 24 points.

Lecture Slides 6-5 to 6-10

In-Lecture #2 Assignment

Consider a second-order discrete-time linear time-invariant (LTI) infinite impulse response (IIR) filter.

The biquad filter has zeros z_0 and z_1 and poles p_0 and p_1 , and its transfer function in the z -domain is

$$H(z) = C \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)}$$

HW 0.4 1.1 2.1 3.1 & 3.3

Labs 2 & 3

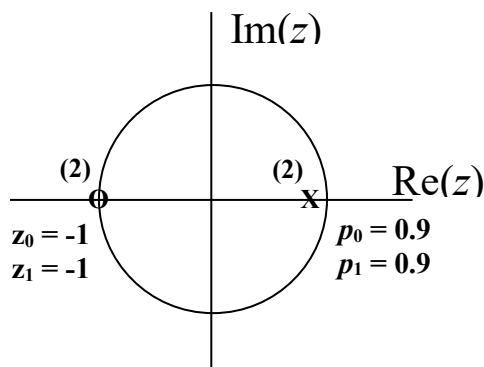
Biquad is short for the “biquadratic” transfer function that is a ratio of two quadratic polynomials.

In this problem, **all of the poles and zeros will be real-valued.**

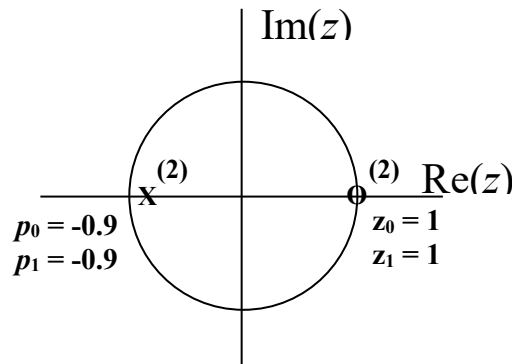
In each part below, design a biquad by placing real-valued poles and zeros to achieve the indicated frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch) or indicate that no such biquad with real-valued poles and zeros could be designed.

Please use O to indicate real-valued zero locations and X to indicate real-valued pole locations.

(a) Lowpass filter



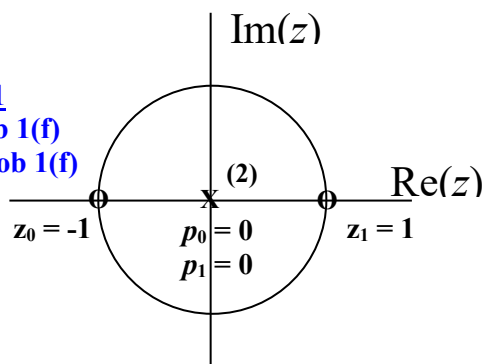
(b) Highpass filter



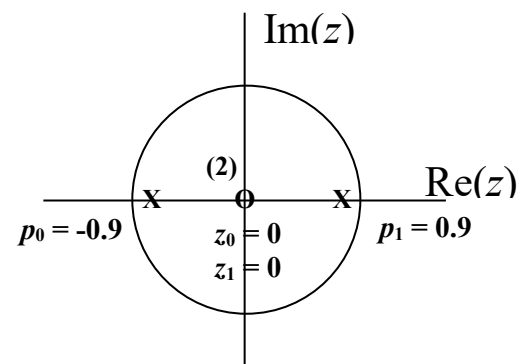
For lowpass, highpass, bandpass and bandstop filters, place the poles at separate angles from the zeros. Angles of poles near unit circle indicate passband frequencies. Angles of zeros on or near unit circle indicate stopband frequencies.

(c) Bandpass filter

Midterm 1
Fall 2019, Prob 1(f)
Spring 2020, Prob 1(f)



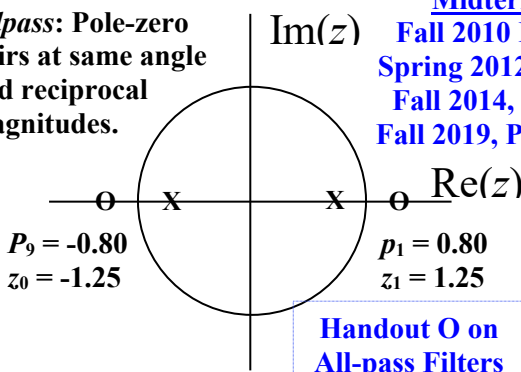
(d) Bandstop filter



(e) Allpass filter

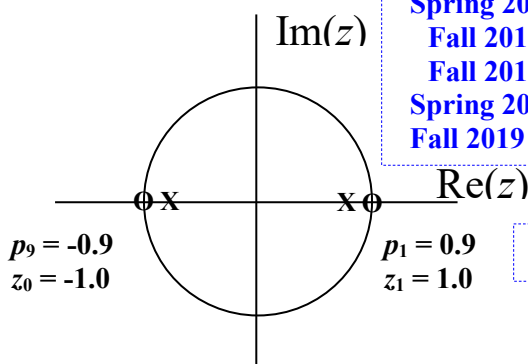
Allpass: Pole-zero pairs at same angle and reciprocal magnitudes.

Midterm 1
Fall 2010 Prob 1
Spring 2012 Prob 1
Fall 2014, Prob 3
Fall 2019, Prob 2(c)



Handout O on All-pass Filters

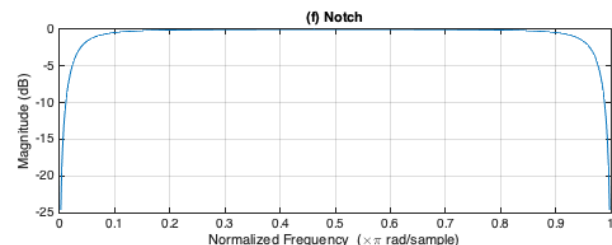
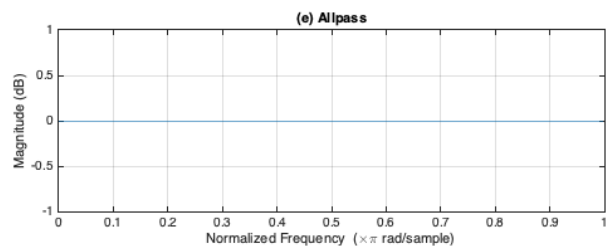
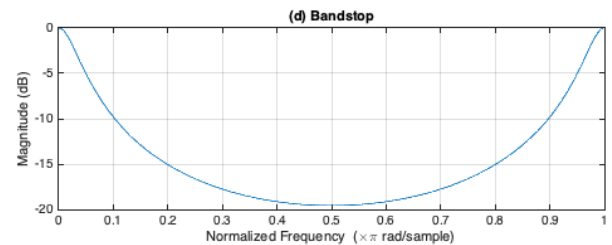
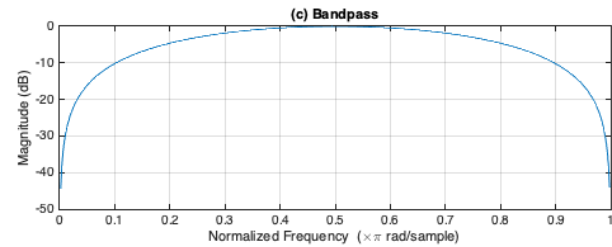
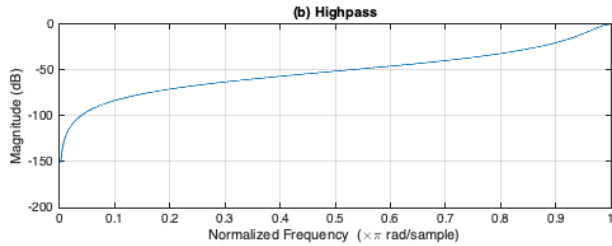
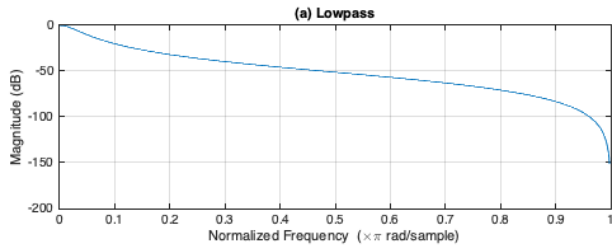
(f) Notch filter



Midterm 1
Spring 2007 Prob 1
Fall 2011 Prob 1
Fall 2016 Prob 1
Spring 2019 Prob 2
Fall 2019 Prob 2(b)

HW 3.1(c)

Epilog. Plots of the magnitude responses of the six filters using the freqz command. All transfer functions were normalized so that maximum magnitude response would be 1 in linear units or 0 dB.



```
figure;
z0 = -1; z1 = -1; p0 = 0.9; p1 = 0.9;
numer = [1 -(z0+z1) z0*z1];
denom = [1 -(p0+p1) p0*p1];
z = 1; zvec = [ 1 z^(-1) z^(-2) ]';
gain = (denom*zvec) / (numer*zvec);
freqz( gain*numer, denom )
title(' (a) Lowpass ');
```

```
figure;
z0 = 1; z1 = 1; p0 = -0.9; p1 = -0.9;
numer = [1 -(z0+z1) z0*z1];
denom = [1 -(p0+p1) p0*p1];
z = -1; zvec = [ 1 z^(-1) z^(-2) ]';
gain = (denom*zvec) / (numer*zvec);
freqz( gain*numer, denom )
title(' (b) Highpass ');
```

```
figure;
z0 = -1; z1 = 1; p0 = 0; p1 = 0;
numer = [1 -(z0+z1) z0*z1];
denom = [1 -(p0+p1) p0*p1];
z = j; zvec = [ 1 z^(-1) z^(-2) ]';
gain = (denom*zvec) / (numer*zvec);
freqz( gain*numer, denom )
title(' (c) Bandpass ');
```

```
figure;
z0 = 0; z1 = 0; p0 = -0.9; p1 = 0.9;
numer = [1 -(z0+z1) z0*z1];
denom = [1 -(p0+p1) p0*p1];
z = 1; zvec = [ 1 z^(-1) z^(-2) ]';
gain = (denom*zvec) / (numer*zvec);
freqz( gain*numer, denom )
title(' (d) Bandstop ');
```

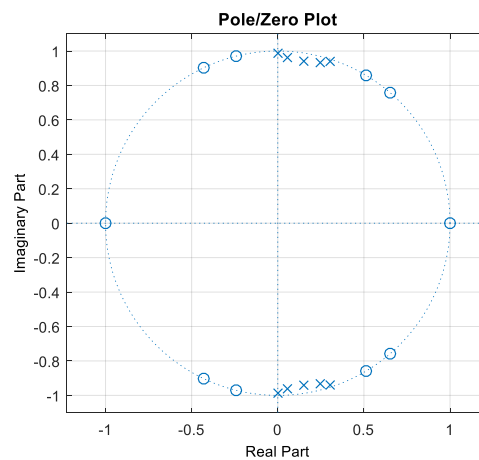
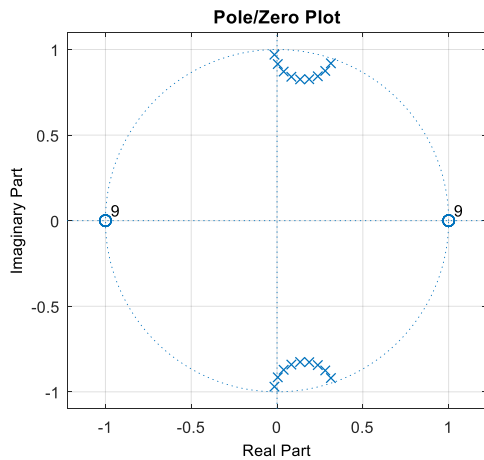
```
figure;
z0 = -1.25; z1 = 1.25; p0 = -0.8; p1 = 0.8;
numer = [1 -(z0+z1) z0*z1];
denom = [1 -(p0+p1) p0*p1];
z = 1; zvec = [ 1 z^(-1) z^(-2) ]';
gain = (denom*zvec) / (numer*zvec);
freqz( gain*numer, denom )
title(' (e) Allpass ');
```

```
figure;
z0 = -1; z1 = 1; p0 = -0.9; p1 = 0.9;
numer = [1 -(z0+z1) z0*z1];
denom = [1 -(p0+p1) p0*p1];
z = j; zvec = [ 1 z^(-1) z^(-2) ]';
gain = (denom*zvec) / (numer*zvec);
freqz( gain*numer, denom )
title(' (f) Notch ');
```

Problem 1.3 Analysis of Filter Designs. 24 points.

Two discrete-time linear time-invariant (LTI) infinite impulse response (IIR) filter designs are below.

Assume that both filters meet the same magnitude specifications. All zeros are on the unit circle.

**Design #1**

Number of complex/real poles: 18 Complex
 Number of complex/real zeros: 18 Real
 Passband behavior: Monotonic
 Stopband behavior: Monotonic
 Design method: Butterworth

Design #2

Number of complex/real poles: 10 Complex
 Number of complex/real zeros: 10 Complex
 Passband behavior: Rippling
 Stopband behavior: Rippling
 Design method: Elliptic

Please **answer** the following questions about the filter designs with **justification**. 3 points each.

	Design #1	Design #2
(a) Filter Order	#poles = 18	#poles = 10
(b) Bounded-Input Bounded Output (BIBO) Stability?	Yes, all poles inside unit circle per Lecture Slides 6-15 and 6-16	Yes, all poles inside unit circle per Lecture Slides 6-15 and 6-16
(c) Approximate range of discrete-time frequencies in the passband in rad/sample	Pole and zero angles separated; estimate outer pole locations gave pole angles from 0.395π to 0.505π	Pole and zero angles separated; estimate outer pole locations gave pole angles from 0.4π to 0.5π
(d) Frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch)	Bandpass due to part (c)	Bandpass due to part (c)
(e) Number of multiplication operations using a cascade of biquads filter structure	9 biquads x 5 mults = 45 mults Fall 2018, Problem 1.4(a) -OR- 9 biquads x 4 mults = 36 mults because all zeros on unit circle per homework 3.3(b)(c)	5 biquads x 5 mults = 25 mults Fall 2018, Problem 1.4(a) -OR- 5 biquads x 4 mults = 20 mults because all zeros on unit circle per homework 3.3(b)(c)
(f) Amount of memory storage (in words) using a cascade of biquads filter structure	9 biquads x 5 words = 45 words Fall 2018, Problem 1.4(a) -OR- 9 biquads x 4 words = 36 words because all zeros on unit circle per homework 3.3(b)(c) solution	5 biquads x 5 words = 25 words Fall 2018, Problem 1.4(a) -OR- 5 biquads x 4 words = 20 words because all zeros on unit circle per homework 3.3(b)(c) solution
(g) Give an advantage of each design, and indicate which design you would choose.	Lower maximum value of group delay due wider transition region per Spring 2020, Prob 1.3(c)	Lower run-time complexity due to fewer multiplications in part (f)
(h) Describe the frequency response if all poles are removed, including selectivity	Bandpass FIR filter with center frequency of $\pi/2$ due to all zeros being at either π or $-\pi$	Weak bandpass FIR filter with same center frequency with six frequencies removed

See work on next page

Storage of input & output samples should have been included

See alternate answer on next page

(c) Poles are separated from the zeros in angles, and the poles are close to the unit circle. The pole angles indicate the passband frequencies. Below, I've estimated the pole angles by zooming into the pole-zero plots in the PDF file and measuring their locations with a ruler.

Design #1. I estimated the pole location with the least positive angle to be at $0.3162 + j*0.9191$, which is an angle of about 0.395π , and the pole with the greatest positive angle to be at $-0.0147 + j*0.9706$, which is an angle of about 0.505π . The pole in the middle of the passband is measured to be $0.1360 + j*0.8235$, which has an angle of about 0.45π rad/sample.

Design #2, I estimated the pole location with the least positive angle to be at $0.3051 + j*0.938$, which is an angle of about 0.4π . The pole with the greatest positive angle approximately resides on the imaginary axis, which is an angle of 0.5π . The pole in the middle of the passband is measured to be $0.1496 + j*0.9382$, which has an angle of about 0.45π rad/sample.

(g) **Butterworth design:** Magnitude response is monotonically decreasing with frequency (no rippling). The attenuation in the stopband increases with increasing frequency. The Butterworth design has nearly linear phase over a wider range of passband frequencies than the Elliptic design (per homework problem 3.3).

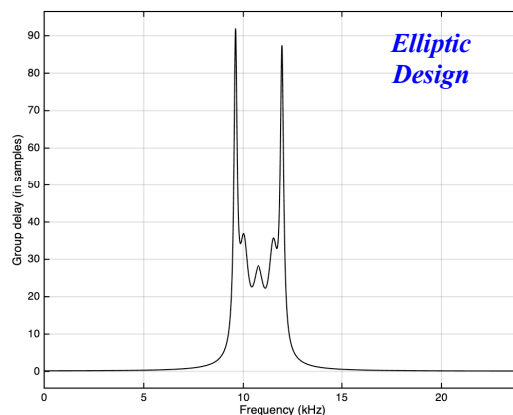
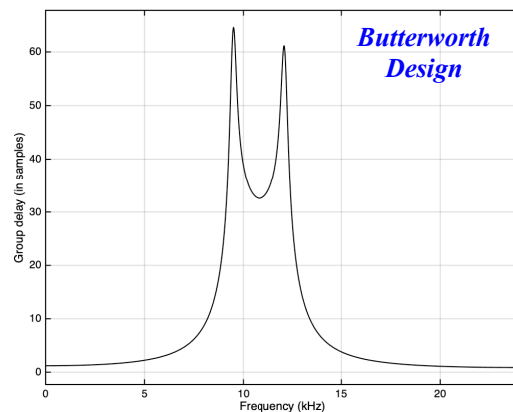
Elliptic design: Magnitude response has much a sharper transition from the passband to either stopband.

Epilog. This filter was designed using the default specification in the filter design and analysis tool in Matlab for bandpass filter design:

- *Bandpass frequency selectivity*
- *IIR filter*
- *Minimum order design*
- $f_s = 48000$ Hz
- $f_{stop1} = 7200$ Hz (0.3π rad/sample)
- $f_{pass1} = 9600$ Hz (0.4π rad/sample)
- $f_{pass2} = 12000$ Hz (0.5π rad/sample)
- $f_{stop2} = 14400$ Hz (0.6π rad/sample)
- $A_{stop1} = 60$ dB
- $A_{pass} = 1$ dB
- $A_{stop2} = 80$ dB

To meet these magnitude specifications with a linear phase FIR filter, a Parks-McClellan design algorithm requires an FIR filter with order 52. An FIR filter of order 52 would require 53 multiplication operations per output sample, compared with 20 for the elliptic IIR filter (implemented as a cascade of biquads) and 36 for the Butterworth IIR filter (implemented as a cascade of biquads). The linear phase FIR filter has a group delay of 26 samples for all frequencies (i.e. order/2 samples).

Group delay vs. frequency



Problem 1.4. Mixer. 24 points.

Sinusoidal amplitude modulation upconverts a baseband (low-frequency) signal into a bandpass (higher frequency) signal.

A block diagram for sinusoidal amplitude modulation is shown on the right. The lowpass filter (LPF) enforces the baseband bandwidth to be f_1 . The bandpass filter (BPF) enforces the transmission bandwidth to be $2f_1$ centered at f_c .

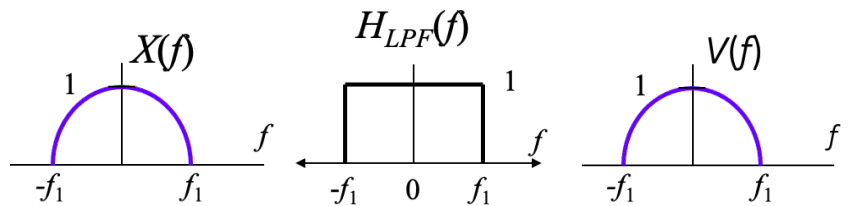
The circuitry can be simplified by replacing the analog multiplier and cosine generator with a sampling block that operates at sampling rate f_s . The sampler could be implemented with a pass transistor and a generator of a simpler periodic waveform.

Assume ideal lowpass and bandpass filters shown on the right.

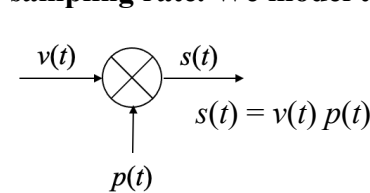
Using the spectrum for $X(f)$ on the right,

(a) Draw $V(f)$. 6 points.

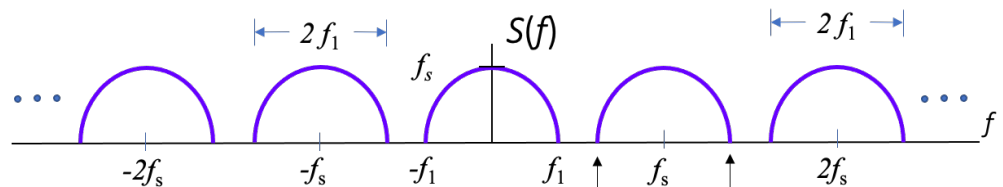
$V(f) = H_{LPF}(f) X(f)$. The lowpass filter only passes frequencies from $-f_1$ to f_1 , and signal $x(t)$ only has frequencies between $-f_1$ and f_1 .



(b) Draw $S(f)$. 6 points. Assume an ideal sampler that closes the switch to gate the input voltage to the output instantaneously and opens switch instantaneously. Switch is closed/opened every T_s seconds where $T_s = 1/f_s$ and f_s is the sampling rate. We model the sampler as modulation by an impulse train $p(t)$.

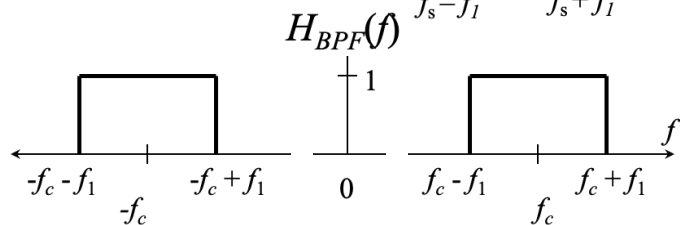


$$S(f) = V(f) * P(f) = V(f) * \left(f_s \sum_{k=-\infty}^{\infty} \delta(f - k f_s) \right) = f_s \sum_{k=-\infty}^{\infty} V(f - k f_s)$$



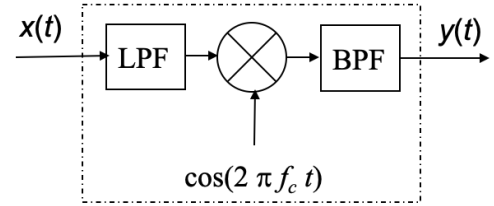
(c) Draw $Y(f)$. 6 points.

$S(f)$ has spectrum of $V(f)$ scaled by f_s plus its replicas at offsets of integer multiples of f_s . We'll set $f_c = k f_s$ so that the BPF selects the k th replica.

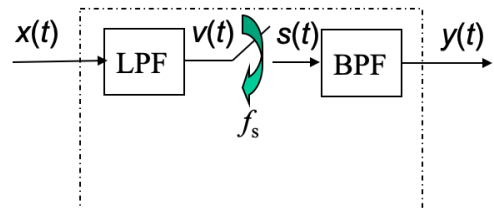


(d) Give formulas that describe all the possible values for f_s so that the mixer implements sinusoidal amplitude modulation. 6 points.

We would like the BPF selects the k th replica that matches the carrier frequency: $f_c = k f_s$
 We would also like to avoid aliasing by keeping gaps between adjacent replicas: $f_s > 2 f_1$



Sinusoidal Amplitude Modulation



Mixer

We covered the mixer implementation in lecture on September 9, 2020, as part of lecture 1, and here's the market board work. The picture is linked into the Web page for lecture 1:

Lecture Slide 1-26 For upconversion, replace multiplication by $\cos(2\pi f_0 t)$ with sampling block (one transistor) running at sampling rate f_s

Homework 0.3: Sampler model

$v(t) \times g(t) = p(t)v(t)$

$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$

$G(f) = P(f) * V(f)$

and

$P(f) = f_s \sum_{k=-\infty}^{\infty} \delta(f - kf_s)$

$G(f) = V(f) * \left(f_s \sum_{k=-\infty}^{\infty} \delta(f - kf_s) \right)$

$G(f) = f_s \sum_{k=-\infty}^{\infty} V(f - kf_s)$

BPF Selects replica centered at $f_c = \frac{1}{2} f_s$

$f_s > 2f_{max}$

Epilog: The idea of using the replicas generated by sampling for upconversion as in this problem can also be applied to downconversion. Upconversion and downconversion using sampling is also called “bandpass sampling”. More info is available on lecture slides 4-15 to 4-17 given below.

Bandpass Sampling

- **Reduce sampling rate**

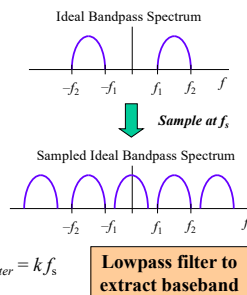
Bandwidth: $f_2 - f_1$
 Sampling rate f_s must be greater than analog bandwidth $f_s > f_2 - f_1$

For replica to be centered at origin after sampling

$f_{center} = \frac{1}{2}(f_1 + f_2) = kf_s$

- **Practical issues**

Sampling clock tolerance: $f_{center} = kf_s$
 Effects of noise



4-15

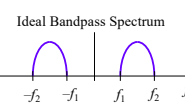
Bandpass Sampling

- **Sampling theorem**

Maximum frequency f_2
 $f_s > 2f_2$

- **Bandpass sampling**

Bandwidth $f_2 - f_1$
 $f_s > f_2 - f_1$



- **2.4 GHz unlicensed band**

$f_1 = 2.4$ GHz
 $f_2 = 2.5$ GHz (2.499 GHz)
 Bandwidth = 0.1 GHz
 $f_c = 2.45$ GHz

- **Sampling theorem**

$f_s = 5.0$ GHz

- **Bandpass sampling**

$f_s = 0.245$ GHz
 20x more efficient

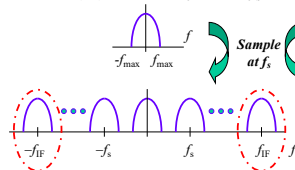


4-16

Sampling for Up/Downconversion

- **Upconversion method**

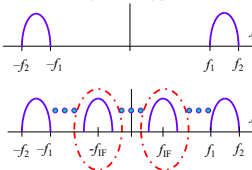
Sampling plus bandpass filtering to extract intermediate frequency (IF) band with $f_{IF} = k_{IF}f_s$



4-17

- **Downconversion method**

Bandpass sampling plus bandpass filtering to extract intermediate frequency (IF) band with $f_{IF} = k_{IF}f_s$



Bandpass sampling is used to accelerate simulations of RF communication systems because of the large reduction in sampling rate and an equal amount of reduction in simulation runtimes.

Problem 1.1. IIR Filter Design. 24 points.

Consider a second-order discrete-time linear time-invariant (LTI) infinite impulse response (IIR) filter.

A second-order section, also known as a biquad, has two zeros z_0 and z_1 and two poles p_0 and p_1 . Its transfer function in the z -domain is

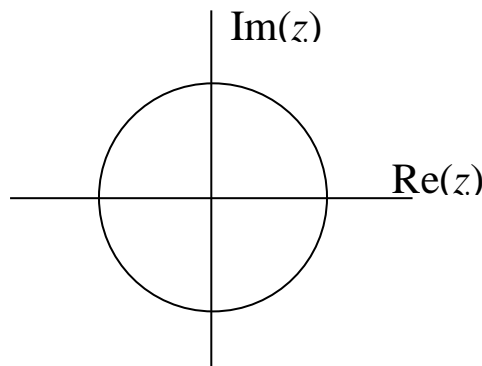
$$H(z) = C \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)}$$

In this problem, **all poles and zeros will be complex-valued but not real-valued**. The imaginary part of the complex number cannot be zero, and the real part of the complex number can be anything.

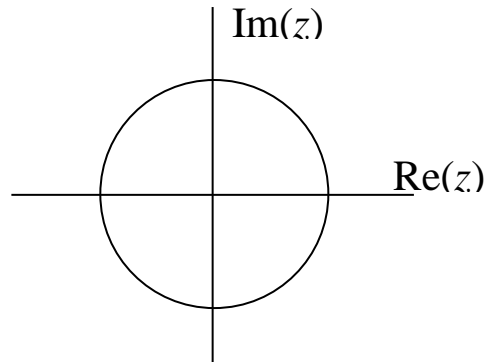
In each part below, design a biquad by placing complex non-real-valued poles and zeros to achieve the indicated frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch) or indicate that no such biquad could be designed. For each filter,

- Please use O to indicate the zero locations and X to indicate pole locations
- Give numeric values for the two poles and two zeros in polar form (i.e. magnitude and phase form)

(a) Lowpass filter



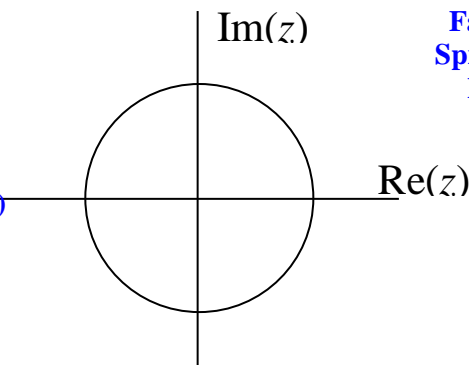
(b) Highpass filter



Please see the next two pages for solutions.

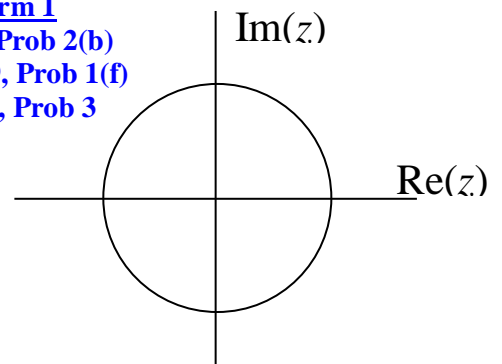
(c) Bandpass filter

Midterm 1
Fall 2008, Prob 2(a)
Spring 2017, Prob 3
Fall 2017, Prob 3
Fall 2019, Prob 1(f)
Spring 2020, Prob 1(f)
Fall 2020, Prob 2(c)



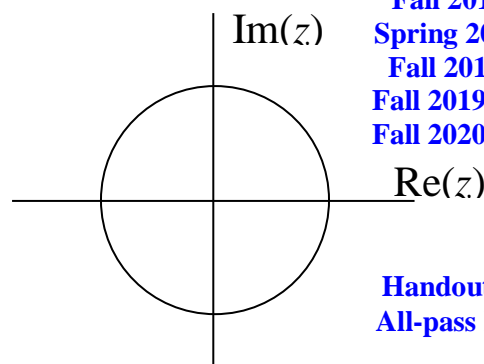
Midterm 1
Fall 2008, Prob 2(b)
Spring 2009, Prob 1(f)
Fall 2016, Prob 3

(d) Bandstop filter



(e) Allpass filter

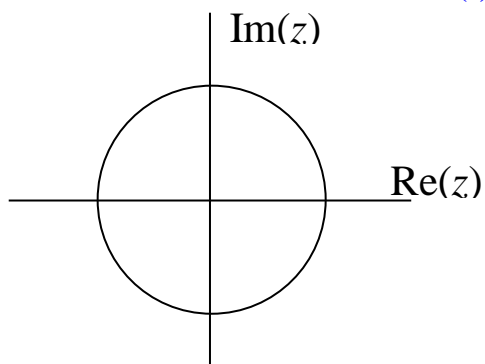
Midterm 1
Fall 2010 Prob 1
Spring 2012 Prob 1
Fall 2014, Prob 3
Fall 2019, Prob 2(c)
Fall 2020, Prob 2(e)



Handout O on All-pass Filters

(f) Notch filter

Midterm 1
HW 3.1(c) Spring 2007, Prob 1
Fall 2010, Prob 4(b)
Spring 2011, Prob 2
Fall 2011, Prob 1
Spring 2012, Prob 2
Fall 2016, Prob 1
Spring 2019, Prob 3
Fall 2019, Prob 2(b)
Fall 2020, Prob 2(f)



Per [lecture slide 6-8](#),

- Angle of pole near unit circle indicates frequency at which peak occurs in magnitude response
- Angle of zero on or near unit circle indicates frequency at which valley occurs in mag. response.

Although not explicitly requested, we will choose poles that are *conjugate symmetric* to give *real-valued* feedback coefficients, and zeros that are *conjugate symmetric* to give *real-valued* feedforward coefficients, as we've been doing throughout the semester to reduce run-time complexity. Using complex coefficients in the difference equation to implement the filter requires **4x** complexity for multiplication-addition operations and **2x** storage for input $x[n]$ and output $y[n]$:

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

```

%%% Lowpass filter example
zeroAngle = 15*pi/16;
z0 = exp(j*zeroAngle);
z1 = exp(-j*zeroAngle);
feedforwardcoeffs = [1 -(z0+z1) z0*z1];

r = 0.9;
poleAngle = pi/16;
p0 = r * exp(j*poleAngle);
p1 = r * exp(-j*poleAngle);
feedbackcoeffs = [1 -(p0+p1) p0*p1];

%%% Normalize frequency response
%%% to 1 at center of passband
z = 1; zvec = [1 z^(-1) z^(-2)]';
C = (feedbackcoeffs * zvec) /
(feedforwardcoeffs * zvec);

figure; zplane(C*feedforwardcoeffs,
feedbackcoeffs);
figure; freqz(C* feedforwardcoeffs,
feedbackcoeffs);

```

Lecture slide 6-7

For lowpass, highpass, bandpass, and bandstop filters, we'll place each pole at the center of each passband as per [lecture slide 6-9](#) (left pole-zero plot) and demos in [lecture slide 6-10](#).

(a) Lowpass: zeros at

$$z_0 = e^{j\theta_1} \text{ and}$$

$$z_1 = e^{-j\theta_1}$$

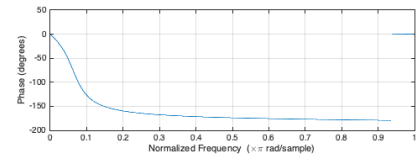
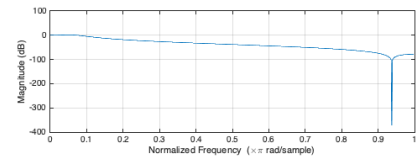
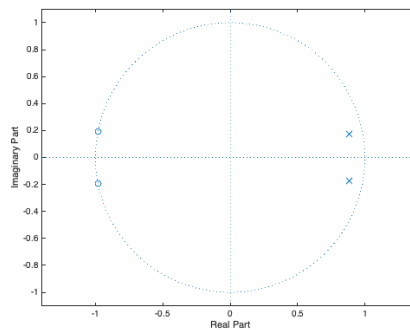
$$\text{with } \theta_1 = \frac{15}{16}\pi \text{ and}$$

poles at

$$p_0 = 0.9 e^{j\theta_2} \text{ and}$$

$$p_1 = 0.9 e^{-j\theta_2}$$

$$\text{with } \theta_2 = \frac{1}{16}\pi$$



(b) Highpass: zeros at

$$z_0 = e^{j\theta_2} \text{ and}$$

$$z_1 = e^{-j\theta_2}$$

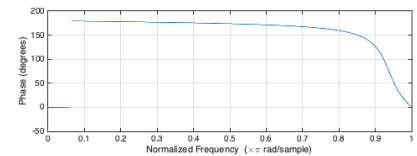
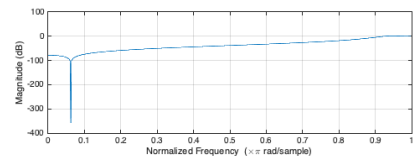
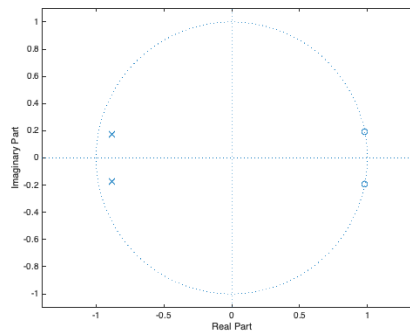
$$\text{with } \theta_2 = \frac{1}{16}\pi \text{ and}$$

poles at

$$p_0 = 0.9 e^{j\theta_1} \text{ and}$$

$$p_1 = 0.9 e^{-j\theta_1}$$

$$\text{with } \theta_1 = \frac{15}{16}\pi$$



(c) Bandpass: zeros at

$$z_0 = 0.1 e^{j\theta_3} \text{ and}$$

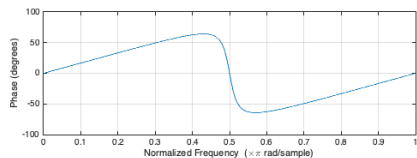
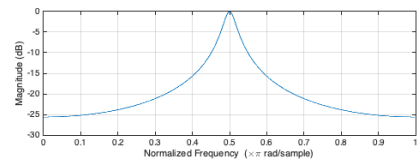
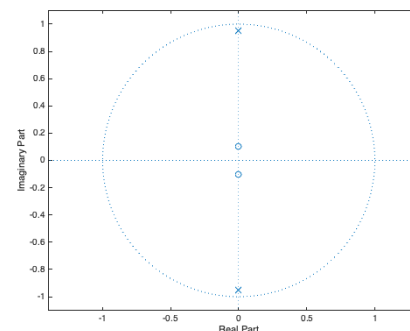
$$z_1 = 0.1 e^{-j\theta_3}$$

$$\text{with } \theta_3 = \frac{\pi}{2} \text{ and}$$

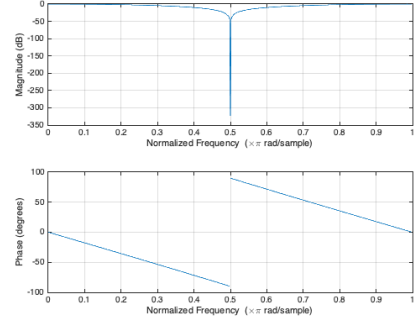
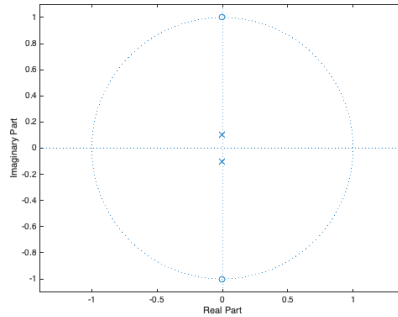
poles at

$$p_0 = 0.95 e^{j\theta_3} \text{ and}$$

$$p_1 = 0.95 e^{-j\theta_3}$$

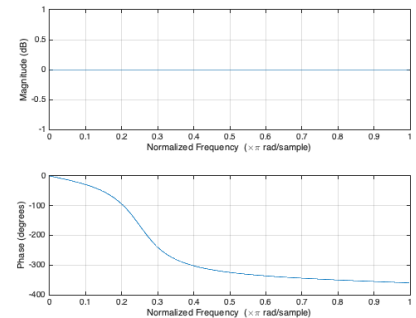
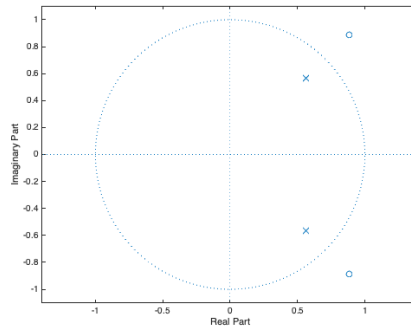


- (d) Bandpass: zeros at
 $z_0 = e^{j\theta_3}$ and
 $z_1 = e^{-j\theta_3}$
 with $\theta_3 = \frac{\pi}{2}$ and
 poles at
 $p_0 = 0.1 e^{j\theta_3}$ and
 $p_1 = 0.1 e^{-j\theta_3}$



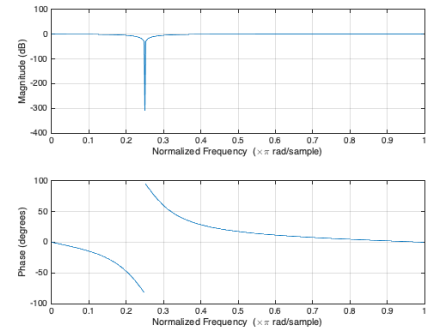
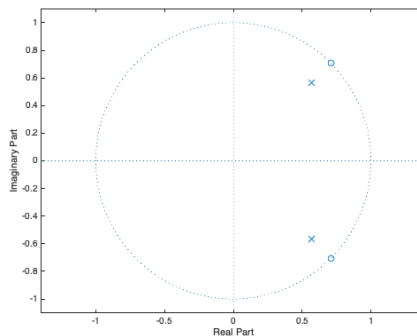
For allpass filters, we'll follow [lecture slide 6-9](#) (right pole-zero plot) and the [all-pass filter handout](#) to place pole-zero pairs at the same angle and reciprocal magnitudes.

- (e) Allpass: zeros at
 $z_0 = 1.25 e^{j\theta_4}$ and
 $z_1 = 1.25 e^{-j\theta_4}$
 with $\theta_4 = \frac{\pi}{4}$ and
 poles at
 $p_0 = 0.8 e^{j\theta_4}$ and
 $p_1 = 0.8 e^{-j\theta_4}$



For notch filters, we'll follow [lecture slide 6-9](#) (middle pole-zero plot), demos in [lecture slide 6-10](#), and homework 3.1 on designing a notch filter to remove narrowband interference.

- (f) Notch: zeros at
 $z_0 = e^{j\theta_4}$ and
 $z_1 = e^{-j\theta_4}$
 with $\theta_4 = \frac{\pi}{4}$ and
 poles at
 $p_0 = 0.8 e^{j\theta_4}$ and
 $p_1 = 0.8 e^{-j\theta_4}$



For reference, and not asked, we obtain the difference equation with feedback coefficients a_1 and a_2 and feedforward coefficients b_0 , b_1 , and b_2 from the transfer function:

$$H(z) = C \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)} = C \frac{z^2 - (z_0 + z_1)z + z_0z_1}{z^2 - (p_0 + p_1)z + p_0p_1} = C \frac{1 - (z_0 + z_1)z^{-1} + z_0z_1z^{-2}}{1 - (p_0 + p_1)z^{-1} + p_0p_1z^{-2}}$$

$$H(z) = C \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} = \frac{Y(z)}{X(z)}$$

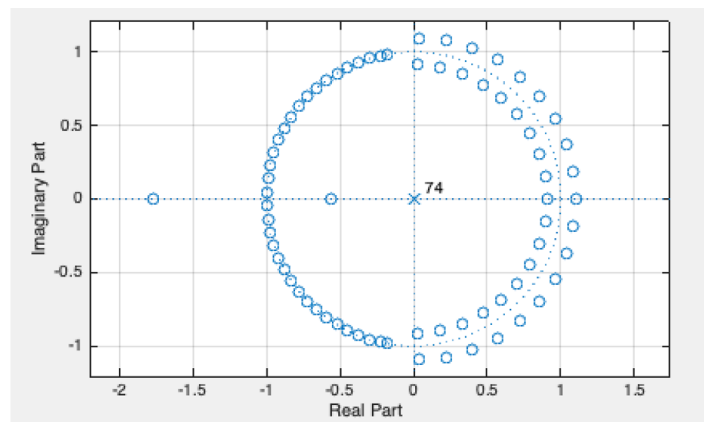
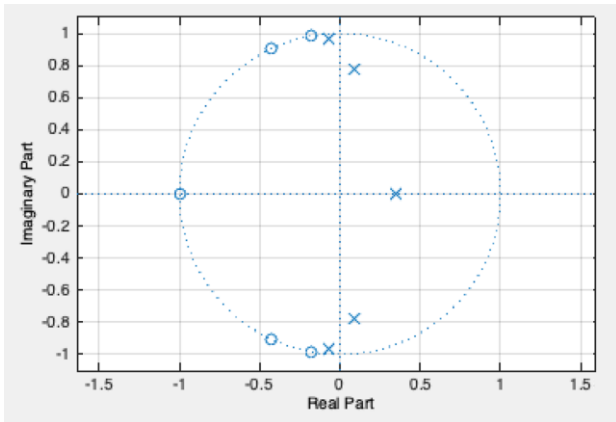
$$(1 + a_1z^{-1} + a_2z^{-2}) Y(z) = (b_0 + b_1z^{-1} + b_2z^{-2}) X(z)$$

$$y[n] = a_1y[n - 1] + a_2y[n - 2] + b_0x[n] + b_1x[n - 1] + b_2x[n - 2]$$

Selectivity	Example Application(s)	Selectivity	Example Application
Lowpass	Anti-aliasing filter before sampler in ADC; demodulation	Bandstop	Alleviate ringing of the ear symptoms (HW 2.3 & 3.3)
Highpass	Enhance edges/texture in images	Allpass	Phase correction after ADC
Bandpass	Reject out-of-band interference and noise (HW 3.1); modulation	Notch	Remove in-band narrowband interference (HW 3.1)

Problem 1.2 Filter Design Tradeoffs. 24 points.

Both discrete-time linear time-invariant (LTI) filters below meet the same filter design specifications based on the magnitude response for an audio application.



Design #1
 Number of complex/real poles: 5
 Number of complex/real zeros: 5
 Passband behavior: Rippling
 Stopband behavior: Rippling
 Design method: Elliptic
 Maximum Group Delay: 33 samples

Design #2
 Number of complex/real poles: 74
 Number of complex/real zeros: 74
 Passband behavior: Rippling
 Stopband behavior: Rippling
 Design method: Parks-McClellan
 Maximum Group Delay: 37 samples

Please **answer** the following questions about the filter designs with **justification**. 3 points each.

	Design #1	Design #2
(a) Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) Filter?	IIR filter because it has at least one non-trivial (non-zero) pole and was designed using the elliptic method.	FIR filter because all poles are zero and it was designed using Parks-McClellan method.
(b) Filter Order	#poles = 5 (Homework 3.3)	#zeros = 74 (Homework 2.3)
(c) Bounded-Input Bounded Output Stable?	Yes, all poles inside unit circle (Lecture Slides 6-13 & 6-14; Homework 2.1 & 3.3)	Yes, all FIR filters are BIBO stable (Lecture Slides 6-13 & 6-14 and BIBO Stability Handout)
(d) Approximate range of discrete-time frequencies in the passband in rad/sample	Pole angles from 0 to 0.52π approx. and zero angles 0.56π approx. to π (Lecture Slide 6-8)	Pole angles from 0 to 0.5π and zero angles 0.58π approx. to π (Lecture Slide 6-8)
(e) Frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch)	Lowpass due to (d)	Lowpass due to (d)
(f) Number of multiplication operations (use cascade of biquads structure if IIR filter)	3 + 2 biquads x 5 = 13 mults Fall 2018, Problem 1.4(a) -OR- 3 + 2 biquads x 4 = 11 mults each pair of zeros on unit circle give 2 of 3 feedforward coefficients with value 1 (Homework 3.3(b)(c))	Number of FIR coefficients is filter order + 1 = 75. 75 mults
(g) Amount of storage (use cascade of biquads structure if IIR filter)	Data: 3 + 2 biquads x 6 = 15 words Coeffs: 3 + 2 biquads x 5 = 13 words Fall 2018, Problem 1.4(a) -OR- 3 + 2 biquads x 4 = 11 words per (f)	Data: 75 words Coeffs: 75 words
(h) Give an advantage of each design, and indicate which design you would choose.	5-6x lower complexity Mildly lower maximum group delay	Linear phase over all freq. BIBO stability regardless of implementation

See work on next page

Include storage of input & output samples

(c) Poles are separated from the zeros in angles, and the poles are close to the unit circle. The pole angles indicate the passband frequencies, and the zero locations on the unit circle indicate the stopband frequencies. I estimated the angle of the pole with greatest positive angle by zooming into the pole-zero plot in the PDF file, measuring their Cartesian in the complex plane with a ruler, and computing the phase. I followed the same approach for the zero with smallest positive angle.

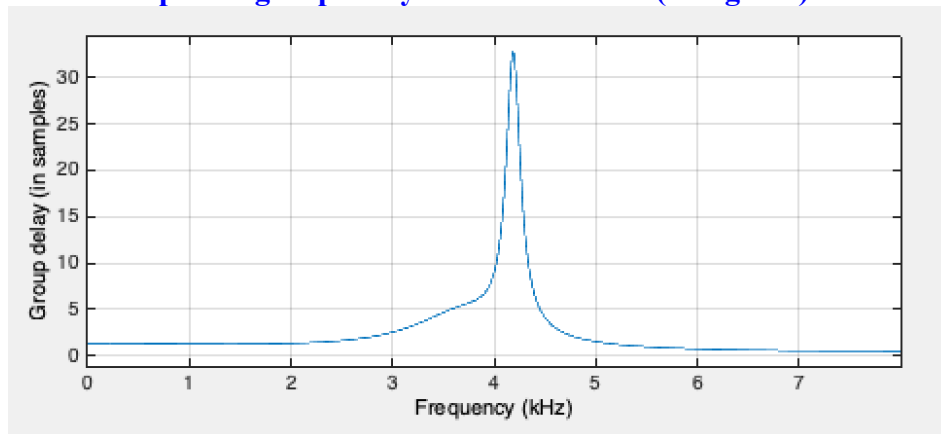
The lowpass filter design specification was to attenuate frequencies above the highest note on an 88-key piano, which is at 4186 Hz (C8). The next note above 4186 Hz is 4434 Hz which will be in the stopband.

Filter specifications

- $F_{pass} = 4186 \text{ Hz}$
- $F_{stop} = 4434 \text{ Hz}$
- $A_{pass} = 1 \text{ dB}$
- $A_{stop} = 30 \text{ dB}$
- $F_s = 16000 \text{ Hz}$

I chose a low stopband attenuation so the FIR filter order wouldn't be so large which in turn would allow one to see all of the zeros on the pole-zero plot. In audio, 80 dB is a common value for stopband attenuation. We'll learn in Lecture 8 on Quantization that 80 dB is equivalent to 13 bits, so 80 dB of stopband attention means that high frequencies will lose the upper 13 bits per sample in strength. The conversion is $SNR_{dB} = 2 + 6B$ where B is the number of bits. When using 80 dB for the stopband attention, the filter orders increase to 11 for the elliptic IIR design method and 164 for the Parks-McClellan FIR design method.

Here's the plot of group delay for the IIR filter (Design #1).



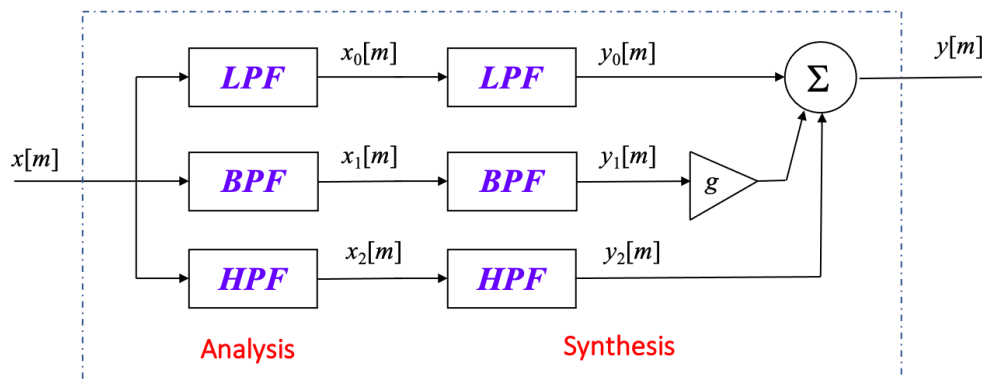
The lowest note on an 88-key piano is 27.5 Hz (A0). If one wanted to also remove frequencies below the lowest note, then one could either add a DC notch filter in cascade with the lowpass filter above, or design a bandpass filter from scratch.

Problem 1.3 FIR Filter Bank Design. 28 points.

A bank of analysis filters divides a signal $x[m]$ into frequency bands for subsequent processing.

A bank of synthesis filters combines the analysis frequency bands into a single signal $y[m]$.

The analysis-synthesis filter bank below has three linear time-invariant (LTI) filters in each bank:



The LPF, BPF, and HPF filters are finite impulse response (FIR) filters with three coefficients each.

Both lowpass filters (LPFs) have impulse response of $h_0[m] = \delta[m] + \delta[m - 1] + \delta[m - 2]$. The LPF coefficients are $[1, 1, 1]$.

- (a) Both bandpass filters (BPFs) are designed by modulating the LPF to shift its frequency response by $\pi/2$ to the right and left: $h_1[m] = \cos\left(\frac{\pi}{2}m\right) h_0[m]$. Give the coefficients for $h_1[m]$. Does the BPF have linear phase? Why or why not? 6 points.

[1, 0, -1]. Yes, BPF has (generalized) linear phase due to odd symmetry about midpoint of impulse response. Odd symmetry: $h[n] = -h[N - 1 - n]$ for $n = 0, 1, \dots, N - 1$. Here, $h[0] = -h[2]$ and $h[1] = -h[1] = 0$. (Modulation is used during the offline filter design procedure.)

- (b) Both highpass filters (HPFs) are designed by modulating the LPF to shift its frequency response by π to the right and left: $h_2[m] = \cos(\pi m) h_0[m]$. Give the coefficients for $h_2[m]$. Does the HPF have linear phase? Why or why not? 6 points.

[1, -1, 1]. Yes, HPF has linear phase due to even symmetry about midpoint of impulse response. Even symmetry: $h[n] = h[N - 1 - n]$ for $n = 0, 1, \dots, N - 1$. Here, $h[0] = h[2]$ and $h[1] = h[1]$. (Modulation is used during the filter design procedure, not implementation.)

- (c) Compute the impulse response $h[m]$ for the overall system with input $x[m]$ and output $y[m]$. The impulse response will include the real-valued gain g . **Hint:** $y[m] = y_0[m] + g y_1[m] + y_2[m]$. 9 points.

Overall impulse response: $h[m] = h_0[m] * h_0[m] + g h_1[m] * h_1[m] + h_2[m] * h_2[m]$

$$h_0[m] * h_0[m]: \quad \text{conv}([1, 1, 1], [1, 1, 1]) = [1, 2, 3, 2, 1]$$

$$g h_1[m] * h_1[m]: \quad g \text{ conv}([1, 0, -1], [1, 0, -1]) = [g, 0, -2g, 0, g]$$

$$h_2[m] * h_2[m]: \quad \text{conv}([1, -1, 1], [1, -1, 1]) = [1, -2, 3, -2, 1]$$

$$\text{Add all term:} \quad [g+2, 0, 6-2g, 0, g+2]$$

- (d) Does the overall system have linear phase for all possible values of g ? Why or why not? 3 points.

Yes, because its impulse response is even symmetric about its midpoint for all values of g .

- (e) Compute the value of g that causes the overall system to act like an ideal delay with gain C , i.e. $y[m] = C x[m - m_0]$. Please give the values of C and m_0 . 4 points.

When $g = -2$, the overall impulse response becomes $[0, 0, 10, 0, 0]$. This equivalent to an ideal delay of $m_0 = 2$ samples with a gain of $C = 10$.

Problem 1.4. Mystery Nonlinearities. 24 points.

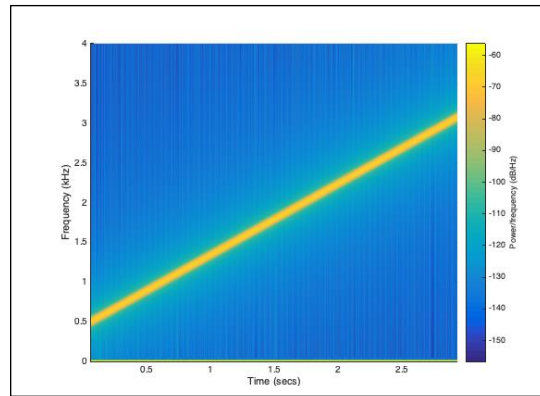
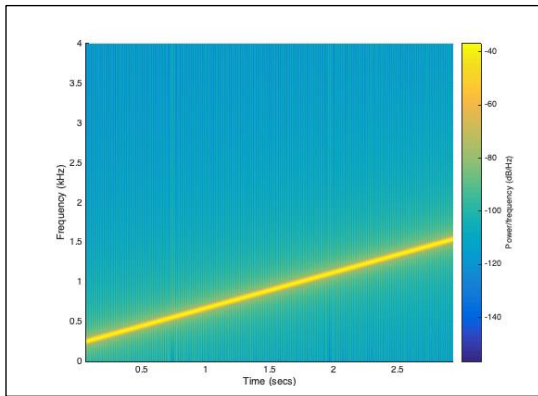
You're trying to determine input-output relationships for discrete-time pointwise nonlinear systems.

For discrete-time pointwise systems, output at discrete-time n only depends on input at discrete-time n .

You input a chirp signal and look at the resulting output signal to figure out what the system is doing.

For the analysis, you decide to use the [chirp signal from in-lecture assignment #1](#), which starts around 240 Hz and ends around 1520 Hz, and uses a sampling rate of 8000 Hz.

- (a) Give a formula for output $y[n]$ in terms of input $x[n]$ by looking at the spectrogram for a chirp input signal (left) and the spectrogram of the output signal (right). 12 points.



The spectrogram of the output shows a strong DC component at all times, and another component shows a linear increase from around 480 Hz to around 3040 Hz over time.

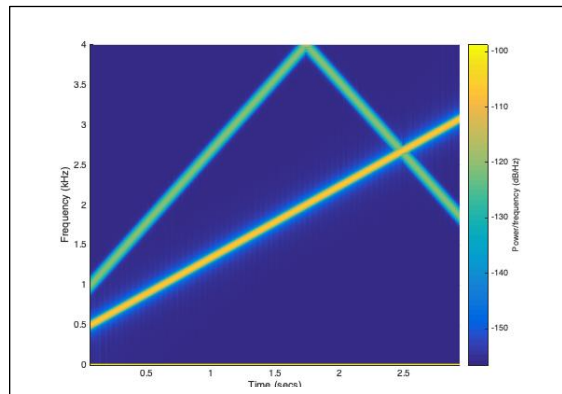
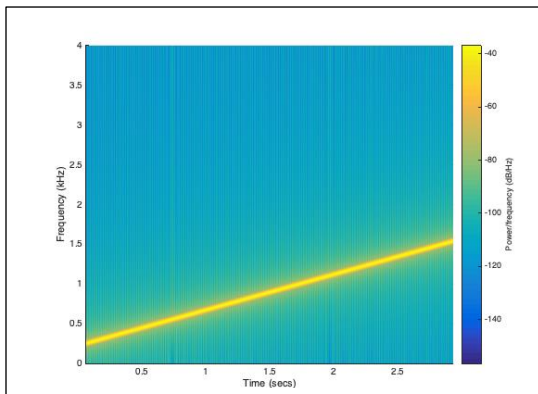
A spectrogram is a plot of the frequency content (y-axis) vs. time (x-axis) of a signal.

At any point in time, the chirp input signal has one principal frequency and the output signal contains double the principal frequency plus a DC component.

This is a squaring block: $y[n] = x^2[n]$.

$$\cos^2(\omega_0 n) = \frac{1}{2} + \frac{1}{2} \cos(2 \omega_0 n)$$

- (b) Give a formula for output $y[n]$ given the input $x[n]$ by looking at the spectrogram for a chirp input signal (left) and the spectrogram of the output signal (right). 12 points.



The spectrogram of the output shows a strong DC component at all times; a second component that shows a linear increase from around 480 Hz to around 3040 Hz; and a third component that starts around 960 Hz, linearly increases to 4000 Hz, and then linearly decreases to around 1880 Hz.

See next page.

At any point in time, the input chirp signal has one principal frequency, and the output contains quadruple and double the input principal frequency as well as a DC component from 0 to 1.8 s. The quadruple component is twice the doubled input principal frequency, which would be consistent with a second squaring block in cascade with the first: $y[n] = x^4[n]$.

For input $\cos(\omega_0 n)$, the first squaring block would give as output

$$\frac{1}{2} + \frac{1}{2} \cos(2 \omega_0 n)$$

The second squaring block would give

$$\left(\frac{1}{2} + \frac{1}{2} \cos(2 \omega_0 n)\right)^2 = \frac{1}{4} + \frac{1}{2} \cos(2 \omega_0 n) + \frac{1}{4} \cos^2(2 \omega_0 n)$$

where

$$\cos^2(2 \omega_0 n) = \frac{1}{2} + \frac{1}{2} \cos(4 \omega_0 n)$$

and hence

$$\left(\frac{1}{2} + \frac{1}{2} \cos(2 \omega_0 n)\right)^2 = \frac{3}{8} + \frac{1}{2} \cos(2 \omega_0 n) + \frac{1}{8} \cos(4 \omega_0 n)$$

Among the three terms, the $\cos(2 \omega_0 n)$ term is strongest, followed by the DC term of 3/8, and finally by the $\cos(4 \omega_0 n)$ term. This is reflected in the color ascribed to these three terms according to the color map show at the right of the spectrogram plot.

We can track the $\cos(4 \omega_0 n)$ component in green from 0 to 3s to see what happens. Let f_0 be the principal input frequency at any point in time. From 0 to 1.8s, $4f_0 < 1/2f_s$ which is 4000 Hz, or equivalently $4\omega_0 < \pi$ rad/sample, and we see a linear increase. From 1.8s to 3s, the $\cos(4 \omega_0 n)$ term aliases because $4f_0 > 1/2f_s$ which accounts for the downward linear trajectory.

Here's the Matlab code to generate the above plots:

```
fs = 8000; % Samples/s
n = 0 : 3*fs; % There are fs samples in 1s
f0 = 220; % A3 (A note at 220 Hz in third octave on Western scale)
w0 = 2*pi*f0/fs;
x = 0.1*cos(w0*n + pi*(0.7*10^(-5))*(n.^2));

blockSize = 1024;
overlap = 1023;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');

y = x .^ 2;

figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');

y = x .^ 4;

figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```


Problem 1.1. Sinusoidal Generation. 24 points.

You're asked to generate one period of a discrete-time cosine signal $y[n]$:

- The continuous-time frequency is 131 Hz ('C' note on the Western scale in the third octave).
- The sampling rate f_s is 8,000 Hz.

(a) What is the discrete-time frequency in rad/sample of the discrete-time cosine signal? *4 points.*

$$x(t) = \cos(2\pi f_0 t) \text{ where } f_0 = 131 \text{ Hz.}$$

$$x[n] = x(nT_s) = x\left(\frac{n}{f_s}\right) = \cos\left(2\pi \frac{f_0}{f_s} n\right) = \cos(\omega_0 n)$$

$$\text{where } \omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{131}{8000} \text{ is the discrete-time frequency in rad/sample.}$$

(b) What is the fundamental period of the discrete-time cosine signal in samples? *4 points.*

Per the [Handout on Discrete-Time Periodicity](#), a sinusoidal signal with discrete-time frequency, where the common factors between integers N and L have been removed,

$$\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{N}{L}$$

has a discrete-time fundamental period of L samples. In the period of L samples, there are N continuous-time periods of a continuous-time sinusoidal signal at frequency f_0 .

- (c) Give a difference equation whose impulse response will generate the discrete-time cosine signal. *4 points.* From homework problem 0.4, $y[n] = (2 \cos \omega_0) y[n-1] - y[n-2] + x[n] - (\cos \omega_0) x[n-1]$ with initial conditions $y[-1]$, $y[-2]$, and $x[-1]$ being zero as necessary conditions for LTI.
- (d) Compare the run-time complexity for the difference equation and the lookup table method. The lookup table would store an entire period of cosine values computed offline. *8 points.*

Method	Total Memory Needed	Multiplications per output sample	Reads per output sample	Writes per output sample
Difference equation	7 words	2	6 words	4 words
Lookup table	8001 words	0	1 word	1 word

The difference equation contains two constants, current input value, previous input value, current output value, and two previous output values. Total memory of 7 words.

The difference equation has two multiplications $(2 \cos \omega_0) y[n-1]$ and $(\cos \omega_0) x[n-1]$ per output sample. To compute $y[n]$, the other six values have to be read once each. Also, we'll need to write the result $y[n]$, and update $y[n-1]$, $y[n-2]$ and $x[n-1]$.

Cosine lookup table stores one period of $L = 8000$ samples. We use $n = 0, 1, \dots, L - 1$ to read the precomputed value from the table for $\cos(\omega_0 n)$ and write it out as $y[n]$.

- (e) How would you use the lookup table for the cosine signal to generate a discrete-time sine signal with the same frequency? *4 points.*

With $\sin(\theta) = \cos(\theta - \pi/2)$, we delay the cosine by $1/4$ of a period: $n_0 = L/4 = 2000$ samples.

To generate $\sin(\omega_0 n) = \cos(\omega_0 (n - n_0))$, we start with $n = 0$ which is index $-n_0$ into the cosine lookup table, which is out of bounds. Due to periodicity, index $-n_0$ is the same as $-n_0 + L = 6000$ into the cosine table. We start with index 6000 into the cosine table. As we increment the index and reach the end of the lookup table, we go to the first entry (index 0).

Note: The above answer in part (d) was my initial answer and is plenty for a timed test. Upon further analysis given on the next page, I learned that the determination of the value of n_0 is a bit more complicated.

(d) We would like to choose the delay n_0 for $x[n] = \cos(\omega_0 n)$ to give $y[n] = \sin(\omega_0 n)$:

$$y[n] = x[n - n_0] = \cos\left(2\pi \frac{N}{L} (n - n_0)\right) = \cos\left(2\pi \frac{N}{L} n - 2\pi \frac{N}{L} n_0\right) = \cos(\omega_0 n - \theta)$$

We would like the phase shift θ to be $\frac{\pi}{2} + 2\pi k$ where k is an integer:

$$2\pi \frac{N}{L} n_0 = \frac{\pi}{2} + 2\pi k \rightarrow n_0 = \left(\frac{1}{4} + k\right) \frac{L}{N} = \left(\frac{1+4k}{4}\right) \frac{L}{N} = \left(\frac{1+4k}{4N}\right) L$$

For example, when $N = 1$ and $k = 0$, we would have $n_0 = \frac{L}{4}$. For $N = 131$ and $L = 8000$:

$$n_0 = \left(\frac{1+4k}{4N}\right) L = \left(\frac{1+4k}{N}\right) \frac{L}{4} = \left(\frac{1+4k}{131}\right) 2000$$

We can try different values for integer k to find integer n_0 . When $k = 98$, $n_0 = 6000$. We would start indexing into the cosine table at -6000 which is same as an index of 2000 due to periodicity.

Deeper dive: We would like to have the term below be an integer, which we'll denote as m :

$$\left(\frac{1+4k}{131}\right) = m \rightarrow 1 + 4k = 131m \rightarrow 1 = 131m - 4k$$

The equation $131m - 4k = 1$ is Bezout's identity which can be solved efficiently by Euclid's algorithm. This is not a topic that would have appeared in the course or its pre-requisites.

Examples: We'll use a shorter discrete-time period $L = 20$ to visualize results. Examples show two different values for $-n_0$ to achieve a phase shift of $-\pi/2$ for the same value of L . In example #1, $n_0 \neq \frac{L}{4}$. N and L must be relatively prime. To satisfy the Sampling Theorem, $L > 2N$.

Let $N = 7$ and $L = 20$:

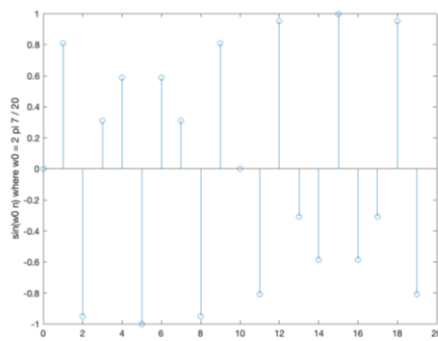
$$\omega_0 = 2\pi \left(\frac{7}{20}\right)$$

$$n_0 = \left(\frac{1+4k}{7}\right) 5$$

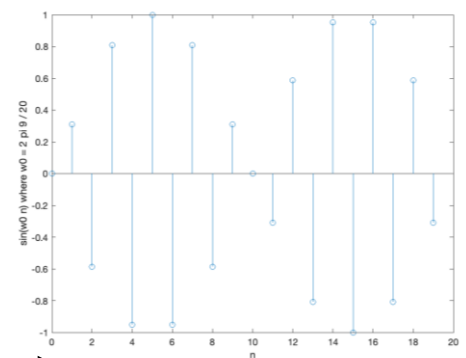
$k = 5$ to give $n_0 = 15$.

Shift by -15 samples is same as shift of 5 samples due to period of 20 samples.

Example #1

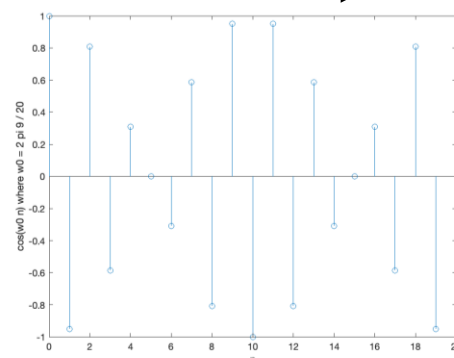
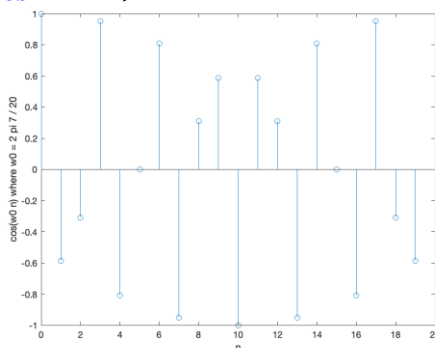


Example #2



5 samples →

15 samples →



$N = 9$ and $L = 20$:

$$\omega_0 = 2\pi \left(\frac{9}{20}\right)$$

$$n_0 = \left(\frac{1+4k}{9}\right) 5$$

$k = 2$ to give $n_0 = 5$

Shift of -5 samples is same as shift of 15 samples due to period of 20 samples

Problem 1.2 FIR Filter Design. 24 points.

You're asked to design a lowpass linear phase finite impulse response (FIR) filter to meet the following specifications:

- Pass frequencies in octave 0, Western scale, from 16.35160 Hz ('C') to 30.8611 Hz ('B')
- Zero out odd harmonics of the 60 Hz powerline frequency (60 Hz, 180 Hz, 300 Hz, etc.)
- Sampling rate is less than 1000 Hz

(a) What sampling rate would you choose? Why? 12 points

From the Sampling Theorem, sample at $f_s > 2 f_{max}$ to be able to reconstruct the original signal from its sampled version. Frequencies captured by sampling are $-\frac{1}{2}f_s < f_{max} < \frac{1}{2}f_s$ and these are the frequencies used in reconstructing a signal

Midterm Problems:
1.2 Sp 19, 1.3(d) Sp 19, 1.2 F16

Lecture Slides
1-16, 1-21 & 3-

Lecture 4

Considerations for choosing the sampling rate $f_s < 1000$ Hz :

- The maximum passband frequency is 30.8611 Hz. $f_s > 2 (30.8611 \text{ Hz})$
- We want to remove odd harmonics of the main powerline frequency of 60 Hz, which are 60 Hz, 180 Hz, 300 Hz, 420 Hz, 540 Hz, 660 Hz, 780 Hz, etc. Some will alias.
- Our sampling rate $f_s < 1000$ Hz, so any odd harmonics of 60 Hz above $\frac{1}{2}f_s$ will alias. The harmonics are 60 Hz, 180 Hz, 300 Hz, 420 Hz, 540 Hz, 660 Hz, 780 Hz, 900 Hz, etc.
- Use the highest sampling rate possible for better audio quality.

If we choose a sampling rate between adjacent odd harmonic frequencies, all the odd harmonics that alias will alias to one of the odd harmonic frequencies that didn't alias.

When $f_s = 240$ Hz, the odd harmonic at 60 Hz does not alias. For the other odd harmonics:

- 180 Hz aliases to $180 \text{ Hz} - 240 \text{ Hz} = -60 \text{ Hz}$ and -180 Hz aliases to $-180 \text{ Hz} + 240 \text{ Hz} = 60 \text{ Hz}$
- 300 Hz aliases to $300 - 240 = 60 \text{ Hz}$ and -300 Hz aliases $-300 + 240 = -60 \text{ Hz}$, etc.

Choose $f_s = 960$ Hz

(b) Give the coefficients of the FIR filter to meet the specifications. 12 points.

Solution #1: Averaging Filter. From the handout [Designing Averaging Filters](#), an averaging filter has a null bandwidth of f_s / N and we would like the first null to be at 60 Hz so that we can pass octave 0:

$$\frac{f_s}{N} = 60 \rightarrow N = \frac{f_s}{60} = \frac{960}{60} = 16$$

The averaging filter would zero out frequencies that are multiples of 60 Hz, which would include the odd and even harmonics. It is zeroing out twice as many frequencies as needed, but still meets the specifications. [See next page for additional analysis.](#)

Solution #2: Manually place zeros. MATLAB command `poly` can convert a list of roots (zeros) to an unfactored polynomial.

We place a zero on the unit circle at the angle of each positive and negative discrete-time frequency to be zeroed out. There are four of each. The zeros

$$e^{\pm j\omega_0}, e^{\pm j3\omega_0}, e^{\pm j5\omega_0}, e^{\pm j7\omega_0} \text{ where } \omega_0 = 2\pi \frac{60}{960} = \frac{\pi}{8} \text{ give coefficients } [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1].$$

Filter has linear phase, but multiband in selectivity instead of lowpass. ([See next page.](#))

Lectures 3 5 & 6

Lab #3

Midterm 1.1

HW 0.1 1.1

F 2020

1.3 2.1 2.2

Sp 2020

2.3 & 3.2

F 2018

F 2016

JSK Ch. 7

Sp 2010

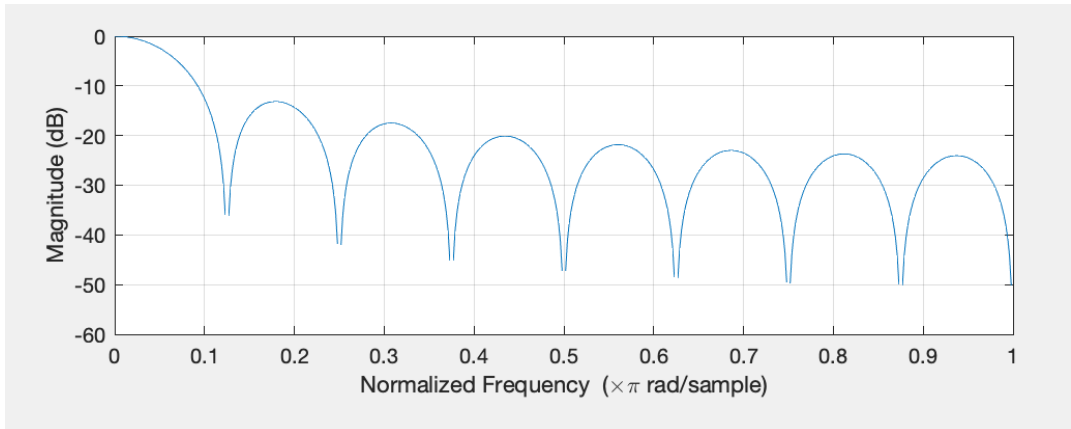
In-Lecture Assignment #2

[Designing Averaging Filters](#)

Additional analysis for problem 1.2(b). This would not be expected on a test.

Solution #1: Averaging filter.

```
h = 1/16)*ones(1,16);      % impulse response is a rectangular pulse
freqz(h);                  % frequency response is a periodic sinc
```



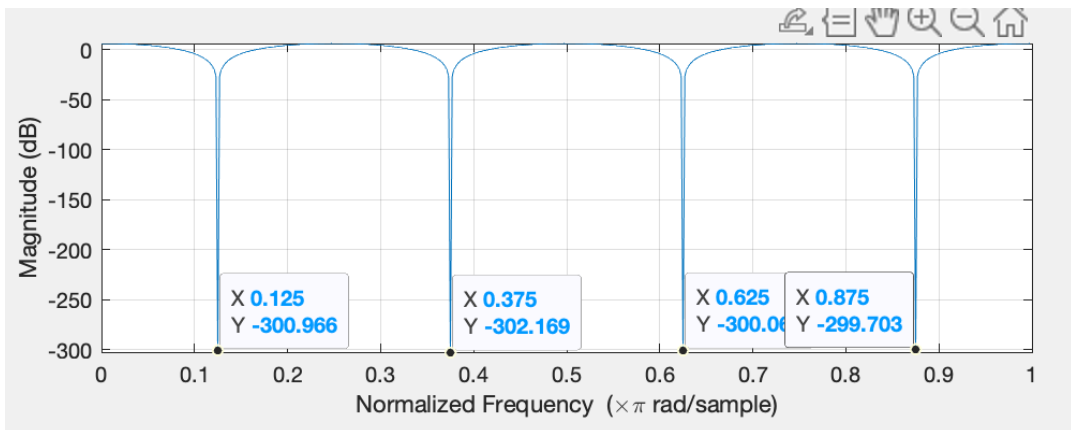
Magnitude response shows discrete-time frequencies at multiples of $\frac{\pi}{8}$ are being zeroed out.

Filter coefficients are even symmetric about the midpoint, which means the FIR filter has linear phase.

Solution #2: Manually place zeros to design the FIR filter.

Matlab code to determine the filter coefficients from the zero locations

```
w0 = pi/8;
zerolist = exp( j * w0 * [1 -1 3 -3 5 -5 7 -7] );
h = poly(zerolist)
>> [ 1 0 0 0 0 0 0 0 1 ]
freqz(h);
```



Magnitude response shows discrete-time frequencies $\frac{1}{8}\pi, \frac{3}{8}\pi, \frac{5}{8}\pi$ and $\frac{7}{8}\pi$ being zeroed out.

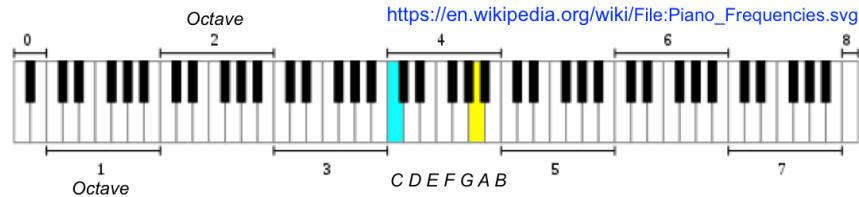
Filter coefficients [1 0 0 0 0 0 0 1] are even symmetric about the midpoint, which means the FIR filter has linear phase.

The magnitude response has multiple passbands. It's not lowpass.

This filter is called a comb filter. We'll see it later in lab #7 on guitar effects.

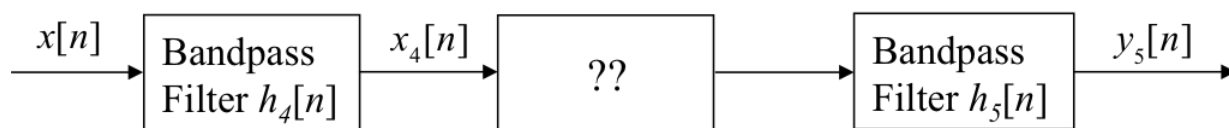
Problem 1.3 Discrete-Time Audio Effects. 28 points.

The notes on the Western scale on an 88-key piano keyboard grouped into octaves follow:



The frequency of note A4 (i.e. 'A' in the 4th octave) at 440 Hz is twice the frequency of A3 at 220 Hz. This type of octave spacing occurs for all the notes on the Western scale.

Design a **discrete-time** audio effects system that will extract the fourth octave of frequencies and then alter that octave of frequencies to be in the next higher octave:



All notes on the fourth octave should appear as the same notes in the next higher octave.

Bandpass filter $h_k[n]$ passes frequencies in the k th octave and attenuates other frequencies.

For signals $x[n]$ and $x_4[n]$, the sampling rate f_s is 8,000 Hz.

- (a) Design a second-order infinite impulse response (IIR) bandpass filter $h_4[n]$ to pass the fourth octave and attenuate the other octaves. In the fourth octave, the lowest note is 262 Hz and highest note is 494 Hz.

12 points.

- Give formulas for the pole and zero locations.
- Plot poles and zeros on the diagram on the right.

Place a pole p_0 at the center frequency in the fourth octave and near but inside the unit circle to define the passband in positive frequencies:

$$f_{center} = \frac{262 \text{ Hz} + 492 \text{ Hz}}{2} = 377 \text{ Hz}$$

$$\omega_{center} = 2\pi \frac{f_{center}}{f_s} = 2\pi \frac{377}{8000}$$

$$p_0 = 0.95 e^{j\omega_{center}} \text{ and } p_1 = 0.95 e^{-j\omega_{center}}$$

- (b) What system would you use for the ?? block? Why? 9 points.

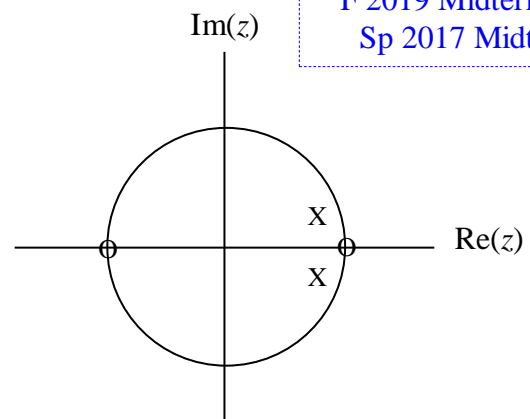
Solution #1: Squaring block. Effect on single input frequency:

$$y[n] = x^2[n] = \cos^2(\omega_0 n) = \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 n)$$

It doubles the input frequency and creates a zero frequency (DC) term.

Any single note in the fourth octave would be doubled to be the same note in the fifth octave, and DC term would be filtered out by the fifth octave bandpass filter.

Solution #2: Downsampling by 2.



F 2019 Midterm 1.3(a)
Sp 2017 Midterm 1.3

Place zeros on the unit circle to define stopbands centered at 0 rad/sample and π rad/sample:

$$z_0 = e^{j0} = 1 \text{ and } z_1 = e^{j\pi} = -1$$

HW 1.3

Sp 2021 Midterm 1.4(a)
Sp 2020 Midterm 1.4(b)
F 2016 Midterm 1.3(c)
F 2015 Midterm 1.3(a)

Lecture slides 3-7 & 3-10

(c) What would the output be for your proposed system if two notes in the fourth octave were being played at the same time? 7 points.

Solution #1: Let the input signal to the squaring block be a sum of two cosine signals at different note frequencies ω_1 and ω_2 in the fourth octave.

$$x[n] = \cos(\omega_1 n) + \cos(\omega_2 n)$$

$$y[n] = x^2[n] = (\cos(\omega_1 n) + \cos(\omega_2 n))^2$$

$$y[n] = \cos^2(\omega_1 n) + 2 \cos(\omega_1 n) \cos(\omega_2 n) + \cos^2(\omega_2 n)$$

We know from part (b) that $\cos^2(\omega_0 n) = \frac{1}{2} + \frac{1}{2} \cos(2 \omega_0 n)$.

The middle term $2 \cos(\omega_1 n) \cos(\omega_2 n)$ is the modulation of one cosine by another.

The resulting frequencies are $\omega_1 + \omega_2$ and $|\omega_1 - \omega_2|$.

$$y[n] = 1 + \frac{1}{2} \cos(2 \omega_1 n) + \cos((\omega_1 + \omega_2) n) + \cos((\omega_1 - \omega_2) n) + \frac{1}{2} \cos(2 \omega_2 n)$$

The output consists of the following frequencies:

$2 \omega_1$ the frequency for note #1 in the fifth octave

$2 \omega_2$ the frequency for note #2 in the fifth octave

$\omega_1 + \omega_2$ which will fall in the fifth octave

$|\omega_1 - \omega_2|$ which fall below the fourth octave

0 frequency

The last two frequencies will be attenuated by the bandpass filter for the fifth octave, but there will be intermodulation distortion (or audio effect?) in the fifth octave at frequency $\omega_1 + \omega_2$.

Solution #2: The two notes in the fourth octave will show up as their respective notes in the fifth octave. Downsampling by 2 will also double the bandwidth around the principal frequency. One can see this is in problem 1.4(b) on this test.

Please note that this discrete-time audio effect system only alters the principal frequencies of notes. It filters out all the harmonics of the note. It's the harmonics that gives richness and texture to the notes. It's the harmonics that allow us to identify what instrument played it.

Problem 1.4. Mystery Systems. 24 points.

You're trying to identify unknown discrete-time systems.

You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 4000 Hz over 0 to 5s

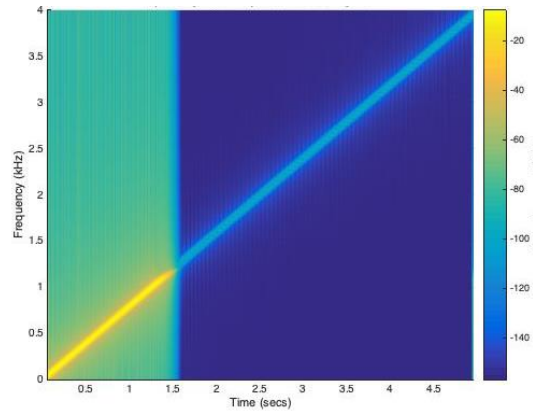
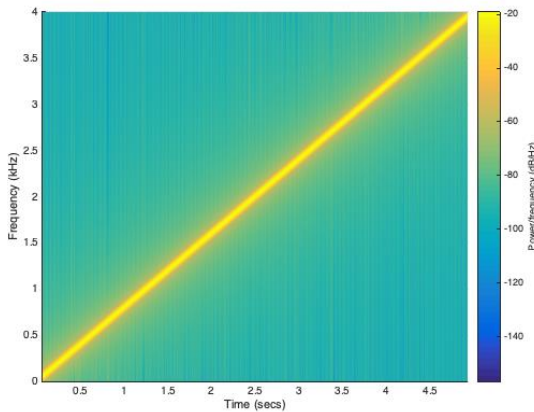
$$x(t) = \cos(2\pi f_1 t + 2\pi\mu t^2)$$

where $f_1 = 0$ Hz, $f_2 = 4000$ Hz, and $\mu = \frac{f_2 - f_1}{2 t_{\max}} = \frac{4000 \text{ Hz}}{10 \text{ s}} = 400 \text{ Hz}^2$. Sampling rate f_s is 8000 Hz.

In each part below, identify the unknown system as one of the following:

1. filter – give selectivity (lowpass, highpass, bandpass, bandstop) and passband/stopband frequencies
2. upsampler – give upsampling factor
3. downsampler – give downsampling factor

(a) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lectures
1 3 5 & 6

Lab #3

HW 0.1 1.1
1.3 2.1 & 2.2

JSK Ch. 7

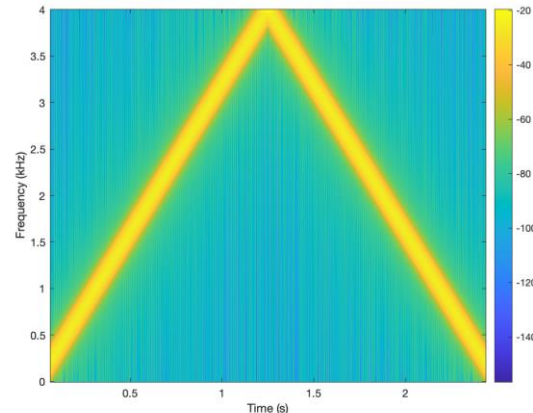
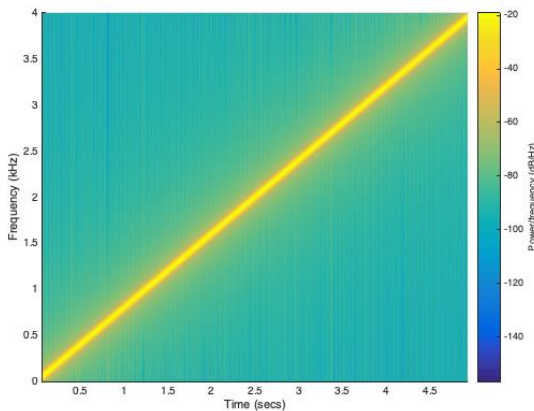
Midterm 1.1
F 2020
Sp 2020
F 2018

Designing
Averaging
Filters

From the output spectrogram, frequencies from 0 Hz to about 1200 Hz are passed. Principal frequencies in the chirp above about 1300 Hz are severely attenuated and their medium blue color indicates at attenuation vs. bright yellow of about 100 dB.

Lowpass Filter with passband frequency ~1200 Hz and stopband frequency ~1300 Hz.

(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lecture

HW 0.3 & 2.2

Midterm
Problems
1.2 F 18
1.2(d) Sp 18
1.2(d) F 09

When compared to the input spectrogram, the output spectrogram has half the duration in time and its principal frequency is increasing. From 1.25s to 2.5s, aliasing occurs.

Downsampling by 2, per homework problem 2.2(d).

Matlab code to generate the spectrograms for problem 1.4.

(a) Lowpass filter

```
fs = 8000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Design lowpass filter
fnyquist = fs/2;
fpass = 1000;
fstop = 1200;
ctfrequencies = [0 fpass fstop fnyquist];
idealAmplitudes = [1 1 0 0];
pmfrequencies = ctfrequencies / fnyquist;
filterOrder = 200;
h = firpm( filterOrder, pmfrequencies, idealAmplitudes );
h = h / sum(h.^2);

y = conv(x, h);

%% Plot spectrogram of signal
blockSize = 1024;
overlap = 1023;
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```

(b) Downsampling by 2

```
fs = 8000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Downsampling by 2
y = x(1:2:end);

blockSize = 1024;
overlap = 1023;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Solutions Version 2.0*

Date: March 9, 2022

Course: EE 445S Evans

Name: _____ **Matrix** _____ **The** _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

	<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
Morpheus	1	24		Sinusoidal Generation
Neo	2	26		Filter Analysis
Trinity	3	26		Chromagram Filter Design
Agent Smith	4	24		Mystery Systems
	<i>Total</i>	100		

Problem 1.1. Sinusoidal Generation. 24 points.

You're asked to generate one period of a discrete-time sine signal $y[n] = \sin(\omega_0 n)$:

- The continuous-time frequency is 196 Hz ('G' note on the Western scale in the third octave).
 - The sampling rate f_s is 8000 Hz.
- (a) What is the discrete-time frequency ω_0 in rad/sample of the discrete-time sine signal? 4 points.

$$y(t) = \sin(2\pi f_0 t) \text{ where } f_0 = 196 \text{ Hz.}$$

$$y[n] = y(nT_s) = y\left(\frac{n}{f_s}\right) = \sin\left(2\pi \frac{f_0}{f_s} n\right) = \sin(\omega_0 n)$$

$$\text{where } \omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{196}{8000} \text{ is the discrete-time frequency in rad/sample.}$$

- (b) What is the fundamental period of the discrete-time sine signal in samples? 4 points.

Per the [Handout on Discrete-Time Periodicity](#), a sinusoidal signal with discrete-time frequency, where the common factors between integers N and L have been removed,

$$\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{N}{L}$$

has a discrete-time fundamental period of L samples:

$$\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{196 \text{ Hz}}{8000 \text{ Hz}} = 2\pi \frac{49}{2000}$$

In the discrete-time period of $L = 2000$ samples, there are $N = 49$ continuous-time periods of a continuous-time sinusoidal signal at frequency f_0 .

- (c) Give a difference equation whose impulse response will generate the discrete-time sine signal. 4 points. From lab #2, $y[n] = (2 \cos \omega_0) y[n-1] - y[n-2] + (\sin \omega_0) x[n-1]$ for $n \geq 0$ with initial conditions $y[-1], y[-2]$, and $x[-1]$ being zero as necessary conditions for LTI to hold. This difference equation comes from $Z\{\sin(\omega_0 n) u[n]\} = \frac{\sin(\omega_0) z^{-1}}{1 - 2 \cos(\omega_0) z^{-1} + z^{-2}}$ for $|z| > 1$.

- (d) An alternate method to compute the amplitude value is to use the Taylor series expansion for the sine function

$$\sin(\theta) = \theta - \frac{1}{3!} \theta^3 + \frac{1}{5!} \theta^5 - \frac{1}{7!} \theta^7 + \frac{1}{9!} \theta^9 - \dots$$

and keep a finite number of terms. For good quality over one period of θ , we'll need 17 terms, i.e. from the θ term to the θ^{33} term.

Compare the run-time complexity for the difference equation and the lookup table method. The lookup table would store an entire period of sine values computed offline. 12 points.

Method	Total Memory Needed	Multiplications per output sample	Reads per output sample	Writes per output sample
Taylor series	18	291	19	1
Difference equation	7	2	6	4
Lookup table	2000	0	1	1

Taylor series method would compute $\theta = \omega_0 n$ using 1 multiplication, and perform a modulo operation (1 multiplication by $1/(2\pi)$ and 1 additional multiplication) to get the value of angle θ in the first period of $[0, 2\pi)$. For a Taylor series expansion with N non-zero terms,

$$\sin(\theta) = \theta - \frac{1}{3!} \theta^3 + \frac{1}{5!} \theta^5 - \frac{1}{7!} \theta^7 + \frac{1}{9!} \theta^9 - \dots$$

we'll assume the constants $1/m!$ have been computed offline. When $N = 1$, $\sin(\theta) = \theta$ does not need any multiplications; $N = 2$ terms needs 3 mults; $N = 3$ needs $3+5=8$ mults; $N = 5$ needs $3+5+7=15$ mults; $N = 7$ needs $3+5+7+9=24$ mults; or $(N + 1)(N - 1)$ mults in general.

Difference equation contains two constants, previous input value, current output value, two previous output values. The current input is stored into the previous input value. Total memory of 7 words. The difference equation has two multiplications $(2 \cos \omega_0) y[n-1]$ and $(\sin \omega_0) x[n-1]$ per output sample. To compute $y[n]$, the other six values have to be read once each. Also, we'll need to write the result $y[n]$, and update $y[n-1]$, $y[n-2]$ and $x[n-1]$.

Lookup table stores one period of $L = 2000$ samples. We use $n = 0, 1, \dots, L - 1$ to read the precomputed value from the table for $\sin(\omega_0 n)$ and write it out as $y[n]$.

Epilogue for part (d): Although not asked, we can reduce the number of multiplications by using Horner's form of the series expansion which results from iteratively factoring out θ^2 :

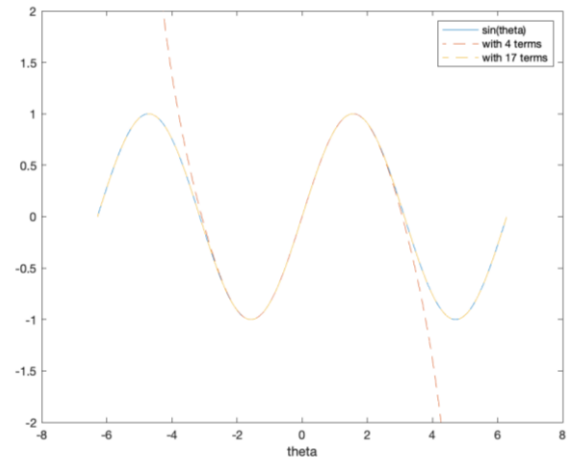
$$\sin(\theta) = \theta - \theta^2 \left(\frac{1}{3!} \theta + \frac{1}{5!} \theta^3 - \frac{1}{7!} \theta^5 + \frac{1}{9!} \theta^7 - \dots \right)$$

$$\sin(\theta) = \theta - \theta^2 \left(\frac{1}{3!} \theta + \theta^2 \left(\frac{1}{5!} \theta - \frac{1}{7!} \theta^3 + \frac{1}{9!} \theta^5 - \dots \right) \right)$$

In this case, Horner's form would have a nested structure of $a_n \theta + b_n \theta^2$ for $n \in [2, N]$. The $N - 1$ nested terms would need $2(N - 1)$ multiplications. We need one multiplication to compute θ^2 , for a total of $2(N - 1) + 1$ multiplications. For $N = 17$ terms, 33 multiplications would be needed for the polynomial calculation, plus 3 multiplications for the calculation of $\theta = \omega_0 n$ and the modulo operation, for a total of 36 multiplications.

Taylor series expansion is about the origin, i.e. $\theta = 0$. For the Taylor series to provide a good fit to $\sin(\theta)$ for $\theta \in [0, 2\pi]$, 17 terms are needed; however, to provide a good fit for $\theta \in [-\pi, \pi]$, only 4 terms are needed (from the θ to the θ^7 terms). This makes sense because the Taylor series expansion is about $\theta = 0$. See https://en.wikipedia.org/wiki/Taylor_series and below.

```
theta = -2*pi : (4*pi)/1000 : 2*pi;
maxi = 17;
sinapprox = zeros(maxi, length(theta));
sinapprox(1,:) = theta;
for i = 2 : maxi
    term = theta.^(2*i-1) / factorial(2*i-1);
    sinapprox(i,:) = sinapprox(i-1,:) + (-1)^(i+1) * term;
end
plot(theta, sin(theta), '-', ...
      theta, sinapprox(4,:), '--', ...
      theta, sinapprox(maxi,:), '--');
xlabel('theta');
ylim([-2, 2]);
legend('sin(theta)', 'with 4 terms', 'with 17 terms');
```



In Matlab, the pattern of three dots ... means that the command continues on the next line.

Problem 1.2 Filter Analysis. 26 points.

In an electric oven, transfer of energy from a heating element to food is governed by the heat equation.

However, an approximate model is given by the following causal linear time-invariant discrete-time filter with input $x[n]$ and output $y[n]$

$$y[n] = \sum_{m=0}^{n-1} \frac{x[m]}{\alpha^{n-m}} \text{ for } n \geq 0$$

where $x[n]$ represents the power delivered to the heating element, α is the thermal diffusivity, and $y[n]$ represents the temperature of the food where a temperature of 0 means room temperature.

We expand the summation as follows

$$y[n] = \frac{1}{\alpha} x[n-1] + \frac{1}{\alpha^2} x[n-2] + \dots \text{ for } n \geq 0$$

and convert $y[n]$ to a recursive difference equation

$$y[n] = \frac{1}{\alpha} x[n-1] + \frac{1}{\alpha} y[n-1] \text{ for } n \geq 0$$

For this problem, assume that $\alpha = 2$. We observe the system starting at $n = 1$.

- (a) Assume that the food is initially at room temperature. What are the initial conditions and their values?

Why? *4 points.* **We observe system starting at $n = 1$: $y[1] = \frac{1}{2}x[0] + \frac{1}{2}y[0]$. Initial conditions $x[0]$ and $y[0]$ should be 0 as necessary conditions for linear and time-invariance (LTI) to hold.**

- (b) Give a formula for the impulse response of the filter $h[n]$. Simplify any summations. *6 points.*

To find the impulse response, input an impulse signal, i.e. let $x[n] = \delta[n]$:

$$h[n] = \frac{1}{2} \delta[n-1] + \frac{1}{2} h[n-1] \text{ for } n \geq 1$$

where the initial condition $h[0] = 0$. We can compute this iteratively for $n \geq 1$

$$h[1] = \frac{1}{2} \delta[0] + \frac{1}{2} h[0] = \frac{1}{2} \text{ and } h[2] = \frac{1}{2} \delta[1] + \frac{1}{2} h[1] = \frac{1}{2^2} \text{ etc.}$$

$$h[n] = \frac{1}{2^n} u[n-1] = \left(\frac{1}{2}\right)^n u[n-1]$$

- (c) Is this a finite impulse response (FIR) or infinite impulse response (IIR) filter? *4 points.*

IIR due to feedback. That is, the output $y[n]$ depends on the previous output value $y[n-1]$.

- (d) Derive a formula for the transfer function in the z -domain and the region of convergence. *6 points.*

Solution #1: Take the z -transform of the impulse response.

$$H(z) = Z \left\{ \left(\frac{1}{2}\right)^n u[n-1] \right\} = \frac{1}{2} Z \left\{ \left(\frac{1}{2}\right)^{n-1} u[n-1] \right\} = \frac{1}{2} z^{-1} Z \left\{ \left(\frac{1}{2}\right)^n u[n] \right\} = \frac{\frac{1}{2} z^{-1}}{1 - \frac{1}{2} z^{-1}} \text{ for } |z| > \frac{1}{2}$$

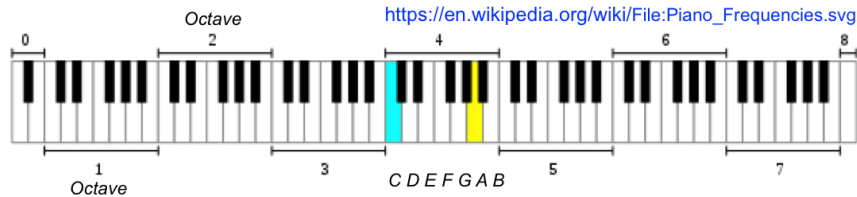
Solution #2: Take the z -transform of the difference equation.

$$Y(z) = \frac{1}{2} z^{-1} X(z) + \frac{1}{2} z^{-1} Y(z) \text{ which gives } H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{1}{2} z^{-1}}{1 - \frac{1}{2} z^{-1}} \text{ for } |z| > \frac{1}{2}$$

- (e) Give the frequency selectivity of filter (lowpass, highpass, bandpass, bandstop, allpass, notch) and explain your reasoning. *6 points.* **The transfer function has a pole at $z = 1/2$ and a zero at $z = 0$. Pole angle of 0 rad/sample gives center of passband. This corresponds to a lowpass filter.**

Problem 1.3 Chromagram Filter Design. 26 points.

The notes on the Western scale on an 88-key piano keyboard grouped into octaves follow:



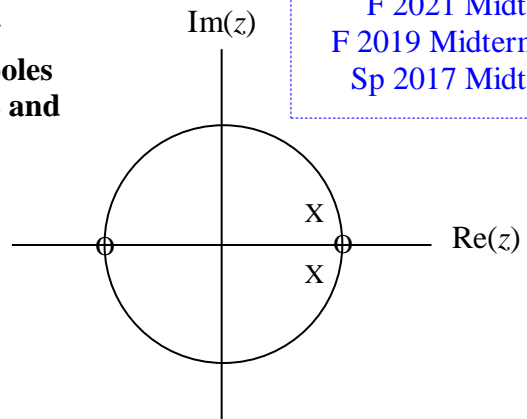
The frequency of note A3 (i.e. ‘A’ in the 3rd octave) at 220 Hz is twice the frequency of A2 at 110 Hz. This type of octave spacing occurs for all the notes on the Western scale.

For this problem, assume that the sampling rate f_s is 16,000 Hz.

When poles and zeros are separated in angle, the angles of poles near but inside the unit circle indicate passband frequencies and the angles of the zeros on or near the unit circle indicate the stopband frequencies, per [lecture 6 slides 6-8 to 6-10](#).

F 2021 Midterm 1.3
F 2019 Midterm 1.3(a)
Sp 2017 Midterm 1.3

- (a) Design a second-order infinite impulse response (IIR) to extract the note A5 (880.0 Hz). The filter should suppress all other frequencies including the neighboring notes G#5 (830.6 Hz) and A#5 (932.3 Hz). 12 points.
- Give formulas for the pole and zero locations.
 - Plot poles and zeros on the diagram on the right.



Two zeros at $z = \pm 1$ and two poles at $z = 0.95 e^{\pm j \omega_{A5}}$ where $\omega_{A5} = 2\pi \times \frac{880}{16000}$

- (b) To extract the note A4, how would the design of the filter in part (a) change? 6 points.

Use $\omega_{A4} = 2\pi \times \frac{440}{16000}$ for placement of the poles, which is half the angles of the poles in (a).

Poles will move toward the real axis but at the same magnitude as the poles in (a).

Lecture Slides
5-21 to 5-25

- (c) Design a first-order discrete-time IIR filter to perform the following smoothing operation

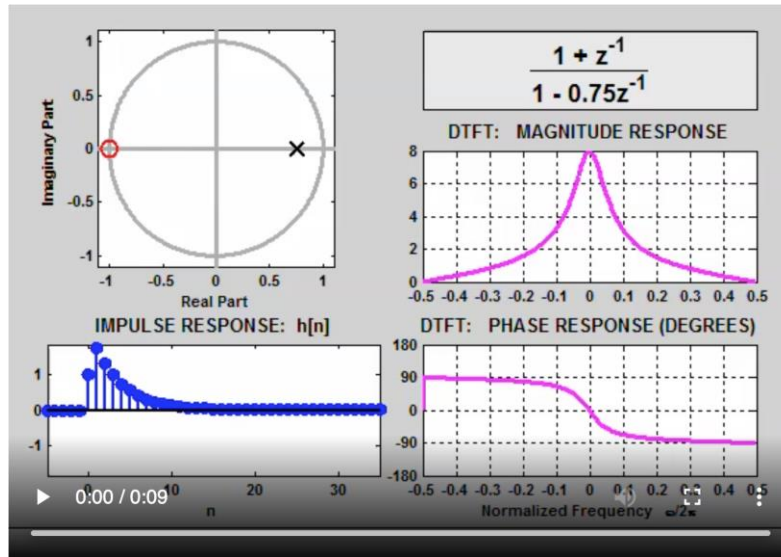
$$y_A[n] = \text{Smooth}\{v[n]\} \text{ where } v[n] = \sum_{k=1}^6 (x[n] * h_{A_k}[n])^2$$

where $h_{A_k}[n]$ is the impulse response of an IIR filter that extracts the note A_k from audio signal $x[n]$, and $y_A[n]$ is the output of the smoothing filter for A notes A1, A2, ... A6. This operation is used to construct a [chromagram to analyze musical recordings](#). 8 points.

A smoothing filter is a lowpass filter; that is, it smooths out sudden changes in the data which corresponding to high-frequency information. In class, we have seen examples of applying an averaging filter, which is a lowpass filter, to create a seven-day moving average of new COVID-19 cases in the [Designing Averaging Filters](#) handout and smooth/blur an image in the “[Cascading Two FIR Filters](#)” DSP First demo mentioned in [lecture 5 slide 5-21](#).

For a first-order IIR filter, place the pole at $z = 0.9$ for a passband centered at 0 rad/sample and the zero at $z = -1$ for a stopband centered at π rad/sample.

We had seen a similar lowpass first-order IIR filter in a DSP First demo (more on next page).

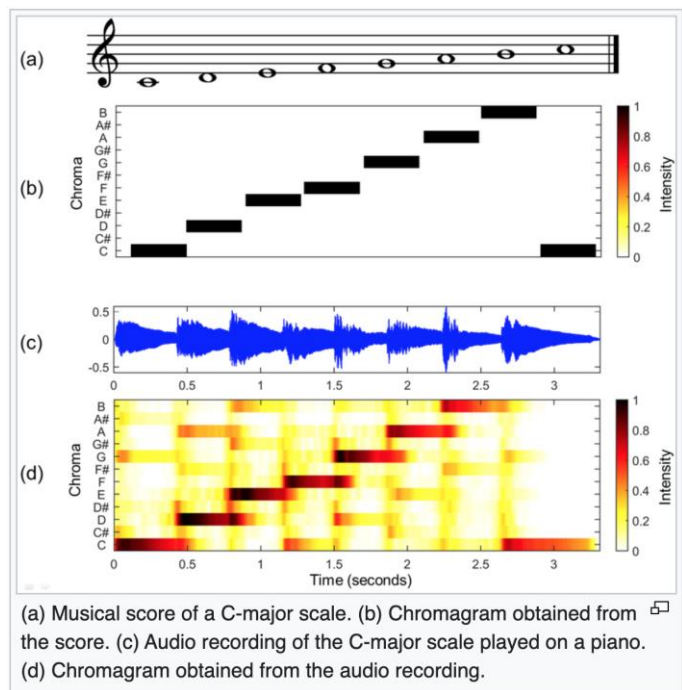


DSP First “[Three-Domain Connections](#)” demo “IIR filter with one pole and one zero.”
 In this demo, the pole is at $z = 0.75$ and the zero is at $z = -1$ for a lowpass response.
 See [lecture 6 slide 6-10](#).

Epilogue: From Wikipedia "[Chroma Feature](#)" (Chromagram):

“In Western [music](#), the term **chroma feature** or **chromagram** closely relates to the twelve different [pitch classes](#). Chroma-based features, which are also referred to as “[pitch class profiles](#)”, are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories) and whose tuning approximates to the [equal-tempered scale](#). One main property of chroma features is that they capture harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation.”

The chromagram (as shown on the right) is a tool used to analyze musical recordings based on the equal temperament scale. It is similar to the spectrogram except that it has exactly twelve frequency bins (corresponding to twelve notes on the Western scale). For each note, all harmonics are combined into the same frequency bin.



Problem 1.4. Mystery Systems. 24 points.

You're trying to identify unknown discrete-time systems.

You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 4000 Hz over 0 to 5s

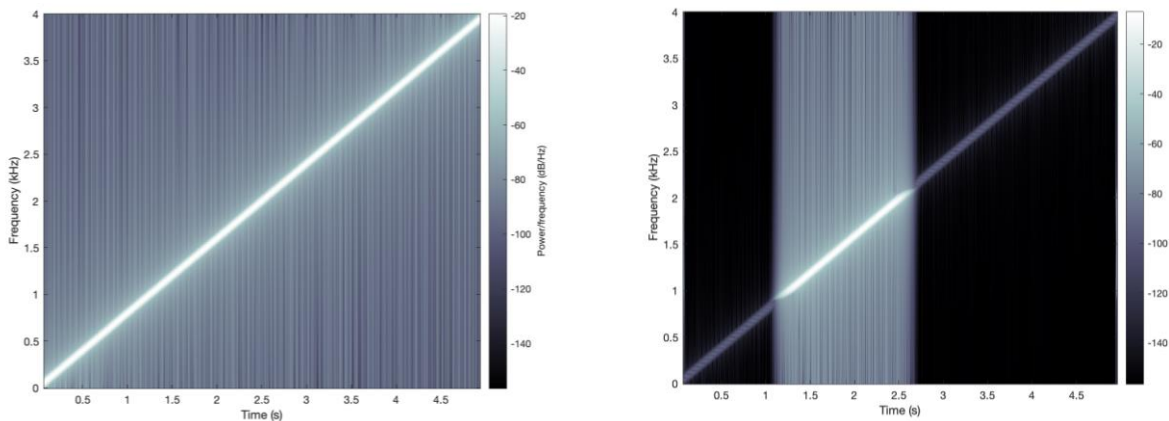
$$x(t) = \cos(2\pi f_1 t + 2\pi \mu t^2)$$

where $f_1 = 0$ Hz, $f_2 = 4000$ Hz, and $\mu = \frac{f_2 - f_1}{2 t_{\max}} = \frac{4000 \text{ Hz}}{10 \text{ s}} = 400 \text{ Hz}^2$. Sampling rate f_s is 8000 Hz.

In each part below, identify the unknown system as one of the following **with justification**:

1. filter – give selectivity (lowpass, highpass, bandpass, bandstop) and passband/stopband frequencies
2. upsampler – give upsampling factor
3. downsampler – give downsampling factor

(a) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lectures
1 3 5 & 6

Lab #3

HW 0.1 1.1
1.3 2.1 & 2.2

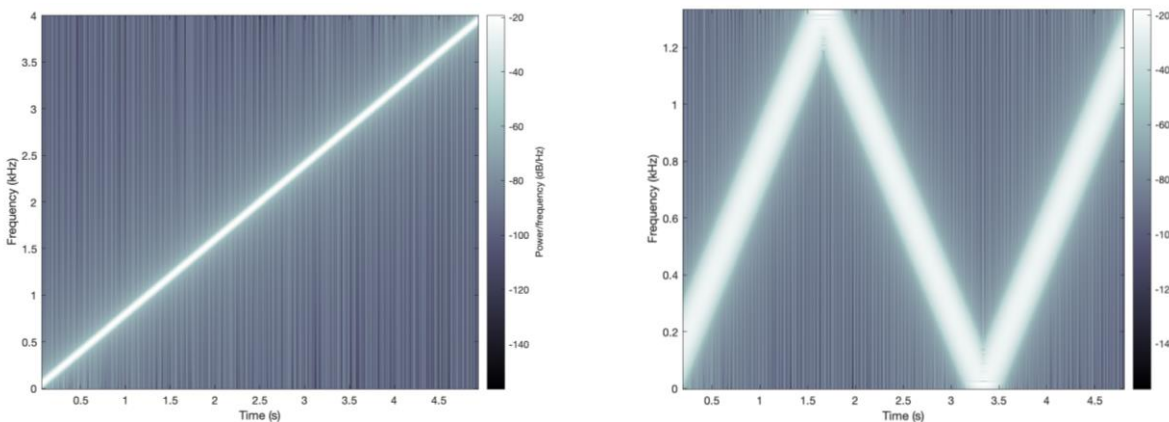
JSK Ch. 7

Midterm 1.1
F 2020
Sp 2020
F 2018

Designing
Averaging
Filters

From the output spectrogram, frequencies from about 900 to 2100 Hz are passed. Principal frequencies below 900 Hz and above 2100 Hz are severely attenuated. In the grayscale color map, white has highest magnitude value. This is a bandpass filter.

(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lecture

HW 0.3 & 2.2

Midterm
Problems
1.2 F 18
1.2(d) Sp 18
1.2(d) F 09

When compared to the input spectrogram, the output spectrogram has about one-third the range of frequencies values and the principal frequency is a chirp pattern that is wider and that has aliasing. Downsampling by 3 per homework problem 2.2(d).

Epilogue: Matlab code to generate the spectrograms for problem 1.4.

(a) Bandpass filter

```
fs = 8000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Design lowpass filter
fnyquist = fs/2;
fstop1 = 900;
fpass1 = 1100;
fpass2 = 1900;
fstop2 = 2100;
ctfrequencies = [0 fstop1 fpass1 fpass2 fstop2 fnyquist];
idealAmplitudes = [0 0 1 1 0 0];
pmfrequencies = ctfrequencies / fnyquist;
filterOrder = 200;
h = firpm( filterOrder, pmfrequencies, idealAmplitudes );
h = h / sum(h.^2);

y = conv(x, h);

%% Plot spectrogram of signal
blockSize = 1024;
overlap = 1023;
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```

(b) Downsampling by 3

```
fs = 8000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;
%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));
%% Downsampling by 3
y = x(1:3:end);
blockSize = 1024;
overlap = 1023;
spectrogram(y, blockSize, overlap, blockSize, fs/3, 'yaxis');
```

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 **Version 4.0**

Date: October 12, 2022

Course: EE 445S Evans

Name: _____ **Set** **Solution**
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	28		FIR Filter Analysis
2	25		IIR Filter Analysis
3	27		Filter Design
4	24		Potpourri
<i>Total</i>	104		

**Instructor caught the error of the points adding to 104 while grading the test.
The original intent was to have problem 1 count as 24 points.**

Problem 1.1 FIR Filter Analysis. 28 points.

Consider the following causal linear time-invariant (LTI) discrete-time finite impulse response (FIR) filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = a x[n] + b x[n-1] - b x[n-3] - a x[n-4]$$

for $n \geq 0$, where a and b are real-valued positive coefficients.

Please note that the coefficient in front of the $x[n-2]$ term is zero.

(a) What are the initial conditions and their values? Why? 6 points.

Let $n=0$: $y[0] = a x[0] + b x[-1] - b x[-3] - a x[-4]$.

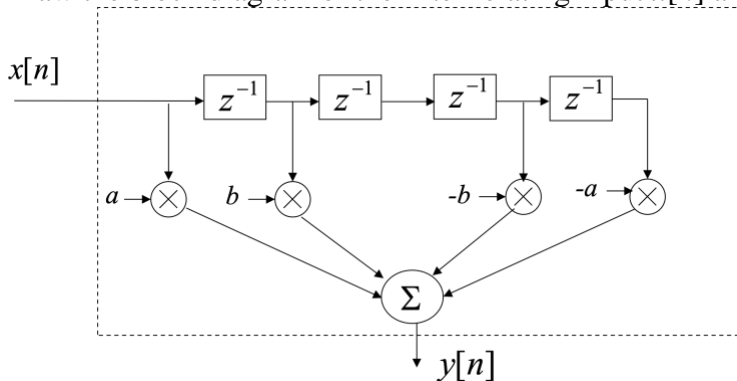
Let $n=1$: $y[1] = a x[1] + b x[0] - b x[-2] - a x[-3]$.

Let $n=2$: $y[2] = a x[2] + b x[1] - b x[-1] - a x[0]$. etc.

Initial conditions are $x[-1], x[-2], x[-3], x[-4]$ which must be zero for linearity and time-invariant properties to hold. $x[0]$ is the first input value and not an initial condition.

Note: A causal system does not depend on future input values or future output values.

(b) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



Note: The four initial conditions are visible here as the initial condition for each unit delay block.

Lecture slides 3-15 & 5-5

(c) Derive a formula for the transfer function in the z -domain and the region of convergence. 4 points.

Z-transform both sides of difference equation, knowing that all initial conditions are zero:

$$Y(z) = a X(z) + b z^{-1} X(z) - b z^{-3} X(z) - a z^{-4} X(z) \text{ which means}$$

$$H(z) = \frac{Y(z)}{X(z)} = a + b z^{-1} - b z^{-3} - a z^{-4} \text{ for } z \neq 0$$

Lecture slide 5-11

(d) Give a formula for the discrete-time frequency response of the filter. 3 points.

We can convert the transfer function $H(z)$ into the discrete-time frequency domain by substituting $z = \exp(j\omega)$ because FIR LTI systems are always Bounded-Input Bounded-Output stable, or equivalently, because the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = a + b e^{-j\omega} - b e^{-3j\omega} - a e^{-4j\omega}$$

(e) Give a formula for the phase response vs. discrete-time frequency. 6 points.

$$H_{freq}(\omega) = e^{-j2\omega} (a e^{j2\omega} + b e^{j\omega} - b e^{-j\omega} - a e^{-j2\omega}) = 2(b \sin(\omega) + a \sin(2\omega)) j e^{-j2\omega}$$

With $j = e^{j\frac{\pi}{2}}$, $\angle H_{freq}(\omega) = \frac{\pi}{2} - 2\omega$ except for phase jumps (discontinuities) of π at frequencies that are zeroed out and don't get through, which is generalized linear phase.

(f) Give a formula for the group delay vs. discrete-time frequency. 3 points.

$$GD(\omega) = -\frac{d}{d\omega} \angle H_{freq}(\omega) = 2 \text{ samples}$$

Problem 1.2 IIR Filter Analysis. 25 points.

For a second-order infinite impulse response (IIR) filter with poles p_0 and p_1 and zeros z_0 and z_1 , the transfer function in the z-domain is

$$H(z) = \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)}$$

The poles will remain at $p_0 = -0.9$ and $p_1 = 0.9$ in this problem. Region of convergence is $|z| > 0.9$. Each question below will ask you to determine the frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass and notch) of the second-order IIR filter with different choices of zero locations.

- (a) From the transfer function $H(z)$, derive an expression for the magnitude response of the filter. 3 points.

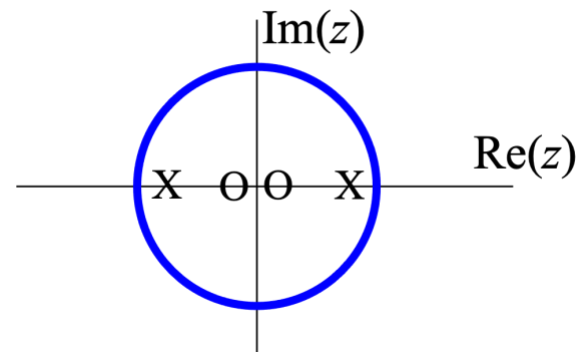
We can convert the transfer function $H(z)$ into the discrete-time frequency domain by substituting $z = \exp(j\omega)$ because the region of convergence includes the unit circle:

$$H(e^{j\omega}) = \frac{(e^{j\omega} - z_0)(e^{j\omega} - z_1)}{(e^{j\omega} - p_0)(e^{j\omega} - p_1)}$$

Then, take the absolute value of both sides:

$$|H(e^{j\omega})| = \frac{|(e^{j\omega} - z_0)(e^{j\omega} - z_1)|}{|(e^{j\omega} - p_0)(e^{j\omega} - p_1)|} = \frac{|e^{j\omega} - z_0||e^{j\omega} - z_1|}{|e^{j\omega} - p_0||e^{j\omega} - p_1|}$$

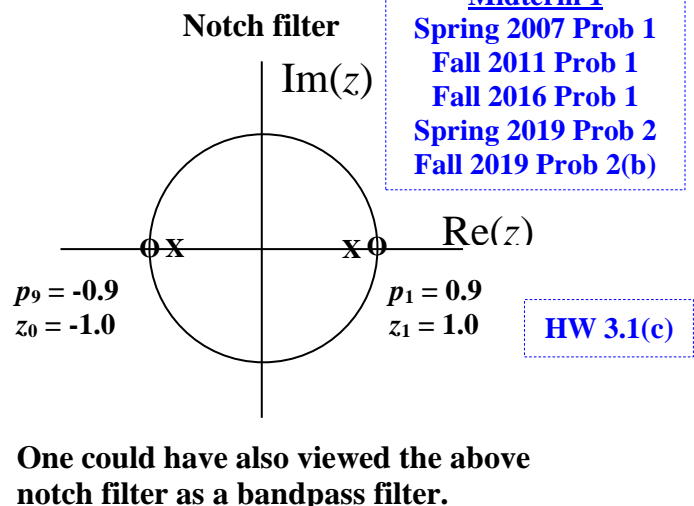
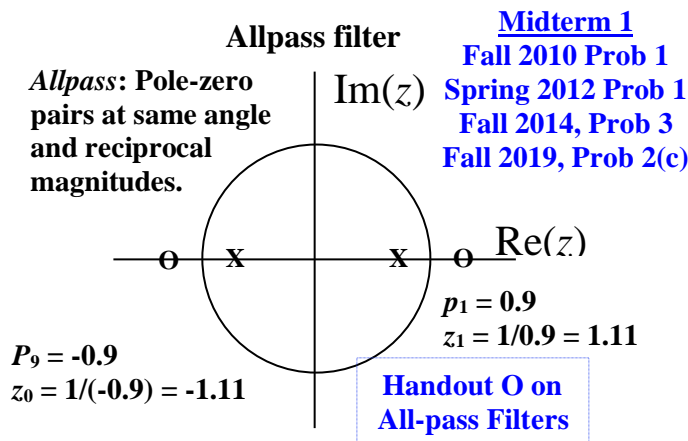
- (b) Let the zeros be $z_0 = -0.1$ and $z_1 = 0.1$ as shown on the right. What is the frequency selectivity? 4 points.



Bandstop. The frequency (angle) of a pole near but inside the unit circle indicates a peak in the magnitude response at that frequency. This comes from the Euclidean distance $|e^{j\omega} - p_0|$ in the denominator of the magnitude response. The minimum distance occurs when ω is equal to the angle of the pole p_0 . The zeros, because they are close to the origin, have little effect on the magnitude response due to the $|e^{j\omega} - z_0|$ term which is close to 1 in value for any ω .

- (c) Let the zeros continue being real-valued and negatives of each other, i.e. $z_0 = -z_1$. For every frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass and notch) that is possible for the second-order IIR filter to achieve, give the values of zeros z_0 and z_1 . You may reuse your answer from part (b). 18 points.

Bandstop filter from part (b) above.



Problem 1.3 Filter Design. 27 points.

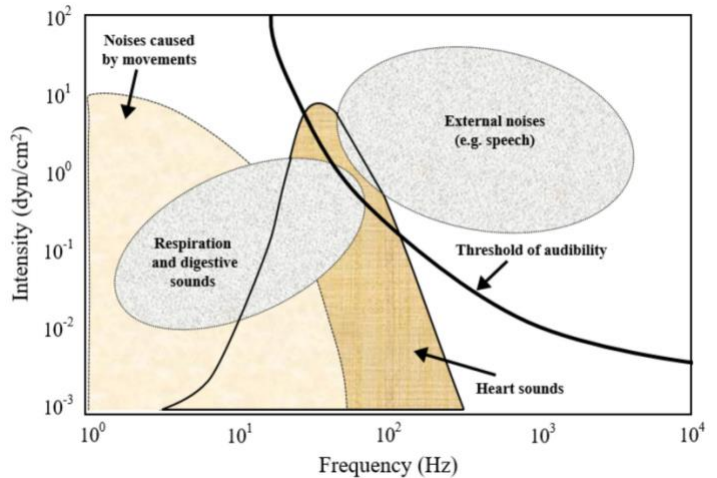
A stethoscope allows a physician to listen to sounds of the circulatory and respiratory system especially the heart and lungs.

A conventional stethoscope uses a tube to directly transmit the vibration to the physician’s ear.

A digital stethoscope uses a microphone and an analog-to-digital converter to record the signal.

This problem will design three discrete-time filters placed in cascade for a digital stethoscope.

Assume a sampling rate of 8000 Hz.

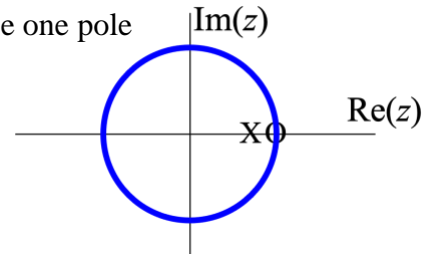


- (a) Noise caused by physical motion of the patient or stethoscope occurs at extremely low frequencies (near 0 Hz). Design a first-order IIR filter to remove noise due to movement by giving the numeric values of the one pole and one zero. Place the pole and zero on the pole-zero diagram. 9 points.

Remove noise at extremely low frequencies centered at 0 rad/sample.

DC notch IIR filter. Pole at $z = 0.9$ and zero at $z = 1$.

This can also be thought of as a highpass filter with a narrowband passband.

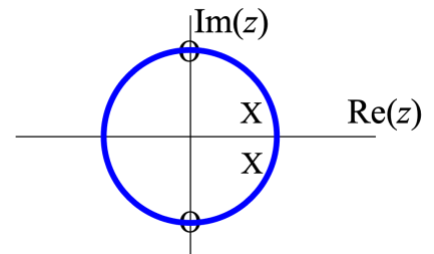


- (b) The sounds produced by the circulatory and respiratory systems occur at low frequencies (<500 Hz). Background noise in a hospital (especially from speech) occurs mostly at frequencies above 500 Hz. Design a second-order IIR filter by specifying the two poles and two zeros to remove background noise above 500 Hz. Place the poles and zeros on the pole-zero diagram on the right. 9 points.

Lowpass filter. Passband is from 0 to 500 Hz. We could place poles

at 250 Hz and -250 Hz. For 250 Hz, the discrete-time frequency is $\omega_{250} = 2\pi \frac{250 \text{ Hz}}{8000 \text{ Hz}}$ and pole locations would be at $p_0 = 0.9 e^{j\omega_{250}}$

and $p_1 = 0.9 e^{-j\omega_{250}}$. Stopband would be from 550 Hz to 4000 Hz, and the zeros could be placed on the unit circle at an angle in this range and its negative. We choose 2000 Hz and -2000 Hz, which would put zeros at $\pi/2$ and $-\pi/2$.



- (c) The digital stethoscope has two modes: One for listening to respiratory and digestive sounds and one for listening to circulatory heart sounds. Design a second-order IIR filter by specifying the two poles and two zeros to extract sounds of the heart between 120 Hz and 500 Hz. Place the poles and zeros on the pole-zero diagram on the right. 9 points.

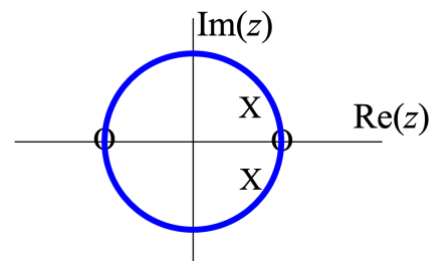
Bandpass filter with passband from 120 Hz to 500 Hz.

Poles must be conjugate symmetric because each frequency Component has a positive and negative component.

Place a pole at $(500 \text{ Hz} + 120 \text{ Hz})/2 = 310 \text{ Hz}$ and at -310 Hz.

$p_0 = 0.9 e^{j\omega_{310}}$ and $p_1 = 0.9 e^{-j\omega_{310}}$ where $\omega_{310} = 2\pi \frac{310 \text{ Hz}}{8000 \text{ Hz}}$.

Put zeros to eliminate 0 and π rad/sample, i.e. let $z = 1$ and zero at $z = -1$.



Problem 1.4. Potpourri. 24 points.

(a) Consider the signal $x[n] = (-1)^n$ observed for all time $-\infty < n < \infty$. 12 points.

I. If we express $x[n] = \cos(\omega_0 n)$ observed for all time $-\infty < n < \infty$, give the value of ω_0 . 4 points.

$\omega = \pi$ because $\cos(\pi n) = (-1)^n$

II. Give a formula for $y[n]$ that is the output of downsampling by 2 applied to $x[n]$. 4 points

Downsampling by 2 of signal $x[n]$ keeps the even-indexed values of $x[n]$.

$y[n] = x[2n] = \cos(\pi (2n)) = \cos(2\pi n) = 1$ for $-\infty < n < \infty$

III. In part II above, explain why the principal frequency of $x[n]$ became the principal frequency in $y[n]$. 4 points

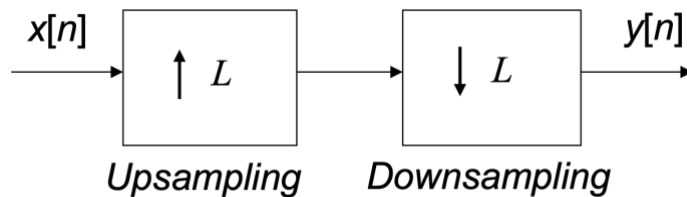
$y[n] = x[2n] = \cos(\pi (2n)) = \cos(2\pi n) = 1$ for $-\infty < n < \infty$.

The principal frequency is π rad/sample in $x[n]$ and 2π rad/sample in $y[n]$.

Downsampling by 2 doubles the frequencies in $x[n]$ which causes the frequencies in the input signal from $\pi/2$ to π to alias. In this case, the input frequency π becomes the frequency 2π rad/sample which aliases to 0 rad/sample (DC).

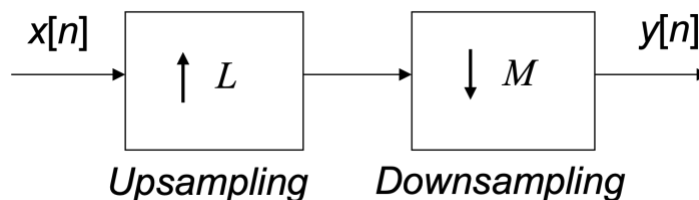
(b) Upsampling by L can be used to increase the sampling rate of the input signal by a factor of L and downsampling by M can be used to decrease the sampling rate of the input signal by a factor of M .

I. What is the sampling rate change from $x[n]$ to $y[n]$ for the system below? 6 points.



Upsampling by L increases the sampling rate by L which is then undone by downsampling by L which decreases the sampling rate by L . There is no sampling rate change from $x[n]$ to $y[n]$.

II. What is the sampling rate change from $x[n]$ to $y[n]$ for the system below? 6 points.



Upsampling by L increases the sampling rate by L and downsampling by M decreases the sampling rate by M . The sampling rate change is L/M from $x[n]$ to $y[n]$.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 Solutions Version 3.0

Date: March 8, 2023

Course: EE 445S Evans

Name: _____ **Place,** _____ **The Good**
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

	<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
<i>Jason</i>	1	25		IIR Filter Analysis
<i>Eleanor</i>	2	24		Increasing the Sampling Rate
<i>Chidi</i>	3	27		Equalizer Design
<i>Tahani</i>	4	24		Potpourri
	<i>Total</i>	100		

Problem 1.1 IIR Filter Analysis. 25 points.

Consider the following causal linear time-invariant (LTI) discrete-time infinite impulse response (IIR) filter with input $x[n]$ and output $y[n]$ described by

$$y[n] - b^2 y[n - 2] = x[n]$$

for $n \geq 0$, where b is a real-valued positive coefficient less than one, i.e. $0 < b < 1$.

Please note that the coefficient in front of the $y[n-1]$ term is zero.

(a) What are the initial conditions and their values? Why? 6 points.

Consider $n = 0$: $y[0] = b^2 y[-2] + x[0]$

Initial Condition

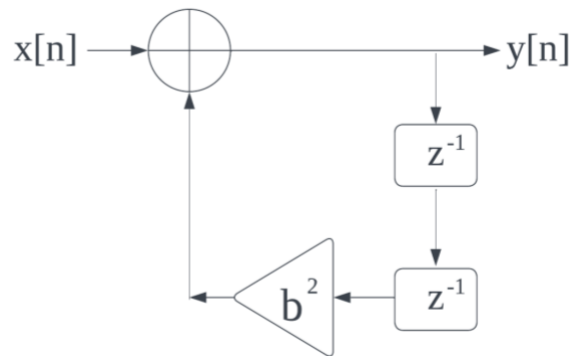
Consider $n = 1$: $y[1] = b^2 y[-1] + x[1]$

Initial Condition

Our initial conditions must equal zero as necessary conditions for LTI to hold:

$$y[-2] = y[-1] = 0$$

(b) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. 6 points.



(c) Derive a formula for the transfer function in the z -domain and the region of convergence. 4 points.

$$Y(z) = b^2 z^{-2} Y(z) + X(z)$$

$$Y(z) - b^2 z^{-2} Y(z) = X(z)$$

$$Y(z)(1 - b^2 z^{-2}) = X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - b^2 z^{-2}}$$

$$H(z) = \frac{1}{(1 - b z^{-1})(1 + b z^{-1})}, \text{ for } |z| > b$$

take z-Transform of $y[n] = b^2 y[n - 2] + x[n]$

combine like terms

recall the transfer function is $H(z) = \frac{Y(z)}{X(z)}$

poles are at $z = b$ and $z = -b$.

final answer

(d) Give a formula for the discrete-time frequency response of the filter. *3 points.*

We substitute $z = e^{j\omega}$, because the unit circle is contained inside the Region of Convergence $|z| > b$ where $0 < b < 1$:

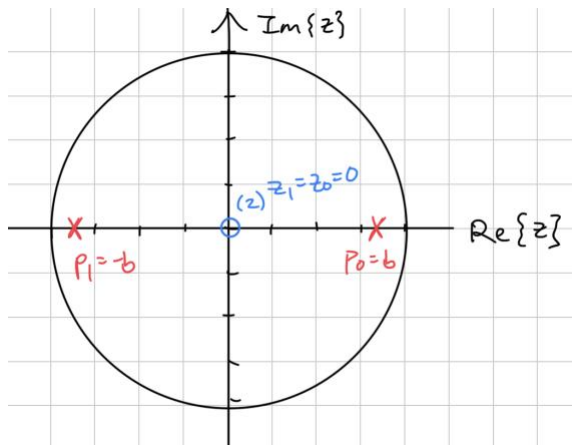
$$H(e^{j\omega}) = \frac{1}{(1 - be^{-j\omega})(1 + be^{-j\omega})} = \frac{1}{1 - b^2 e^{-j2\omega}}$$

(e) What frequency responses are possible among lowpass, highpass, bandpass, bandstop, allpass and notch? For each possibility, give a value of b that would give that response. *6 points.*

From our transfer function, $H(z) = \frac{1}{1 - b^2 z^{-2}} = \frac{z^2}{z^2 - b^2}$

Thus, we know that...

- 2 zeros exist at $z = 0$
- 2 poles exist at $p = \pm b$



Consider as $b \rightarrow 1$

The 2 poles $p_0 \rightarrow 1, p_1 \rightarrow -1$

(Note that the 2 poles are not equal to 1! Because then the poles would be on the unit circle, and we would encounter Bounded-Input Bounded-Output instability issues.)

With these pole locations, it means the poles are amplifying low and high frequencies.

Thus, we can see that this as a bandstop filter.

Consider as $b \rightarrow 0$

The 2 poles approach $p_0 = p_1 = 0$

Now the 2 zeros and the 2 poles are on the same location, cancelling out.

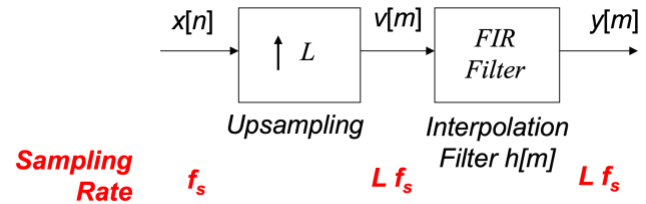
Thus, we can see that this as an allpass filter.

Problem 1.2 Increasing the Sampling Rate. 24 points.

Upsampling by L can be used to increase the sampling rate of the input signal by a factor of L .

A lowpass finite impulse response (FIR) filter can then be applied to the output of the upsampler to attenuate the high frequencies introduced by upsampling.

On the right, discrete-time index n is associated with sampling rate f_s and discrete-time index m is associated with sampling rate $L f_s$.



- (a) What is the maximum continuous-time frequency f_{max} that is present in $x[n]$? What discrete-time frequency does f_{max} correspond to? 6 points.

$$x(t) = \cos(2\pi f_0 t), \quad \text{for } -\infty < t < \infty$$

$$x(t) = \cos\left(2\pi f_0 \left(\frac{n}{f_s}\right)\right)$$

$$\omega_0 = 2\pi \frac{f_0}{f_s}$$

From Nyquist Theorem, we require that $f_s > 2f_{max}$

Thus, we have the continuous-time frequency is:

$$f_{max} < \frac{1}{2} f_s$$

And the corresponding discrete-time frequency is:

$$\omega_{max} = 2\pi \frac{f_{max}}{f_s} = 2\pi \frac{\left(\frac{1}{2} f_s\right)}{f_s} = \pi$$

Thus, from sampling, we can capture a frequency range of...

$$-\frac{1}{2} f_s < f < \frac{1}{2} f_s \text{ (continuous time frequencies)}$$

$$-\pi < \omega < \pi \text{ (discrete - time frequencies)}$$

- (b) What discrete-time frequency in $v[m]$ corresponds to the maximum continuous-time frequency f_{max} that is present in $x[n]$? 6 points.

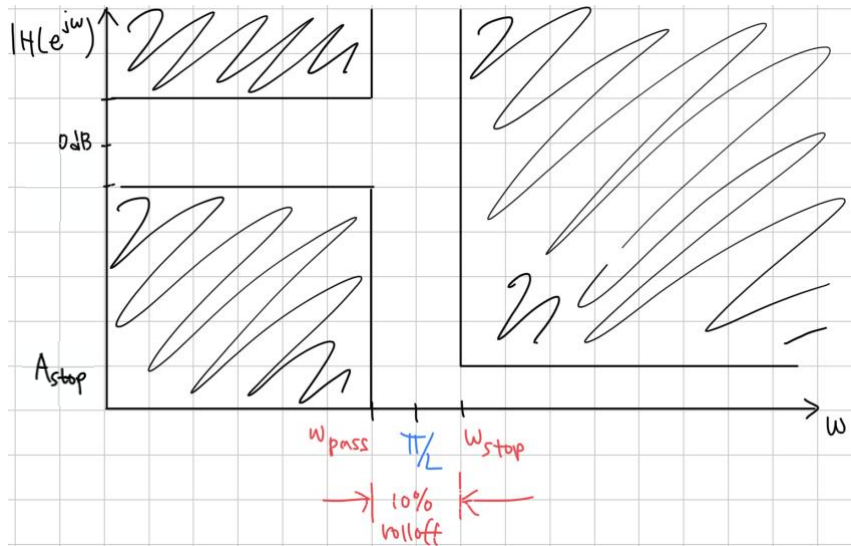
From our solution in part (a), $f_{max} < \frac{1}{2} f_s$

And we are given that at $v[m]$, our new sampling rate is $L f_s$

Using the same equation for discrete-time frequency, we find that our solution is:

$$\omega_{\text{max_at_}v[m]} = 2\pi \frac{f_{max}}{f_{s_at_}v[m]} = 2\pi \frac{\left(\frac{1}{2} f_s\right)}{L f_s} = \frac{\pi}{L}$$

- (c) Any discrete-time frequencies present in $v[m]$ higher than your answer in part (b) but less than π rad/sample correspond to frequencies introduced by upsampling. Give the discrete-time passband frequency ω_{pass} and stopband frequency ω_{stop} you would use for the lowpass filter design. *6 points.*



To determine our ω_{pass} and ω_{stop} values, we want to consider a reasonable value of 10% for the roll-off factor. Meaning, $\omega_{stop} = 1.1\omega_{pass}$

Thus, our values result in...

$$\omega_{pass} = 0.95 \frac{\pi}{L}$$

$$\omega_{stop} \cong 1.05 \frac{\pi}{L}$$

- (d) Give two ways to design a lowpass FIR filter with linear phase to meet the specifications of a discrete-time passband frequency ω_{pass} and stopband frequency ω_{stop} . A “way” to design the filter could be a formula, algorithm, etc. *6 points.*

Solution I:

Design an averaging filter with N coefficients

The first zero should occur at $\frac{2\pi}{N}$

To determine N, we know that $\frac{2\pi}{N} = \frac{\pi}{L}$

Thus, $N = 2L$

Solution II:

Use filterDesigner from MATLAB

Use the design methods such as Parks-McClellan, Least-Squares, or Kaiser Window

Solution III:

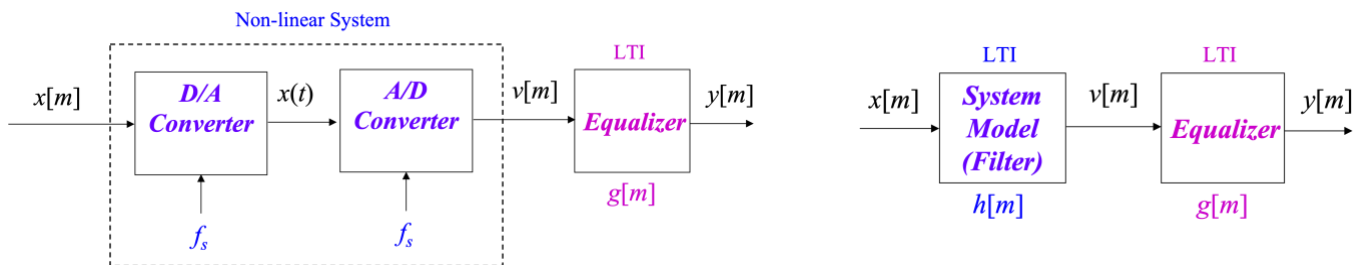
Place zeros on a pole-zero plot for a low-pass filter

Problem 1.3 Equalizer Design. 27 points.

Many applications use a digital-to-analog (D/A) converter and an analog-to-digital (A/D) converter.

- An audio system would use a D/A converter for playback over earbuds or speakers and an A/D converter in a microphone for recording and mixing.
- A digital communication system would use a D/A converter in the transmitter and an A/D converter in the receiver.

You'll design a linear time-invariant (LTI) bound-input bounded-output (BIBO) stable discrete-time equalizer to compensate the distortion in the nonlinear system (on the left) using an LTI model of the nonlinear system (on right):



- (a) Give a formula for a test signal to use for $x[m]$ that would be of finite duration. The test signal should have all discrete-time frequencies in it. The test signal would be used to estimate the impulse response $h[m]$ for the LTI model of the cascade of the D/A and A/D converters. 12 points.

Solution I: Use a chirp signal: $x(t) = \cos(\underbrace{2\pi f_0 t + \pi \mu t^2}_{\theta(t)})$

The chirp will sweep continuous-time frequencies:

$$0 \text{ Hz} \rightarrow f_{max} = \frac{1}{2} f_s \text{ Hz}$$

where the instantaneous frequency is:

$$\frac{d\theta(t)}{dt} = 2\pi f_0 + 2\pi \mu t = 2\pi(f_0 + \mu t) \text{ rad/s}$$

$$f_i(t) = f_0 + \mu t, f_0 = 0 \text{ Hz}$$

$$f_{max} = \frac{1}{2} f_s = \mu t_{max}$$

This chirp signal would sweep discrete-time frequencies from 0 rad/sample \rightarrow π rad/sample

Solution II:

Use a PN (Pseudo-Random Noise) Sequence because it covers all frequencies

- r^{th} -order connection polynomial
- $2^r - 1$ bits for fundamental period

We subtract 1 because the state of all zeros is a lockout state that does not change

Solution III:

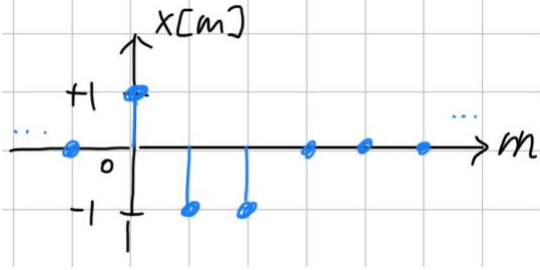
Use an impulse response: $x[m] = \delta[m]$

In theory this is a good/feasible idea.

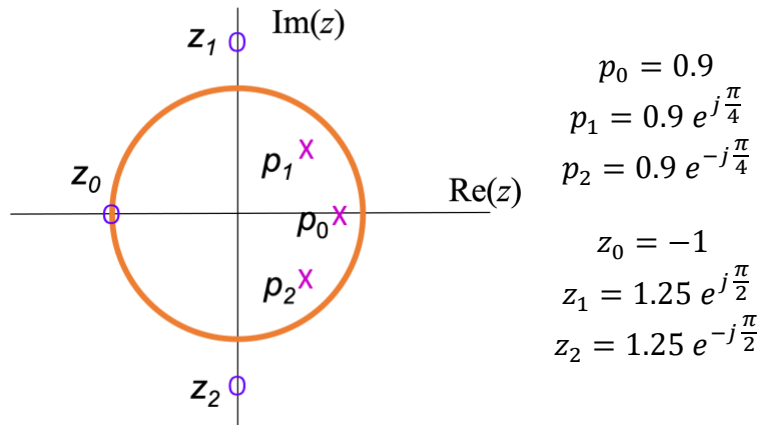
In practice, it does not capturing variability over time in the unknown system.

Note: Impulsive events used in seismic modelling of acoustic layers (eg., earth layers)

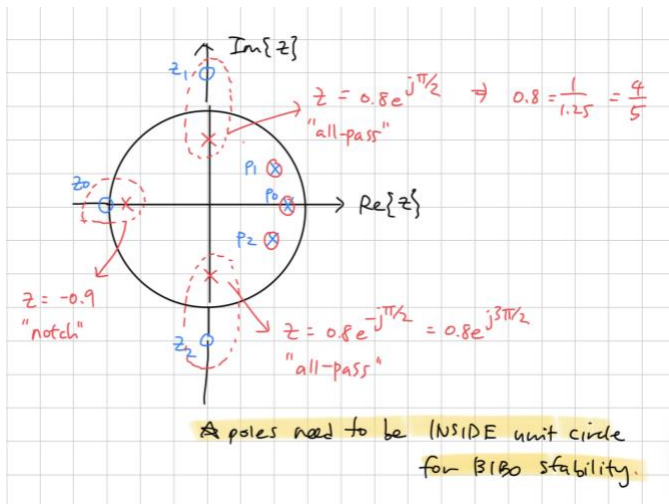
Solution IV: Use a special case of a short PN sequence



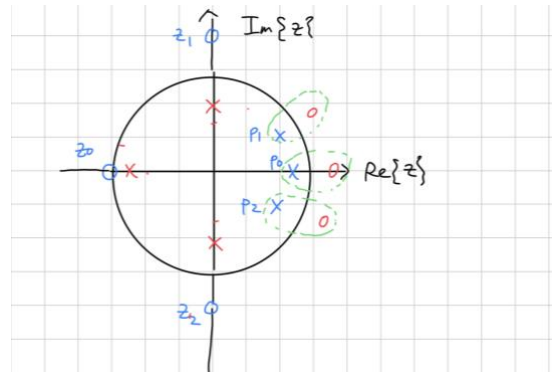
(b) Given the poles and zeros of $H(z)$ below, give the values for the poles and zeros of $G(z)$ and draw them on the pole-zero diagram on the right for an LTI equalizer $G(z)$ that would make the cascade of the LTI system model $H(z)$ and the LTI equalizer $G(z)$ be allpass. 15 points.



Solution I



Solution II



Problem 1.4. Potpourri. 24 points.

(a) An integrator is a common building block in systems. The discrete-time version of the integrator is the running summation, which is defined for input $x[n]$ and output $y[n]$ for $n \geq 0$ as follows:

$$y[n] = \sum_{m=0}^n x[m]$$

The summation requires unbounded memory as $n \rightarrow \infty$. A more efficient implementation is

$$y[n] = y[n - 1] + x[n] \text{ for } n \geq 0$$

I. Give the initial condition(s) for the more efficient implementation to be linear and time-invariant (LTI). 4 points.

Rather than using the summation equation,

$$y[n] = \sum_{m=0}^n x[m]$$

It is simpler to solve for the initial conditions using the equivalent recursive difference equation instead, $y[n] = y[n - 1] + x[n]$ for $n \geq 0$

Consider $n = 0$: $y[0] = \underbrace{y[-1]} + x[0]$

Initial Condition

In order for LTI to hold, our initial conditions must equal zero:

$$y[-1] = 0$$

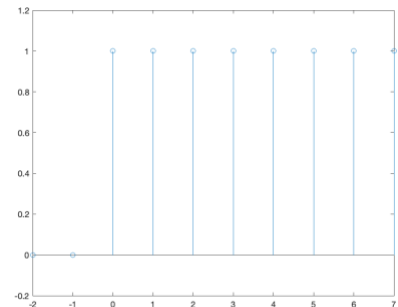
II. Give a formula for and plot the impulse response $h[n]$ of an LTI running summation. 4 points

This time, let's use the summation form,

$$y[n] = \sum_{m=0}^n x[m]$$

Let $x[m] = \delta[m]$ and then $h[n] = y[n]$:

$$h[n] = \sum_{m=0}^n \delta[m] = \begin{cases} 1, & \text{if } n \geq 0 \\ 0, & \text{else} \end{cases}$$



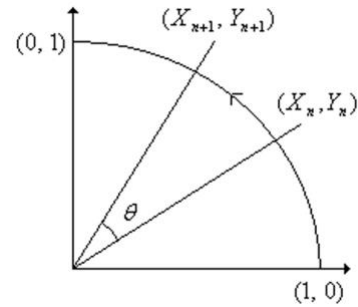
III. The running summation only gives an unbounded output for a bounded input when the input has a non-zero constant component. What filter would you apply before the running summation to prevent the running summation being bounded-input bounded-output unstable? 4 points

Use a DC notch filter to remove the non-zero constant at 0 radians/sample

- (b) There are several algorithms [1] to generate a cosine signal $x[n] = \cos(\omega_0 n)$ and a sine signal $y[n] = \sin(\omega_0 n)$ at the same time using rotation. For each signal, the argument is $0, \omega_0, 2\omega_0, \dots$, for $n = 0, 1, 2, \dots$

Using the visual representation on the right, the next cosine value X_{n+1} and next sine value Y_{n+1} are computed from the current cosine value X_n and current sine value Y_n using a rotation operation:

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix}$$



Matrix Rotation [1]

Using rotation keeps the values of the cosine and sine for the same angle on the unit circle.

With $\theta = \omega_0$, we use the rotation approach starting at $n = 0$ and pre-compute $\cos(\theta)$ and $\sin(\theta)$.

Note: This is called the Coordinate Rotation Digital Computer (CORDIC) Algorithm. “The algorithm was used in the navigational system of the Apollo program's Lunar Roving Vehicle to compute bearing and range, or distance from the Lunar module.”

- I. What are the values of X_0 and Y_0 ? *4 points.*

Consider at $n = 0$: $\cos(\omega_0 n) = \cos(0) = 1$

$\sin(\omega_0 n) = \sin(0) = 0$

Thus, $X_0 = 1, Y_0 = 0$

- II. How many multiplications per output sample are needed to compute the cosine and sine signals? *4 points.*

For the matrix operation, we have a 2x2 matrix multiplied by a 2x1 vector.

To compute X_{n+1} (cosine component), we need 2 multiplications

$$X_{n+1} = \cos(\theta) X_n + (-\sin(\theta)) Y_n$$

To compute Y_{n+1} (sine component), we need 2 multiplications

$$Y_{n+1} = \sin(\theta) X_n + \cos(\theta) Y_n$$

So, we require a total of 4 multiplications (and 4 additions) per output sample.

- III. Compare your answer in part II to using second-order difference equations to compute cosine and sine signals separately. *4 points.*

Consider the difference equation for the cosine signal:

$$y[n] = (2\cos(\omega_0)) y[n - 1] - y[n - 2] + x[n] - (\cos(\omega_0)) x[n - 1]$$

Consider the difference equation for the sine signal:

$$y[n] = (2\cos(\omega_0)) y[n - 1] - y[n - 2] + (\sin(\omega_0)) x[n - 1]$$

We see that with this format, both cosine and sine require around 2 multiplications and 4 additions, for a total of 4 multiplications and 8 additions.

In comparison, the CORDIC algorithm only requires 4 multiplications and 4 additions, and so gives us computational savings.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Version 4.0*

Date: Oct. 18, 2023

Course: EE 445S Evans

Name: _____ **Lupin** _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

	<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
<i>Raoul</i>	1	25		FIR Filter Analysis
<i>Youssef Guedira</i>	2	24		Decreasing the Sampling Rate
<i>Assane Diop</i>	3	27		System Identification
<i>Benjamin Ferel</i>	4	24		Mystery Systems
	<i>Total</i>	104		

Problem 1.1 FIR Filter Analysis. 25 points.

Consider a causal linear time-invariant (LTI) discrete-time finite impulse response (FIR) filter with input $x[n]$ and output $y[n]$ observed for $n \geq 0$. The transfer function in the z -domain is

$$H(z) = 1 - z^{-1}$$

- (a) Give the equation for output $y[n]$ in terms of the input $x[n]$ in the discrete-time domain. 6 points.

$$H(z) = \frac{Y(z)}{X(z)} = 1 - z^{-1} \text{ which means } Y(z) = (1 - z^{-1}) X(z) = X(z) - z^{-1} X(z)$$

Lecture Slide 5-12

Take the inverse z -transform of both sides to obtain $y[n] = x[n] - x[n - 1]$ for $n \geq 0$.

- (b) What are the initial condition(s) and their value(s)? Why? 6 points.

Consider $n = 0$: $y[0] = x[0] - x[-1]$ which has one initial condition $x[-1]$.

Consider $n = 1$: $y[1] = x[1] - x[0]$ which does not have any initial conditions.

Our initial condition must equal zero as necessary conditions for LTI to hold: $x[-1] = 0$

- (c) Derive a formula for the discrete-time frequency response of the filter. 3 points.

We first determine the region of convergence for $H(z) = 1 - z^{-1}$ which is $z \neq 0$ because we need to avoid division by zero due to z^{-1} term. Another way to see this is

$$H(z) = 1 - z^{-1} = \frac{z - 1}{z}$$

Because the unit circle is in the region of convergence, we can substitute $z = e^{j\omega}$ into $H(z)$

$$H_{freq}(\omega) = H(z)|_{z=e^{j\omega}} = 1 - e^{-j\omega}$$

- (d) Consider implementing a fourth-order version of this filter which would have the transfer function

$$H_4(z) = (1 - z^{-1})^4$$

Assume the input $x[n]$ and output $y[n]$ values are stored in 32-bit IEEE floating-point format.

In terms of run-time implementation complexity, which of the following designs would you advocate using? Please fill out the table to justify your answer. 10 points.

Filter Structure	# multiplications	# additions	# words of memory
Cascade of four first-order sections	0	4	12 or 9 (see below)
Cascade of two second-order sections	2	4	10 or 9 (see below)
Single fourth-order section	3	4	9

For a cascade of four first-order sections, each section would have input-output relationship $y[n] = x[n] - x[n - 1]$ which requires 0 mults, 1 add, and 3 words of memory. If reuse the same memory location for first section output and second section input, etc., 9 total words.

For a cascade of two second-order sections, each second-order section would have an input-output relationship $y[n] = x[n] - 2x[n - 1] + x[n - 2]$ which requires 1 mults, 2 adds, and 5 words of memory to store the coefficient -2 as well as $x[n], x[n - 1], x[n - 2]$ and $y[n]$.

We can obtain the second-order coefficients by conv([1 -1], [1 -1]) or expand $(1 - z^{-1})^2$. If reuse same memory location for first section output and second section input, 9 total words

A single fourth-order section would have the input-output relationship

$$y[n] = x[n] - 4x[n - 1] + 6x[n - 2] - 4x[n - 3] + x[n - 4]$$

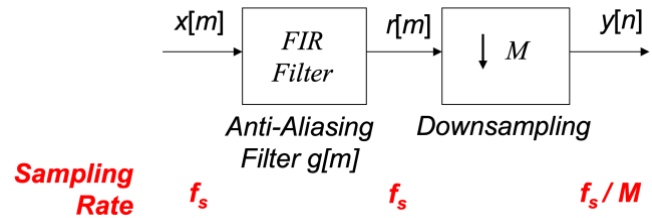
which requires 3 mults, 4 adds, and 9 words of memory to store the coefficients -4, -6, -4 and input values $x[n], x[n - 1], x[n - 2], x[n - 3]$, and $x[n - 4]$ and output $y[n]$.

Problem 1.2 Decreasing the Sampling Rate. 24 points.

Downsampling by M can be used to decrease the sampling rate of the input signal by a factor of M .

A lowpass finite impulse response (FIR) filter can be placed before the downsampler to reduce the aliasing caused by resampling at the lower sampling rate.

On the right, discrete-time index m is associated with sampling rate f_s and discrete-time index n is associated with sampling rate $\frac{f_s}{M}$.



- (a) What is the maximum continuous-time frequency f_{max} that is present in $y[n]$? What discrete-time frequency does f_{max} correspond to? 6 points. **The sampling rate for $y[n]$ is f_s/M .**

Sampling Theorem says to choose the sampling rate $\frac{f_s}{M} > 2 f_{max}$ so $f_{max} \rightarrow \frac{1}{2} \frac{f_s}{M}$.

Let $y(t) = \cos(2\pi f_{max} t)$ for $-\infty < t < \infty$ and sample it at sampling rate $\frac{f_s}{M}$:

$$y[n] = y(t) \Big|_{t=\frac{n}{\frac{f_s}{M}}} = \cos\left(2\pi f_{max} \left(\frac{n}{\frac{f_s}{M}}\right)\right) = \cos\left(2\pi \frac{1}{2} \frac{f_s}{M} n\right) = \cos(\pi n) \text{ for } -\infty < n < \infty$$

The continuous-time frequency f_{max} corresponds to discrete-time frequency π rad/sample.

- (b) What discrete-time frequency in $r[m]$ corresponds to the maximum continuous-time frequency f_{max} that is present in $y[n]$? 6 points.

Since the sampling rate for $r[n]$ is f_s , continuous-time frequency f_{max} in $r[n]$ corresponds to

$$2\pi \frac{1}{2} \frac{f_s}{M} = \frac{\pi}{M} \text{ rad/sample}$$

- (c) Any discrete-time frequencies present in $r[m]$ higher than your answer in part (b) but less than π rad/sample correspond to frequencies that will alias due to downsampling. Give the discrete-time passband frequency ω_{pass} and stopband frequency ω_{stop} you would use for the lowpass filter design.

6 points. **With 10% rolloff from passband frequency ω_{pass} and stopband frequency ω_{stop} , we could use $\omega_{pass} = 0.95 \frac{\pi}{M}$ and $\omega_{stop} = 1.05 \frac{\pi}{M}$.**

- (d) Here are two possible lowpass FIR filter with linear phase to meet the specifications of a discrete-time passband frequency ω_{pass} and stopband frequency ω_{stop} .

- i. Averaging filter. How many FIR filter coefficients would you use? Why? What are the filter coefficient values? 3 points. **Averaging FIR filter with N coefficients has null bandwidth of $\frac{2\pi}{N}$ rad/sample. With $\frac{2\pi}{N} = \frac{\pi}{M}$, we use $N = 2M$ coefficients each with a value of $\frac{1}{N}$.**
- ii. Impulse response is a truncated sinc pulse. How would you choose the number of samples in the sinc pulse? 3 points. **An infinite sinc pulse would be an ideal lowpass filter. For a discrete-time cutoff frequency of $\frac{\pi}{M}$, the sinc pulse would have zero crossings every M samples except the origin. To keep even symmetry, we would keep $2Mk + 1$ samples where k is a positive integer. For causality, we'd delay the sinc by Mk samples. Larger k means better lowpass response but higher run-time complexity and group delay.**

Lecture 1

Labs 2 & 3

HW 2.3 & 3.3

Designing
Averaging Filters

Problem 1.3 System Identification. 27 points.

You are given several causal discrete-time linear time-invariant (LTI) systems with unknown impulse responses but you know the response of each system when the input is a unit step function $u[n]$ where

$$u[n] = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

The z-transform of $u[n]$ is $\frac{1}{1-z^{-1}}$ for $|z| > 1$.

(a) When the input is $x[n] = u[n]$, the output $y[n] = \delta[n]$ where $\delta[n]$ is the discrete-time impulse.

$$\delta[n] = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{otherwise} \end{cases}$$

HW 1.1(b) and 2.1(b)

Find the impulse response $h[n]$. 9 points.

Time-domain approach. $y[n] = \delta[n] = u[n] - u[n-1]$. Since $x[n] = u[n]$, we can write $y[n] = x[n] - x[n-1]$ and hence $h[n] = \delta[n] - \delta[n-1]$.

Deconvolution approach. Assume the LTI system is an FIR filter observed for $n \geq 0$:

$$y[0] = h[0] x[0] + h[1] x[-1] + h[2] x[-2] + \dots + h[N-1] x[-(N-1)]$$

All initial conditions are zero as a necessary condition for LTI properties to hold:

$$y[0] = h[0] x[0] \text{ so } 1 = h[0] \text{ because } y[0] = 1 \text{ and } x[0] = 1 \text{ so } h[0] = 1$$

$$y[1] = h[0] x[1] + h[1] x[0] \text{ which is } 0 = h[0] + h[1] \text{ so } h[1] = -1$$

$$y[2] = h[0] x[2] + h[1] x[1] + h[2] x[0] \text{ which is } 0 = h[0] + h[1] + h[2] \text{ so } h[2] = 0$$

We can check to see that $h[n] = \delta[n] - \delta[n-1]$ convolved with $u[n]$ is $\delta[n]$.

Z-domain approach. An equalizer problem in disguise. We are trying to find an LTI system $h[n]$ so that $h[n] * u[n] = \delta[n]$. In the z-domain, $H(z) U(z) = 1$ which means that

$$H(z) = \frac{1}{U(z)} = \frac{1}{\frac{1}{1-z^{-1}}} = 1 - z^{-1} \text{ for } z \neq 0. \text{ Inverse z-transform is } h[n] = \delta[n] - \delta[n-1].$$

(b) When the input is $x[n] = u[n]$, the output is $y[n] = n u[n]$. Find the impulse response $h[n]$. 9 points. We rewrite $y[n] = n u[n] = (n-1) u[n-1]$.

HW 1.1(c) and 2.1(c)

Deconvolution approach. Assume the LTI system is an FIR filter observed for $n \geq 0$:

$$y[0] = h[0] x[0] + h[1] x[-1] + h[2] x[-2] + \dots + h[N-1] x[-(N-1)]$$

All initial conditions are zero as a necessary condition for LTI properties to hold:

$$y[0] = h[0] x[0] \text{ so } 0 = h[0] \text{ because } y[0] = 0 \text{ and } x[0] = 1 \text{ so } h[0] = 0$$

$$y[1] = h[0] x[1] + h[1] x[0] \text{ which is } 1 = h[0] + h[1] \text{ so } h[1] = 1$$

$$y[2] = h[0] x[2] + h[1] x[1] + h[2] x[0] \text{ which is } 2 = h[0] + h[1] + h[2] \text{ so } h[2] = 1$$

$$y[3] = h[0] x[3] + h[1] x[2] + h[2] x[1] + h[3] x[0] \text{ which is}$$

$$3 = h[0] + h[1] + h[2] + h[3] \text{ so } h[3] = 1$$

We check to see that $h[n] = u[n-1]$ convolved with $u[n]$ is $n u[n]$. So, $h[n] = u[n-1]$.

Z-domain approach. Rewrite $y[n] = n u[n] = (n-1) u[n-1]$. We're finding LTI system $h[n]$ so that $h[n] * u[n] = (n-1) u[n-1]$. In z-domain, $H(z) U(z) = \frac{z^{-1}}{(1-z^{-1})^2}$ for $|z| > 1$:

$$H(z) = \frac{z^{-1}}{\frac{1}{(1-z^{-1})^2}} = \frac{z^{-1}}{1-z^{-1}} \text{ for } |z| > 1$$

Taking the inverse z-transform gives $h[n] = u[n-1]$.

(c) When the input is $x[n] = u[n]$, the output is $y[n]$ is a rectangular pulse of L samples in duration:

$$y[n] = \begin{cases} 1 & \text{for } 0 \leq n \leq L - 1 \\ 0 & \text{otherwise} \end{cases}$$

Find the impulse response $h[n]$. 9 points.

Time-domain approach. $y[n] = \delta[n] = u[n] - u[n - L]$. Since $x[n] = u[n]$, we can write $y[n] = x[n] - x[n - L]$ and hence $h[n] = \delta[n] - \delta[n - L]$.

Deconvolution approach. Assume the LTI system is an FIR filter observed for $n \geq 0$:

$$y[0] = h[0] x[0] + h[1] x[0 - 1] + h[2] x[0 - 2] + \dots + h[N - 1] x[0 - (N - 1)]$$

All initial conditions are zero as a necessary condition for LTI properties to hold:

$$y[0] = h[0] x[0] \text{ so } 1 = h[0] \text{ because } y[0] = 1 \text{ and } x[0] = 1 \text{ so } h[0] = 1$$

$$y[1] = h[0] x[1] + h[1] x[0] \text{ which is } 1 = h[0] + h[1] \text{ so } h[1] = 0$$

$$y[2] = h[0] x[2] + h[1] x[1] + h[2] x[0] \text{ which is } 1 = h[0] + h[1] + h[2] \text{ so } h[2] = 0$$

If $h[n] = \delta[n]$, then $h[n] * u[n] \neq y[n]$. So, we keep computing $h[n]$ values.

$$y[L] = h[0] x[L] + h[1] x[L - 1] + \dots + h[L] x[0] \text{ which is}$$

$$0 = h[0] + h[1] + \dots + h[L] \text{ so } h[L] = 0$$

We check to see that $h[n] = \delta[n] - \delta[n - L]$ convolved with $u[n]$ is $y[n]$.

Z-domain approach. We're finding LTI system $h[n]$ so that $h[n] * u[n]$ is rectangular pulse of L samples in duration. In the z -domain, $H(z) U(z) = 1 + z^{-1} + \dots + z^{-(L-1)}$ which means

$$H(z) = \frac{1 + z^{-1} + \dots + z^{-(L-1)}}{1 - z^{-1}} = (1 + z^{-1} + \dots + z^{-(L-1)})(1 - z^{-1}) = 1 - z^{-L} \text{ for } z \neq 0$$

Taking the inverse z -transform gives $h[n] = \delta[n] - \delta[n - L]$. In Matlab, polynomial multiplication is computed using the conv command, e.g. conv([1 1 1 1 1 1], [1 -1]).

```
%% Deconvolution by Prof. Brian L. Evans.
%% Keep in mind the first element in a
%% MATLAB vector has index 1 and not 0.

%% USAGE
%% FIR filters convolve the input signal
%% and the FIR filter impulse response
%% (which is equal to the filter coeffs).
%% When input signal has finite length,
%% the output is finite length:
%%
%% LengthOfy = LengthOfx + NumCoeffs - 1
%%
%% Given finite-length signals x and y,
%% we can determine how many filter
%% coefficients b there are.
%%
%% If the input signal is infinite in
%% length, then the output could be
%% either infinite or finite in length.

%% Define input and output signals. Give
%% an equal number of x and y values if
%% x is to be considered infinite length.
```

```
x = [ 1 1 1 1 1 1 ]; % Midterm 1.3(a)
y = [ 1 0 0 0 0 0 ];

% Determine Nmax based on input signal
% Finite-length length(y) - length(x) + 1
% Infinite-length length(x)
if ( length(x) == length(y) )
    Nmax = length(x);
else
    Nmax = length(y) - length(x) + 1;
end

b = zeros(1, Nmax);
b(1) = y(1) / x(1);
for k = 2:Nmax
    numer = y(k);
    n = k;
    for m = 1:(k-1)
        if (n >= 1)
            numer = numer - b(m) * x(n);
        end
        n = n - 1;
    end
    b(k) = numer / x(1);
end
```

Problem 1.4. Mystery Systems. 24 points.

You're trying to identify unknown discrete-time systems.

You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 8000 Hz over 0 to 5s

$$x(t) = \cos(2\pi f_1 t + 2\pi\mu t^2)$$

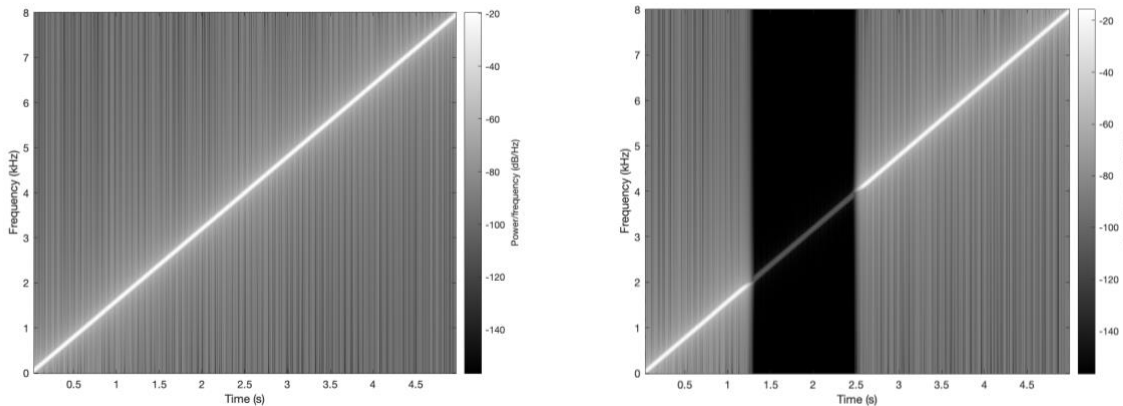
where $f_1 = 0$ Hz, $f_2 = 8000$ Hz, and $\mu = \frac{f_2 - f_1}{2 t_{\max}} = \frac{8000 \text{ Hz}}{10 \text{ s}} = 800 \text{ Hz}^2$. Sampling rate f_s is 16000 Hz.

In each part below, identify the unknown system as one of the following **with justification**:

1. filter – give selectivity (lowpass, highpass, bandpass, bandstop) and passband/stopband frequencies
2. upsampler – give upsampling factor
3. downsampler – give downsampling factor
4. pointwise nonlinearity – give the integer exponent k to produce the output $y[n] = x^k[n]$

In the grayscale color map, white has highest magnitude value.

(a) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 pts.



Lectures
1 3 5 & 6

Lab #3

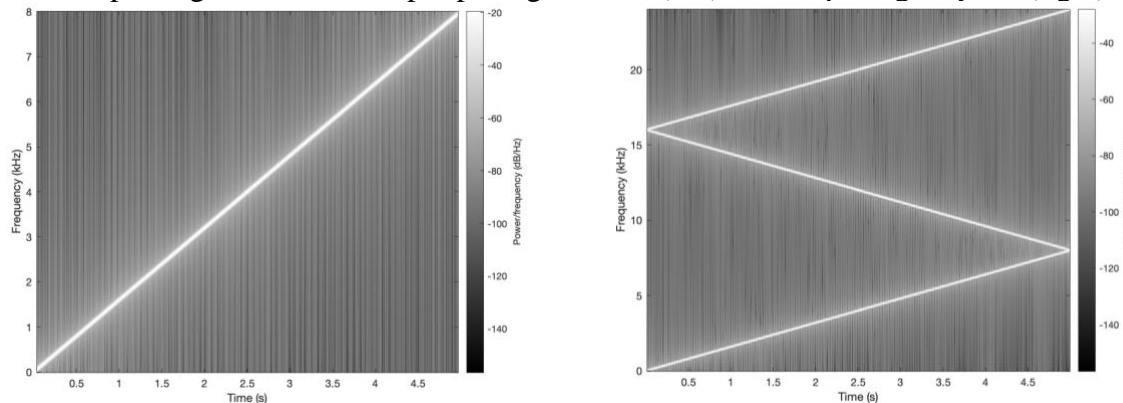
HW 1.1 1.3
2.1 2.2 & 2.3

JSK Ch. 7

Midterm 1.4
Sp22

In the output spectrogram, principal frequencies between 2000 and 4000 Hz are severely attenuated and other principal frequencies are passed. No new frequencies are created, so it is likely an LTI filter. This is a bandstop filter similar to HW 2.3. See next page for the Matlab code.

(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lecture 4

HW 0.3 & 2.2

Midterm
Problems

1.2 Sp23

1.2 F18

1.2(d) Sp18

Output spectrogram has about 3x the frequency values on vertical axis over the same time interval vs. input spectrogram. Upsampling by 3 gives input chirp signal plus replicas every 16 kHz. Two replicas are visible -- a replica from 16 to 24 kHz and a replica from -32 to -40 kHz which aliases to 16 to 8 kHz due to a sampling rate of 48 kHz. Upsampling by 3 per HW 2.2(e). See next page.


```

%% Midterm Problem 2.4(a) bandstop filter

fs = 16000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Design lowpass filter
fnyquist = fs/2;
fstop1 = 1800;
fpass1 = 2000;
fpass2 = 4000;
fstop2 = 4200;
ctfrequencies = [0 fstop1 fpass1 fpass2 fstop2 fnyquist];
idealAmplitudes = [1 1 0 0 1 1];
pmfrequencies = ctfrequencies / fnyquist;
filterOrder = 400;
h = firpm( filterOrder, pmfrequencies, idealAmplitudes );
h = h / sum(h.^2);

y = conv(x, h);

%% Spectrogram parameters
blockSize = 1024;
overlap = 1023;
figure;

%% Plot spectrogram of input signal
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;

%% Plot spectrogram of output signal
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;

```

```
%% Midterm Problem 2.4(b) upsampling by 3

fs = 16000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Upsample by 3
upsampleFactor = 3;
y = zeros(1, upsampleFactor*length(x));
y(1:3:end) = x;

%% Spectrogram parameters
blockSize = 1024;
overlap = 1023;
figure;

%% Plot spectrogram of input signal
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;

%% Plot spectrogram of output signal
figure;
spectrogram(y, blockSize, overlap, blockSize, upsampleFactor*fs, 'yaxis');
colormap gray;
```

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Version 4.0*

Date: March 6, 2024

Course: EE 445S Evans

Name: _____ **Dune** _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **No AI tools allowed.** As mentioned on the course syllabus, you may not use GPT or other AI tools during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

	<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
<i>Leto Atreides</i>	1	25		FIR Filter Analysis
<i>Chani</i>	2	24		Predistortion
<i>Paul Atreides</i>	3	27		Detection of Radar
<i>Lady Jessica</i>	4	24		Mystery Systems
	<i>Total</i>	100		

Problem 1.1 FIR Filter Analysis. 25 points.

Consider a causal linear time-invariant (LTI) discrete-time finite impulse response (FIR) filter with input $x[n]$ and output $y[n]$ observed for $n \geq 0$. The transfer function in the z -domain is

$$H(z) = a + b z^{-1} + c z^{-2} \quad \text{for } z \neq 0$$

where a , b , and c are real-valued constants.

- (a) Give the equation for output $y[n]$ in terms of the input $x[n]$ in the discrete-time domain for $n \geq 0$.
6 points.

$$H(z) = \frac{Y(z)}{X(z)} = a + b z^{-1} + c z^{-2} \text{ which means}$$

Lecture Slide 5-12

$$Y(z) = (a + b z^{-1} + c z^{-2}) X(z) = a X(z) + b z^{-1} X(z) + c z^{-2} X(z)$$

Taking inverse z -transforms of both sides, $y[n] = a x[n] + b x[n - 1] + c x[n - 2]$ for $n \geq 0$

- (b) What are the initial condition(s) and their value(s)? Why? 3 points.

For $n = 0$: $y[0] = a x[0] + b x[-1] + c x[-2]$ which has initial conditions $x[-1]$ & $x[-2]$

For $n = 1$: $y[1] = a x[1] + b x[0] + c x[-1]$ which has initial condition $x[-1]$

For $n = 2$: $y[2] = a x[2] + b x[1] + c x[0]$ which does not have any initial conditions

A necessary condition for LTI to hold is for the system to initially be at rest, i.e., the initial conditions must be zero: $x[-1] = 0$ and $x[-2] = 0$.

- (c) In managing the memory for storing the previous input values, would you advocate for a linear buffer or a circular buffer? Why? 3 points

For this system, $y[n] = a x[n] + b x[n - 1] + c x[n - 2]$.

After calculating $y[n]$ for the current value of n ,

- **A linear buffer in this case would need two writes and two reads to update the memory to previous input values: $x[n - 2] = x[n - 1]$ and $x[n - 1] = x[n]$.**
- **A circular buffer maintains the index to the oldest data value, needs to read the index to the oldest data value, update the oldest data value with $x[n]$, and increment the index to the oldest data value and apply a modulo operation, which requires two writes, one read, one addition, and one modulo operation, regardless of the length of the circular buffer.**

Using the above, one could argue either way. When compared to the circular buffer, the linear buffer requires an extra read but does not require an addition or modular operation.

- (d) Derive a formula for the discrete-time frequency response of the filter. 3 points.

Because the unit circle is in the region of convergence $z \neq 0$, we can substitute $z = e^{j\omega}$ into system transfer function in the z -domain $H(z)$:

$$H_{freq}(\omega) = H(z)_{z=e^{j\omega}} = a + b e^{-j\omega} + c e^{-2j\omega}$$

- (e) Give all possible conditions on the constants a , b , and c so that the FIR filter has constant group delay. 10 points.

With $Group\ Delay(\omega) = -\frac{d}{d\omega} \angle H_{freq}(\omega)$, a constant group delay means linear phase.

In order for a filter to have linear phase, the impulse response has to be either

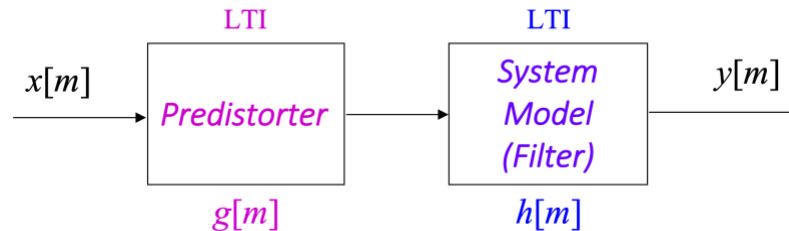
- **even symmetric about the midpoint: $a = c$**
- **odd symmetric about the midpoint: $a = -c$ and $b = 0$**

Problem 1.2 Predistortion. 24 points.

Predistortion is a technique used to compensate the distortion in another system.

An example is applying predistortion to an audio signal before being played by an audio speaker.

The block diagram below illustrates the use of predistortion when the predistorter is a linear time invariant (LTI) system and the other system is modeled as LTI:



In this problem, $g[m]$ is the impulse response of the discrete-time LTI predistorter, and $h[m]$ is the impulse response of a discrete-time LTI model of an audio speaker.

Each LTI system is a second-order infinite impulse response (IIR) filter with two zeros and two poles.

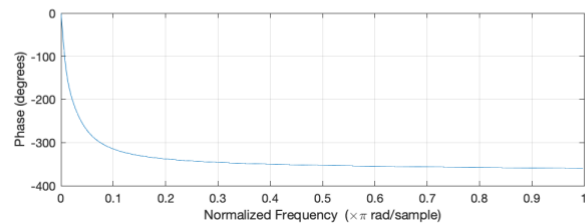
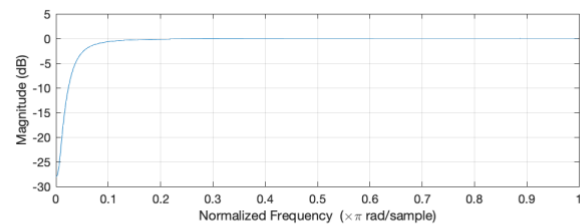
The sampling rate f_s is 44100 Hz.

(a) The LTI model of the audio speaker $h[m]$ has the frequency response on the right. 12 points.

- i. The two poles are located at $p_0 = 0.9 e^{j\omega_p}$ and $p_1 = 0.9 e^{-j\omega_p}$. Please give the value for discrete-time frequency ω_p that corresponds to continuous-time frequency 441 Hz.

$$\omega_p = 2\pi \frac{f_p}{f_s} = 2\pi \frac{441 \text{ Hz}}{44100 \text{ Hz}} = 2\pi \frac{1}{100}$$

- ii. The two zeros are located at $z_0 = 1.02 e^{j\omega_z}$ and $z_1 = 1.02 e^{-j\omega_z}$. Please give the value for discrete-time frequency ω_z that corresponds to continuous-time frequency 100 Hz.



$$\omega_z = 2\pi \frac{f_z}{f_s} = 2\pi \frac{100 \text{ Hz}}{44100 \text{ Hz}} = 2\pi \frac{1}{441}$$

(b) Give the two poles and two zeros for a bounded-input bounded-output (BIBO) stable LTI predistorter $g[m]$ to compensate the distortion $h[m]$. 12 points.

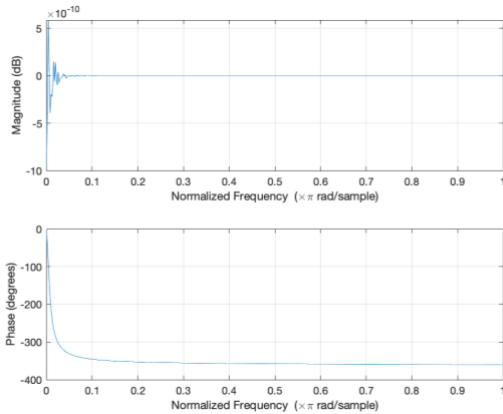
We use predistorter $g[m]$ to compensate distortion in $h[m]$: $G(z)H(z) = 1$ or $G(z) = \frac{1}{H(z)}$.

A predistorter plays the same role as an equalizer, but appears before the system.

With $G(z) = \frac{1}{H(z)} = \frac{(1-p_0 z^{-1})(1-p_1 z^{-1})}{(1-z_0 z^{-1})(1-z_1 z^{-1})}$, the poles for $G(z)$ are outside the unit circle, and hence, the filter would be BIBO unstable.

We can reflect the poles of $G(z)$ inside the unit circle to create an all-pass configuration with the zeros of $H(z)$: $p_2 = \frac{1}{1.02} e^{j\omega_z}$ and $p_3 = \frac{1}{1.02} e^{-j\omega_z}$.

Magnitude and phase responses of the cascade (allpass response):



```
clear all;
close all;

%% Audio sampling rate
fs = 44100;

%% Distortion h[m]
fp = 441;
fz = 100;
wp = 2*pi*fp/fs;
wz = 2*pi*fz/fs;
rz = 1.02;
z0 = rz*exp(j*wz);
z1 = rz*exp(-j*wz);;
rp = 0.9;
p0 = rp*exp(j*wp);
p1 = rp*exp(-j*wp);
numerh = [1 -(z0+z1) z0*z1];
denomh = [1 -(p0+p1) p0*p1];

%% Predistortion g[m]
z2 = p0;
z3 = p1;
p2 = (1/rz)*exp(j*wz);
p3 = (1/rz)*exp(-j*wz);
numerg = [1 -(z2+z3) z2*z3];
denomg = [1 -(p2+p3) p2*p3];

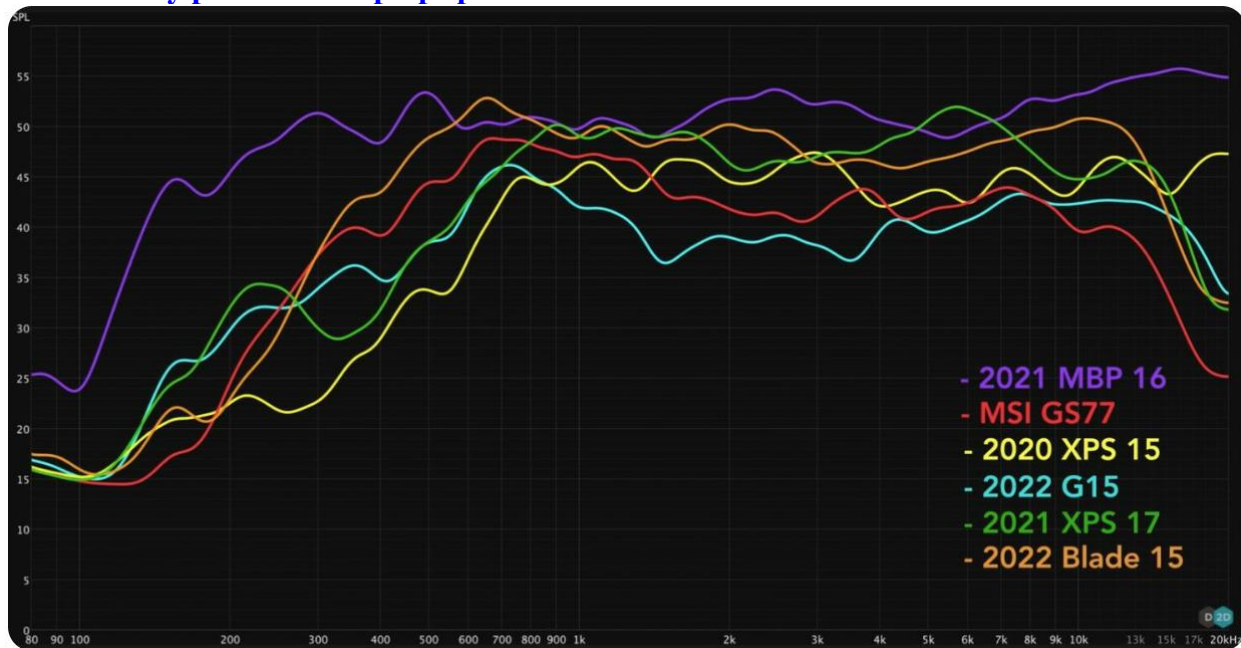
%% Cascade of G(z) and H(z)
%% conv is polynomial mult
numer = conv(numerg, numerh);
denom = conv(denomg, denomh);

%% Normalize response at w = pi
%% to 1 in linear units by
setting
%% H(z)= 1 and solving for C
zval = -1;
zvec = zval .^ [0 -1 -2 -3 -4];
C = (denom*zvec)/(numer*zvec);

figure;
freqz(C*numer, denom);
```

The distortion $h[m]$ in problem 1.2 is based on measurements of frequency responses in audio speakers. $h[m]$ is a biquad model of the response for the 2021 MBP 16 audio speaker below.

The plot below shows the frequency responses from 80 Hz to 10 kHz on a log-log scale measured for audio speakers in several laptop models, e.g. MBP means Mac Book Pro. You'll notice that responses are very poor in the laptop speakers below 100 Hz.



“The comparison was conducted by the [YouTube channel Dave2D](#).... While it is difficult to analyze sound quality, Dave used a calibrated measurement microphone. The calibrated mic allowed him to measure and quantify the quality of sound coming out of the MacBook as well as Windows laptops.” [1]

[1] [Ali Salman](#), “YouTuber Shows How MacBook Pro Speakers are Superior Than the Most Expensive and Premium Windows Laptops”, April 16, 2022. <https://wccftech.com/youtuber-shows-how-macbook-pro-speakers-are-superior-than-the-most-expensive-and-premium-windows-laptops/>

Many possible answers to this problem.

Lectures 1 4 5 & 6		HW 0.1, 0.2, 1.1, 1.3, 2.1, 2.2, 3.1
Lab #2	Lab #3	Midterm 1.3 Sp07, 1.3 Sp23

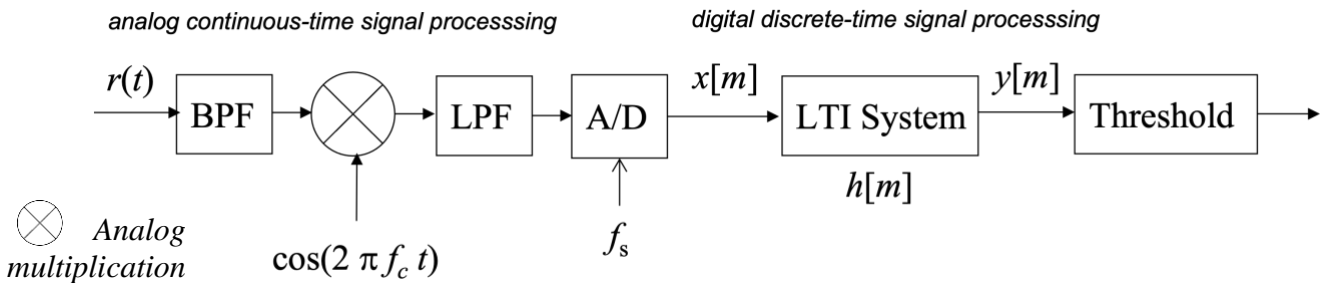
Problem 1.3 Detection of Radar. 27 points.

In certain wireless communication bands, radar systems might also be operating from time to time.

The Citizens Broadband Radio Service (CBRS) band, which is between 3550 MHz and 3700 MHz, is used for cellular communications and US military radar, including shipborne radar off the US coasts.

If a communication system in the CBRS band in the US detects radar operating in the transmission band, then the communication system must stop transmitting within one second.

In this problem, you'll design a system to detect the presence of an SPN-43 radar operating at a center frequency of 3570 MHz with a bandwidth of 1.6 MHz based on the block diagram below:

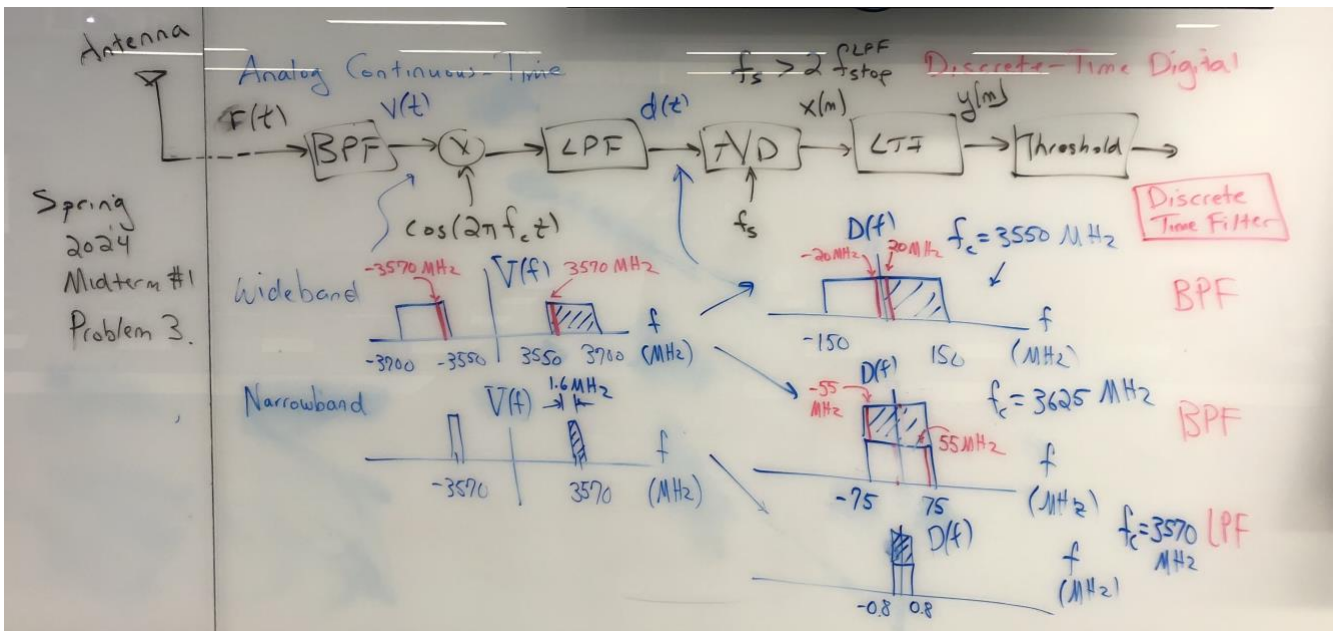


A/D: Analog-to-digital converter operating at sampling rate f_s . BPF: Bandpass filter. LPF: Lowpass filter. LTI System: linear time-invariant system. $r(t)$: received radio frequency signal.

The threshold block will compare its input $y[m]$ to a constant and output true or false to indicate whether or not there is an SPN-43 radar operating at a center frequency of 3570 MHz.

In your answers below, please explain why you chose the numerical values that you did.

Prologue. Here's a visual analysis of the receiver that goes along with the solution that follows:



- (a) Give passband and stopband frequencies for the bandpass filter (BPF). 6 points.

HW 3.1

The role of bandpass filter is to attenuate out-of-band noise and interference like HW 3.1.

Wideband Solution: Pass the entire CBRS band which has bandwidth of 150 MHz. Allow 10% rolloff from stopband to passband and vice-versa. $f_{stop1} = 3535$ MHz, $f_{pass1} = 3550$ MHz, $f_{pass2} = 3700$ MHz, $f_{stop2} = 3715$ MHz.

Narrowband Solution: Only pass the frequency band for an SPN-43 radar operating at a center frequency of 3570 MHz. Bandwidth 1.6 MHz. $f_{stop1} = 3569.2$ MHz, $f_{pass1} = 3569.4$ MHz, $f_{pass2} = 3570.8$ MHz, $f_{stop2} = 3571$ MHz.

Other Solutions: continuum of solutions between the narrowband and wideband solutions.

- (b) Give a value for f_c . 3 points.

The role of the modulator and lowpass filter is to demodulate the RF signal to a lower center frequency and possibly all the way down to baseband like HW 0.2 and 1.3.

HW 0.2 & 1.3

Wideband Solution #1: $f_c = \frac{3700 \text{ MHz} + 3550 \text{ MHz}}{2} = \frac{7250 \text{ MHz}}{2} = 3625$ MHz

Wideband Solution #2: $f_c = 3550$ MHz

Narrowband Solution: $f_c = 3570$ MHz

- (c) Give the passband and stopband frequencies for the lowpass filter (LPF). 3 points

HW 0.2 & 1.3

Wideband Solution #1: $f_{pass} = 75$ MHz and $f_{stop} = 82.5$ MHz. Allows 10% rolloff.

Wideband Solution #2: $f_{pass} = 150$ MHz and $f_{stop} = 165$ MHz. Allows 10% rolloff.

Narrowband Solution: $f_{pass} = 0.8$ MHz and $f_{stop} = 0.88$ MHz. Allows 10% rolloff.

- (d) Give a sampling rate f_s for the analog-to-digital (A/D) converter. 3 points.

The sampling rate f_s has to be chosen to be greater than twice the stopband frequency in (c).

Wideband Solution #1: $f_s = 200$ MHz

Wideband Solution #2: $f_s = 400$ MHz

Narrowband Solution: $f_s = 2$ MHz

- (e) Design a second-order LTI system with infinite impulse response $h[m]$ by giving the formulas for the two poles and two zeros to enable detection of an SPN-43 radar system if one is operating. 12 points

We have to figure out what discrete-time frequencies ω_{3570} and ω_{-3570} in $x[m]$ correspond to the continuous-time frequencies of 3570 MHz and -3570 MHz in $r(t)$ and design the appropriate lowpass or bandpass filter.

HW 1.1(c) & 2.1(c)

Wideband Solution #1: Bandpass filter. Poles at $0.95 e^{\pm j \omega_{3570}}$ and zeros at ± 1 .

$$\omega_{3570} = 2\pi \frac{3570 \text{ MHz} - f_c}{f_s} = 2\pi \frac{3570 \text{ MHz} - 3625 \text{ MHz}}{200 \text{ MHz}} = -2\pi \frac{55 \text{ MHz}}{200 \text{ MHz}}$$

$$\omega_{-3570} = 2\pi \frac{-3570 \text{ MHz} + f_c}{f_s} = 2\pi \frac{-3570 \text{ MHz} + 3625 \text{ MHz}}{200 \text{ MHz}} = 2\pi \frac{55 \text{ MHz}}{200 \text{ MHz}}. \text{ Bandpass filter.}$$

Wideband Solution #2: Bandpass filter. Poles at $0.95 e^{\pm j \omega_{3570}}$ and zeros at ± 1 .

$$\omega_{3570} = 2\pi \frac{3570 \text{ MHz} - f_c}{f_s} = 2\pi \frac{3570 \text{ MHz} - 3550 \text{ MHz}}{400 \text{ MHz}} = 2\pi \frac{20 \text{ MHz}}{400 \text{ MHz}}$$

$$\omega_{-3570} = 2\pi \frac{-3570 \text{ MHz} + f_c}{f_s} = 2\pi \frac{-3570 \text{ MHz} + 3550 \text{ MHz}}{400 \text{ MHz}} = -2\pi \frac{20 \text{ MHz}}{400 \text{ MHz}}$$

Narrowband Solution: Lowpass filter. Poles at 0.95 and 0.95. Zeros at -1 and -1.

$$\omega_{3570} = 2\pi \frac{3570 \text{ MHz} - f_c}{f_s} = 2\pi \frac{3570 \text{ MHz} - 3570 \text{ MHz}}{2 \text{ MHz}} = 0$$

$$\omega_{-3570} = 2\pi \frac{-3570 \text{ MHz} + f_c}{f_s} = 2\pi \frac{-3570 \text{ MHz} + 3570 \text{ MHz}}{2 \text{ MHz}} = 0$$

A student came up with better solution which is to pass frequencies from -0.8 MHz to 0.8 MHz which are equivalent to discrete time frequencies of

$$\omega_{-0.8 \text{ MHz}} = -2\pi \frac{0.8 \text{ MHz}}{f_s} = -2\pi \frac{0.8 \text{ MHz}}{2.0 \text{ MHz}} = -0.8\pi$$

$$\omega_{0.8 \text{ MHz}} = 2\pi \frac{0.8 \text{ MHz}}{f_s} = 2\pi \frac{0.8 \text{ MHz}}{2.0 \text{ MHz}} = 0.8\pi$$

Poles at $0.75 e^{\pm j \omega_{0.8 \text{ MHz}}}$. Zeros at -1 and -1.

Note: For this more general case of an SPN-43 radar operating anywhere in the CBRS band, a wideband solution is preferred. Wideband solution #2 is a software-defined radio solution that allows the simultaneous operation of a communication system and detection of an operating radar in the CBRS band. For a software-defined radio solution, see Midterm 1.3 from Fall 2018.

[1] AN/SPN-43 radar tutorial, <https://www.radartutorial.eu/19.kartei/07.naval/karte041.en.html>

[2] R. Caromi, M. Souryal, W.-B. Yang, "Detection of Incumbent Radar in the 3.5 GHz CBRS Band", *IEEE Global Signal Proc. Conf.*, 2018. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=926222. This approach uses a spectrogram to detect an SPN-43 radar operating anywhere in the CBRS band.

Problem 1.4. Mystery Systems. 24 points.

You're trying to identify unknown discrete-time systems.

You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 8000 Hz over 0 to 5s

$$x(t) = \cos(2\pi f_1 t + 2\pi\mu t^2)$$

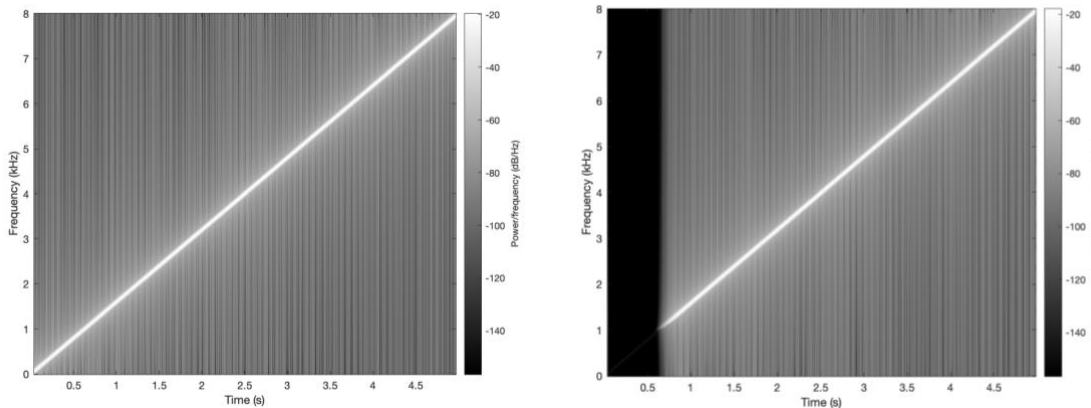
where $f_1 = 0$ Hz, $f_2 = 8000$ Hz, and $\mu = \frac{f_2 - f_1}{2 t_{\max}} = \frac{8000 \text{ Hz}}{10 \text{ s}} = 800 \text{ Hz}^2$. Sampling rate f_s is 16000 Hz.

In each part below, identify the unknown system as one of the following **with justification**:

1. filter – give selectivity (lowpass, highpass, bandpass, bandstop) and passband/stopband frequencies
2. upsampler – give upsampling factor
3. downsampler – give downsampling factor
4. pointwise nonlinearity – give the integer exponent k to produce the output $y[n] = x^k[n]$

In the grayscale color map, white has highest magnitude value.

(a) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lectures
1 3 5 & 6

Lab #3

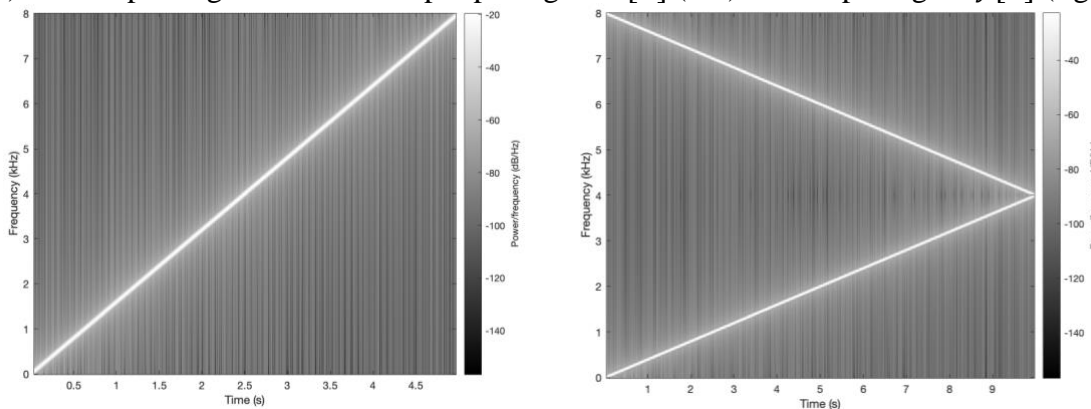
HW 1.1 1.3
2.1 2.2 & 2.3

JSK Ch. 7

Midterm 1.4
Sp22 & F23

In the output spectrogram, principal frequencies of the chirp input signal between 0 and 1 kHz are severely attenuated and other principal frequencies are passed. No new frequencies are created, so it is likely an LTI filter. Highpass filter. See next page for the Matlab code.

(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lecture 4

HW 0.3 & 2.2

Midterm Problems
1.4 F23 & 1.2 Sp23
1.2 F18 & 1.2(d) Sp18

Although there is no change in the frequency (vertical) axis, the duration of time on horizontal axis has doubled, which means upsampling by 2 when keeping the sampling rate the same for the purpose of plotting the output spectrogram. Upsampling by 2 per HW 2.2(e). See next pages.

```

%% Midterm Problem 2.4(a) highpass filter

clear all

fs = 16000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Design lowpass filter
fnyquist = fs/2;
fstop = 1000;
fpass = 1500;
ctfrequencies = [0 fstop fpass fnyquist];
idealAmplitudes = [0 0 1 1];
pmfrequencies = ctfrequencies / fnyquist;
filterOrder = 250;
h = firpm( filterOrder, pmfrequencies, idealAmplitudes );
h = h / sum(h.^2);

y = conv(x, h);

%%% Spectrogram parameters
blockSize = 1024;
overlap = 1023;
figure;

%%% Plot spectrogram of input signal
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;

%%% Plot spectrogram of output signal
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;

```

```
% Midterm Problem 2.4(b) upsampling by 2

fs = 16000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Upsample by 2
upsampleFactor = 2;
y = zeros(1, upsampleFactor*length(x));
y(1:upsampleFactor:end) = x;

%%% Spectrogram parameters
blockSize = 1024;
overlap = 1023;
figure;

%%% Plot spectrogram of input signal
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;

%%% Plot spectrogram of output signal
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
colormap gray;
```

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 *Solutions 2.0*

Date: October 16, 2024

Course: EE 445S Evans

Name: _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **No AI tools allowed.** As mentioned on the course syllabus, you may not use GPT or other AI tools during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

<i>Problem</i>	<i>Point Value</i>	<i>Your score</i>	<i>Topic</i>
1	25		FIR Filter Analysis
2	27		Predistortion
3	24		Discrete-time Feedback System
4	24		Mystery Systems
<i>Total</i>	100		

Problem 1.1 FIR Filter Analysis. 25 points.

Consider a causal linear time-invariant (LTI) discrete-time finite impulse response (FIR) filter with input $x[n]$ and output $y[n]$ observed for $n \geq 0$. The transfer function in the z -domain is

$$H(z) = a + b z^{-100} \quad \text{for } z \neq 0$$

where a and b are real-valued, non-zero constants.

- (a) Give the equation for output $y[n]$ in terms of the input $x[n]$ in the discrete-time domain for $n \geq 0$.
6 points.

$$H(z) = \frac{Y(z)}{X(z)} = a + b z^{-100} \quad \text{which means } Y(z) = a X(z) + b z^{-100} X(z)$$

So, $y[n] = a x[n] + b x[n - 100]$ for $n \geq 0$.

- (b) What are the initial condition(s) and their value(s)? Why? 3 points.

We can see the initial conditions by starting to compute the first few values of $y[n]$.

$$y[0] = a x[0] + b x[-100]$$

$$y[1] = a x[1] + b x[-99]$$

The initial conditions are

$$x[-1] = x[-2] = \dots = x[-100] = 0$$

This will satisfy the necessary (but not sufficient) conditions for linearity and time-invariance properties to hold.

- (c) In managing the memory for storing the previous input values, would you advocate for a linear buffer or a circular buffer? Why? 3 points

To update the storage of the current and previous 100 input samples each time a new input sample arrives, a linear buffer would require 100 reads and 100 writes

$$x[-100] = x[-99]; x[-99] = x[-98]; \dots; x[-1] = x[0]$$

whereas a circular buffer would require one read and two writes. The circular buffer update would need to read the pointer to the oldest value, write the current input value into the memory location for the oldest sample, and then update the pointer to the oldest value. I would use a circular buffer due to its much lower complexity.

- (d) Derive a formula for the discrete-time frequency response of the filter. 3 points.

Since the transfer function $H(z)$ includes the unit circle in the region of convergence, we can substitute $z = e^{j\omega}$ to convert the transfer function into a frequency response:

$$H(e^{j\omega}) = a + b e^{-j 100 \omega}$$

- (e) Give all possible conditions on the constants a and b so that the FIR filter has constant group delay. Compute the constant group delay. 10 points.

$$H(e^{j\omega}) = a + b e^{-j 100 \omega} = e^{-j 50 \omega} (a e^{j 50 \omega} + b e^{-j 50 \omega})$$

Constant group delay means the phase response has constant slope: $GD(\omega) = -\frac{d}{d\omega} \angle H(e^{j\omega})$

Only possible if there's symmetry in impulse response about its midpoint (lecture slide 5-17):

Even symmetry: $a = b$. $H(e^{j\omega}) = e^{-j 50 \omega} (2 a \cos(50 \omega))$ and $GD(\omega) = 50$ samples

Odd symmetry: $a = -b$. $H(e^{j\omega}) = e^{-j 50 \omega} (2 j a \sin(50 \omega))$ and $GD(\omega) = 50$ samples

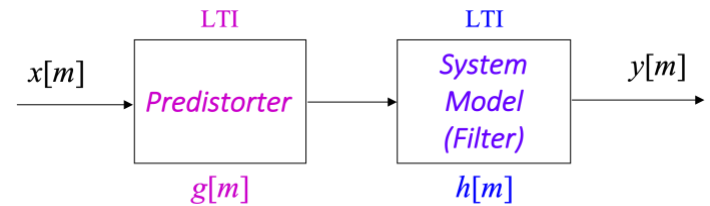
Note: If the predistorter were placed after the system, then it would be called an equalizer

Problem 1.2 Predistortion. 27 points.

Predistortion is a technique used to compensate the distortion in another system.

An example is applying predistortion to an audio signal before being played by an audio system.

The block diagram illustrates the use of predistortion when the predistorter is a linear time invariant (LTI) system and the distortion is modeled as LTI.

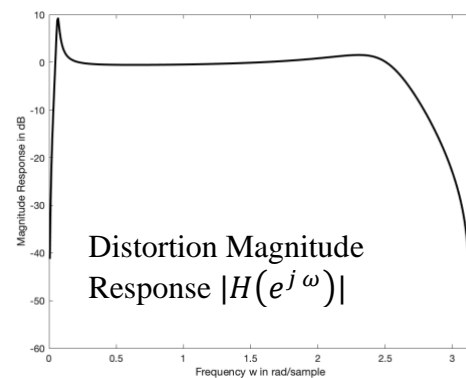


- sampling rate f_s is 44100 Hz.
- $g[m]$ is the impulse response of the discrete-time LTI predistorter
- $h[m]$ is the impulse response of a discrete-time LTI model of the distortion in a playback system.

A predistorter is bounded-input bounded-output stable.

Distortion. Its poles and zeros are given below and its magnitude response $|H(e^{j\omega})|$ is on given on the right.

<u>Poles</u>	<u>Zeros</u>
$p_0 = 0.75 e^{j 0.8 \pi}$	$z_0 = 1$
$p_1 = 0.75 e^{-j 0.8 \pi}$	$z_1 = -1$
$p_2 = 0.99 e^{j 0.02 \pi}$	$z_2 = 1$
$p_3 = 0.99 e^{-j 0.02 \pi}$	$z_3 = -1.25$



An all-pass cascade would mean $|G_{freq}(\omega) H_{freq}(\omega)| = g$ where g is a positive constant. Per lecture slide 6-15, a pole-zero pair in an all-pass configuration has two cases: (1) if the zero is inside the unit circle, then the pole and zero have the same value and cancel or (2) if the zero is outside the unit circle, the pole would have the same angle and reciprocal magnitude. If the zero is on the unit circle, place pole inside the unit circle at the zero angle for a notch configuration.

(a) What continuous-time frequency corresponds to pole location p_2 ? 3 points.

From sampling a continuous-time sinusoidal signal $x(t) = \cos(2\pi f_0 t)$ at sampling rate f_s , $x[n] = x(n T_s) = \cos\left(2\pi \frac{f_0}{f_s} n\right)$ where $\hat{\omega}_0 = 2\pi \frac{f_0}{f_s}$.

The discrete-time frequency $\hat{\omega}_2$ association with pole location p_2 is 0.02π , so

$$\hat{\omega}_2 = 2\pi \frac{f_2}{f_s} \text{ which means } f_2 = \frac{\hat{\omega}_2}{2\pi} f_s = \frac{0.02\pi}{2\pi} (44100 \text{ Hz}) = 441 \text{ Hz. Close to note 'A4'.$$

(b) Give the zeros of the predistorter. Explain how you determined each one. 12 points.

Answer #1: The zeros of the predistorter would be the values of the poles of the distortion. This will lead to pole-zero cancellations. Zeros do not cause BIBO stability.

$$z_0 = p_0 = 0.75 e^{j 0.8 \pi} \text{ and } z_1 = p_1 = 0.75 e^{-j 0.8 \pi}$$

$$z_2 = p_2 = 0.99 e^{j 0.02 \pi} \text{ and } z_3 = p_3 = 0.99 e^{-j 0.02 \pi}$$

Answer #2: Put the zeros in an all-pass configuration where each zero is at the same angle as the pole but the magnitude is inverted. Lecture Slide 6-15 and Handout O on all-pass filters.

$$z_0 = \frac{1}{0.75} e^{j 0.8 \pi} = \frac{4}{3} e^{j 0.8 \pi} \text{ and } z_1 = \frac{1}{0.75} e^{-j 0.8 \pi} = \frac{4}{3} e^{-j 0.8 \pi}$$

$$z_2 = \frac{1}{0.99} e^{j 0.02 \pi} \text{ and } z_3 = \frac{1}{0.99} e^{-j 0.02 \pi}$$

(c) Give the poles of the predistorter. Explain how you determined each one. *12 points.*

Notch configuration: $p_0 = 0.9$ and $p_1 = -0.9$ and $p_2 = 0.9$.

Reciprocal magnitudes for an all-pass configuration: $p_3 = -0.8$.

%% Matlab code to generate the magnitude response in Problem 1.2

`clear all;`

`fs = 44100;`

`z0 = 1;`

`z1 = -1;`

`numer1 = [1 -(z0+z1) z0*z1];`

`poleAngle = 0.8*pi;`

`r0 = 0.75;`

`p0 = r0 * exp(j*poleAngle);`

`p1 = r0 * exp(-j*poleAngle);`

`denom1 = [1 -(p0+p1) p0*p1];`

`z2 = 1;`

`z3 = -1.25;`

`numer2 = [1 -(z2+z3) z2*z3];`

`r = 0.99;`

`fpole = 441;`

`poleAngle = 2*pi*fpole/fs;`

`p2 = r * exp(j*poleAngle);`

`p3 = r * exp(-j*poleAngle);`

`denom2 = [1 -(p2+p3) p2*p3];`

%% Normalize the DC response to 1 in linear units by

%% setting H(z) evaluated at z = exp(j pi/2) to be 1

`numer = conv(numer1, numer2); % polynomial multiplication`

`denom = conv(denom1, denom2); % polynomial multiplication`

`zval = exp(j*pi/2);`

`zvec = zval .^ [0 -1 -2 -3 -4];`

`C = (denom * zvec) / (numer * zvec);`

`figure;`

`[h,w] = freqz(C*numer, denom);`

`p = plot(w,20*log10(abs(h)), 'k');`

`p(1).LineWidth = 2;`

`xlim([0 pi]);`

`xlabel('Frequency w in rad/sample');`

`ylabel('Magnitude Response in dB');`

Problem 1.3. Discrete-Time Feedback System. 24 points.

Consider a discrete-time linear time-invariant (LTI) system with input signal $x[n]$ and output signal $y[n]$ that is governed by the following second-order difference equation for $n \geq 0$:

$$y[n] = 1.6 y[n - 1] - K y[n - 2] + x[n]$$

where K is a real-valued constant.

- (a) What are the initial conditions of the system and what values should they have? 6 points.

A necessary condition for a system to have LTI properties is that it must be “at rest”.

That is, the initial conditions of the system must be zero.

We can find the initial conditions of the system by computing the first output values:

$$y[0] = 1.6 y[-1] - K y[-2] + x[0]$$

$x[0]$ and $y[0]$ are initial values of input signal $x[n]$ and output signal $y[n]$, respectively.

Neither is an initial condition of the system. Hence $y[-1] = 0$ and $y[-2] = 0$.

- (b) Derive the transfer function $H(z)$ for the system, which will depend on K . 6 points.

Take the z-transform of both sides of the equation:

$$Y(z) = 1.6 z^{-1} Y(z) - K z^{-2} Y(z) + X(z)$$

Next, collect $Y(z)$ terms on left-hand side of the equation:

$$Y(z) - 1.6 z^{-1} Y(z) + K z^{-2} Y(z) = X(z)$$

$$(1 - 1.6 z^{-1} + K z^{-2}) Y(z) = X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - 1.6 z^{-1} + K z^{-2}} = \frac{z^2}{z^2 - 1.6 z + K}$$

$H(z)$ has two zeros at the origin ($z = 0$) and poles at

$$\frac{-1.6 \pm \sqrt{(-1.6)^2 - 4K}}{2} = \frac{-1.6 \pm \sqrt{(-2 \times 0.8)^2 - 4K}}{2} = 0.8 \pm \sqrt{0.64 - K}$$

Region of convergence is complex z plane outside a circle of the larger pole radius: $|z| > \max\{|p_0|, |p_1|\}$

- (c) Give the range of values for K for which the system is bounded-input bounded-output (BIBO) stable. 6 points.

Both poles must be inside unit circle for BIBO stability. For $K \leq 0.64$, poles are real-valued:

$$-1 < 0.8 - \sqrt{0.64 - K} \text{ and } 0.8 + \sqrt{0.64 - K} < 1$$

The left inequality gives $K > -2.6$ and the right one gives $K > 0.6$. Hence, $K > 0.6$.

For $K > 0.64$, the poles are complex-valued: $0.8 \pm j \sqrt{K - 0.64}$.

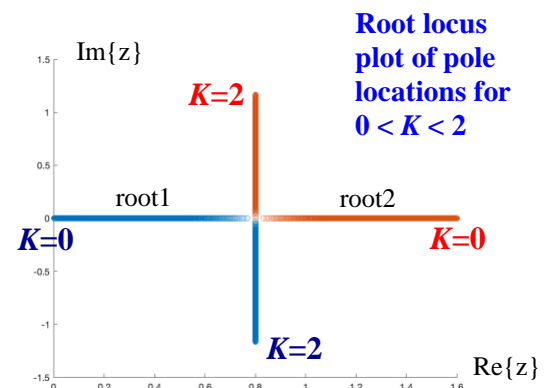
$$|0.8 + j \sqrt{K - 0.64}| < 1 \text{ which means } \sqrt{(0.8)^2 + (K - 0.64)} < 1$$

By squaring both sides, $(0.8)^2 + (K - 0.64) < 1$ which means $K < 1$. So, $0.64 < K < 1$.

The full range of K for BIBO stability is $0.6 < K < 1$.

- (d) Describe the possible frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch) that the system could exhibit for different values of K for which the system is BIBO stable. 6 points.

For $0.6 < K < 0.64$, poles are real-valued between 0.8 and 1.0, not inclusive. Lowpass. At $K = 0.64$, there is double pole at $z = 0.8$, which is also means a lowpass response. As K increases from 0.64 to 1, the pole separation increases and the response becomes bandpass.



```
K = 0 : 0.001 : 2;
root1 = 0.8 - sqrt(0.64 - K);
root2 = 0.8 + sqrt(0.64 - K);
hold on;
scatter(real(root1), imag(root1));
scatter(real(root2), imag(root2));
hold off;
```

Problem 1.4. Mystery Systems. 24 points.

You're trying to identify unknown discrete-time systems.

You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 8000 Hz over 0 to 5s

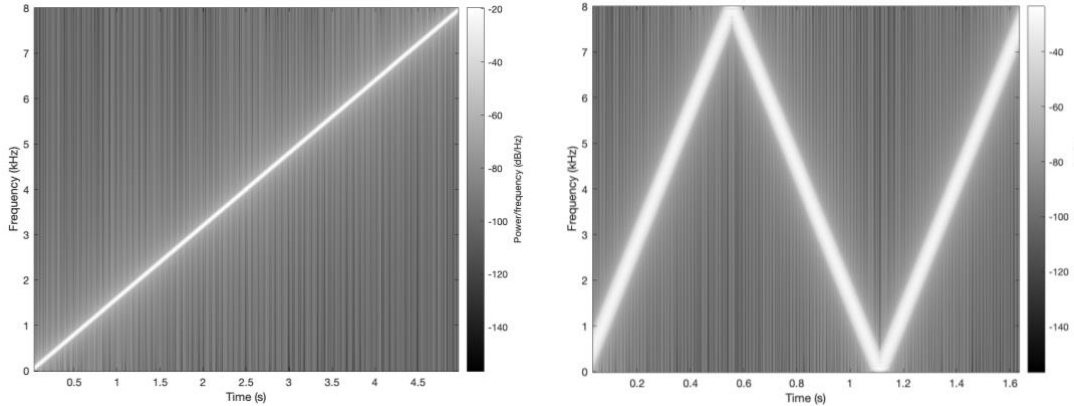
$$x(t) = \cos(2\pi f_1 t + 2\pi \mu t^2)$$

where $f_1 = 0$ Hz, $f_2 = 8000$ Hz, and $\mu = \frac{f_2 - f_1}{2 t_{\max}} = \frac{8000 \text{ Hz}}{10 \text{ s}} = 800 \text{ Hz}^2$. Sampling rate f_s is 16000 Hz.

In each part below, identify the unknown system as one of the following **with justification**:

1. filter – give selectivity (lowpass, highpass, bandpass, bandstop) and passband/stopband frequencies
2. upsampler – give upsampling factor
3. downsampler – give downsampling factor
4. pointwise nonlinearity – give the integer exponent k to produce the output $y[n] = x^k[n]$

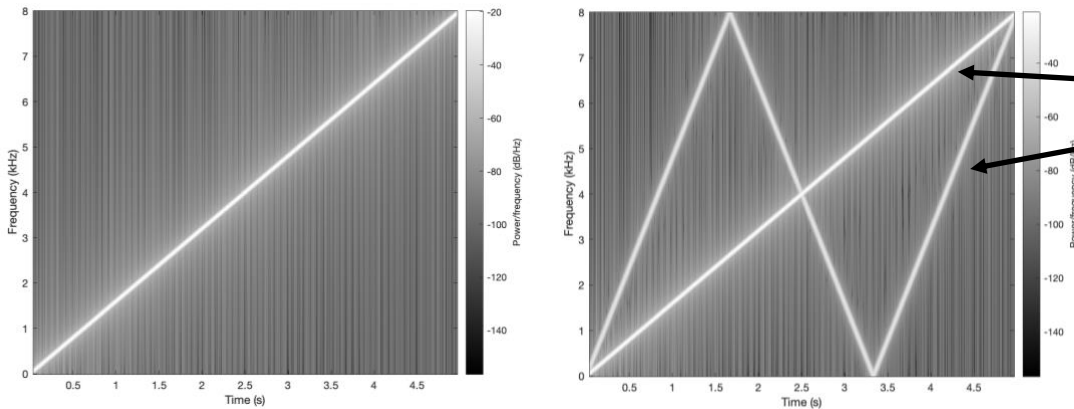
(a) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Midterm
Problems
1.4 Sp 22
1.2 F 18
1.2(d) Sp 18

When compared to the input spectrogram, the output spectrogram has the same range of frequencies along the vertical axis but one-third the duration in time along the horizontal axis. The principal frequency in the output spectrogram is a chirp pattern that is wider and has aliasing. Downsampling by 3 per homework problem 2.2(d).

(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). 12 points.



Lecture 4

Cubic nonlinearity gives frequencies $+f_0$ and $+3f_0$ as well as their negative counterparts:

$$\cos^3(2\pi f_0 t) = \cos^2(2\pi f_0 t) \cos(2\pi f_0 t) = \left(\frac{1}{2} + \frac{1}{2} \cos(2\pi(2f_0)t) \right) \cos(2\pi f_0 t)$$

$$\cos^3(2\pi f_0 t) = \frac{1}{2} \cos(2\pi f_0 t) + \frac{1}{2} \cos(2\pi(2f_0)t) \cos(2\pi f_0 t)$$

The product of two cosines gives a sum and difference of the frequencies, scaled by (1/2):

$$\cos^3(2\pi f_0 t) = \frac{3}{4} \cos(2\pi f_0 t) + \frac{1}{4} \cos(2\pi(3f_0)t)$$

The frequency at $+3f_0$ will alias once $f_0 \geq \frac{1}{6} f_s$. Folding for $\frac{1}{6} f_s \leq f_0 < \frac{1}{3} f_s$.

```
%% Matlab code to generate the spectrograms for Problem 1.4
```

```
fs = 16000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;
```

```
%% Create chirp signal
```

```
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));
```

```
%% (a) Downsample by 3
```

```
downsamplingFactor = 3;
xLength = length(x);
y = x(1:downsamplingFactor:xLength);
```

```
%%% Spectrogram parameters
```

```
blockSize = 1024;
overlap = 1023;
```

```
%%% Plot spectrogram of input signal
```

```
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
```

```
%%% Plot spectrogram of output signal
```

```
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```

```
%% (b) Cubic nonlinearity
```

```
y = x .^ 3;
```

```
%%% Spectrogram parameters
```

```
blockSize = 1024;
overlap = 1023;
```

```
%%% Plot spectrogram of input signal
```

```
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
```

```
%%% Plot spectrogram of output signal
```

```
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```

UNIVERSITY OF TEXAS AT AUSTIN
Dept. of Electrical and Computer Engineering

Quiz #2

Date: December 5, 2001

Course: EE 345S

Name: _____
Last, First

- The exam will last 75 minutes.
- Open textbooks, open notes, and open lab reports.
- Calculators are allowed.
- You may use any standalone computer system, i.e., one that is not connected to a network.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- Fully justify your answers.

Problem	Point Value	Your Score	Topic
1	30		True/False Questions
2	20		PAM & QAM
3	20		Pulse Shaping
4	15		ADSL Modems
5	15		Potpourri
Total	100		

Problem 2.1 True/False Questions. 30 points.

Please determine whether the following claims are true or false and **support each answer with a brief justification**. If you put a true/false answer without any justification, then you will get 0 points for that part.

- (a) The receiver (demodulator) design for digital communications is always more complicated than the transmitter (modulator) design.
- (b) Pulse shaping filters are designed to contain the spectrum of a digital communication signal. They will introduce ISI except that we sample at certain particular time instances.
- (c) The eye diagram does not tell you anything about intersymbol interference but only tells you how noisy or how clean the channel is.
- (d) QAM is more popular than PAM because it is easier to build a QAM receiver than a PAM receiver.
- (e) It is less accurate to use a DSP to realize in-phase/quadrature (I/Q) modulation and demodulation than to use an analog I/Q modulator and demodulator due to the quantization errors.
- (f) A low-cost DSP cannot be used for doing in-phase/quadrature (I/Q) modulation and demodulation for high carrier frequency (e.g., 100 MHz) since it does not have enough MIPS to implement it.
- (g) PAM and QAM will have the same bit-error-rate (BER) performance given the same signal-to-noise ratio (SNR).
- (h) Digital communication systems are better than analog communication systems since digital communication systems are more reliable and more immune to noise and interference.
- (i) FM and Spread Spectrum communications are examples of wideband communications. The excess frequency makes transmission more resistant to degradation by the channel.
- (j) Analog PAM generally requires channel equalization.

Problem 2.2 PAM and QAM. 20 points.

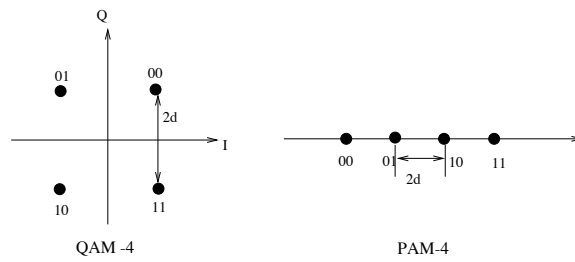


Figure 1: PAM-4 and QAM-4 (QPSK) constellation

Assume that the noise is additive white Gaussian noise with variance σ^2 in both the in-phase and quadrature components.

Assuming that 0's and 1's appear with equal probability.

The symbol error probability formula for PAM-4 is

$$P_e = \frac{3}{2} Q\left(\frac{d}{\sigma}\right)$$

- (a) Derive the symbol error probability formula for QAM-4 (also known as QPSK) shown in Figure 1. 10 points.

- (b) Please accurately calculate the power of the QPSK signal given d . Please compare the power difference of PAM-4 and QAM-4 for the same d . 5 points.
- (c) Are the bit assignments for the PAM or QAM optimal in Figure 1? If not, then please suggest another assignment scheme to achieve lower bit error rate given the same scenario, *i.e.*, the same SNR. The optimal bit assignment is commonly referred to as Gray coding. 5 points.

Problem 2.3 Pulse Shaping. 20 points.

Consider doing pulse shaping for a 2-PAM signal also known as BPSK signal. Assume the pulse shaping filter has 24 coefficients $\{h_0, \dots, h_{23}\}$ and the oversampling rate is 4.

- (a) Draw a block diagram of a filter bank scheme to implement the pulse shaping. Please also specify the number of the filters in the filter bank and express the coefficients of each filter in terms of h_0, \dots, h_{23} . 5 points.

- (b) Evaluate the number of MACs and the amount of RAM space required to accomplish the pulse shaping via the approach in (a). 5 points.

- (c) Since the data symbols coming into the filters in (a) are 1's or -1's (BPSK) and the filter coefficients are fixed, the pulse shaping filter can be implemented via a lookup table approach on a DSP (similar to the implementation of sine and cosine signals). Please describe one way of implementing the lookup table approach, including how to build the lookup table. 5 points.
- (d) If a bit shift operation and a MAC instruction each takes one instruction cycle (omitting the data move instructions), how many instructions and how much RAM space are required to implement the pulse shaping via the lookup table approach. Please compare the results with those in (b). 5 points.

Problem 2.4 ADSL Modems. 15 points.

- (a) What does the fast Fourier transform implement? 2 points.

- (b) Estimate the number of multiply-accumulates per second for the upstream and downstream fast Fourier transform. 4 points.

- (c) Before each symbol is transmitted, a cyclic prefix is transmitted. 3 points.
 - 1. How is the cyclic prefix chosen?

 - 2. Give two reasons why a cyclic prefix is used.

- (d) Compare discrete multitone (DMT) modulation, such as the ADSL standards, with orthogonal frequency division multiplexing (OFDM), such as for the physical layer of the IEEE 802.11a wireless local area network standard.
 - 1. Give three similarities between DMT and OFDM. 3 points.

 - 2. Give three differences between DMT and OFDM. 3 points.

Problem 2.5 Potpourri. 15 points.

- (a) You are evaluating two DSP processors, the TI TMS320C6200 and the TI TMS320C30, for use in a high-end laser printer that has to process 40 MB/s. Which of the two processors would you choose? Give at least three reasons to support your choice. 6 points.
- (b) You are designing an A/D converter to produce audio sampled at 96 kHz with 24 bits per sample. When an analog sinusoid is input to the A/D converter, the converter should produce one sinusoid at the right frequency and no harmonics. The converter should give true 24 bits of precision at low frequencies, but can give lower resolution at higher frequencies. Draw a block diagram of the A/D converter you would design. 9 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 3, 2003

Course: EE 345S

Name: _____
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework and solution sets.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network.
- Please turn off all cell phones and pagers.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- Fully justify your answers unless instructed otherwise.

Problem	Point Value	Your score	Topic
1	15		Phase Modulation
2	15		Equalizer Design
3	30		8-QAM
4	20		ADSL Receivers
5	20		Potpourri
Total	100		

Problem 2.1 Phase Modulation. 15 points.

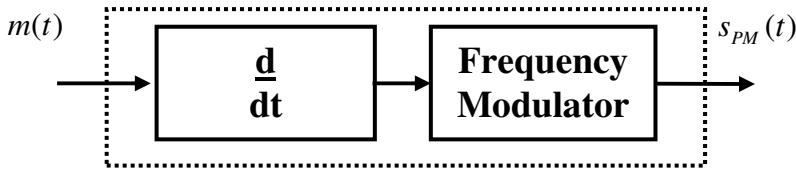
Phase modulation at a carrier frequency f_c of a causal message signal $m(t)$ is defined as

$$s_{PM}(t) = A_c \cos(2\pi f_c t + 2\pi k_p m(t))$$

Frequency modulation is defined as

$$s_{FM}(t) = A_c \cos\left(2\pi f_c t + 2\pi k_f \int_0^t m(\lambda) d\lambda\right)$$

- (a) Show that one can use a frequency modulator to generate a phase modulated signal using the block diagram below. Give k_p in terms of frequency modulation parameters. 5 points.



- (b) Give a version of Carson's rule for the transmission bandwidth of phase modulation. You do not have to derive it. 10 points.

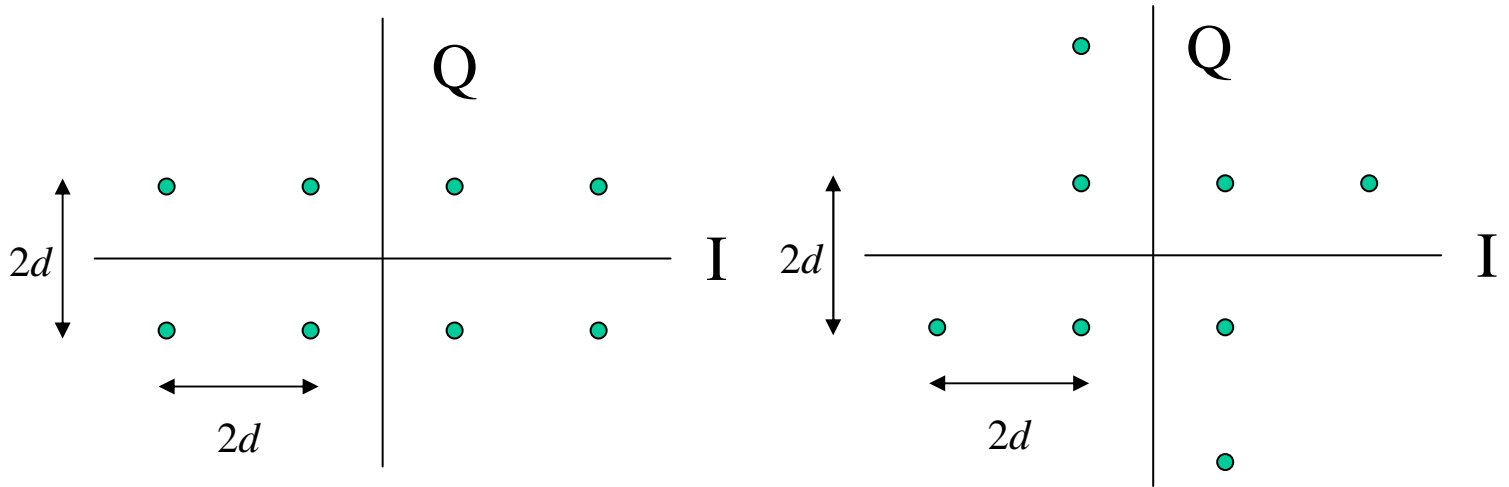
Problem 2.3 8-QAM. 30 points.

This problem asks you to compare the two different 8-QAM constellations below.

(i) Assume that the channel noise is additive white Gaussian noise with variance σ^2 in both the in-phase and quadrature components.

(ii) Assume that 0's and 1's occur with equal probability.

(iii) Assume that the symbol period T is equal to 1.



(a) Compute the average power for each 8-QAM constellation. 5 points.

(b) Compute the formula for probability of symbol error for each 8-QAM in terms of the Q function. Draw the decision regions you are using on the above constellations. 15 points.

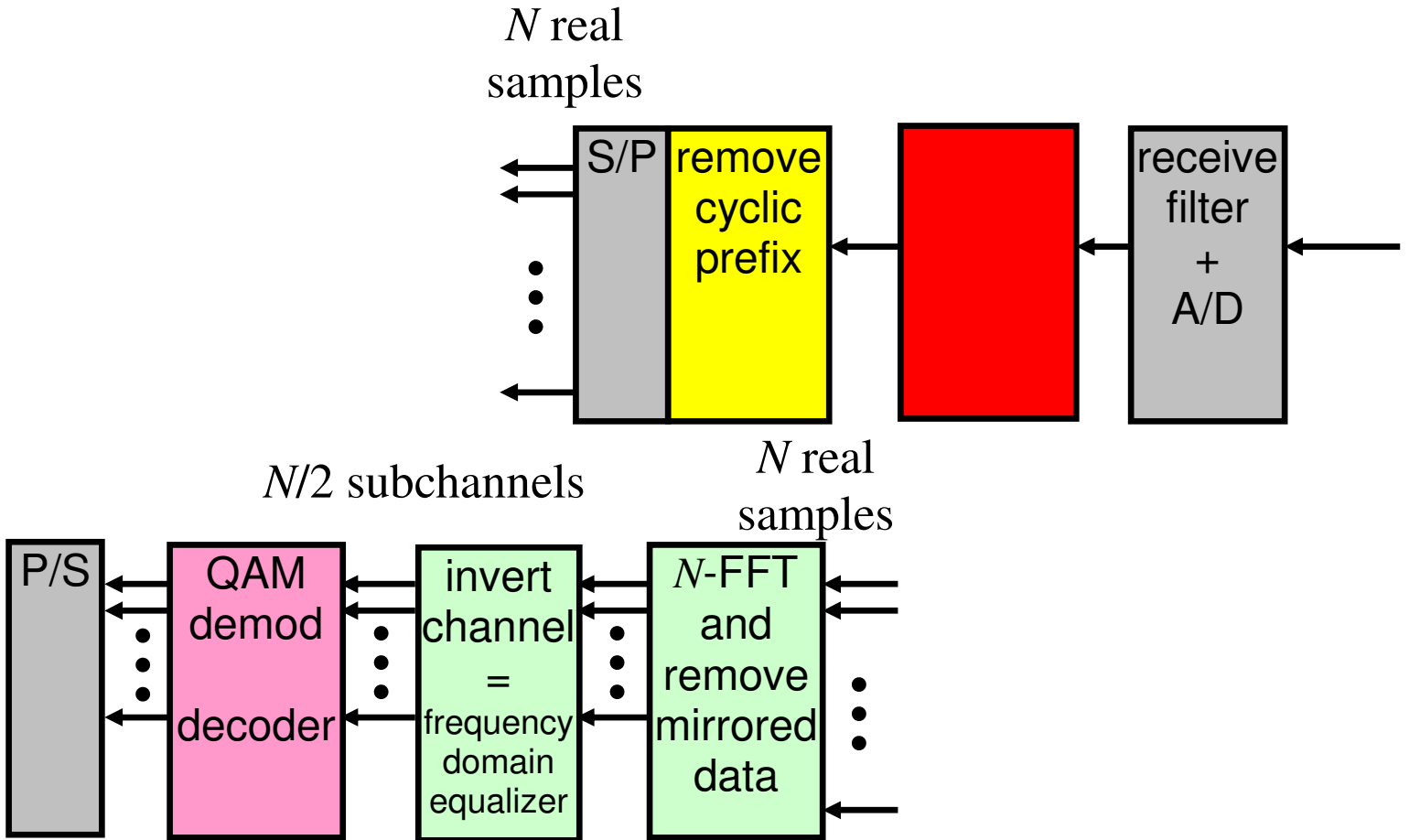
(c) Draw the optimal bit assignments for each symbol you would use for each of the constellations above on the constellations directly. 5 points.

(d) How would you choose which 8-QAM constellation to use in a modem? 5 points.

Problem 2.4 ADSL Receivers. 20 points.

Downstream ADSL transmission uses a symbol length N of 512, a cyclic prefix ν of 32 samples, and a sampling rate of 2.208 MHz. There are $N/2$ or 256 subchannels.

A downstream ADSL receiver for data transmission is shown below. The D/A converter has 16 bits of resolution. Use a word size of 16 bits for the analysis. The time-domain equalizer is a 32-tap FIR filter. Please calculate the computational complexity and memory usage of the each function shown except for the receive filter and A/D converter. (From slide 18-8).



Function	Multiply-accumulates	Compares	Words of memory
Time domain equalizer			
Remove cyclic prefix			
Serial-to-parallel converter			
Fast Fourier Transform			
Remove mirrored data			
Frequency domain equalizer			
QAM decoder			
Parallel-to-serial converter			

Problem 2.5 Potpourri. 20 points.

Please determine whether the following claims are true or false and **support each answer with a brief justification**. If you give a true or false answer without any justification, then you will receive zero points for that answer.

- (a) In a communication system design, digital communication should always be chosen over analog communications because digital communication systems are more reliable and more immune to noise and interference. 4 points.

- (b) Digital QAM is more popular than Digital PAM because it is easier to build a Digital QAM transmitter than a Digital PAM transmitter. 4 points.

- (c) Pulse shaping filters are designed to contain the spectrum of a digital communication signal. They are chosen to aid the receiver in locking onto the carrier frequency and phase. 4 points.

- (d) IEEE 802.11a wireless LAN modems and ADSL/VDSL wireline modems employ multicarrier modulation. 802.11a modems achieve higher bit rates than ADSL/VDSL because 802.11a systems deliver the highest bits/s/Hz of transmission bandwidth. 4 points.

- (e) FM radio uses excess frequency to make transmission more resistant to fading in wireless channels. 4 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 13, 2005

Course: EE 345S

Name: _____
Last, First

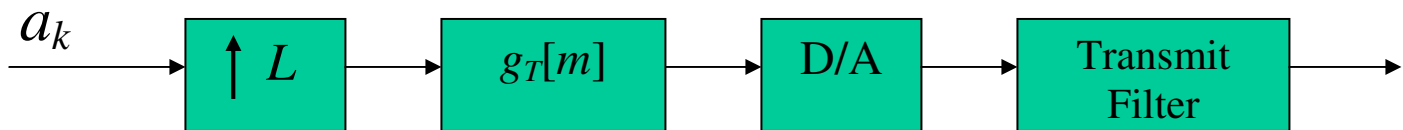
- The exam is scheduled to last 90 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network.
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the J&S and Tretter textbooks, course reader, and course handouts. Please be sure to reference the page/slide.

Problem	Point Value	Your score	Topic
1	15		Digital PAM Transmission
2	15		Equalizer Design
3	40		8-PAM vs. 8-QAM
4	15		Multicarrier Communications
5	15		Potpourri
Total	100		

Problem 2.1. Digital PAM Transmission. 15 points.

Shown below is a block diagram for baseband pulse amplitude modulation (PAM) transmission. The system parameters include the following:

- M is the number of points in the constellation
- $2d$ is the constellation spacing in the PAM constellation.
- $g_T[m]$ is the pulse shape (i.e. the impulse response of the pulse shaping filter)
- N_g is the length in symbols of the non-zero extent of the pulse shape
- f_{sym} is the symbol rate



- (a) Give a formula using the appropriate system parameters for the bit rate being transmitted. 2 points.
- (b) The leftmost block performs upsampling by L samples. What *communication* system parameter does L represent? 2 points.
- (c) Give a formula using the appropriate system parameters for the sampling rate for the D/A converter. 3 points.
- (d) Give a formula using the appropriate system parameters for the number of multiplication-accumulation operations per second that would be required to compute the two leftmost blocks in the above block diagram (i.e., the blocks before the D/A converter)
- I. as shown in the block diagram? 4 points.
 - II. using a filter bank implementation? 4 points.

Problem 2.2 *Equalizer Design*. 15 points.

You are given a discretized communication channel defined by the following sampled impulse response, where $|a| < 1$:

$$h[n] = a^{n-1} u[n-1]$$

- (a) Give the transfer function of the channel. 2 points.

- (b) Does the channel have a lowpass, highpass, bandpass, bandstop, allpass, or notch response? 2 points.

- (c) Design a causal stable discrete-time filter to equalize the above channel for a single-carrier communication system. 4 points.

- (d) Does the channel equalizer you designed in part (c) have a lowpass, highpass, bandpass, bandstop, allpass, or notch response? 2 points.

- (e) Using your answer in part (c), give the impulse response of the equalized channel. 5 points.

Problem 2.3 8-PAM vs. an 8-QAM constellation. 40 points.

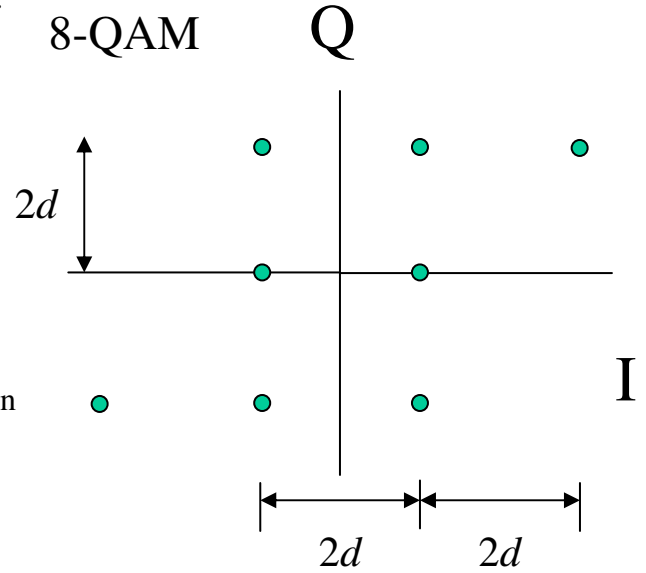
In this problem, assume that

(i) the channel noise for both in-phase and quadrature components is additive white Gaussian noise with variance of σ^2 and mean of zero.

(ii) 0's and 1's occur with equal probability.

(iii) the symbol period T is equal to 1.

Please complete the comparison below of 8-PAM and the version of 8-QAM shown on the right.



(a) Compute the average power for the 8-QAM constellation on the right. 5 points.

(b) Draw your decision regions on the 8-QAM constellation shown above. 5 points.

(c) Based on the decision regions in part (b), derive the formula for the probability of symbol error at the sampled output of the matched filter for the 8-QAM constellation in terms of the Q function and the SNR. 10 points.

(d) On the blank PAM constellation on the right, draw the 8-PAM constellation with spacing between adjacent constellation points of $2d$. 5 points.

8-PAM

(e) Compute the average power for the 8-PAM constellation. 5 points.

(f) *Derive* the formula for the probability of symbol error at the sampled output of the matched filter in the receiver for the 8-PAM constellation in terms of the Q function and the SNR. 5 points.

(g) Which constellation, the 8-PAM constellation on this page or the 8-QAM constellation on the previous page, is better to use and why? 5 points.

Problem 2.4 Multicarrier Communications. 15 points.

Here are some of the system parameters for a standard-compliant ADSL transceiver:

- Transmission bandwidth $B_T = 1.104$ MHz
 - Sampling rate $f_{\text{sampling}} = 2.208$ MHz
 - Symbol rate $f_{\text{symbol}} = 4$ kHz (same symbol rate in both downstream and upstream directions)
 - Number of subcarriers: $N_{\text{downstream}} = 256$ and $N_{\text{upstream}} = 32$
 - Cyclic prefix length is 1/16 of the symbol length
- (a) What is the ratio of the computational complexity of the downstream fast Fourier transform to the upstream fast Fourier transform in terms of real multiplication-accumulation (MAC) operations per second? 4 points.
- (b) During data transmission, what is the longest time domain equalizer that could be computed in real time on the C6701 digital signal processing board you have been using in lab? 4 points.
- (c) Which block in a multicarrier transceiver implements pulse shaping? What is the pulse shape? 4 points.
- (d) Every 69th frame in an ADSL transmission is a synchronization frame. For use between synchronization frames, describe in words a method for symbol synchronization. 3 points.

Problem 2.5 Potpourri. 15 points.

Please determine whether the following claims are true or false and **support each answer with a brief justification**. If you give a true or false answer without any justification, then you will be awarded zero points for that answer.

- (a) In a modem, as much of the processing as possible in the baseband transceiver should be performed in the digital, discrete-time domain because digital communications is more reliable and more immune to noise and interference than is analog communications. 3 points.

- (b) Pulse shaping filters are designed to contain the spectrum of a transmitted signal in a communication system. In a communication system, the pulse shape should be zero at non-zero integer multiples of the symbol duration and have its maximum value at the origin. 3 points.

- (c) Although wired and wireless channels have impulse responses of infinite duration, each can be modeled as an FIR filter. Wired channel impulse responses do not change over time, whereas wireless channel impulse responses change over time. 3 points.

- (d) A receiver in a digital communication system employs a variety of adaptive subsystems, including automatic gain control, carrier recovery, and timing recovery. A transmitter in a digital communication system does not employ any adaptive systems. 3 points.

- (e) All consumer modems for high-speed Internet access (i.e. capable of bit rates at or above 1 Mbps) employ multicarrier modulation. 3 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 7, 2007

Course: EE 345S

Name: _____
Last, First

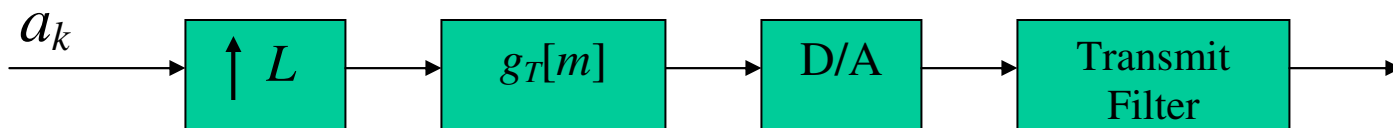
- The exam is scheduled to last 60 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your stand-alone computer system.**
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson & Sethares and Tretter textbooks, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	28		Digital PAM Transmission
2	24		Digital PAM Reception
3	24		Equalizer Design
4	24		Potpourri
Total	100		

Problem 2.1. Digital PAM Transmission. 28 points.

Shown below is a block diagram for baseband digital pulse amplitude modulation (PAM) transmission. The system parameters (in alphabetical order) include the following:

- $2d$ is the constellation spacing in the PAM constellation.
- f_{sym} is the symbol rate.
- $g_T[m]$ is the pulse shape (i.e. the impulse response of the pulse shaping filter).
- L is the upsampling factor, a.k.a. the oversampling ratio
- M is the number of points in the constellation, where $M = 2^J$.
- N_g is the length in symbols of the non-zero extent of the pulse shape.



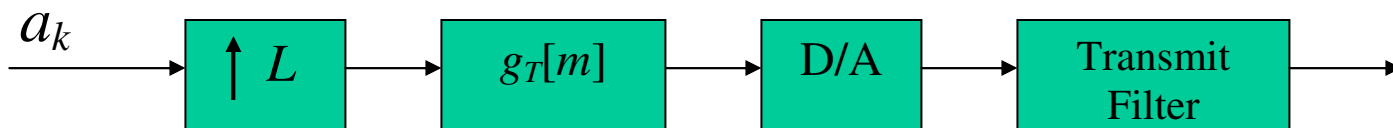
- (a) What does a_k represent? Give a formula using the appropriate system parameters for the values that a_k could take. 3 points
- (b) What *communication* system parameter does the upsampling factor L represent? 3 points.
- (c) Give a formula for the data rate in bits per second of the transmitter. 4 points.
- (d) Give formulas using the appropriate system parameters for the implementation complexity measures in the table below that would be required to compute the two leftmost blocks in the above block diagram (i.e., the blocks before the D/A converter). 18 points.

	Multiplication-accumulation operations per second	Memory Usage in Words	Memory Reads and Writes in words/second
As shown above			
Using a filter bank			

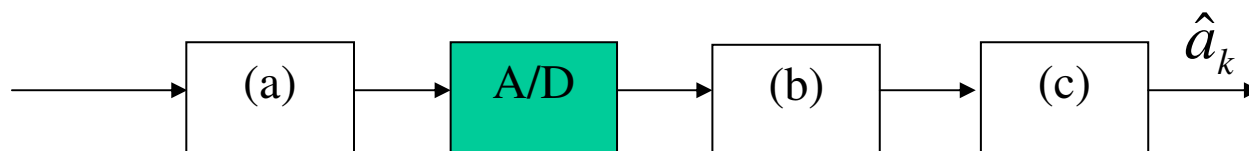
Problem 2.2 Digital PAM Reception. 24 points

As in problem 2.1, shown below is a block diagram for baseband digital pulse amplitude modulation (PAM) transmission. The system parameters (in alphabetical order) include the following:

- $2d$ is the constellation spacing in the PAM constellation.
- f_{sym} is the symbol rate.
- $g_T[m]$ is the pulse shape (i.e. the impulse response of the pulse shaping filter).
- L is the upsampling factor, a.k.a. the oversampling ratio
- M is the number of points in the constellation, where $M = 2^J$.
- N_g is the length in symbols of the non-zero extent of the pulse shape.



Here is a block diagram for baseband digital PAM reception:



The blocks in the baseband digital PAM receiver are analogous to the blocks in the digital baseband PAM transmitter. The hat above the a_k term in the receiver means an estimate of a_k in the transmitter. Assume that the channel only consists of additive white Gaussian noise. Assume synchronization.

Please describe in words each of the missing blocks (a)-(c) and how to choose the parameters (e.g. filter coefficients) for each block. Each part is worth 8 points.

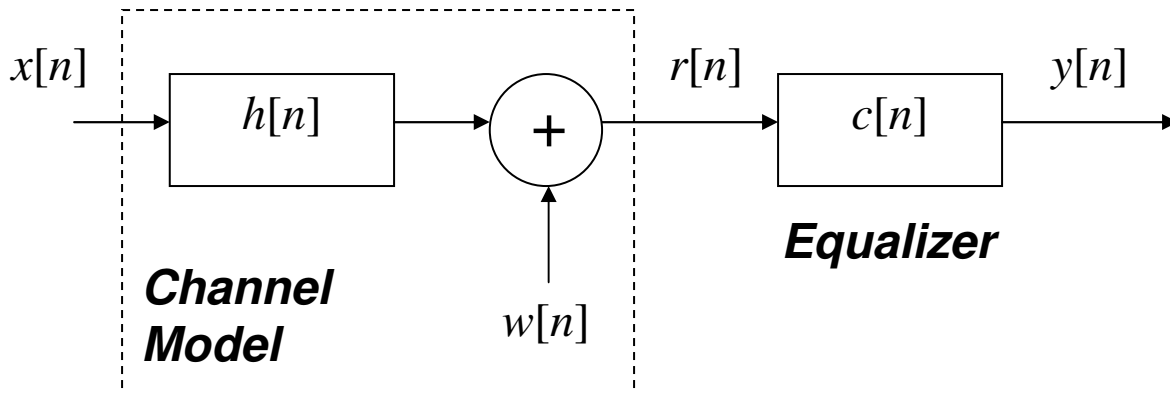
(a)

(b)

(c)

Problem 2.3 Equalizer Design. 24 points.

Consider a discrete-time baseband model of a communication channel that consists of a linear time-invariant finite impulse response (FIR) filter with impulse response $h[n]$ plus additive white Gaussian noise $w[n]$ with zero mean, as shown below:



During modem training, the transmitter transmits a short training signal that is a pseudo-noise sequence of length seven that is known to the receiver. The bit pattern is 1 1 1 0 1 0 0. The bits are encoded using 2-level pulse amplitude modulation (but without pulse shaping) so that

- for $x[n]$ equal to the sequence 1, 1, 1, -1, 1, -1, -1,
 - the channel output $r[n]$ is equal to the sequence 0.982, 2.04, 2.02, -0.009, 0.040, -2.03, -0.891.
- (a) Assuming that $h[n]$ has two non-zero coefficients, i.e. $h[0]$ and $h[1]$, estimate their values to three significant digits. 6 points.
- (b) Using the result in (a), estimate the coefficients for a two-tap FIR filter $c[n]$ to equalize the channel. What value of the delay are you assuming? 6 points.
- (c) Without changing the training sequence, describe an algorithm that the receiver can use to estimate the true length of the FIR filter $h[n]$. You do not have to compute the length. 6 points.
- (d) In a receiver, for a training sequence of 8000 samples and an FIR equalizer of 100 coefficients, would you advocate using a least-squares equalizer design algorithm or an adaptive equalizer design algorithm for real-time implementation on a DSP processor? Why? 6 points.

Problem 2.4 Potpourri. 24 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) A common baseband model for wired and wireless channels is as an FIR filter plus additive white Gaussian noise. 4 points.

- (b) When a Gaussian random process is input to a linear time-invariant system, the output is also a Gaussian random process where the mean is scaled by the DC response of the linear time-invariant system and the variance is scaled by twice the bandwidth. 4 points.

- (c) Frequency shift keying is another type of multicarrier modulation method in which one or more subcarriers are “turned on” to represent the digital information being transmitted. The only use of frequency shift keying in a consumer electronics product is in telephone touchtone dialing (i.e. dual-tone multiple frequency signaling). 4 points.

- (d) All modems in currently available consumer electronics products for *very high-speed Internet access* (i.e. capable of bit rates at or above 5 Mbps) employ multicarrier modulation. 4 points.

- (e) The TI TMS320C6713 digital signal processing board you have been using in lab can compute the fast Fourier transform operation for downstream ADSL reception in real time using single-precision floating-point arithmetic. 4 points.

- (f) In ADSL, the pulse shape used in the transmitter is a square root raised cosine. 4 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 4, 2009

Course: EE 345S

Name: _____
Last, First

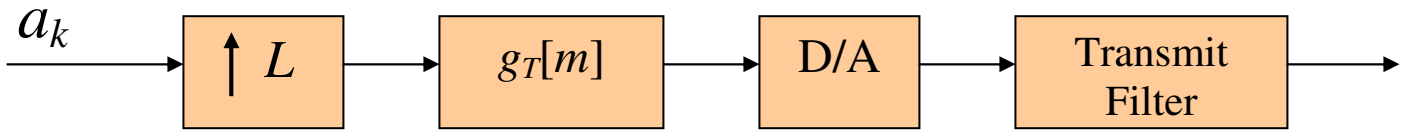
- The exam is scheduled to last 60 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your stand-alone computer system.**
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Tretter lab manual, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	27		Baseband Digital PAM Transmission
2	27		Digital PAM Reception
3	30		QAM
4	16		Equalizer Design
Total	100		

Problem 2.1. Baseband Digital PAM Transmission. 27 points.

Shown below is part of a baseband digital pulse amplitude modulation (PAM) transmitter. The system parameters (in alphabetical order) include the following:

- $2d$ is the constellation spacing in the PAM constellation.
- f_s is the sampling rate.
- $g_T[m]$ is the pulse shape (i.e. the impulse response of the pulse shaping filter).
- L is the upsampling factor, a.k.a. the oversampling ratio
- M is the number of points in the constellation, where $M = 2^J$.
- N_g is the length in symbol periods of the non-zero extent of the pulse shape.



- (a) What does a_k represent? Give a formula using the appropriate system parameters for the values that a_k could take. 3 points
- (b) What *communication* system parameter does the upsampling factor L represent? 3 points.
- (c) Give a formula for the bit rate in bits per second of the transmitter. 3 points.
- (d) Give formulas using the appropriate system parameters for the implementation complexity measures in the table below that would be required to compute the two leftmost blocks in the above block diagram (i.e., the blocks before the D/A converter). 18 points.

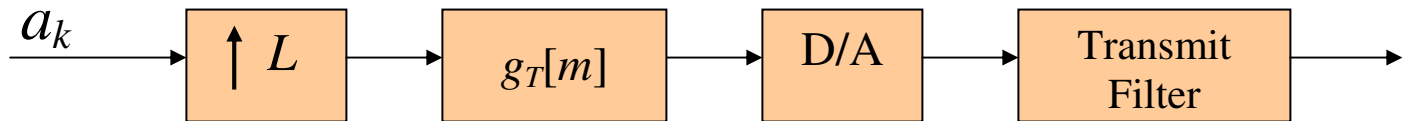
	Multiplication-accumulation operations per second	Memory usage in words	Memory reads and writes in words/second
As shown above			
Using a filter bank			

Problem 2.2 *Baseband Digital PAM Reception.* 27 points

As in problem 2.1, shown below is part of a baseband digital pulse amplitude modulation (PAM) transmitter. The system parameters (in alphabetical order) include the following:

- $2d$ is the constellation spacing in the PAM constellation.
- f_s is the sampling rate.
- $g_T[m]$ is the pulse shape (i.e. the impulse response of the pulse shaping filter).
- L is the upsampling factor, a.k.a. the oversampling ratio
- M is the number of points in the constellation, where $M = 2^J$.
- N_g is the length in symbols of the non-zero extent of the pulse shape.

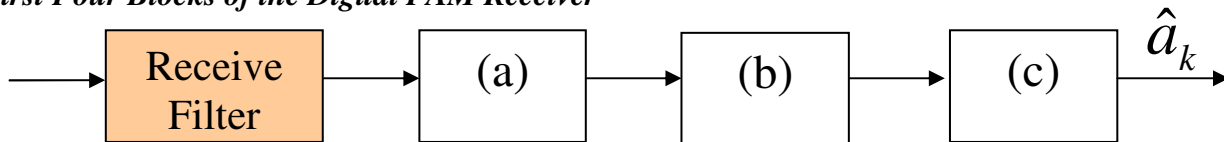
Last Four Blocks of the Digital PAM Transmitter



Channel Model

Additive white Gaussian noise.

First Four Blocks of the Digital PAM Receiver



The hat above the a_k term in the receiver means an estimate of a_k in the transmitter.

Assume that the receiver is synchronized to the transmitter.

Each block in the baseband digital PAM receiver is **analogous** to one block in the digital baseband PAM transmitter; e.g., the receive filter is **analogous** to the transmit filter.

Please describe in words each of the missing blocks (a)-(c) and how to choose the parameters (e.g. filter coefficients) for each block. Each part is worth 9 points.

(a)

(b)

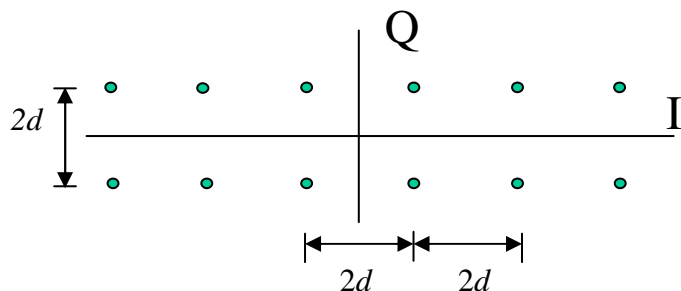
(c)

Problem 2.3 QAM. 30 points.

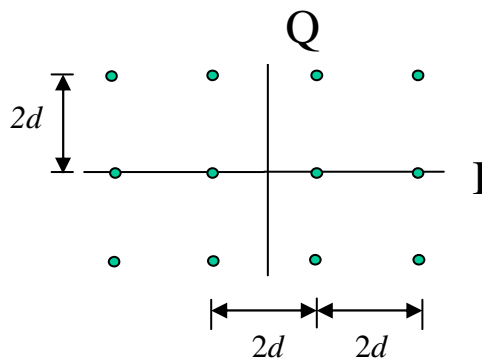
This problem asks you to evaluate two different 12-QAM constellations. Assumptions follow:

- (i) Each symbol is equally likely
- (ii) Channel only consists of additive white Gaussian noise with zero mean and variance σ^2 in both the in-phase (I) and quadrature (Q) components
- (iii) Perfect carrier frequency/phase recovery
- (iv) Perfect symbol timing recovery
- (v) Constellation spacing of $2d$
- (vi) Symbol duration $T_{\text{sym}} = 1$

Constellation #1



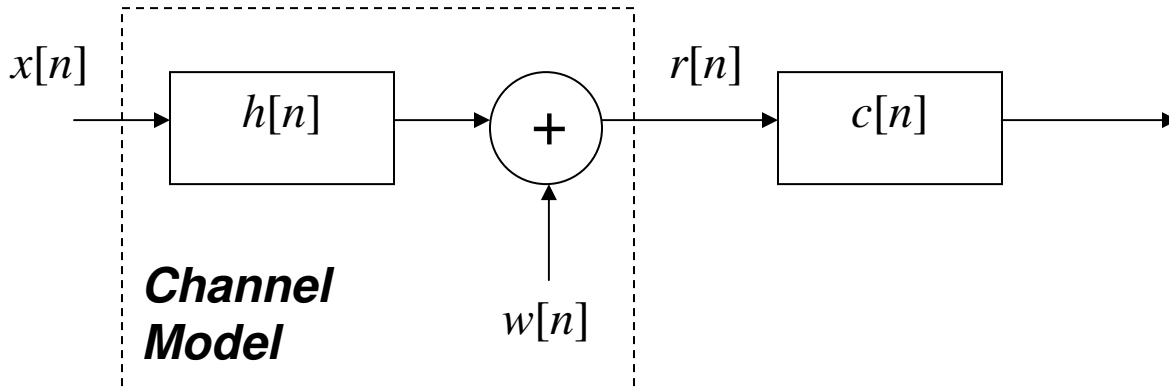
Constellation #2



- (a) Compute the **average signal power** for each of the QAM constellations above. 6 points.
- (b) Draw your **decision regions** on the 12-QAM constellations shown above. 6 points.
- (c) *Based on your decision regions in part (b), give* a formula for the probability of symbol error at the sampled output of the matched filter for each of the 12-QAM constellations in terms of the Q function, i.e. $Q(d/\sigma)$. 12 points.
- (d) Given the above assumptions and answers, which 12-QAM constellation would you choose? 6 points.

Problem 2.4 Equalizer Design. 16 points.

Consider a discrete-time baseband model of a communication system with transmitted signal $x[n]$ and received signal $r[n]$. The channel model is a linear time-invariant (LTI) finite impulse response (FIR) filter with impulse response $h[n]$ plus additive white Gaussian noise process with zero mean $w[n]$:



During modem training, the transmitter transmits a short training signal that is a pseudo-noise sequence of length seven that is known to the receiver. The bit pattern is 1 1 1 0 1 0 0. The bits are encoded using 2-level pulse amplitude modulation (but without pulse shaping) so that

- for $x[n]$ equal to the sequence 1, 1, 1, -1, 1, -1, -1,
- $r[n]$ is equal to the sequence 0.982, 2.04, 2.02, -0.009, 0.040, -2.03, -0.891,

(a) Assume that the equalizer is a two-tap LTI FIR filter. Compute an equalizer impulse response $c[n]$ for a transmission delay of zero. 7 points.

(b) In a receiver, for a training sequence of 8000 symbols and an FIR equalizer of 100 coefficients, would you advocate using a least-squares equalizer design algorithm or an adaptive equalizer design algorithm for real-time implementation on a digital signal processor? Why? 9 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 2, 2011

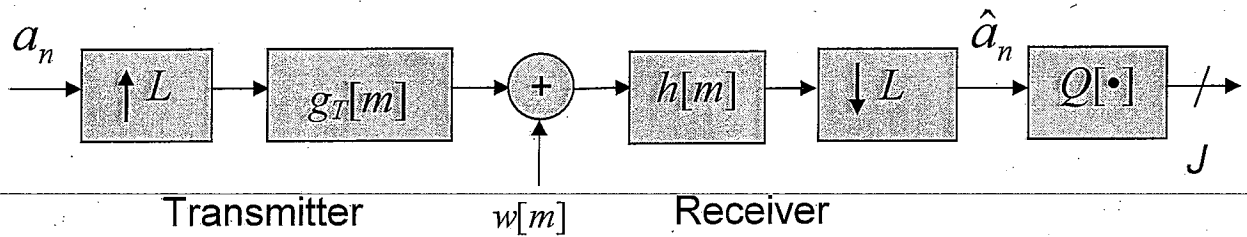
Course: EE 445S

Name: Set, Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your stand-alone computer system.**
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Tretter lab manual, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	28		Baseband Digital PAM Transceiver
2	27		Signal Quality
3	27		Automatic Gain Control
4	18		Potpourri
Total	100		

Problem 2.1 Baseband Digital PAM Transceiver in Discrete Time. 28 points.



The system parameters (in alphabetical order) include the following:

- $2d$ is the constellation spacing in the PAM constellation.
- f_{sym} is the symbol rate.
- $g_T[m]$ is the pulse shape (i.e. the impulse response of the pulse shaping filter).
- J is the number of bits per symbol.
- L is the number of samples per symbol duration.
- N_g is the length in symbols of the non-zero extent of the pulse shape.

The hat above the a_n term in the receiver means an estimate of a_n in the transmitter.

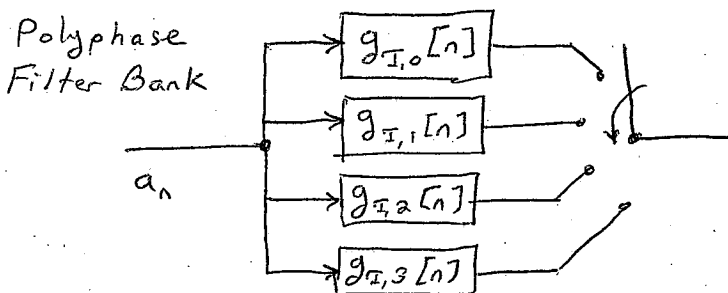
The only channel impairment is additive Gaussian noise given by $w[m]$.

Assume that the receiver is synchronized to the transmitter.

(a) What is the bit rate in bits per second? 4 points.

$$J f_{\text{sym}}$$

(b) Draw a block diagram for a more efficient implementation of the cascade of the upsampler by L for $L=4$ and the filter $g_T[m]$. What is the loss (if any) in signal quality? What is the reduction in computational complexity in terms of multiplications per second? 9 points.



No loss in signal quality - polyphase filter bank avoids multiplication by zero.
Reduction in computational complexity by a factor of L .

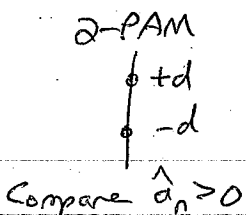
(c) What operation is represented by the filtering block described by impulse response $h[m]$? Give a formula for the best choice of $h[m]$. What measure of signal quality does it optimize? 9 points

Matched filter.

$$h_{\text{opt}}[m] = k g_T^*[L-m]$$

Optimizes peak pulse SNR at the output of the downsampler by L .

(d) Describe a fast algorithm for the quantizer $Q[\cdot]$ that does not involve any multiplications or additions. What is the computational complexity? 6 points.

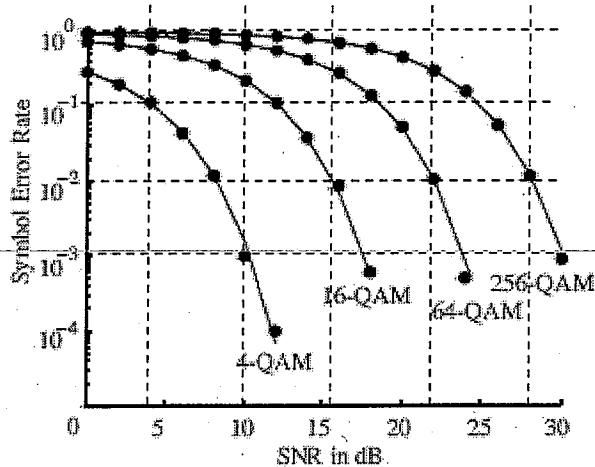


- 4-PAM
1. Compare $\hat{a}_n > 0$
 2. Then compare \hat{a}_n with $2d$ or $-2d$

Use divide-and-conquer.
Each comparison rules out half the remaining constellation points.
 J comparisons.

Problem 2.2. Signal Quality. 27 points.

Fig. 16.12 on page 380 of *Software Receiver Design* is shown on the right with dashed lines superimposed. The SNR is measured in the receiver at the decision device input. Error correction is not considered.



- (a) What is the probability of symbol error for 4-QAM when no transmitted signal power makes it to the receiver; i.e., the receiver only receives noise (SNR = $-\infty$ dB). 6 points.

Receiver would be reduced to randomly guessing which 4-QAM symbol had been transmitted.

$$P_e^{4\text{-QAM}} = \frac{3}{4} = 0.75$$

- (b) From the plot on the right, estimate the probability of symbol error for 4-QAM at an SNR of 0 dB. An SNR of 0 dB indicates an equal amount of signal and noise power. 6 points.

$$P_e^{4\text{-QAM}} \approx 10^{-0.5} = 0.316$$

- (c) At a symbol error rate of 10^{-2} , please give the difference in SNR in dB using the above plot
- Between 16-QAM and 4-QAM (addition of two bits/symbol). 3 points.

$$16\text{ dB} - 8\text{ dB} = 8\text{ dB}$$

- Between 64-QAM to 16-QAM (addition of two bits/symbol). 3 points.

$$22\text{ dB} - 16\text{ dB} = 6\text{ dB}$$

- (d) A formula to convert the number of bits J in a symbol to SNR in dB is $C_0 + C_1 J$.

- Using the above plot, give values of C_0 and C_1 in dB for QAM at symbol error rate of 10^{-2} . 3 points 256-QAM at 10^{-2} requires 28 dB.

$$C_0 = 4\text{ dB} \quad C_1 = 3\text{ dB/bit}$$

- What is C_1 for PAM? Why the difference? 6 points

$$C_1 = 6\text{ dB/bit from Quantization Lecture.}$$

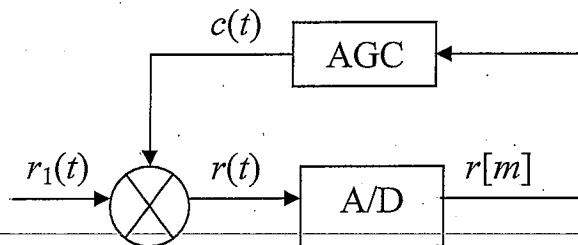
Since PAM isn't as efficient with the spectrum as QAM, more power is needed for PAM to send the same number of bits with same probability of symbol error. QAM can transmit two PAM signals in same transmission bandwidth.

Problem 2.3. Automatic Gain Control. 27 points.

Part of a 256-PAM receiver is shown on the right.

The analog multiplier computes $r(t) = c(t) r_1(t)$.

The analog/digital (A/D) converter outputs a signed two's complement 8-bit integer (i.e. 256 levels).



The automatic gain control (AGC) block outputs the gain $c(t)$ to be used in the analog multiplier:

- If the gain $c(t)$ is zero, all of the A/D output values will be 0.
- If the gain $c(t)$ is infinite, all of the A/D output values will be either the most positive value (127) or the most negative value (-128).

The AGC block computes how frequently A/D output values 127, 0, and -128 occur, denoted as f_{127} , f_0 , and f_{-128} , respectively.

Assume that the PAM symbol amplitudes are equally likely to occur.

- (a) Develop a formula for the AGC output $c(t)$ based on the values of f_{127} , f_0 , and f_{-128} . This formula could be used to adapt $c(t)$ over time in order to maximize the number of the 256-PAM symbol amplitudes uniquely represented in future A/D output values. 18 points.

If the gain $c(t)$ is too low, f_0 will be high.

If the gain $c(t)$ is too high, f_{127} and f_{-128} will be high.

We can update the gain as often as every sample taken by A/D.

Solution #1

$$c(t) = (1 + 2f_0 - f_{127} - f_{-128}) c(t - \tau_1) \text{ where } \tau_1 = T_s$$

Solution #2

$$c(t) = \frac{2f_0}{f_{127} + f_{-128}} c(t - \tau_2) \text{ where } \tau_2 = T_{\text{sym}}$$

Sanity check solutions by substituting initial values.

- (b) For your formula in (a), what are the initial values f_{127} , f_0 , and f_{-128} ? Hint: None is initially zero. 9 points.

Assume that all output values of the A/D are equally likely. Initial values are

$$f_0 = f_{127} = f_{-128} = \frac{1}{256}$$

Problem 2.4. Potpourri. 18 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) For a digital communications system, assuming that a channel equalizer is needed, the least squares channel equalizer should always be used instead of an adaptive channel equalizer. 6 points. FALSE.

Least Squares Equalizer

- Computes correlation matrix
- Computes matrix inverse and matrix multiplications (for more MACs)
- Gives best average equalizer over training sequence
- Requires floating point.

Adaptive Equalizer

- Computes FIR filter output for each training sequence value
- Updates FIR coefficients using vector adds & multiplies
- Tracks channel changes during training sequence
- Works in fixed-point or floating

- (b) In communication channel, additive noise is a Gaussian random process with constant mean and constant variance. 6 points.

True. Due to many independent noise sources adding together, the result tends toward a Gaussian distribution due to the Central Limit Theorem.

False. Gaussian random process models thermal noise in the system, which changes variance with temperature.

False. Other distributions model other noise sources, e.g. in powerline communication systems.

- (c) In his guest lecture, Prof. Andrews claimed that the best way to meet the exponential increase in mobile data traffic is to increase the transmission power of tower-mounted traditional wireless base stations. 6 points.

False. Smaller cells (femtocells and picocells) can be used to target high usage areas better, and are easier and cheaper to deploy than macrocell base stations. The smaller cells allow for better spectral reuse, which gives the highest increase in communication capacity.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: May 4, 2012

Course: EE 445S

Name: Set, Solution
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your stand-alone computer system.**
- Please turn off all cell phones, pagers, and personal digital assistants (PDAs).
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Welch, Wright and Morrow lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	27		Quadrature Amplitude Modulation
2	27		Channel Estimation
3	27		Pulse Amplitude Modulation Receiver
4	19		Noise Shaping
Total	100		

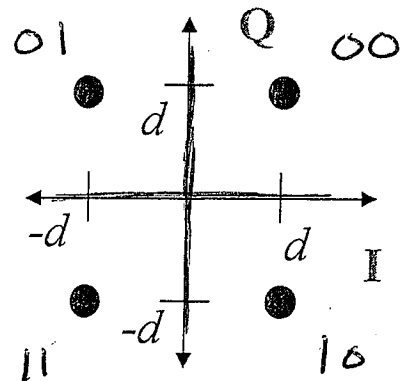
Problem 2.1 Quadrature Amplitude Modulation (QAM). 27 points.

A 4-level QAM constellation is shown on the right.

Assume that the symbol time is 1s.

Assume that the energy in the pulse shape is 1.

$$T_{\text{sym}} = 1$$



(a) On the 4-QAM constellation on the right, please specify an encoding for each level that minimizes the number of bit errors when a symbol error occurs. 6 points. *Gray coding* →

(b) Compute the average and peak transmitted power. 3 points.

$$\text{Total power: } 2d^2 + 2d^2 + 2d^2 + 2d^2 = 8d^2$$

$$\text{Average power: } \frac{8d^2}{4} = 2d^2$$

$$\text{Peak power: } 2d^2$$

(c) Draw decision regions at the receiver on the above constellation. 6 points. *I axis and Q axis.*

(d) Based on your decision regions in (c), give the fastest algorithm possible to decode/quantize the estimated symbol amplitude in the receiver into a symbol of bits. 6 points.

Symbol of bits has 2 bits $s_0 s_1$. Symbol amplitude (estimated) $= \hat{a}_n + j \hat{b}_n$

$$\text{If } (\hat{a}_n > 0) s_1 = 0 \text{ else } s_1 = 1$$

$$\text{If } (\hat{b}_n > 0) s_0 = 1 \text{ else } s_0 = 0$$

Two comparisons using divide-and-conquer strategy.

(e) Based on the decision regions in (c), give a formula for the probability of symbol error. 6 points.

Based on QAM transmitter lecture slides 15-13 and 15-14,

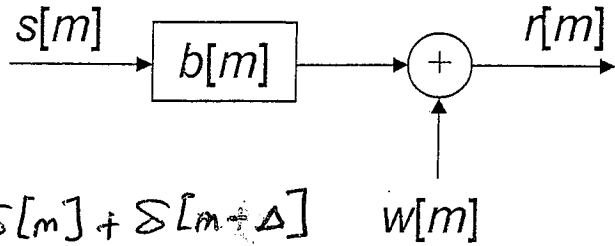
4-QAM decision regions are type-3 QAM regions (i.e. corner regions that are not edges).

$$P_3(c) = (1 - Q(\frac{d}{\sigma}))^2$$

$$P(e) = 1 - P_3(c) = 1 - (1 - Q(\frac{d}{\sigma}))^2 = 2Q(\frac{d}{\sigma}) - Q^2(\frac{d}{\sigma})$$

Problem 2.2. Channel Estimation. 27 points.

A sparse communication channel is modeled as a linear time-invariant (LTI) finite impulse response (FIR) comb filter $b[m]$ plus additive white Gaussian noise $w[m]$.



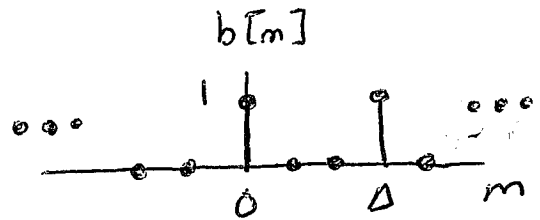
~~$b[m] = 1 + \delta[m - \Delta]$~~ where $\Delta > 1$ $b[m] = \delta[m] + \delta[m - \Delta]$

$w[m]$ has zero mean and variance σ^2 .

- (a) Give an equation in the discrete-time domain for the received signal $r[m]$ for the transmitted signal $s[m]$ and the channel model. 6 points.

$$r[m] = s[m] * b[m] + w[m]$$

$$r[m] = s[m] + s[m - \Delta] + w[m]$$



- (b) Give a training signal $s[m]$ that would enable accurate estimation of Δ for $\Delta > 1$. How would you determine the length of the training signal? 6 points.

Use a long maximal-length pseudo-noise sequence for the training signal $s[m]$. PN sequences are robust to frequency selective channels and to additive noise.

- (c) Using your answer in (b), give an algorithm in the receiver to estimate Δ . 6 points

In the receiver, we would correlate the received signal $r[m]$ against the training sequence. Two peaks should result at $m=0$ and at $m=\Delta$.

sequence's will better results in (c). PN length can be less than, equal to,

- (d) Give a formula for the impulse response or transfer function of a channel equalizer in the receiver or to compensate for the frequency selectivity of the channel. You may ignore the noise. 9 points.

$$b[m] = \delta[m] + \delta[m - \Delta]$$

$$B(z) = 1 + z^{-\Delta} \quad \leftarrow \Delta \text{ zeros on the unit circle}$$

Equalizer filter $G(z)$

We can't use $G(z) = \frac{1}{B(z)}$ because

$G(z)$ would have Δ poles on unit circle.

$$\text{Let } G(z) = \frac{1}{1 + 0.95z^{-\Delta}} \quad \leftarrow \Delta \text{ poles with radius } (0.95)^\Delta$$

IIR Comb Filter

Cascade of $B(z)$ and $G(z)$ gives LTI system with Δ notches.

Problem 2.3. Pulse Amplitude Modulation (PAM) Receiver. 27 points.

For a discrete-time baseband PAM receiver when the channel is modeled as additive white Gaussian noise, the first two blocks are:

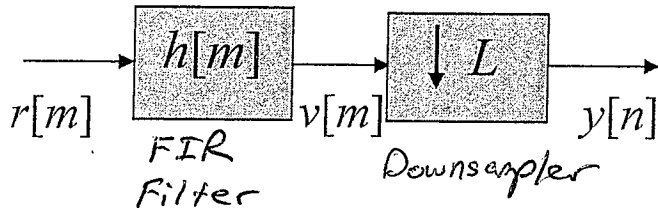
$r[m]$ is the discrete-time received signal.

$g[m]$ is the pulse shape used in the transmitter.

N_g is the number of symbol periods in the pulse shape.

f_{sym} is the symbol rate.

Note that $v[m] = h[m] * r[m]$ and $y[n] = v[Ln]$



$$h_{causal}[m] = k g^*[LN_g - m] \leftarrow$$

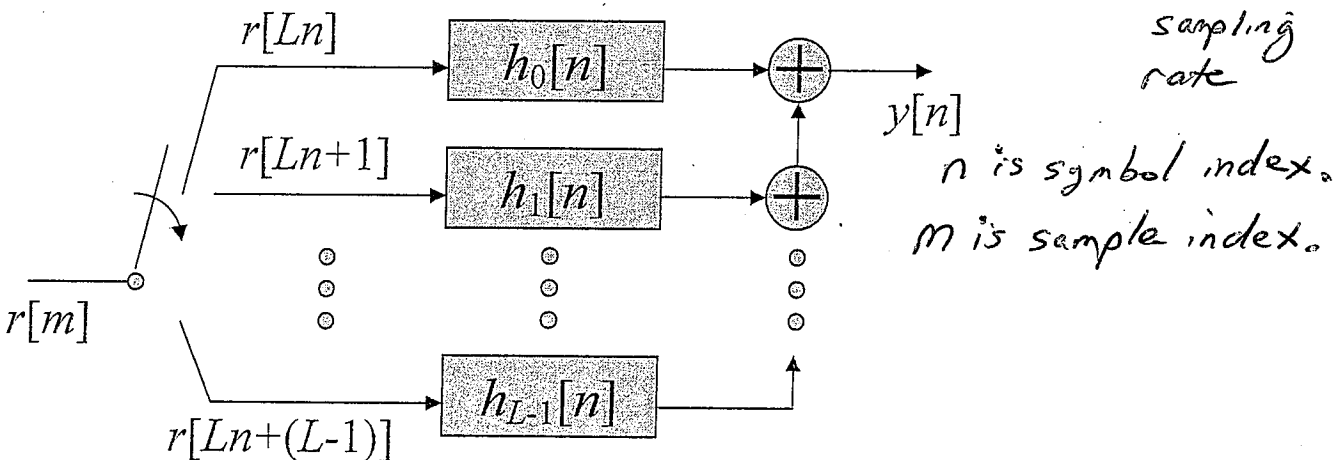
(a) Give a formula for the causal impulse response $h[m]$ that maximizes a measure of signal-to-noise ratio at $y[n]$. 6 points.

Matched filter $h[m] = k g^*[L-m]$ where $k \in \mathbb{R}$. Delay to make causal.

(b) How many multiplication-accumulation operations per second are needed for the two blocks above? 6 points.

For an FIR filter of LN_g coefficients, LN_g multiplication-accumulation (MAC) operations are needed for each output sample. $MACs/s (LN_g)(L f_{sym})$

The above cascade can be efficiently implemented as a polyphase filter bank as follows:



(c) Give a formula of $h_0[n]$ in terms of $h[n]$. Hint: Compare $y[N_g]$ for the direct form with $y[N_g]$ of the polyphase filter bank. 6 points.

From midterm #2 review slide 10:

$$h_0[n] = h[Ln] \text{ for } n = 0, 1, \dots, N_g - 1$$

$$y[l] = v[L] = h[0]r[L] + h[1]r[L-1] + \dots + h[L-1]r[1] + h[L]r[0]$$

(d) How many multiplication-accumulation operations per second are needed to implement the above polyphase filter bank? 9 points.

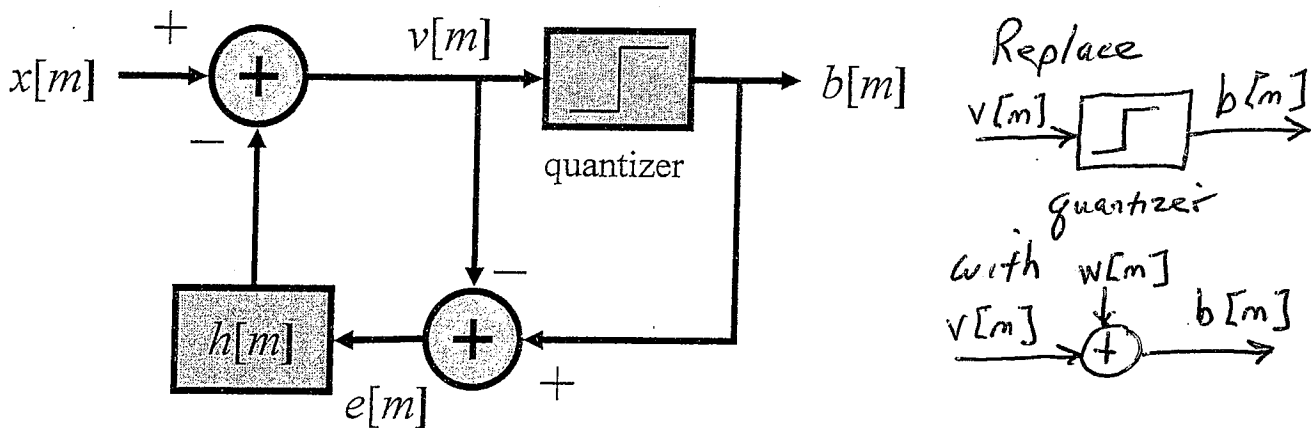
L filters with N_g coefficients each. Executes at symbol rate.

$$MACs/s = LN_g f_{sym}$$

Computational savings over direct implementation by a factor of L .

Problem 2.4. Noise Shaping. 19 points.

Here is a block diagram of a noise-shaping feedback coder used in data conversion.



$h[m]$ is the impulse response of a linear time-invariant (LTI) finite impulse response (FIR) filter.

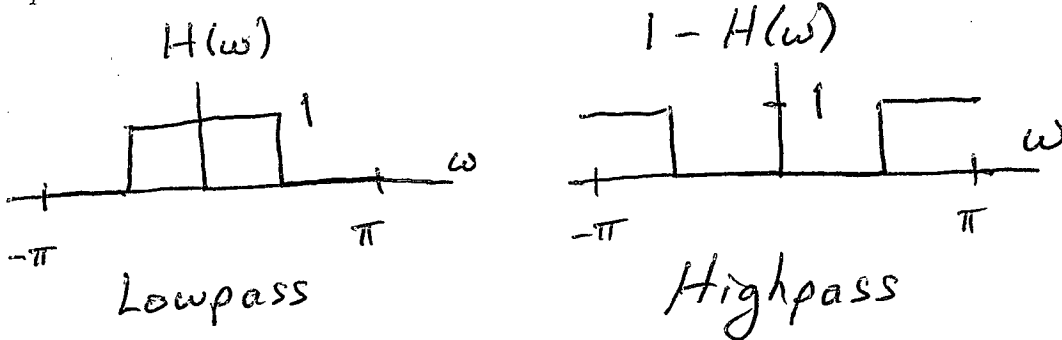
This problem asks you to analyze the noise shaping.

- (a) Replace the quantizer with an additive noise source $w[m]$, i.e. $b[m] = v[m] + w[m]$, and derive the transfer function in the frequency domain from the noise source $w[m]$ to the output $b[m]$. Assume that the input $x[m]$ is zero. 10 points. Set $x[m] = 0$.

$$\begin{aligned}
 b[m] &= v[m] + w[m] \\
 v[m] &= \cancel{x[m]} - h[m] * e[m] \\
 e[m] &= w[m]
 \end{aligned}
 \quad \left\{ \begin{aligned}
 B(\omega) &= V(\omega) - W(\omega) \\
 V(\omega) &= -H(\omega)W(\omega) \\
 &\Downarrow \\
 B(\omega) &= -H(\omega)W(\omega) + W(\omega) \\
 &\Leftarrow B(\omega) = (1 - H(\omega))W(\omega)
 \end{aligned} \right.$$

$$\frac{B(\omega)}{W(\omega)} = 1 - H(\omega)$$

- (b) If the frequency selectivity of $h[m]$ were lowpass, what is the frequency selectivity of the noise transfer function? Sketch example frequency responses for both to help justify your answers. 9 points.



$$\frac{B(\omega)}{W(\omega)} = 1 - H(\omega) \text{ is highpass.}$$

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 7, 2012

Course: EE 445S

Name: Umizoomi, Team
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all cell phones and other personal communication devices.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Welch, Wright and Morrow lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	30		Quadrature Amplitude Modulation
2	23		Channel Equalization
3	27		Analog-to-Digital Conversion
4	20		Potpourri
Total	100		

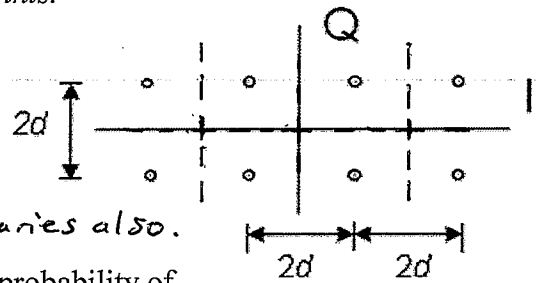
Bot
Milli
Unifriend
Geo

Problem 2.1 Quadrature Amplitude Modulation (QAM). 30 points.

An 8-level QAM constellation is shown on the right.

T_{sym} is the symbol time

Energy in the pulse shape is 1.



- (a) Draw decision regions at the receiver on the above constellation. 6 points. I and Q axis are boundaries also.
- (b) Based on the decision regions in (a), give a formula for the probability of symbol error. 6 points.

Two decision region types: four are corners and four are edge regions but not corners. Eight total regions.

$$P(e) = \frac{4}{8} (1 - Q(\frac{d}{\sigma} \sqrt{I_{\text{sym}}})) (1 - 2Q(\frac{d}{\sigma} \sqrt{I_{\text{sym}}})) + \frac{4}{8} (1 - Q(\frac{d}{\sigma} \sqrt{I_{\text{sym}}}))^2$$

$$P(e) = 1 - P(c) = \frac{5}{2} Q(\frac{d}{\sigma} \sqrt{I_{\text{sym}}}) - \frac{3}{2} Q^2(\frac{d}{\sigma} \sqrt{I_{\text{sym}}})$$

- (c) When increasing the value of d , does each of the following increase, decrease or stay the same? Why? 9 points.

$Q(x)$ is a monotonically decreasing function of x .

probability of symbol error? $\uparrow d \downarrow P(e)$ because $\uparrow d$ causes increase in argument of Q function.

symbol rate? There is no dependence for the symbol rate on d . Symbol rate stays the same.

implementation complexity/cost in transmitter Increasing d increases the average and peak transmit power, which increases cost in power

- (d) When increasing the value of T_{sym} , does each of the following increase, decrease or stay the same? Why? 9 points.

probability of symbol error? $\uparrow I_{\text{sym}} \downarrow P(e)$ because $\uparrow I_{\text{sym}}$ causes increase in argument of Q function

symbol rate? Decreases because $f_{\text{sym}} = \frac{1}{T_{\text{sym}}}$

implementation complexity/cost in transmitter

Decreases. Increase in T_{sym} causes decrease in f_{sym} , which causes a decrease in the sampling rate in the D/A converter. There are fewer MACs per second, as well.

because it has to operate in a linear region over a larger voltage range.

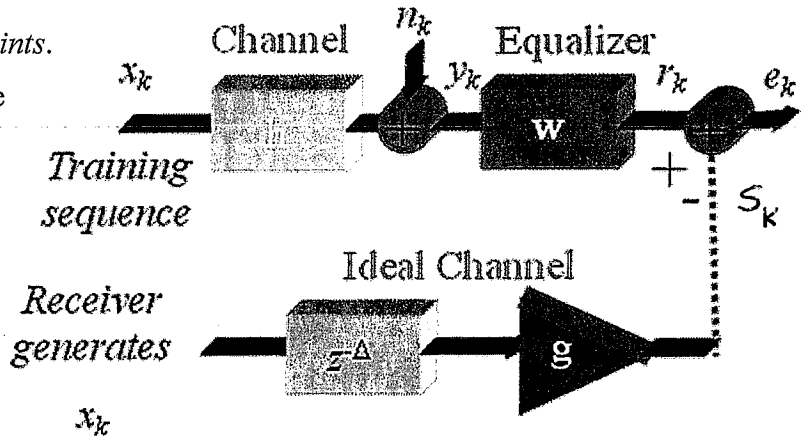
Problem 2.2. Channel Equalization. 23 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate.

Equalizer has two real-valued coefficients, and the first one is fixed to be one:

$$w[k] = \delta[k] + w_1 \delta[k-1] \quad \text{Impulse Response}$$

You may ignore the noise signal n_k .



- (a) During training, derive the update equation for w_1 to implement an adaptive least mean squares equalizer. 15 points.

$$w_1[k+1] = w_1[k] - \mu \frac{\partial J_{LMS}[k]}{\partial w_1} \Big|_{w_1 = w_1[k]}$$

$$J_{LMS}[k] = \frac{1}{2} e^2[k]$$

$$e[k] = r[k] - s[k] \quad \text{and} \quad s[k] = g x[k-\Delta]$$

$$r[k] = y[k] + w_1[k] y[k-1]$$

$$w_1[k+1] = w_1[k] - \mu e[k] y[k-1]$$

- (b) What values of Δ and g would you use? Why? 4 points.

Δ is the transmission delay through the equalized channel, and is between 0 and equalizer length - 1, inclusive. Let $\Delta = 0$.

g is the gain of the equalized channel. Letting $g = 1$ simplifies

- (c) What value of μ would you advocate? Why? 4 points.

To minimize the objective function $J_{LMS}[k]$, $\mu > 0$.

For stability, $0 < \mu < 1$.

We want μ to be small enough

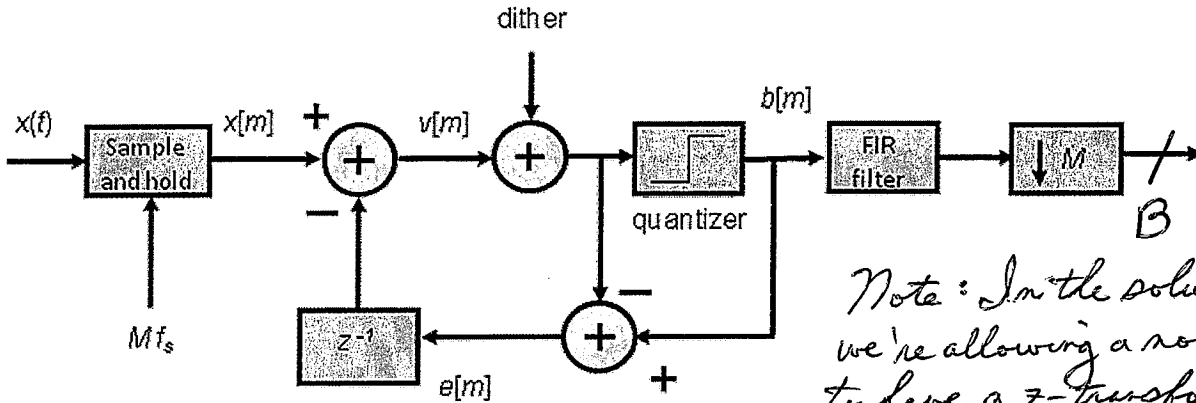
not to overshoot but large enough to converge to a good answer in fewer iterations (which reduces complexity).

From homework problem 7.2, $\mu = 0.001$. $\mu = 0.01$ is also okay.

the complexity of the adaptive update equation and allows exact recovery of transmitted signal without having to multiply by $\frac{1}{g}$.

Problem 2.3. Analog-to-Digital Conversion. 27 points.

For an analog-to-digital converter running at sampling rate of f_s and quantizing to B bits, here is a block diagram of a sigma-delta modulation implementation of the A/D:



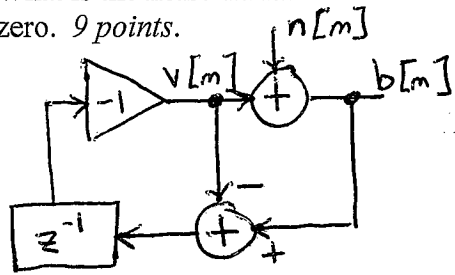
Note: In the solution for (b), we're allowing a noise signal to have a z-transform. This is only true for a finite length sequence.

The internal clock runs at Mf_s .

- (a) How would you efficiently generate an unsigned two-bit dither signal with a triangular probability density function? 6 points.

dither = $p_1 + p_2$ where p_1 and p_2 are independent 1-bit pseudo-noise sequences with very long periods. Adding two independent random variables yields pdf that is a convolution of

- (b) For analyzing noise shaping, we can replace the quantizer with an additive noise source $n[m]$. What is the noise transfer function from $n[m]$ to $b[m]$? Please set the dither signal and $x[m]$ to zero. 9 points.



$$B(z) = V(z) + N(z)$$

$$V(z) = (B(z) - V(z)) z^{-1} (-1)$$

$$= -N(z) z^{-1}$$

$$B(z) = -N(z) z^{-1} + N(z) \Rightarrow \frac{B(z)}{N(z)} = 1 - z^{-1}$$

pdfs p_1 and p_2 .

- (c) The output stage contains a finite impulse response (FIR) filter and a downsampler by M . If the internal quantizer gives 5 bits (signed) and if the A/D converter output is 12 bits (signed), how many 4-bit FIR coefficients (signed) are there for the following cases so that there is no loss of precision? 12 points.

For signed multiplication of 5-bit and a 4-bit multiplicands, the result is 8 bits (signed).

Direct implementation of the FIR filter (without inclusion of downsampler effects)

Adding two 8-bit signed values gives a 9-bit signed value worst case.

Adding two 9-bit signed values gives a 10-bit signed value in the

Polyphase filter bank implementation of cascade of FIR filter and downsampler worst case...

A polyphase filter bank would have M polyphase FIR filters.

Each filter can be $2^4 = 16$ coefficients long.

$2^4 = 16$

$16M$

Problem 2.4. Potpourri. 20 points.

Shown below are five common impairments in communication systems. For each impairment:

- Give the name of the receiver block or subsystem that would attempt to compensate it.
- Give the name of a design method for each block or subsystem and any assumptions made in the design method

(a) Additive noise. 4 points. Matched filter. Assumes noise is Gaussian.

For pulse shape $g[m]$ used in transmitter, matched filter has impulse response $h_{opt}[m] = k g^*[L-m]$ where L is number of samples per symbol time. (JSK p. 244.)

(b) Linear time-invariant distortion. 4 points. Channel equalizer.

Method #1: Least squares equalizer. (JSK pp. 273-284)

Method #2: Adaptive least mean squared equalizer.

Assumes transmitter sends training signal known by the

(c) Fading. 4 points. Automatic gain control. receiver.

Method #1: Squared difference adaptive element (JSK p. 122)

Method #2: Use counters of occurrence of A/D output values of maximum integer, 0, and minimum integer and adapt

(d) Carrier mismatch. 4 points. Carrier recovery. gain. (Midterm #2 from fall 2011.)

Use a phase locked loop (JSK p. 202).

Assumes that the carrier frequency reference is accurate.

Small frequency differences are tracked as phase variations.

(e) Symbol timing mismatch. 4 points. Timing recovery.

Method #1: Directed timing recovery assumes the combination of pulse shape, channel and matched

filter has Nyquist property. (JSK p. 256)

Method #2: Use two single-pole bandpass filters in parallel tuned to $\omega_c + 0.5\omega_{sym}$ and $\omega_c - 0.5\omega_{sym}$ respectively. Use nonlinearity and smoothing. See appendix M in reader.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: May 3, 2013

Course: EE 445S

Name: _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all cell phones and other personal communication devices.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Welch, Wright and Morrow lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	30		Channel Equalization
2	24		Quadrature Amplitude Modulation
3	24		Data Conversion
4	22		Potpourri
Total	100		

Problem 2.1. Channel Equalization. 30 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate.

Equalizer has real coefficients w_0 and w_1 :

$$r[k] = w_0 y[k] + w_1 y[k-1]$$

You may ignore the noise signal n_k .

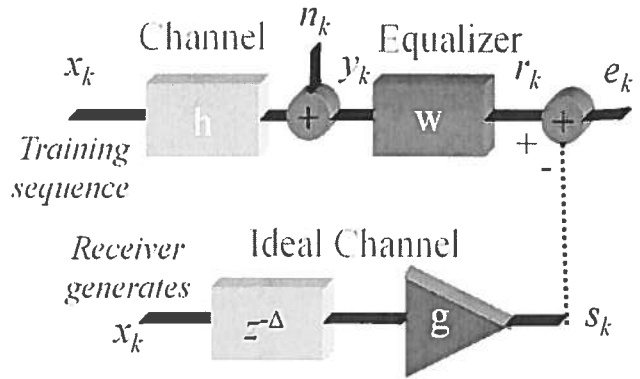
For the adaptive least mean squares (LMS) equalizer, the objective function is

$$J_{LMS}[k] = \frac{1}{2} e^2[k]$$

During training, the update equation for w_1 for iteration $k+1$ is

$$w_1[k+1] = w_1[k] - \mu e[k] y[k-1]$$

where μ is the constant step size.



(a) Derive the update equation for w_0 for an adaptive LMS equalizer. 12 points.

$$e[k] = r[k] - s[k] = w_0 y[k] + w_1 y[k-1] - g x[k-\Delta]$$

$$w_0[k+1] = w_0[k] - \mu \left. \frac{d}{dw_0} J_{LMS}[k] \right|_{w_0 = w_0[k]}$$

$$\frac{d}{dw_0} J_{LMS}[k] = e[k] y[k]$$

$$w_0[k+1] = w_0[k] - \mu e[k] y[k]$$

(b) Prior to training, what initial values would you give w_0 and w_1 ? Why? 6 points.

Initially, we can set the equalizer to match the ideal channel.

If $\Delta=0$, $w_0 = g$ and $w_1 = 0$. If $\Delta=1$, $w_0 = 0$ and $w_1 = g$.

(c) Let the vector of equalizer coefficients be $\vec{w} = [w_0 \ w_1]$. Using the result from (a), write the update in one equation in vector form. Please define any new vectors that you introduce. 6 points.

$$\vec{w}[k+1] = \vec{w}[k] - \mu e[k] \vec{y}[k]$$

$$\text{where } \vec{y}[k] = [y[k] \ y[k-1]]$$

(d) For an adaptive LMS equalizer with n coefficients, how many multiplications are needed per training sample? 6 points.

Vectors $\vec{w}[k+1]$, $\vec{w}[k]$ and $\vec{y}[k]$ have n entries.

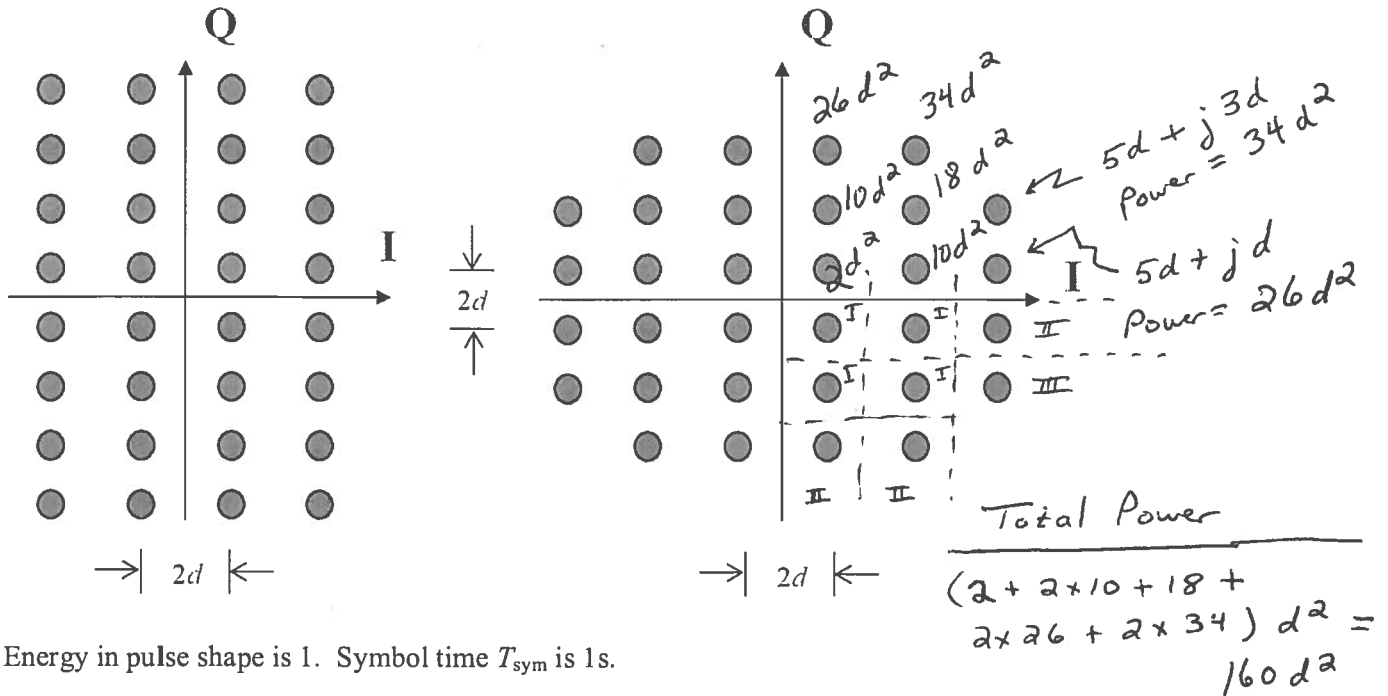
$e[k]$ takes $n+1$ multiplications to compute.

$\mu e[k]$ takes one multiplication.

$\mu e[k] \vec{y}[k]$ takes n multiplications. Total: $2n+2$ mults.

Problem 2.2 Quadrature Amplitude Modulation (QAM). 24 points.

Consider the two 32-QAM constellations below. Constellation spacing is $2d$.



	Left Constellation	Right Constellation
(a) Peak power	$56d^2$ $58d^2$	$34d^2$
(b) Average power	$25.875d^2$	$20d^2$
(c) Number of type I regions	12	16
(d) Number of type II regions	16	12
(e) Number of type III regions	4	4

Fill in each entry (a)-(e) for the right constellation. Each entry is worth 3 points.

Due to quadrant symmetry, average power can be computed over one quadrant.

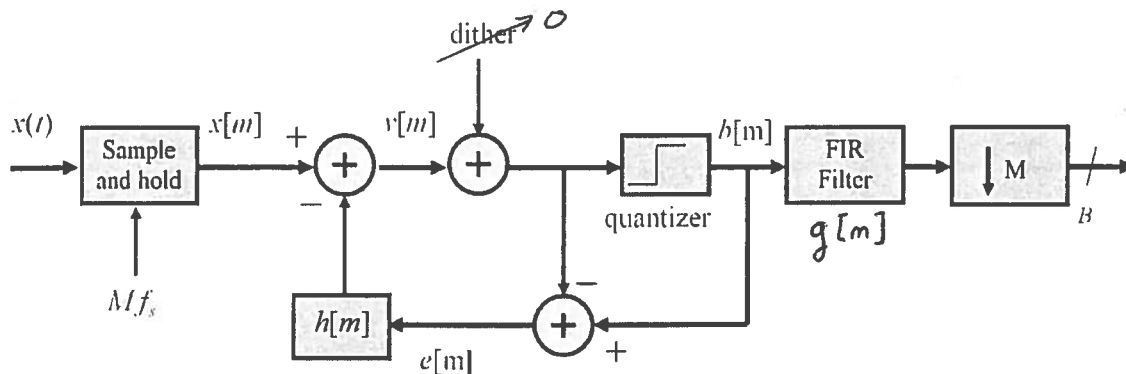
Which of the two constellations would you advocate using? Why? 9 points.

Pick the right constellation because it has lower peak power, lower average power, and lower peak-to-average power ratio.

Note: It is true that the left constellation would have a lower probability of symbol error as a function of $Q(\frac{d}{\sigma})$. Once $\frac{d}{\sigma}$ is put in terms of SNR, right constellation would have lower symbol error prob.

Problem 2.3. Data Conversion. 24 points.

For an analog-to-digital converter running at sampling rate of f_s and quantizing to B bits, here is a block diagram of a sigma-delta modulation implementation of the A/D:



The internal clock runs at Mf_s .

- (a) Replace the quantizer with a constant gain of K and assume $K \geq 2$. Derive the signal transfer function in the z -domain for input $x[m]$ and output $b[m]$. Please set the dither to zero. 12 points.

$$B(z) = K V(z)$$

$$E(z) = B(z) - V(z) = B(z) - (1/K) B(z) = \frac{K-1}{K} B(z)$$

$$V(z) = X(z) - H(z) E(z)$$

$$\frac{1}{K} B(z) = X(z) - H(z) \frac{K-1}{K} B(z) \Rightarrow \frac{B(z)}{X(z)} = \frac{K}{1 + (K-1)H(z)}$$

- (b) We can design the FIR filter prior to downsampling as a cascade of an equalizer and an anti-aliasing filter. Assuming that $h[m]$ is an FIR filter, please define the equalizer as the FIR filter that cancels the poles in the signal transfer function found in (a). 6 points.

$$G(z) = 1 + (K-1)H(z)$$

Note: In practice, we would put the equalizer after the downsampling by M for implementation complexity reduction.

- (c) Give a filter specification for the anti-aliasing filter. 6 points.

$$\omega_{\text{stop}} = \frac{\pi}{M}$$

$$\omega_{\text{pass}} = 0.9 \omega_{\text{stop}}$$

$$A_{\text{pass}} = 20 \log_{10} \Delta$$

$$A_{\text{stop}} = 6B + C_0$$

Note: An FIR filter of length $M+1$ coefficients that performs averaging would give 13.5 dB of

stopband attenuation (not much).

where $\Delta = \frac{1}{2^B - 1}$ constant that depends on the application.

Problem 2.4. Potpourri. 22 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) In a certain QAM system, pseudo-noise sequence is sent at the beginning of transmission. In the receiver, one would correlate against the known the PN sequence to determine when transmission has begun instead of an energy detector because the correlator has lower complexity. 7 points.

An energy detector requires 2 multiplications per sample.

A correlator requires n multiplications for a PN sequence of length n ($n > 2$).

FALSE: Energy detector has a lower complexity.

Note: An energy detector can be used until enough energy is

- (b) The symbol recovery method based on appendix M of the course reader and discussed in lecture 16 on QAM Receivers uses the Fourier property that a shift in time corresponds to a shift in detected frequency. That is why the method locks onto frequencies $\omega_c - \omega_{sym}$ and $\omega_c + \omega_{sym}$ for QAM to then symbol recovery. 7 points.

FALSE: The Fourier transform property of a shift in time leads to a phase shift in frequency.

FALSE: The symbol recovery method locks onto frequencies $\omega_c - \frac{1}{2}\omega_{sym}$ and $\omega_c + \frac{1}{2}\omega_{sym}$.

run the correlator. This saves power/energy.

- (c) In communication channel modeling, we model the frequency selectivity using a finite impulse response (FIR) filter because the linear time-invariant properties of all physical channels are FIR. 8 points.

FALSE: The physical channels have an infinite impulse responses when modeled as linear time-invariant systems. (a) Wireline channels can be modeled as RLC circuits. (b) Wireless channels can be modeled as having multiple propagation paths from transmitter to receiver (direct path, 1 reflection, 2 reflections, etc.). The infinite impulse response dies out. We truncate the response to be finite length.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 6, 2013

Course: EE 445S

Name: Test, Family
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all cell phones and other personal communication devices.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Welch, Wright and Morrow lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	25		Channel Equalization
2	27		Receiver Design
3	30		Pre-emphasis
4	18		Potpourri
Total	100		

Johnny
Susan
Mary
Dukey

Problem 2.1. Channel Equalization. 25 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate.

Equalizer is a finite impulse response (FIR) filter with two real coefficients w_0 and w_1 :

$$r[k] = w_0 y[k] + w_1 y[k-1]$$

You may ignore the noise signal n_k .

- (a) For the adaptive FIR equalizer, derive the update equation for w_1 for the following objective function: 12 points.

$$J(e[k]) = \frac{1}{4} e^4[k]$$

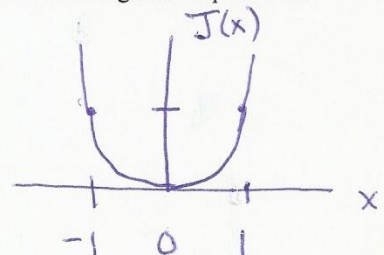
$$w_1[k+1] = w_1[k] - \mu \left. \frac{\partial J(e[k])}{\partial w_1} \right|_{w_1 = w_1[k]} \quad e[k] = r[k] - s[k] = w_0 y[k] + w_1 y[k-1] - s[k]$$

$$w_1[k+1] = w_1[k] - \mu e^3[k] y[k-1]$$

- (b) One of the problems with the adaptive FIR equalizer in part (a) is that its convergence depends on the initial value for w_1 .

Consider finding the roots of the polynomial

$$J(x) = \frac{1}{4} x^4$$



- i. Give the iterative update equation for estimates for x . 6 points.

Minimize $J(x)$

$$x[k+1] = x[k] - \mu \left. \frac{\partial J(x)}{\partial x} \right|_{x=x[k]} = x[k] - \mu x^3[k]$$

- ii. From the iterative update equation in part i, give the range of initial values of x to guarantee convergence. The range of values may depend on the step size μ . 7 points

$$x[k+1] = f(x[k]) = x[k] - \mu x^3[k]$$

Convergence occurs when $|f'(x)| < 1$.

$$f'(x) = 1 - 3\mu x^2$$

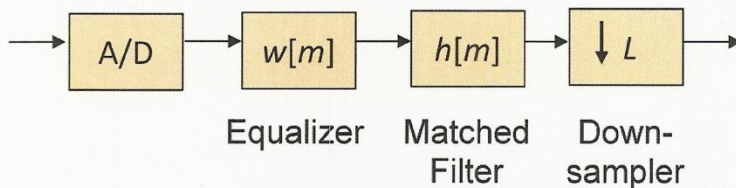
$$-1 < 1 - 3\mu x^2 < 1 \Rightarrow -2 < -3\mu x^2 < 0$$

$$|x| < \sqrt{\frac{2}{3\mu}} \Leftrightarrow 0 < 3\mu x^2 < 2$$

Problem 2.2 Receiver Design. 27 points.

Consider the baseband pulse amplitude modulation (PAM) receiver blocks below with

- sampling rate f_s
- downsampling factor $L = 6$ samples/symbol where $f_s = L f_{sym}$
- square root raised cosine pulse shape $g[m]$ with rolloff parameter $\alpha = 1$:



- a) Why is placing the FIR equalizer immediately after the A/D converter inefficient? Completing steps (b)-(d) below might help you here. 6 points.

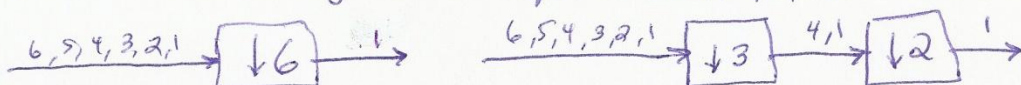
Transmission bandwidth is $\frac{1}{2}(1+\alpha)f_{sym} = f_{sym}$.
 Equalizer unnecessarily equalizes over $[0, 3f_{sym}]$ and operates at the sampling rate of $6f_{sym}$. Equalizer is followed by matched filter with bandwidth of f_{sym} .

- b) The first step to remove the inefficiency is to swap the order of the equalizer and matched filter. How can this be justified? 6 points.

After training, the equalizer is linear and time-invariant (LTI) if we set the initial conditions to zero. The matched filter is also LTI if initial conditions are zero. We can swap the order of two LTI systems in cascade under the assumption of exact

- c) Show in the discrete time domain that downsampling by 6 is the same as downsampling by 3 followed by downsampling by 2. 6 points.

Consider causal signal with amplitudes 1, 2, 3, ... precision calculations.

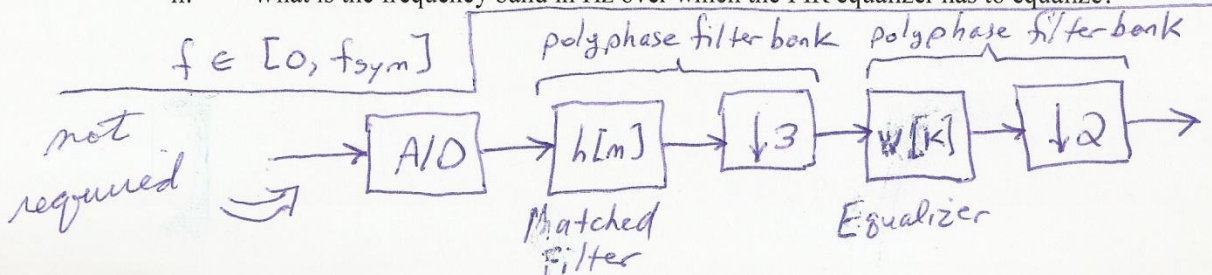


- d) The second step to reduce the inefficiency is to exchange the FIR equalizer with the downsampling by 3. 9 points.

- i. How can this exchange be justified?

Matched filter is a lowpass filter with bandwidth f_{sym} , which serves as anti-aliasing filter for downsampling by 3.

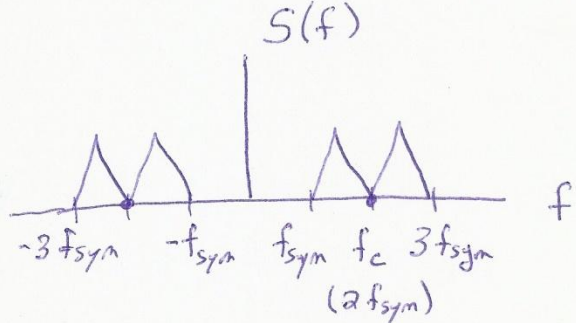
- ii. What is the frequency band in Hz over which the FIR equalizer has to equalize?



Problem 2.3. Pre-emphasis. 30 points.

Consider an upconverted baseband 2-PAM signal $x(t) = s(t) \cos(2\pi f_c t)$ where $s(t)$ is a baseband 2-PAM signal with

Constellation spacing	$2d$
Symbol rate	f_{sym}
Sampling rate	f_s
Samples per symbol	$L = 20$
Rolloff factor	$\alpha = 1$

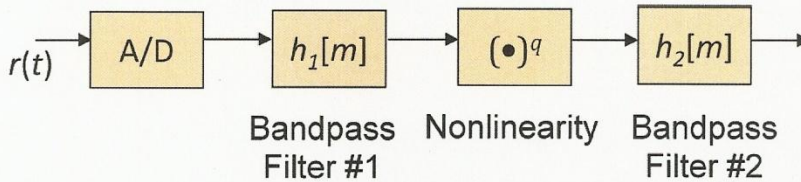


and where

Carrier frequency	$f_c = 2f_{sym}$
Transmission bandwidth	$B = 2f_{sym}$

The received signal is $r(t) = x(t) + n(t)$ where $n(t)$ is spectrally-flat Gaussian noise.

Here is the block diagram for pre-emphasis filtering where q is an integer and $q > 1$:



The nonlinearity raises the input to the q th power.

(a) Give the passband and stopband frequencies for bandpass filter (BPF) #1. 6 points.

This filter enforces the transmission band.

$f_{stop_1} = 0.9 f_{pass_1}$, $f_{pass_1} = f_{sym}$, $f_{pass_2} = 3 f_{sym}$, $f_{stop_2} = 1.1 f_{pass_2}$

(c) Pre-emphasis of carrier frequency f_c . 12 points.

i. give all possible values for q

q must be even. Highest frequency becomes $3q f_{sym}$. Aliasing if $q > 3$.

ii. which value of q would you use and why?

$q = 2$ for computational efficiency and less wordlength

Carrier frequency becomes $2q f_{sym}$. Aliasing if $q > 4$. q is 2 or 4.

iii. give the center frequency for BPF #2

$2q f_{sym}$ or $4 f_{sym}$ for $q = 2$.

(c) Pre-emphasis of symbol clock f_{sym} . 12 points.

i. give all possible values for q

q must be even. Symbol clock at $\frac{3}{2} q f_{sym}$.

ii. which value of q would you use and why?

$q = 2$ for computational efficiency

Aliasing if $q > 6$.

iii. give the center frequency for BPF #2

$\frac{3}{2} q f_{sym}$ or $3 f_{sym}$

for $q = 2$.

Symbol clock corresponds to frequency $\frac{1}{2} f_{sym}$ at baseband or $f_c \pm \frac{1}{2} f_{sym}$ for transmission

Let's lock onto $f_c - \frac{1}{2} f_{sym} = \frac{3}{2} f_{sym}$

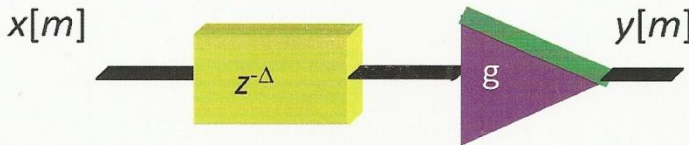
Problem 2.4. Potpourri. 18 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

- (a) Applying a lowpass filter to a spectrally-flat Gaussian noise signal always produces an output signal with lower average noise power than that of the input signal. *6 points.*

False. Applying a lowpass filter with bandwidth B to an input signal that is a spectrally-flat Gaussian noise signal with zero mean and variance σ^2 produces a Gaussian noise signal with zero mean and variance $2B\sigma^2$ (noise power).

- (b) In discrete time, an ideal channel can be modeled as

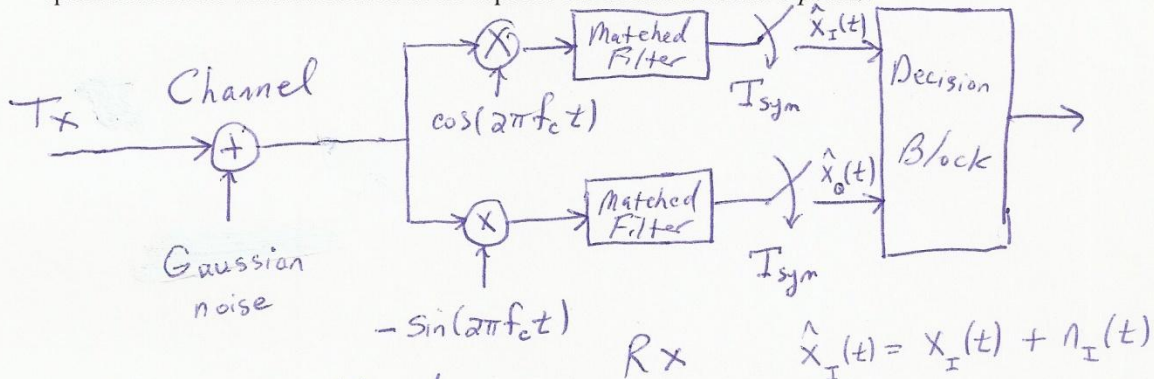


Noise power increases only when $B > \frac{1}{2}$.

The input $x[m]$ can always be exactly recovered by discarding the first Δ samples of $y[m]$ and scaling each subsequent sample by $1/g$. *6 points.*

False, for the case $g=0$, $\frac{1}{g}$ is undefined.
 or True, assuming that $\Delta \geq 0$ and $g \neq 0$ and $g \cdot \frac{1}{g} = 1$ to the precision of the arithmetic being used.

- (c) For a synchronized quadrature amplitude modulation (QAM) receiver and an additive spectrally-flat Gaussian noise channel, the in-phase noise will always be statistically independent of the quadrature noise when measured at the input to the decision block. *6 points.*



In-phase noise $n_I(t)$ and quadrature noise $n_Q(t)$ have the same source — Gaussian channel noise. They cannot be statistically independent.

$$\hat{X}_I(t) = X_I(t) + n_I(t)$$

$$\hat{X}_Q(t) = X_Q(t) + n_Q(t)$$

where $t = n T_{sym}$.

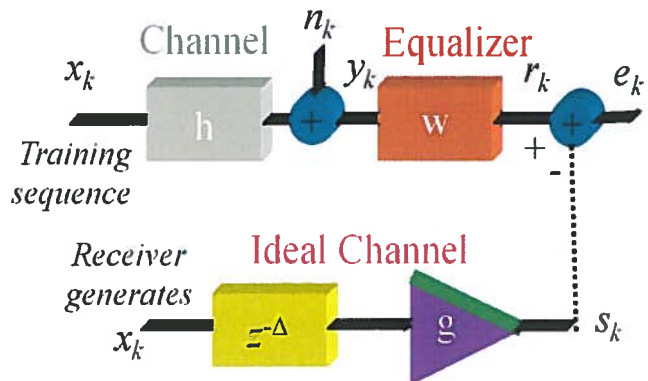
Problem 2.1. Channel Equalization. 27 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate.

The equalizer is a finite impulse response (FIR) filter with two real coefficients w_0 and w_1 :

$$r[k] = w_0 y[k] + w_1 y[k-1]$$

Channel model is an FIR filter with impulse response h in cascade with additive spectrally flat noise n_k .



(a) What training sequence would you use? Why? 6 points.

Maximal length pseudo-noise (PN) sequence. It's robust to frequency distortion and additive noise in the channel. It's easy to generate (feedback shift register plus exclusive or operations).

(b) Using your training sequence in part (a), describe how you would estimate the delay parameter Δ in the ideal channel model. 6 points.

The receiver would correlate the received signal y_k against the generated PN sequence x_k and take the location of the first peak to be the value of Δ .

(c) For an adaptive FIR equalizer, derive the update equation for w_1 for the objective function

$$J(e[k]) = e^2[k]. \quad 9 \text{ points.}$$

$$w_1[k+1] = w_1[k] - \mu \left. \frac{\partial J(e[k])}{\partial w_1} \right|_{w_1 = w_1[k]} \quad \begin{matrix} \# \\ \# \\ \# \\ \# \end{matrix} \quad \begin{matrix} e[k] = r[k] - g x[k-\Delta] \\ r[k] = w_0 y[k] + w_1 y[k-1] \end{matrix}$$

$$w_1[k+1] = w_1[k] - 2\mu e[k] y[k-1]$$

(d) Derive the values of the step size parameter μ that guarantees convergence of the adaptive algorithm? 6 points.

$$w_1[k+1] = f(w_1[k]) \text{ which has the form } v_{k+1} = f(v_k).$$

Convergence occurs if $|f'(v)| < 1$ for all $v \in [a, b]$,

where $[a, b]$ is the interval of all iteration values for v_k .

$$f'(w_1[k]) = 1 - 2\mu y^2[k-1]$$

$$|f'(w_1[k])| < 1 \implies |1 - 2\mu y^2[k-1]| < 1$$

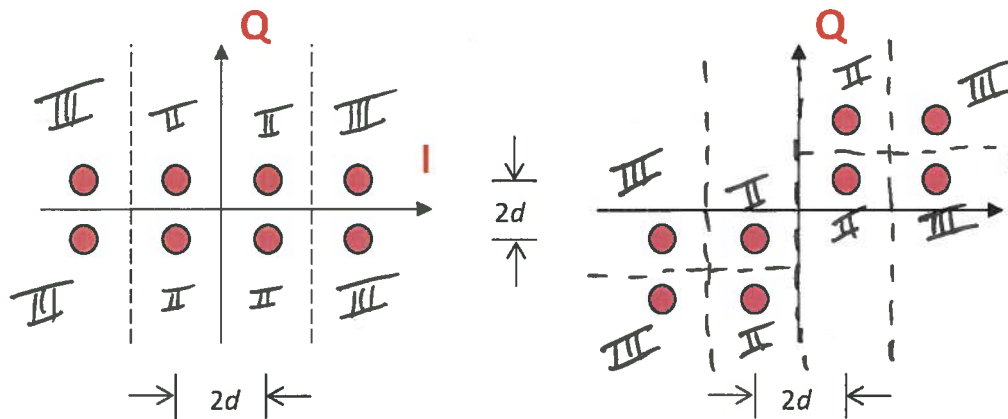
$$-2 < -2\mu y^2[k-1] < 0 \iff -1 < 1 - 2\mu y^2[k-1] < 1$$

$$0 < \mu < \frac{1}{\max_k y^2[k]}$$

$$0 < \mu < \frac{1}{y^2[k-1]}$$

Problem 2.2 Communication Performance. 27 points.

Consider the two 8-QAM constellations below. Constellation spacing is $2d$.



Power in the upper four points:

$$\begin{matrix} 10d^2 & 18d^2 \\ 2d^2 & 10d^2 \end{matrix}$$

Peak power = $18d^2$

Average power = $\frac{40d^2}{4} = 10d^2$

Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left shows the decision regions whose boundaries are the I axis, Q axis and dashed lines.

	Left Constellation	Right Constellation
(a) Peak power	$10d^2$	$18d^2$
(b) Average power	$6d^2$	$10d^2$
(c) Number of type I regions	0	0
(d) Number of type II regions	4	4
(e) Number of type III regions	4	4

→ Several possible correct answers.

Draw the decision regions for the right constellation on top of the right constellation. 3 points.

↳ Must be non-overlapping and cover the entire plane. Always have four type III regions (corners).
 Fill in each entry (a)-(e) for the right constellation. Each entry is worth 3 points.

Which of the two constellations would you advocate using? Why? 9 points.

The left constellation has several advantages over the one on the right. The left constellation

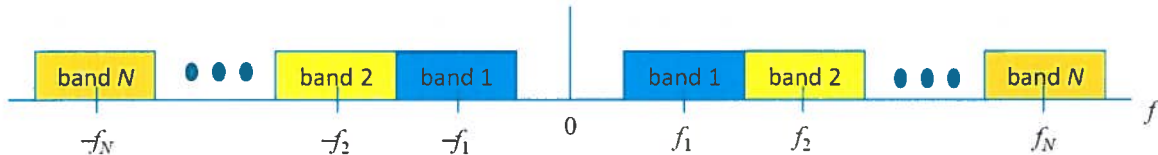
1. has lower peak power
2. has lower average power, and
3. has a Gray coding (the right one does not).

Problem 2.3. Multicarrier Communications. 24 points.

Multicarrier communications uses multiple carrier frequencies to transmit information in parallel.

Examples include IEEE 802.11a/g Wi-Fi, cellular LTE, DSL, and powerline communication systems.

In multicarrier communications, the transmission bandwidth W is divided into N equally spaced bands, shown below as band 1, band 2, ..., band N . A separate modulated signal is placed in each band.



The center frequency for each band is given as f_1, f_2, \dots, f_N .

- (a) Would you advocate for using pulse amplitude modulation (PAM) or quadrature amplitude modulation (QAM) in each band? Why? 6 points. Use QAM.

QAM would have a much better tradeoff in symbol error rate vs. SNR than PAM. QAM puts two orthogonal PAM signals in the same bandwidth as one PAM signal.

- (b) For each band, we can adapt the constellation size based on the signal-to-noise ratio (SNR) in that band. Based on your choice of modulation in (a), give a formula to determine the number of bits in a band based on the SNR measured in that band. 6 points.

$$\text{For PAM: } \text{SNR}_{dB} = \bar{C}_0 + 6B \Rightarrow B = \left\lfloor \frac{\text{SNR}_{dB} - \bar{C}_0}{6} \right\rfloor$$

$$\text{For QAM: } \text{SNR}_{dB} = C_0 + 3B \Rightarrow B = \left\lfloor \frac{\text{SNR}_{dB} - C_0}{3} \right\rfloor$$

Floor operation

- (c) For part (b), the SNR measurement for a particular band would be taken in the receiver at the equalizer output. From the block diagram of the channel equalizer in problem 2.1 on this test, give a formula to estimate the SNR at the equalizer output $r[k]$ for a given training signal $x[k]$. 9 points.

Signal is $x[k]$. Noise at the equalizer output is $r[k] - g x[k-\Delta]$.

$$\text{SNR} = \frac{\sum_k |x[k]|^2}{\sum_k |e[k]|^2}$$

- (d) If $f_1 = W/N$, give a formula for f_2 and f_N in terms of W and N . 3 points.

$$f_2 = 2 \frac{W}{N} \quad \text{and} \quad f_N = N \frac{W}{N} = W$$

Problem 2.4. Potpourri. 22 points.

Please determine whether the following claims are true or false. If you believe the claim to be false, then provide a **counterexample**. If you believe the claim to be true, then give **supporting evidence** that may include formulas and graphs as appropriate. If you give a true or false answer without any justification, then you will be awarded **zero points** for that answer. If you answer by simply rephrasing the claim, you will be awarded **zero points** for that answer.

(a) Adding noise to a system always reduces signal quality. 8 points. **False.**

For the case of analog-to-digital converters, one can add noise in the form of triangular pdf noise at the input to the quantizer to bury harmonics due to quantization in the noise. This allows better signal quality - sinusoidal input sounds

(b) Additive noise in a system is always spectrally flat. 7 points. **False.**

like the sinusoid in a bed of noise.

1. Counterexample #1: Consider additive spectrally-flat noise that models thermal noise in a system. Once it passes through a lowpass filter (e.g. a matched filter) the additive noise is lowpass.
2. Counterexample #2: As shown in the channel modeling lecture, additive noise may be narrowband or may have $e^{-\alpha f}$ shape, e.g. in

(c) The noise floor in a discrete-time digital system is always due to thermal noise. 7 points.

False. Quantization noise ^{power} can be greater than the thermal noise ^{power}. It depends on the number of bits used in quantization:

powerline communication channels.

$$SNR_{dB} = C_1 + 6B$$

When the quantization noise power is greater than the thermal noise power, the noise floor is due to the quantization noise.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 5, 2014

Course: EE 445S

Name: Asterix and Obelix
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all cell phones and other personal communication devices.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein textbook, the Welch, Wright and Morrow lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content you are using in your justification.

Problem	Point Value	Your score	Topic
1	21		Energy Detection
2	27		Communication Performance
3	30		Echo Cancellation
4	22		Transceiver Design
Total	100		

Problem 2.1. Energy Detection. 21 points.

Fall 2014

Energy detection is useful in a wide variety of signal processing applications.

Definition of energy for a causal discrete-time signal $r[m]$ follows:

$$\text{Energy} = \sum_{m=0}^{\infty} |r[m]|^2$$

Several linear time-invariant energy detectors are given below. Assume all initial conditions are zero.

For each energy detector, input $x[m]$ is the instantaneous power $|r[m]|^2$. Output is $y[m]$.

Give the transfer function for each system and give the primary advantage and disadvantage of each.

(a) Running sum: $y[m] = y[m-1] + x[m]$. 7 points.

$$Y(z) = z^{-1} Y(z) + X(z)$$

$$(1 - z^{-1}) Y(z) = X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - z^{-1}}$$

Advantage

- Computationally simple (1 add/sample)
- Little memory usage (1 word)

Disadvantage

- BIBO Unstable
- Same as (c)

(b) Sum of current input and previous $M-1$ inputs: $y[m] = x[m] + x[m-1] + \dots + x[m-(M-1)]$. 7 points.

$$y[m] = \sum_{n=0}^{M-1} x[m-n]$$

$$Y(z) = \sum_{n=0}^{M-1} z^{-n} X(z)$$

$$H(z) = 1 + z^{-1} + \dots + z^{-(M-1)}$$

- BIBO Stable

- More Complex Implementation (M adds/sample and M words of memory) vs. (a)

(c) Weighted combination using constant c where $0 < c < 1$: $y[m] = c y[m-1] + (1-c) x[m]$.

$$Y(z) = c z^{-1} Y(z) + (1-c) X(z)$$

$$(1 - c z^{-1}) Y(z) = (1-c) X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1-c}{1 - c z^{-1}}$$

- BIBO Stable
- Good tradeoff in complexity (2 mults/sample, 1 add/sample, 3 words memory)

- Number of bits needed for $y[m]$ grows without bound as $m \rightarrow \infty$ and no clear way to reset the system

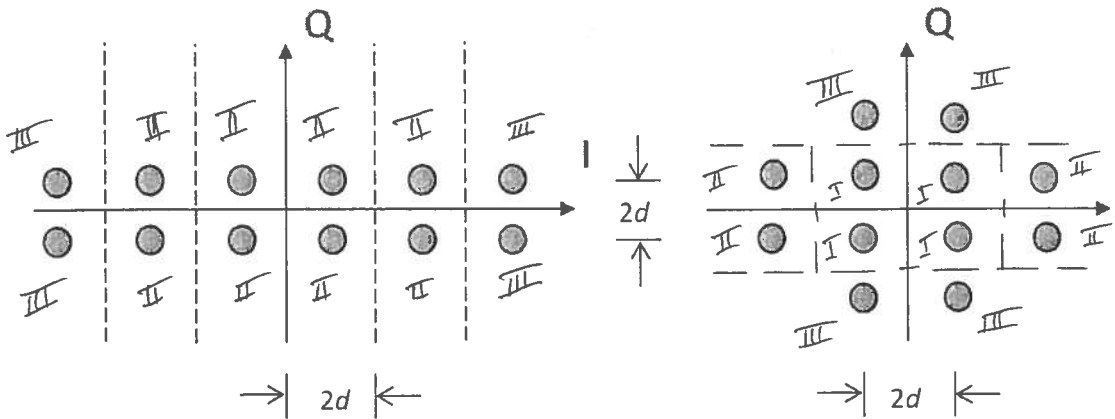
Note: Constant c is real-valued.

All three LTI systems smooth the instantaneous power calculation of $|r[m]|^2$. LTI systems (b) and (c) are filters.

Problem 2.2 Communication Performance. 27 points.

Fall 2014

Consider the two 12-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the I axis, Q axis and dashed lines.

	Left Constellation	Right Constellation
(a) Peak power	$26 d^2$	$10 d^2$
(b) Average power	$12.67 d^2$	$7.33 d^2$
(c) Number of type I regions	0	4
(d) Number of type II regions	8	4
(e) Number of type III regions	4	4

Draw the decision regions for the right constellation on top of the right constellation. 3 points.

The I axis and Q axis are also boundaries of decision regions.

Fill in each entry (a)-(e) in the above table for the right constellation. Each entry is worth 3 points.

Due to symmetry, we can compute the average power in one quadrant:

Which of the two constellations would you advocate using? Why? 9 points.

The right constellation has

- Lower peak power
- Lower average power
- Lower peak power to average power ratio

$$\frac{2d^2 + 10d^2 + 10d^2}{3} = 7.33d^2$$

The left constellation has a lower probability of symbol error as a function of d . Once we normalize in terms of SNR, the right constellation will have a lower probability of symbol error. Both constellations can be gray coded.

Correction: Only the left constellation can be gray coded.

Fall 2014

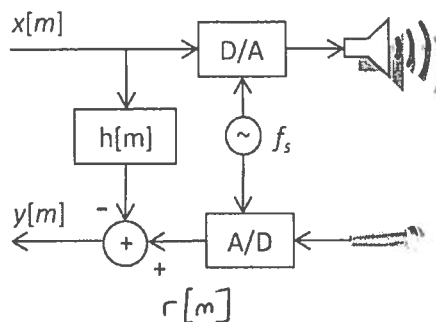
Problem 2.3. Echo Cancellation. 30 points.

A block diagram of a speakerphone is shown to the right.

During a phone call, the received speech plays out over the speaker. The sound from the speaker is captured by the microphone, and the caller will hear an echo of his/her voice.

$x[m]$ is a reference signal. It is either the digitized speech from the other person on the call or a training signal.

$h[m]$ is the impulse response of a finite impulse response filter.



The analog-to-digital (A/D) and digital-to-analog (D/A) converters are synchronized by use of a common sampling clock. The A/D and D/A converters quantize to B bits per sample.

Design the finite impulse response (FIR) filter to reduce echo by using an adaptive method.

(a) Give a reference signal for $x[m]$ to use for training $h[m]$. Why did you choose it? 6 points.

Use a maximal length pseudo-noise sequence. This sequence is robust to frequency distortion and additive noise, and is easy to generate using a shift register and exclusive-OR operations.

(b) During training, the ideal value for $y[m]$ is 0, which would mean that all echo has been removed.

i. Give an objective function $J(y[m])$ to be minimized. 9 points.

$$J(y[m]) = \frac{1}{2} y^2[m]$$

We seek to minimize $J(y[m])$ so that its minimum value corresponds to all echo being removed.

ii. Give the update equation for the vector \vec{h} of FIR coefficients. 9 points.

$$\vec{h} = [h_0 \quad h_1 \quad h_2 \quad \dots \quad h_{m-1}]$$

$$\vec{h}[m] = [h_0[m] \quad h_1[m] \quad h_2[m] \quad \dots \quad h_{m-1}[m]]$$

$$\vec{h}[m+1] = \vec{h}[m] - \mu \left. \frac{J(y[m])}{d\vec{h}} \right|_{\vec{h}=\vec{h}[m]} \quad \checkmark$$

iii. What values would you recommend for the step size μ ? 6 points.

- $\mu > 0$ and very small based on observations in homework problems (e.g. $\mu = 0.001$).
- Reusing the solution from midterm problem 2.1 in spring 2014:

$$0 < \mu < \frac{2}{x^2[m]}$$

$$\begin{aligned} y[m] &= r[m] - x[m] * h[m] \\ &= r[m] - \\ &\quad h_0 x[m] - \\ &\quad h_1 x[m-1] \dots \end{aligned}$$

$$\vec{h}[m+1] = \vec{h}[m] + \mu y[m] \vec{x}[m]$$

$$\vec{x}[m] = [x[m] \quad x[m-1] \quad \dots]$$

Spring 2014 objective function was $y^2[m]$; replace 2 u in spring 2014 solution with u

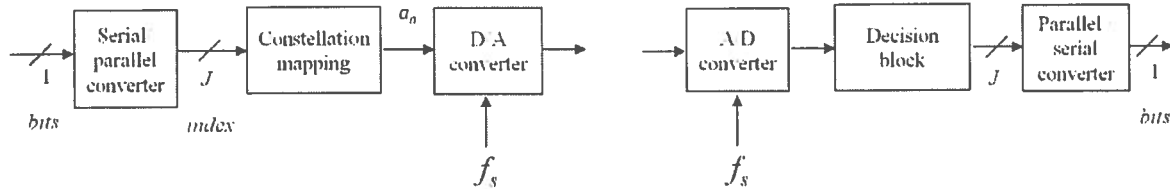
Problem 2.4. Transceiver Design. 22 points.

Fall 2014

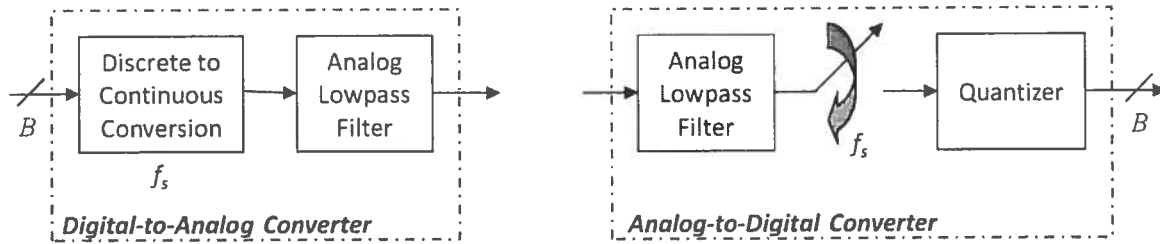
In certain discrete-time baseband transceivers, we remove the pulse shaping filter in the transmitter and the matched filter in the receiver to reduce complexity.

We also remove the upsampler in the transmitter and downsampler in the receiver.

Here is the resulting transmitter (left) and receiver (right) for baseband pulse amplitude modulation:



Here are block diagrams for the analog-to-digital (A/D) and the digital-to-analog (D/A) converters:



(a) What is the formula relating the symbol rate f_{sym} and the sampling rate f_s ? 4 points

$$f_s = f_{sym}$$

Alternate answer to (b)-i: The discrete-to-continuous conversion block modulates its input with an impulse train with impulses separated by T_s and then applies an interpolating (pulse-shaping) filter.

(b) Consider a channel model of only additive spectrally-flat Gaussian noise.

i. Which block in the block diagrams acts as a pulse shaping filter? In what way? 4 points.
If the analog lowpass filter in the A/D converter is the matched filter, then the analog lowpass filter in the D/A converter is the pulse shaping filter.

ii. Which block in the block diagrams acts as a matched filter? In what way? 4 points.
The matched filter increases the SNR at the input to the decision block, primarily by attenuating out-of-band noise. This role is played by the

iii. How close does the matched filter you identified in part (b)-ii above come to the optimal analog matched filter? 4 points. The optimal matched filter has an impulse response of $h_{opt}(t) = k g_I^*(T_{sym} - t)$ where $g_I(t)$ is the pulse shape. $|H_{opt}(f)| = |G_I(f)|$. The analog lowpass filter in the A/D converter in the receiver.

(c) Consider a binary phase shift keying (BPSK) system, a.k.a. two-level Pulse Amplitude Modulation, with symbol amplitudes of $-d$ and $+d$. Give a formula for d . Why? 6 points.

We would like to have as much transmit power as possible; hence, we would like d to be as large as possible.

$$d = 2^{B-1} - 1$$

B is the number of bits in the D/A converter in the transmitter

filters in the A/D converter and the D/A converter have the same magnitude design specification.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: May 8, 2015

Course: EE 445S

Name: _____ **House,** **Full**
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

	Problem	Point Value	Your score	Topic
<i>Danny</i>	1	21		Equalizer Design
<i>D.J.</i>	2	27		QAM Communication Performance
<i>Stephanie</i>	3	30		Automatic Gain Control
<i>Michelle</i>	4	22		Data Converter Design
	Total	100		

Full House is a TV show.

Problem 2.1. Equalizer Design. 21 points.

This problem asks you to design an equalizer to compensate for the magnitude and phase distortion of a discrete-time linear time-invariant (LTI) system.

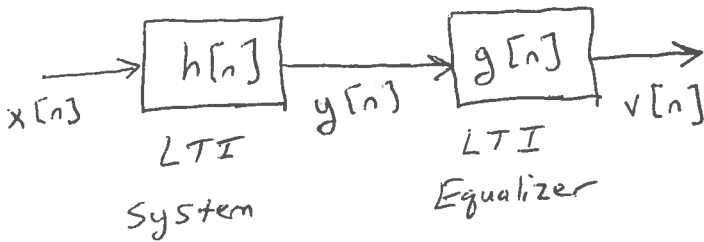
(a) Describe how you would estimate the impulse response of the discrete-time LTI system. 6 points.

A discrete-time LTI system is uniquely characterized by its impulse response, $h[n]$.



For an unknown $h[n]$, input a maximal-length pseudo-noise sequence $x[n]$ and observe $y[n]$: and either backsolve for $h[n]$ in $y[n] = x[n] * h[n]$ or compute $H(\omega) = \frac{Y(\omega)}{X(\omega)}$ and take the inverse discrete-time Fourier transform.

(b) For a discrete-time LTI system with impulse response $h[n] = \delta[n] - a\delta[n-1]$ where a is a real number, design a stable discrete-time LTI equalizer so that the impulse response of the cascade of the discretized channel and equalizer yields a delayed impulse. Your approach must handle all possible values of a . 15 points.



A maximal-length pseudo-noise sequence has all frequencies in it, and hence $X(\omega)$ never equals zero.

We want the cascade of $h[n]$ and $g[n]$ to be an all-pass

LTI system: $h[n] * g[n] = C_0 \delta[n - n_0]$
constant gain constant delay

$$H(z)G(z) = C_0 z^{-n_0}$$

$$G(z) = C_0 \frac{z^{-n_0}}{H(z)}; \quad H(z) = 1 - az^{-1}$$

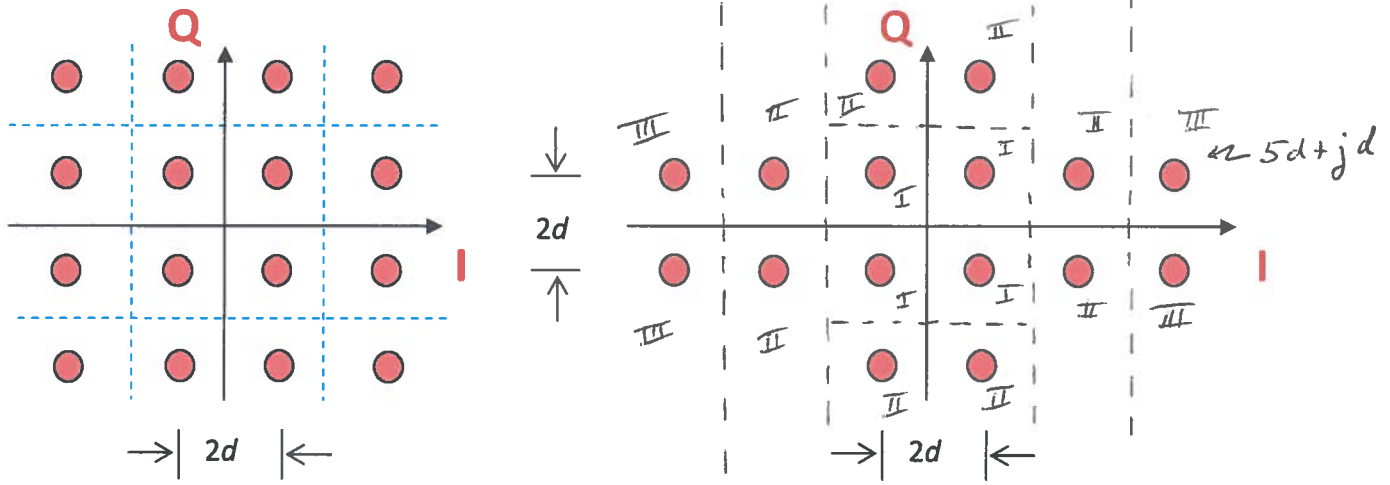
Case I: $|a| < 1$. Use pole-zero cancellation for cascade of $H(z)$ and $G(z)$: $G(z) = \frac{C_0 z^{-n_0}}{1 - az^{-1}}$

Case II: $|a| = 1$. Zero of $H(z)$ is on the unit circle; hence, the frequency of the zero is eliminated and cannot be recovered. Use notch configuration: $G(z) = \frac{C_0 z^{-n_0}}{1 - 0.95 \operatorname{sgn}(a) z^{-1}}$

Case III: $|a| > 1$. Use all-pass configuration for cascade of $H(z)$ and $G(z)$: $G(z) = \frac{C_0 z^{-n_0}}{1 - \frac{1}{a} z^{-1}}$

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$26d^2$
(b) Average transmit power	$10d^2$	$12d^2$
(c) Number of type I regions	4	4
(d) Number of type II regions	8	8
(e) Number of type III regions	4	4
(f) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$

Draw the decision regions for the right constellation on top of the right constellation. 3 points.

I and Q axes are also decision boundaries.

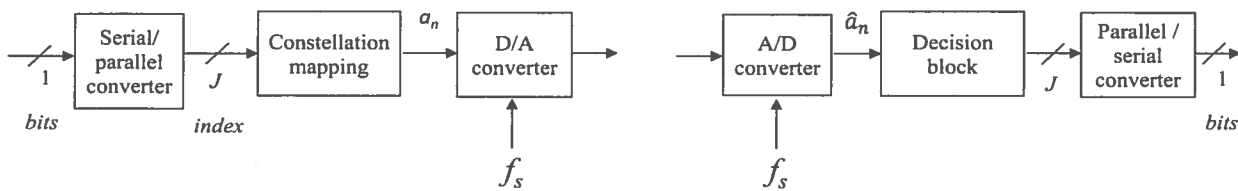
Fill in each entry (a)-(f) in the above table for the right constellation. Each entry is worth 3 points.

Which of the two constellations would you advocate using? Why? 6 points.

The left constellation is the better choice because it has

1. Lower peak transmit power
2. Lower average transmit power
3. Lower peak-to-average power ratio
4. Lower probability of symbol error vs. SNR
5. Gray coding whereas the right one does not.

Problem 2.3. Automatic Gain Control. 30 points. Using decision-directed feedback.
 Consider the simplified transmitter (left) and receiver (right) for baseband pulse amplitude modulation:



System uses J bits per symbol and a constellation spacing of $2d$ in units of Volts.

Your goal is to design an automatic gain control system for the receiver to compensate for fading:

- Fading is modeled as an unknown time-varying gain $g(t)$ or $g[n]$.
- The decision block will feed back the following signal to the automatic gain control system

$$v[n] = \hat{a}_n - a_n$$
 where \hat{a}_n is the received symbol amplitude and a_n is the transmitted symbol amplitude.
- The automatic gain control system will adapt its gain $c[n]$ so that estimated symbol amplitudes will become closer in value to the transmitter symbol amplitudes over time.

(a) Determine an objective function $J(v[n])$. 6 points.

$$J(v[n]) = \frac{1}{2} v^2[n]$$

$$\begin{aligned} v[n] &= \hat{a}_n - a_n \\ \hat{a}_n &= g[n] c[n] a_n \\ v[n] &= (g[n] c[n] - 1) a_n \end{aligned}$$

(b) Based on your objective function in (a), derive an update equation to adapt $c[n]$. 9 points.

$$c[n+1] = c[n] - \bar{\mu} \frac{dJ(v[n])}{dc[n]} \quad \frac{dJ(v[n])}{dc[n]} = g[n] a_n v[n]$$

$$c[n+1] = c[n] - \mu v[n] a_n$$

where $\mu = g[n] \bar{\mu}$ but μ is treated as a constant. ~~unknown~~

(c) For the answer in (b), what value of the step size would you recommend? Why? 3 points.

We want $0 < \bar{\mu} < 1$. In practice, keep $\bar{\mu}$ small, e.g. 0.001.

Then, μ is the minimum value of $|g[n]|$ times $\bar{\mu}$.

(d) Propose an algorithm to find an initial accurate value of $c[n]$. 6 points

1. Use a training sequence so that a_n is known to the receiver and apply the above algorithm, or

2. Use the automatic gain control method from QAM receiver lecture:

$$c[n] = (1 + 2f_0 - f_{\max} - f_{\min}) c[n-1] \text{ where } f_0 \text{ is frequency of } a_n = 0,$$

(e) Modify your update equation for $c[n]$ in (b) to improve convergence for $c[n]$ when $g[n]$ is varying quickly with time. 6 points.

1. Use a smaller step size, or

2. Replace $v[n]$ with an average of the current $v[n]$ and its previous M values (via a lowpass filtering of $v[n]$).

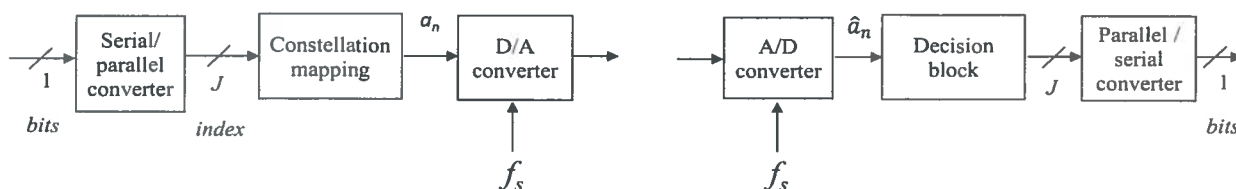
and f_{\max} is frequency of $a_n = 2^{J-1} - 1$,
 and f_{\min} is frequency of $a_n = -2^{J-1}$.

$$\text{Initial values: } f_0 = f_{\min} = f_{\max} = \frac{1}{2^J}$$

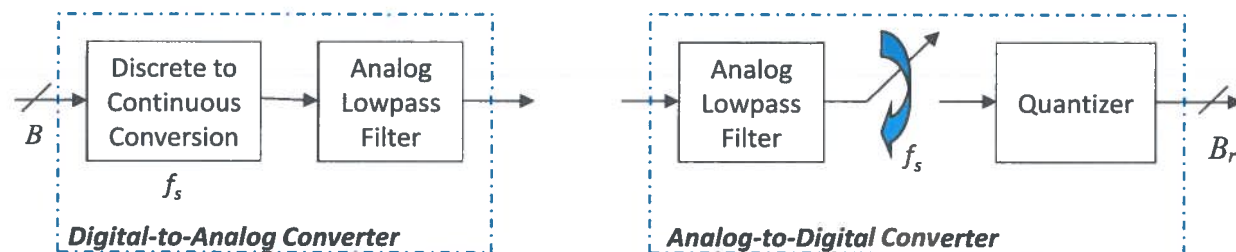
Note that $N_g = 1$ symbols/pulse
and $L = 1$ sample/symbol.

Problem 2.4. Data Converter Design. 22 points

Consider the simplified transmitter (left) and receiver (right) for baseband pulse amplitude modulation



Here are block diagrams for the analog-to-digital (A/D) and the digital-to-analog (D/A) converters:



Communication system uses J bits per symbol and a constellation spacing of $2d$ in units of Volts.

The channel model consists of additive spectrally-flat Gaussian noise with zero mean and variance σ^2 .

(a) In the transmitter, what is the smallest number of bits B needed for the D/A Converter? 6 points.

$B = J$

Note: In this case, each constellation point is separated by one quantization level, and $2d$ is the step size: $\frac{V_{max} - V_{min}}{\text{Levels} - 1}$ where Levels = 2^J .

(b) In the transmitter, what is the second smallest number of bits that could be used for the D/A Converter? 6 points.

$B = J + 1$

Note: In this case, each constellation point is separated by two quantization levels.

(c) In the receiver, what is the minimum number of bits B_r needed for the A/D Converter so that the quantization noise power at the quantizer output in the A/D Converter is less than or equal to the system noise power at the quantizer input? 10 points.

$\sigma^2 \geq \sigma_Q^2$ where $\sigma_Q^2 = \frac{1}{3} V_{max}^2 2^{-2B_r}$ on slide 8-13.

System noise power Quantization noise power

$$\sigma^2 \geq \frac{1}{3} V_{max}^2 2^{-2B_r} \Rightarrow \frac{3\sigma^2}{V_{max}^2} \geq 2^{-2B_r} \Rightarrow \log_2 \frac{3\sigma^2}{V_{max}^2} \geq -2B_r$$

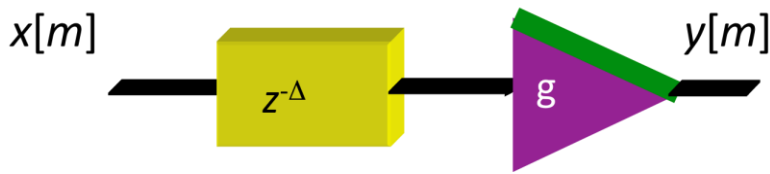
non-negative non-negative

$$B_r \geq \left\lceil \log_2(V_{max}) - \log_2(\sqrt{3}\sigma) \right\rceil$$

ceiling operation

Problem 2.1. Ideal Channel Model. 21 points.

Consider the following block diagram of an ideal channel model:



Assume that the delay Δ is positive and the gain g is not zero.

- (a) Give an algorithm to recover $x[m]$ from $y[m]$ assuming that the values of Δ and g are known. *6 points.*

$$y[m] = g x[m-\Delta] \text{ or equivalently } x[m] = (1/g) y[m+\Delta]$$

Discard the first Δ samples of $y[m]$ and scale by $1/g$

- (b) For a pseudo-noise training sequence known to the transmitter and receiver, give an algorithm for the receiver to estimate the delay Δ and gain g . *9 points.*

Assume that pseudo-noise training sequence $x[m]$ has values of +1 and -1.

Correlate $y[m]$ with $x[m]$.

Location of the first peak gives Δ .

Discard the first Δ samples of $y[m]$ to obtain $y_I[m]$.

Compute g as the average value of $|y_I[m]|$ divided by the average value of $|x[m]|$.

Using averaging of all samples gives a more accurate estimate the ideal channel model for an actual physical channel.

- (c) Given a sequence of -1 and +1 values, how could you verify whether or not it is a maximal length pseudo-noise sequence? *6 points.*

The sequence length N must be $2^r - 1$ where r is an integer and $r > 1$

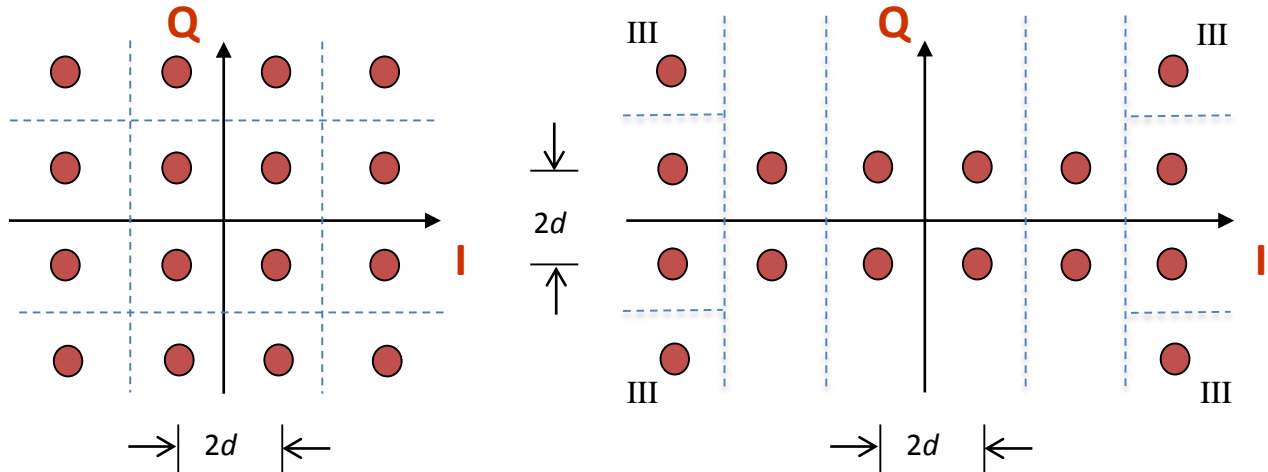
The normalized autocorrelation $R[m]$ of the sequence is 1 at the origin and $-1/N$ otherwise.

$$R[m] = \frac{1}{N} \sum_{k=0}^{N-1} x[k]x[m+k]$$

The magnitude of the discrete Fourier transform of the sequence is constant except at DC where it has a much smaller value of 1.

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

	Left Constellation	Right Constellation
(a) Peak transmit power	$18 d^2$	$34 d^2$
(b) Average transmit power	$10 d^2$	$18 d^2$
(c) Number of type I regions	4	0
(d) Number of type II regions	8	12
(e) Number of type III regions	4	4
(f) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$\frac{11}{4}Q\left(\frac{d}{\sigma}\right) - \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$

Draw the decision regions for the right constellation on top of the right constellation. 3 points.
The I and Q axes are decision boundaries. The decision regions must cover the entire I-Q plane.

Fill in each entry (a)-(f) in the above table for the right constellation. Each entry is worth 3 points.

Due to symmetry in the constellation, one can compute parts (a) and (b) from one quadrant.

Transmit power for each constellation point is $2 d^2$, $10 d^2$, $26 d^2$ and $34 d^2$.

Peak power is $34 d^2$ and average power is $18 d^2$.

For part (f), $P(\text{correct}) = 1 - P(\text{error})$ and

$$P(\text{error}) = \frac{12}{16} \left(1 - Q\left(\frac{d}{\sigma}\right)\right) \left(1 - 2Q\left(\frac{d}{\sigma}\right)\right) + \frac{4}{16} \left(1 - Q\left(\frac{d}{\sigma}\right)\right)^2 = 1 - \frac{11}{4}Q\left(\frac{d}{\sigma}\right) + \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$$

Which of the two constellations would you advocate using? Why? Please give at least two reasons. 6 points. The left constellation is the better choice because it has

Lower peak transmit power

Lower average transmit power

Lower peak-to-average transmit power

Lower probability of symbol error vs. signal-to-noise ratio (SNR)

Gray coding whereas the right constellation does not

Problem 2.3. Narrowband Interference. 28 points.

Consider a baseband pulse amplitude modulation communication system in which the narrowband interference is stronger than the transmitted signal and additive noise at the receiver input.

Design a causal second-order adaptive infinite impulse response (IIR) filter to remove the interference:

- Zero locations are at $\exp(j\omega_0)$ and $\exp(-j\omega_0)$
- Pole locations are at $r \exp(j\omega_0)$ and $r \exp(-j\omega_0)$

Relationship between input $x[m]$ and output $y[m]$ is

$$y[m] = x[m] - (2 \cos \omega_0) x[m-1] + x[m-2] + (2 r \cos \omega_0) y[m-1] - r^2 y[m-2]$$

To guarantee stability, we will set the pole radius r to a constant value so that $0 < r < 1$.

We will adapt the frequency location of the notch, ω_0 .

- (a) Determine an objective function $J(y[m])$. 6 points.

To minimize the power of the narrowband interferer, let $J(y[m]) = \frac{1}{2} y^2[m]$

- (b) What initial value of ω_0 would you use? Why? 6 points

If the narrowband interferer were outside the transmission band, then the matched filter in the receiver would attenuate it. Let the initial value of ω_0 be in the middle of the transmission band in order to quickly adapt it to the location of the interferer.

- (c) Compute the partial derivative of $y[m]$ with respect to ω_0 . You may assume that the partial derivative of $y[m]$ with respect to ω_0 is 0 for $m < 0$. 6 points.

System is causal, so $x[m] = 0$ for $m < 0$ and $y[m] = 0$ for $m < 0$.

$$y[m] = x[m] - (2 \cos \omega_0) x[m-1] + x[m-2] + (2 r \cos \omega_0) y[m-1] - r^2 y[m-2]$$

$$y[m-1] = x[m-1] - (2 \cos \omega_0) x[m-2] + x[m-3] + 2 r \cos \omega_0 y[m-2] - r^2 y[m-3]$$

$$y[m-2] = x[m-2] - (2 \cos \omega_0) x[m-3] + x[m-4] + (2 r \cos \omega_0) y[m-3] - r^2 y[m-4]$$

$$\frac{dy[m]}{d\omega_0} = 2 \sin(\omega_0) x[m-1] - 2 r \sin(\omega_0) y[m-1] + (2 r \cos(\omega_0)) \frac{dy[m-1]}{d\omega_0} - r^2 \frac{dy[m-2]}{d\omega_0}$$

Let $v[m] = \frac{dy[m]}{d\omega_0}$ where $v[-1] = 0$ and $v[-2] = 0$,

$$v[m] = 2 \sin(\omega_0) x[m-1] - 2 r \sin(\omega_0) y[m-1] + 2 r \cos(\omega_0) v[m-1] - r^2 v[m-2]$$

- (d) Based on your answers in (a), (b), and (c), derive an update equation to adapt ω_0 . 6 points.

$$\omega_0[m+1] = \omega_0[m] - \mu \left. \frac{dJ(y[m])}{d\omega_0} \right|_{\omega_0=\omega_0[m]} = \omega_0[m] - \mu y[m] \left. \frac{dy[m]}{d\omega_0} \right|_{\omega_0=\omega_0[m]}$$

$$\omega_0[m+1] = \omega_0[m] - \mu y[m] v[m]$$

$$y[m] = x[m] - 2 \cos(\omega_0[m]) x[m-1] + x[m-2] + 2 r \cos(\omega_0[m]) y[m-1] - r^2 y[m-2]$$

$$v[m] = 2 \sin(\omega_0[m]) x[m-1] - 2 r \sin(\omega_0[m]) y[m-1] + 2 r \cos(\omega_0[m]) v[m-1] - r^2 v[m-2]$$

- (e) For the answer in (d), what value of the step size would you recommend? Why? 4 points.

We need a very small $\mu > 0$, e.g. $\mu = 0.001$, to ensure that the adaptation converges.

Or apply fixed-point theorem to $\omega_0[m+1] = f(\omega_0[m])$ and solve for μ for $|f'(\omega_0[m])| < 1$

Simulation of Problem 2.3 is provided below for additional information and insight. A Matlab simulation was neither required nor expected to answer this or any other problem on the test.

Using the Matlab code below to simulate the solution to problem 2.3, we generate 10,000 samples of a narrowband interferer centered at a discrete-time frequency of 0.65π rad/sample for $x[m]$ and pass it through the adaptive IIR notch filter to generate $y[m]$. We set $\mu = 0.001$. The adaptive IIR notch filter properly adapts the notch frequency ω_0 from 0.5π rad/sample to 0.65π rad/sample (about 2.04). After the adaptive IIR notch filter converges, the reduction in average power was about 240 dB. Average power in $x[m]$ is 5×10^{-5} , and average power in $y[m]$ after sample index 3000 is 4.1×10^{-29} .

```

%%% Adaptive IIR Notch Filter
%%% Date: December 7, 2015
%%% Programmer: Prof. Brian L. Evans
%%% Affiliation: The University of Texas at Austin

%%% Generate narrowband interferer
w0 = 0.65*pi;
mmax = 10000;
mmax = mmax + 2;          %%% two init conditions
mindex = 3 : mmax;
x = [0 0 cos(w0*mindex)]; %%% two init conditions

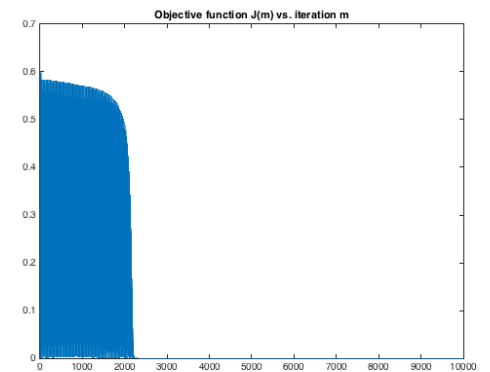
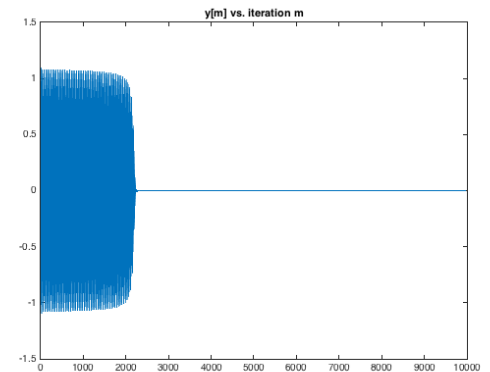
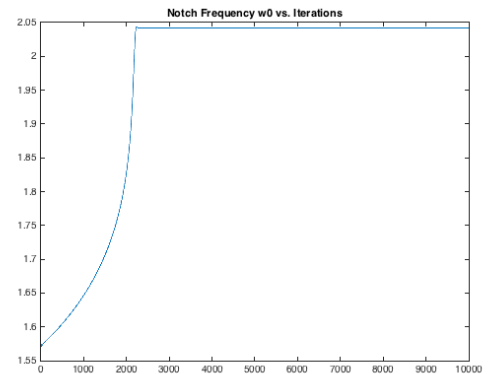
%%% IIR Notch Filter with input x(m) and output y(m)
%%% notch frequency is w0
%%% zeros: exp(j w0) and exp(-j w0)
%%% poles: r exp(j w0) and r exp(-j w0)
w0 = 0.5*pi;
r = 0.9;
y = zeros(1, length(x)); %%% two init conditions

%%% Settings for adaptive update
mu = 0.001;
v = zeros(1, length(x));
w0vector = zeros(1, length(x));
Jvector = zeros(1, length(x));

for m = 3 : mmax
    y(m) = x(m) - 2*cos(w0)*x(m-1) + x(m-2) + ...
           2*r*cos(w0)*y(m-1) - r^2*y(m-2);
    v(m) = 2*sin(w0)*x(m-1) - 2*r*sin(w0)*y(m-1) + ...
           2*r*cos(w0)*v(m-1) - r^2*v(m-2);
    w0 = w0 - mu*y(m)*v(m);

    %%% Storage of values for w0 and objective fun
    w0vector(m) = w0;
    Jvector(m) = 0.5*(y(m)^2);
end

```



The adaptive IIR notch filter converged in about 300 iterations when $\mu = 0.01$.

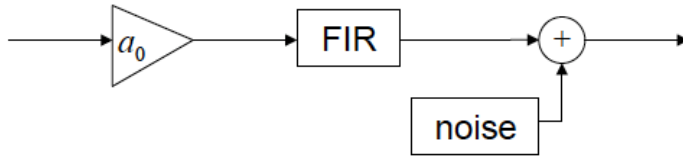
The above Matlab code is available online at

<http://users.ece.utexas.edu/~bevans/courses/realtime/lectures/AdaptiveIIRNotchFilter.m>

Problem 2.4. Potpourri. 24 points

In a communication receiver, a finite impulse response (FIR) channel equalizer may be designed by various methods. For this problem, the channel equalizer will operate at the sampling rate.

Consider the following channel model:



Here, a_0 represents time-varying fading gain and the FIR filter models the channel impulse response.

(a) Describe why the channel impulse response is modeled by a finite impulse response. 6 points.

Wireless channels – **Model reflection and absorption of propagating electromagnetic waves in air. Truncate infinite impulse response after significant decay has occurred.**

Wired channels – **Model transmission line as RLC circuit. Truncate the infinite impulse response after significant decay has occurred.**

(b) Describe what a channel equalizer tries to do. 6 points.

Time domain – **Shortens channel impulse response to reduce intersymbol interference. Impulse response of the cascade of the channel and the channel equalizer would ideally be a single impulse at index Δ with gain g . See the ideal channel model in problem 2.1.**

Frequency domain – **Compensates for frequency distortion in the channel. Frequency response of the cascade of the channel and the channel equalizer would ideally be allpass.**

(c) When the transmitter is transmitting a known training sequence, would you recommend using a least squares method or an adaptive least mean squares method for the channel equalizer? Please justify your answer for each criterion below. 6 points. **Adaptive LMS method.**

Communication performance – **The channel model has time-varying fading gain. The adaptive LMS equalizer tracks the channel during training. LS equalizer gives best average equalizer over the training sequence and may be mismatched to the current state of the channel.**

Implementation complexity – **The adaptive LMS method is based on vector additions and scalar-vector multiplications, whereas the LS equalizer is based on matrix multiplication and inversion. The adaptive LMS method requires less computation and far less memory.**

(d) If the noise contained a narrowband interferer in the transmission band, would a separate notch filter be needed in addition to the channel equalizer? Why or why not? 6 points.

An adaptive LMS equalizer or an LS equalizer seeks to make the cascade of the channel and the equalizer have an allpass frequency response. The equalizer will seek to equalize not only the channel impulse response but fading, noise and interference. Hence, the equalizer will try to notch out narrowband interference; however, because it is an FIR filter, the equalizer's notch will only have a mild reduction when compared to an IIR notch filter. See next page.

Note: Certain multicarrier systems will simply not transmit data over parts of the transmission band corrupted by narrowband interference. This is known as interference avoidance.

Simulation of Problem 2.4(d) provided below for additional information and insight. A Matlab simulation was neither required nor expected to answer this or any other problem on the test.

To simulate problem 2.4(d), we can add a narrowband interferer to the channel equalizer design problems in homework assignment 7.1 (least squares equalizer) and 7.2 (adaptive least mean squares equalizer) from the spring 2014 version of the course:

<http://users.ece.utexas.edu/~bevans/courses/rtdsp/homework/solution7.pdf>

We add the following code to add a sinusoidal narrowband interferer at 0.65π rad/sample:

```
w0 = 0.65*pi;
samples = length(r);
nindex = 0 : (samples-1);
r = r + cos(w0*nindex);
```

after the line

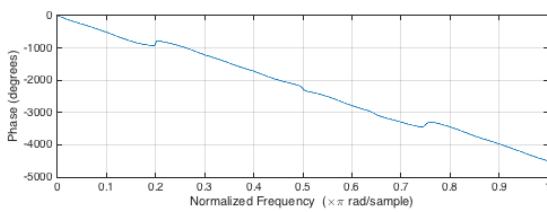
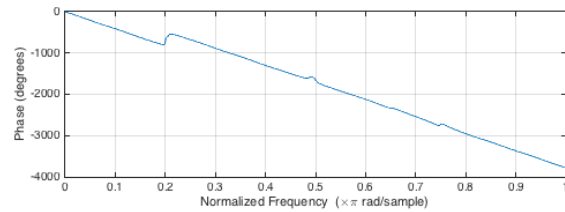
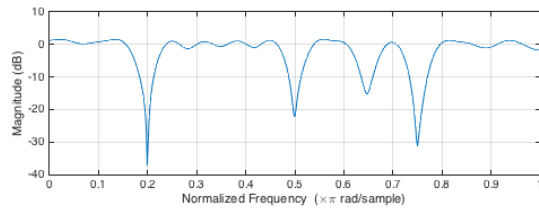
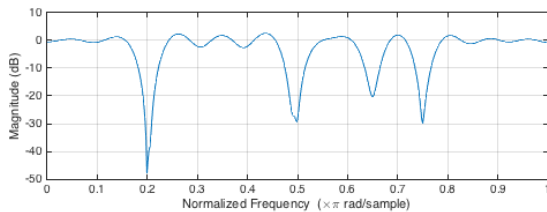
```
r=filter(b,1,s); % output of channel
```

The average power in the narrowband interferer is one-seventh of the average power of the transmitted signal after passing through the FIR filter in the channel model.

The best LS equalizer has a length of 40 and delay $\Delta = 23$, and gave 46 symbol (bit) errors. Without the narrowband interferer, there are 0 bit errors.

The best adaptive LMS equalizer has a length of 40 and delay $\Delta = 29$, and gave 128 symbol (bit) errors. Without the narrowband interferer, there are 0 bit errors.

Here are the magnitude responses of the equalized channels. Please note the notches at the narrowband frequency of 0.65π rad/sample:



LS equalized channel
(20 dB notch at 0.65π)

Adaptive LMS equalized channel
(15 dB notch at 0.65π)

An IIR notch filter can deliver 50 dB (or more) of attenuation at the narrowband frequency. See the simulations for an adaptive IIR notch filter in problem 2.3.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: May 6, 2016

Course: EE 445S

Name: _____ **Scooby-Doo** _____
Last, First

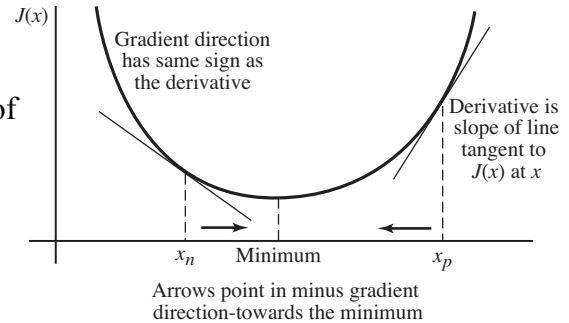
- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Shaggy
Velma
Daphne
Fred

Problem	Point Value	Your score	Topic
1	21		Steepest Descent Algorithm
2	27		QAM Communication Performance
3	28		Estimating SNR at a Receiver
4	24		Acoustics of a Concert Hall
Total	100		

Problem 2.1. Steepest Descent Algorithm. 21 points.

The steepest descent algorithm seeks to find a minimum value of an objective function by descending into a valley of an objective function $J(x)$, as shown on the right.



The optimum value occurs when the first derivative of the objective function is zero. When the first derivative is zero, the steepest descent algorithm will stop updating.

- (a) We seek to minimize $J(x) = \frac{1}{2} (x - x_0)^2$ where x_0 is a constant. Write the update equation for $x[k+1]$ in terms of $x[k]$. 6 points.

$$x[k+1] = x[k] - \mu \left. \frac{\partial J(x)}{\partial x} \right|_{x=x[k]}$$

$$x[k+1] = x[k] - \mu (x[k] - x_0)$$

Homework 5.1 solution and its in-class discussion

- (b) The update equation in part (a) can be interpreted as a first-order linear time invariant (LTI) system with output $x[k+1]$ and previous output $x[k]$ for $k \geq 0$.

$$x[k+1] = (1 - \mu) x[k] + \mu x_0$$

- i. Give a formula for the input signal for the linear time-invariant system? 3 points.

$$\mu x_0 u[k]$$

- ii. What is the initial guess of x , i.e. $x[0]$? 3 points.

$$x[0] = 0 \text{ to satisfy linear and time-invariant properties}$$

- iii. What is the pole location? 3 points.

$$1 - \mu$$

Homework 5.1 solution and its in-class discussion

- iv. Give the range of step size values that make the LTI system bounded-input bounded-output stable. 3 points.

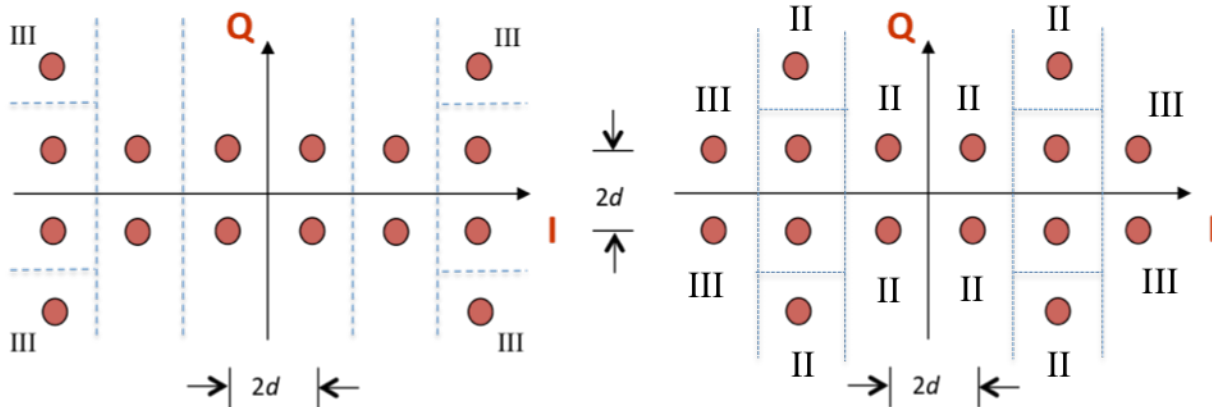
$$|1 - \mu| < 1 \Rightarrow -1 < 1 - \mu < 1 \Rightarrow -2 < -\mu < 0 \Rightarrow 0 < \mu < 2$$

- v. What values of the step size lead to the first-order LTI system being a lowpass filter? 3 points.

The angle of a pole near the unit circle indicates the center of the passband. A real positive pole near the unit circle indicates a lowpass filter. With the pole at $1 - \mu$, we would like μ to be small and positive ($0 < \mu < 0.25$).

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$34d^2$	$26d^2$
(b) Average transmit power	$18d^2$	$14d^2$
(c) Draw the decision regions for the right constellation on top of the right constellation.		
(d) Number of type I regions	0	4
(e) Number of type II regions	12	8
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$\frac{11}{4}Q\left(\frac{d}{\sigma}\right) - \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$
(h) Gray coding possible?	No	No

For parts (a) and (b), due to symmetry in the constellation on the right, we can look at upper right quadrant for power calculations: $d+jd, 3d+jd, 3d+j3d, 5d+jd$. Power is proportional to $I^2 + Q^2$, i.e. $2d^2, 10d^2, 18d^2$, and $26d^2$, respectively. Peak power is $26d^2$. Average power is $14d^2$.

For part (g), the constellation on the right has the same number of type I, II and III constellation regions as a standard 16-QAM constellation (i.e. 4-PAM in in-phase and 4-PAM in quadrature directions). We can reuse symbol error probability from the standard 16-QAM constellation.

(i) Give a fast algorithm to decode a received symbol amplitude into a symbol of bits using the left constellation above. **Using divide & conquer, we need J comparisons for a constellation of J bits.**

Bit #1: $Q > 0$

Bit #2: $I > 0$. Now we know which of the four quadrants we're in.

Upper right quadrant:

Bit #3 is set to $I > 4d$.

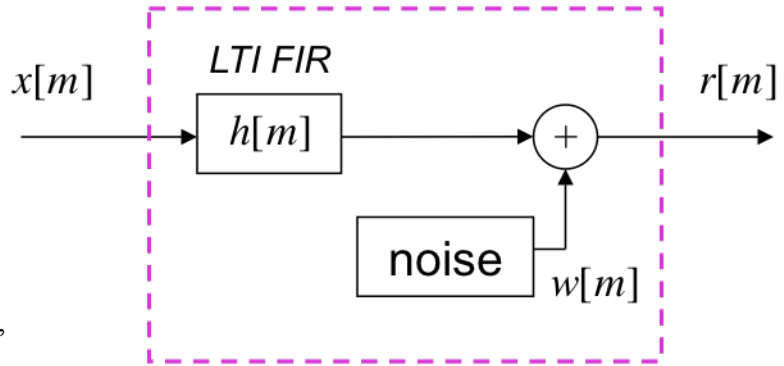
Bit #4 is set to $I > 2d$ if Bit #3 set and set to $Q < 2d$ otherwise.

Problem 2.3. Estimating SNR at a Receiver. 28 points.

Signal-to-noise ratio (SNR) is application-independent measure of signal quality.

Consider a signal $x[m]$ passing through an unknown system that is received as $r[m]$.

We model the unknown system as a linear time-invariant (LTI) finite impulse response (FIR) filter plus an additive Gaussian noise signal $w[m]$ with zero mean and variance σ^2 , as shown on the right.



Impulse response of the LTI FIR system is $h[m]$.

- (a) An application-independent way to estimate the SNR at $r[m]$ is to send signal $x_1[m]$ of M samples to receive $r_1[m]$, wait for a very short period of time, and send $x_1[m]$ again to receive $r_2[m]$:

$$r_1[m] = h[m] * x_1[m] + w_1[m] \quad \text{for } m = 0, 1, \dots, M-1$$

$$r_2[m] = h[m] * x_1[m] + w_2[m] \quad \text{for } m = 0, 1, \dots, M-1$$

- i. Derive an algorithm to estimate σ^2 by subtracting $r_2[m]$ and $r_1[m]$. 9 points.

$$r_2[m] - r_1[m] = (h[m] * x_1[m] + w_2[m]) - (h[m] * x_1[m] + w_1[m]) = w_2[m] - w_1[m] = v[m]$$

$$\sigma_v^2 = E\{v^2[m]\} - E^2\{v[m]\}$$

$$E\{v^2[m]\} = E\{(w_2[m] - w_1[m])^2\} = E\{w_2^2[m]\} - 2E\{w_1[m]w_2[m]\} + E\{w_1^2[m]\} = 2\sigma^2$$

$$E\{v[m]\} = \frac{1}{M} \sum_{m=0}^{M-1} v[m] = \frac{1}{M} \sum_{m=0}^{M-1} (w_2[m] - w_1[m]) = \frac{1}{M} \sum_{m=0}^{M-1} w_2[m] - \frac{1}{M} \sum_{m=0}^{M-1} w_1[m] = 0$$

Note: $E\{w_1[m]w_2[m]\} = \frac{1}{M} \sum_{m=0}^{M-1} w_1[m]w_2[m] = 0$ because $w_1[m]$ and $w_2[m]$ have zero mean.

- ii. How long should we wait between the two transmissions of $x_1[m]$? 5 points.

Group delay of FIR filter $h[m]$ to prevent overlap in the two transmissions at receiver.

- (b) In a pulse amplitude modulation (PAM) system, the received signal $r[m]$ goes through a matched filter and downsampler. The downsampler output is the received symbol amplitude.

- i. During training, transmitted and received symbol amplitudes are known. Use this fact to estimate the SNR for each training symbol at the downsampler output. 9 points.

Let a_n be the transmitted symbol amplitude and \hat{a}_n be the received symbol amplitude for symbol index n . (Index m is the sample index.) Noise power is $(\hat{a}_n - a_n)^2$.

$$\text{SNR} = \frac{\text{Signal Power}}{\text{Noise Power}} = \frac{a_n^2}{(\hat{a}_n - a_n)^2}$$

- ii. Based on the noise power at the downsampler output, give a formula for σ^2 . 5 points

A continuous-time matched filter changes the additive noise power σ^2 by $1/T_{\text{sym}}$.

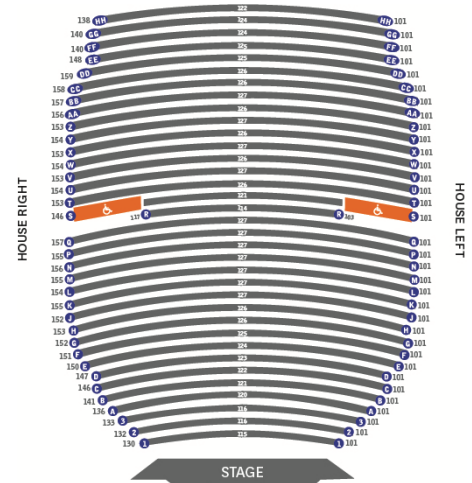
In discrete-time, a matched filter changes the additive noise power by $1/L$ where L is the number of samples per symbol period. So, $\sigma^2 = L(\hat{a}_n - a_n)^2$.

Problem 2.4. Acoustics of a Concert Hall. 24 points

Some audio playback systems have the option to emulate a specific concert hall.

One implementation is to convolve an audio track with the impulse response $h[m]$ of the concert hall.

To estimate the impulse response of the concert hall, we place a speaker on stage and a microphone at one of the seats.



*Orchestra Seating, Bass Concert Hall,
The University of Texas at Austin*

- (a) Give two examples of the training signal $x[m]$ you could use. Why? 6 points.

- Pseudo-noise sequence (maximal length preferred)**
- Chirp signal that sweeps over all audio frequencies**
- Audio track**

Homework 4.2, 4.3 & 5.2

- (b) Set up steepest descent algorithm to update $h[m]$ so that $r[m]$ is as close to $h[m] * x[m]$ as possible.
- i. Give an objective function to be minimized. 6 points.

$$y[m] = r[m] - h[m] * x[m]$$

$$J(y[m]) = (1/2) y^2[m]$$

*Homework 7.2
Fall 2014 Midterm 2.3*

- ii. Give the update equation for the vector \vec{h} of FIR coefficients. 6 points.

$$h[m] * x[m] = h[0] x[m] + h[1] x[m-1] + h[2] x[m-2] + \dots + h[M-1] x[m-(M-1)]$$

$$\text{Let } \vec{h}[m] = [h[0] \quad h[1] \quad h[2] \quad \dots \quad h[M-1]]$$

$$\text{and } \vec{x}[m] = [x[m] \quad x[m-1] \quad x[m-2] \quad \dots \quad x[m-(M-1)]]$$

$$\vec{h}[m+1] = \vec{h}[m] - \mu \frac{\partial J(y[m])}{\partial \vec{h}[m]} = \vec{h}[m] - \mu y[m] \frac{\partial y[m]}{\partial \vec{h}[m]} = \vec{h}[m] + \mu y[m] \vec{x}[m]$$

*Homework 7.2
Fall 2014 Midterm 2.3*

- iii. What values would you recommend for the step size μ ? 6 points.

We would like small positive μ values, e.g. $\mu = 0.001$.

Homework 6.1, 6.2, 7.2 & 7.3

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 5, 2016

Course: EE 445S

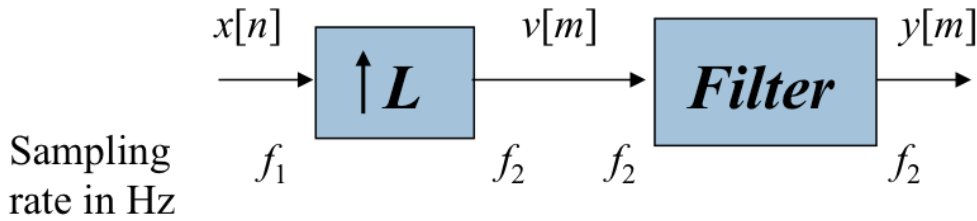
Name: _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Problem	Point Value	Your score	Topic
1	21		Interpolation
2	27		QAM Communication Performance
3	28		Phase Locked Loop (PLL)
4	24		Communication System Design
Total	100		

Problem 2.1. Interpolation. 21 points.

Interpolation can change the sampling rate of discrete-time signal $x[n]$ through discrete-time operations of upsampling by L and then filtering:



(a) Give a formula for f_2 in terms of f_1 . 3 points.

$$f_2 = L f_1$$

(b) Specify the filter's passband frequency ω_{pass} and stopband frequency ω_{stop} in rad/sample to pass as many frequencies in $x[n]$ as possible and reduce artifacts due to upsampling. 6 points.

$x[n]$ contains frequencies from $-(\frac{1}{2})f_1$ to $(\frac{1}{2})f_1$ due to the sampling theorem $f_1 > 2 f_{\text{max}}$.

For sampling rate f_1 , $\cos(2\pi (\frac{1}{2})f_1 t)$ becomes $\cos(\pi n)$, which does not alias.

The filter operates at sampling rate $f_2 = L f_1$.

Answer #1: $\omega_{\text{pass}} = 2\pi \frac{\frac{1}{2}f_1}{f_2} = 2\pi \frac{\frac{1}{2}f_1}{L f_1} = \frac{\pi}{L}$ and $\omega_{\text{stop}} = 1.1\omega_{\text{pass}}$ to allow 10% rolloff.

Answer #1 makes sure that all frequencies in $x[n]$ are passed, but a frequency band equal to 10% of $(\frac{1}{2})f_1$ of artifacts due to upsampling also passes.

Answer #2: $\omega_{\text{pass}} = 0.9\omega_{\text{stop}}$ to allow 10% rolloff and $\omega_{\text{stop}} = 2\pi \frac{\frac{1}{2}f_1}{f_2} = 2\pi \frac{\frac{1}{2}f_1}{L f_1} = \frac{\pi}{L}$.

Answer #2 makes sure that all artifacts due to upsampling fall into the stopband, but with a loss in the upper 10% of the frequency content in $x[n]$.

(c) If $x[n]$ is represented with B bits, specify the passband tolerance A_{pass} in dB and the stopband attenuation A_{stop} in dB. 6 points.

With a representation of B bits, $\text{SNR}_{\text{dB}} = C_0 + 6 B = P_{\text{signal}} - P_{\text{noise}}$.

Passband magnitude response will be approximately 1 in linear units, which is 0 dB.

$$A_{\text{stop}} = -\text{SNR}_{\text{dB}} = -C_0 - 6 B$$

$$A_{\text{pass}} = 20 \log_{10}(1-\Delta) \text{ where } \Delta \text{ is the quantization step size: } \Delta = \frac{1}{2^{B-1}}$$

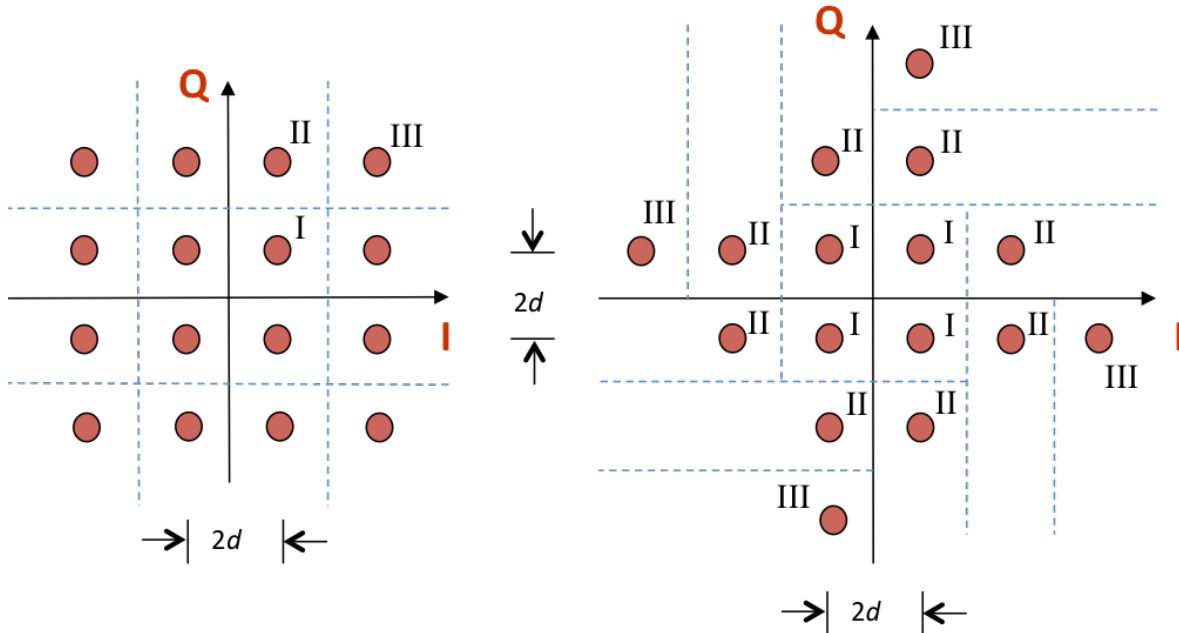
(d) Give an advantage for each type of interpolation filter below. 6 points.

i. Finite impulse response filter. (a) Always bounded-input bounded-output stable; (b) Has efficient polyphase filter bank form that saves factor of L in multiplication-add and read operations/second, factor of 2 in write operations/second, and factor of L in storage of current and previous values $x[n]$; (c) Can have linear phase over all frequencies if impulse response is symmetric or anti-symmetric about its midpoint.

ii. Infinite impulse response filter. (a) Small group delay; (b) Lower order for same magnitude specification with 10% rolloff, but the polyphase filter bank form for the FIR filter would have a more efficient implementation.

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$26d^2$
(b) Average transmit power	$10d^2$	$12d^2$
(c) Draw the decision regions for the right constellation on top of the right constellation.		
(d) Number of type I regions	4	4
(e) Number of type II regions	8	8
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$

The right constellation has symmetry. The upper right quadrant of constellation points is rotated by 90 degrees in the upper left quadrant, 180 degrees in the lower left quadrant, etc.

In part (b), the average transmit power can be computed using the upper right quadrant.

In part (c), the boundaries of the decision regions for the right constellation are drawn are the in-phase (I) axis, quadrature (Q) axis and dashed lines.

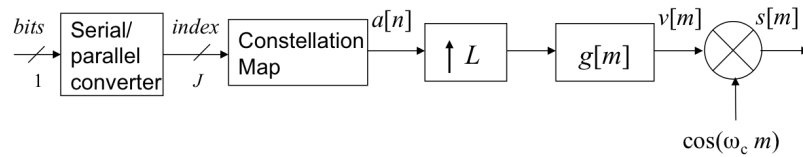
In part (g), the probability of symbol error expression for the right constellation is the same as that of the left because the number of type I, type II and type III regions are the same.

(h) Which of the constellations would you advocate using? Why? Please give two reasons. 6 points.

Left constellation has lower peak transmit power, lower average transmit power, and lower transmit peak-to-average power ratio. It can be gray coded, but right constellation cannot.

Problem 2.3. Phase Locked Loop (PLL). 28 points.

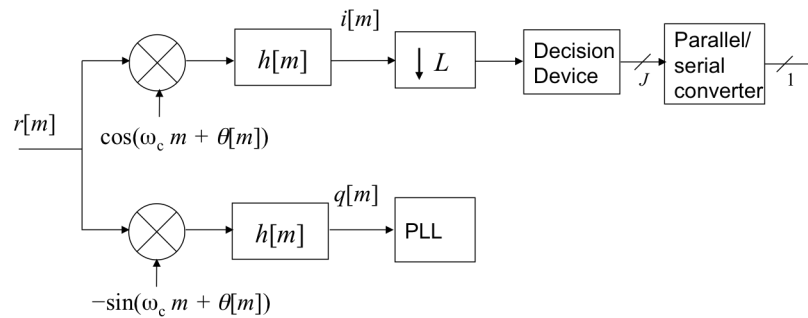
The discrete-time transmitter below is for pulse amplitude modulation (PAM) with upconversion:



where

$a[n]$ symbol amplitude f_s sampling rate $g[m]$ pulse shape
 J bits/symbol L samples/symbol ω_c carrier frequency

The discrete-time PAM receiver below has two downconversion paths, and one feeds into the PLL:



$\theta[m]$ is the carrier phase offset, and $h[m]$ represents a lowpass finite impulse response (FIR) filter.

(a) When the receiver carrier phase matches the transmitter carrier phase, i.e. when $\theta[m] = 0$, show that $q[m]$ is zero. 6 points.

$$q[m] = \text{LPF}\{x[m]\} = \text{LPF}\{-r[m] \sin(\omega_c m + \theta[m])\} = \text{LPF}\{-v[m] \cos(\omega_c m) \sin(\omega_c m + \theta[m])\}$$

Here, $v[m]$ is a lowpass signal because the pulse shaping filter $g[m]$ is lowpass.

The lowpass filter (LPF) $h[m]$ has the same bandwidth as lowpass filter $g[m]$.

$$q[m] = \text{LPF}\left\{-\frac{1}{2} v[m] (\sin(\theta[m]) + \sin(2\omega_c m + \theta[m]))\right\} = \left(-\frac{1}{2}\right) v[m] \sin(\theta[m])$$

Hence, when $\theta[m] = 0$, $q[m] = 0$. Please note that the receiver does not know $v[m]$.

(b) Develop a steepest descent algorithm to estimate the carrier phase offset, $\theta[m]$, per the steps below.

i. Give an objective function. 6 points.

$$J(q[m]) = \frac{1}{2} q^2[m]$$

ii. Give an update equation for $\theta[m+1]$ in terms of $\theta[m]$. 9 points.

$$\theta[m+1] = \theta[m] - \mu \frac{dJ(q[m])}{d\theta[m]} = \theta[m] - \mu q[m] \text{LPF}\{-r[m] \cos(\omega_c m + \theta[m])\}$$

$$\theta[m+1] = \theta[m] + \mu q[m] i[m]$$

Note: This is a version of the Costas loop.

iii. Give an initial value of $\theta[m]$. 3 points.

Objective function can go to zero at $\theta[m]$ at $0, 90^\circ, 180^\circ$, etc. Let $\theta[0] = 0$.

iv. What values of the step size, μ , would you use. Why? 4 points.

Small positive values to guarantee convergence, e.g. $\mu = 0.01$ or $\mu = 0.001$.

Problem 2.4. Communication System Design. 24 points

For M -level pulse amplitude modulation systems, the probability of a symbol error in the receiver is

$$P_{error} = \frac{2(M-1)}{M} Q\left(\frac{d}{\sigma} \sqrt{T_{sym}}\right)$$

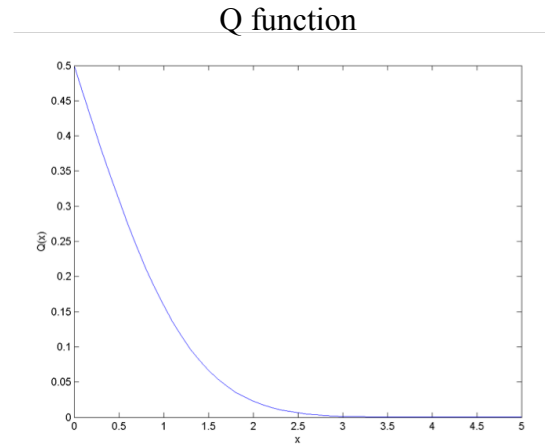
where

$2d$ is spacing between adjacent constellation points in Volts,

σ^2 is variance of the noise in the communication channel,

T_{sym} is symbol time, and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy$$



Solution prolog: Bit rate is $f_{sym} \log_2(M)$ where $f_{sym} = 1 / T_{sym}$. Peak transmit power is $(M-1)^2 d^2$. M has minor impact on symbol error probability because $2(M-1) / M$ is in interval $[1, 2)$.

- (a) How would you choose M , d and T_{sym} for a high-speed communication link with probability of symbol error of 10^{-3} . 9 points.

High-speed communication link means a large f_{sym} or equivalently a small T_{sym} .

High-speed communication link would need a large value of M .

Choose a large value of d to offset the small value of T_{sym} .

The end result will be a high-power, high-speed communication link.

- (b) How would you choose M , d and T_{sym} for a low-speed control channel with probability of symbol error of 10^{-7} . The control channel would allow the feedback of information from receiver to transmitter, such as estimated SNR and channel impulse responses, with high accuracy. 9 points.

Low-speed communication link means a small f_{sym} or equivalently a large T_{sym} .

To get 10^{-7} symbol error probability, Q function argument needs to be slightly more than 5.

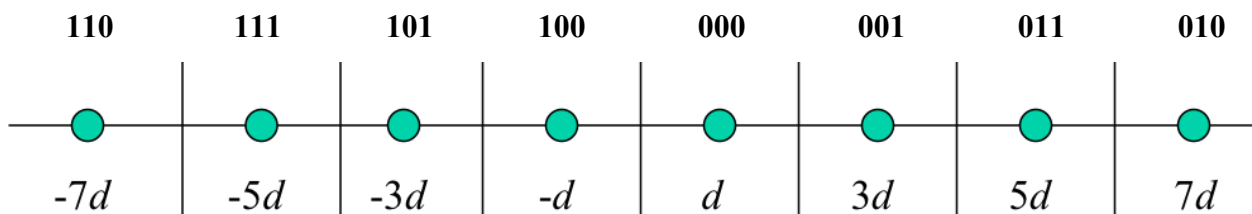
By setting $M = 2$ for 2-PAM or BPSK, we can make d as large as possible.

The peak and average transmit power will be proportional to d^2 .

- (c) Give an optimal encoding for symbols of bits for the 8-PAM constellation below. In what sense is your encoding optimal? 6 points.

Gray coding minimizes the number of bit errors when there is a symbol error.

One of many possible Gray codings is shown below.



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: May 5, 2017

Course: EE 445S

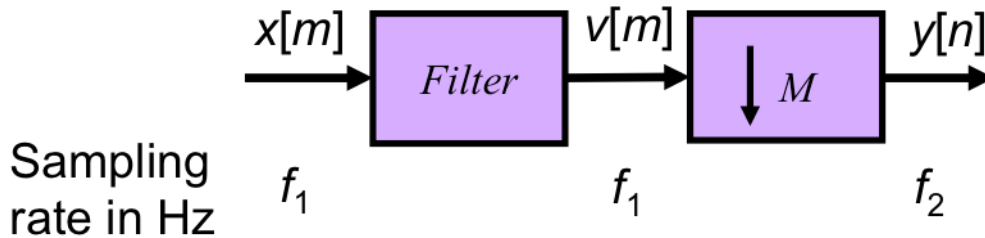
Name: Mini-Soigneurs Les
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Problem	Point Value	Your score	Topic
1	21		Decimation
2	27		QAM Communication Performance
3	28		Blind Channel Equalization
4	24		Channel Equalization With Training
Total	100		

Problem 2.1. Decimation. 21 points.

Decimation can change the sampling rate of discrete-time signal $x[n]$ through discrete-time operations of filtering and then downsampling by M .



(a) Give a formula for $y[n]$ in terms of $v[]$. 3 points.

The downsampler accepts a block of M samples as input, and then outputs the first sample and discards the rest. Hence, $y[n] = v[Mn]$.

(b) Give a formula for f_2 in terms of f_1 . 3 points.

M times as many samples are at the downsampler input than at the downsampler output:

$$f_2 = \frac{1}{M} f_1$$

(c) Specify the filter's passband frequency ω_{pass} and stopband frequency ω_{stop} in rad/sample to pass as many frequencies in $x[m]$ as possible and reduce as many artifacts due to downsampling in $y[n]$ as possible. 6 points.

$y[n]$ contains frequencies from $-(\frac{1}{2})f_2$ to $(\frac{1}{2})f_2$ due to the sampling theorem $f_2 > 2f_{max}$

The lowpass filter operates at sampling rate $f_1 = Mf_2$.

Answer #1: $\omega_{pass} = 2\pi \frac{\frac{1}{2}f_2}{f_1} = 2\pi \frac{\frac{1}{2}f_1}{Mf_1} = \frac{\pi}{M}$ and $\omega_{stop} = 1.1\omega_{pass}$ to allow 10% rolloff.

Answer #1 makes sure that the filter passes all frequencies in $y[n]$, but a frequency band equal to 10% of $(\frac{1}{2})f_2$ of artifacts due to downsampling also passes.

Answer #2: $\omega_{pass} = 0.9\omega_{stop}$ to allow 10% rolloff and $\omega_{stop} = 2\pi \frac{\frac{1}{2}f_2}{f_1} = 2\pi \frac{\frac{1}{2}f_1}{Mf_1} = \frac{\pi}{M}$.

Answer #2 makes sure that all artifacts due to downsampling fall into the stopband, but with a loss in the upper 10% of the frequency content in $y[n]$.

(d) In converting an audio signal sampled at 48 kHz to a speech signal sampled at 8 kHz,

i. What is the value of M ? 3 points. **$M = \frac{48 \text{ kHz}}{8 \text{ kHz}} = 6$**

ii. Would you use a finite impulse response filter or an infinite impulse response filter. Why? 6 points

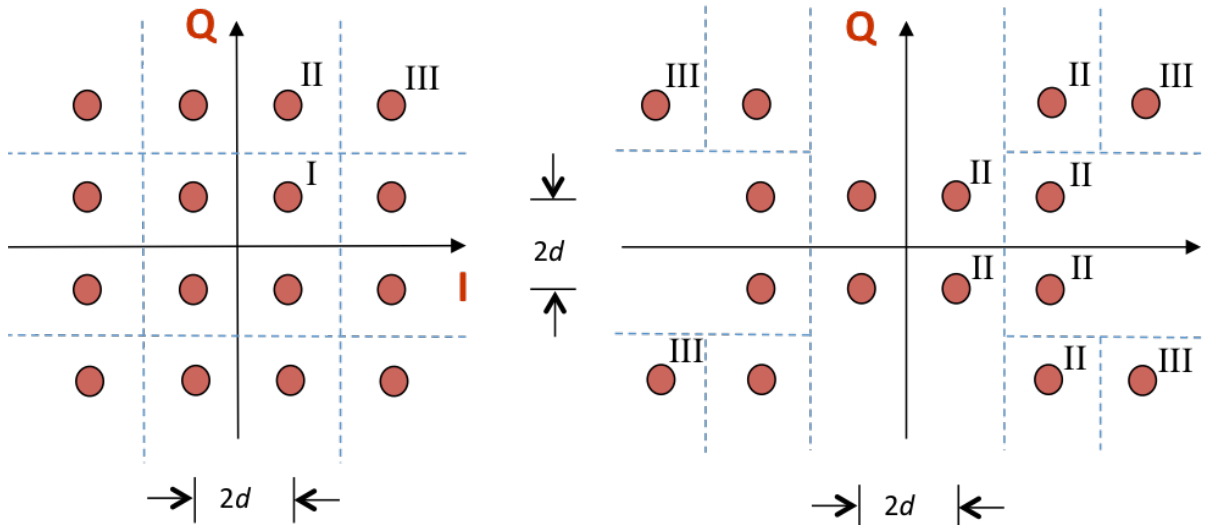
In audio systems, phase response is important. For real-time audio systems, having low group delay and low implementation complexity are also important.

Answer #1: Linear phase FIR filter— linear phase over all frequencies and low complexity due to polyphase form. May have high group delay. Always BIBO stable.

Answer #2: IIR filter— approximate linear phase over passband, low complexity due to low order, and low group delay. Implementation may become BIBO unstable.

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.

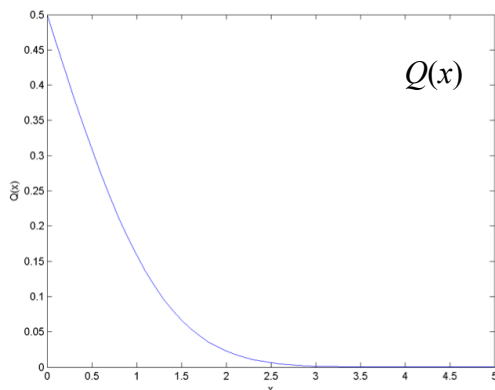


Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$34d^2$
(b) Average transmit power	$10d^2$	$16d^2$
(c) Draw the decision regions for the right constellation on top of the right constellation.		
(d) Number of type I regions	4	0
(e) Number of type II regions	8	12
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$\frac{11}{4}Q\left(\frac{d}{\sigma}\right) - \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$

(h) Which constellation has a lower probability of symbol error vs. signal-to-noise ratio? Why? 6 points.



$$\text{SNR}^{\text{left}} = \frac{\text{Transmit Signal Power}}{\text{Average Noise Power}} = \frac{10d^2}{\sigma^2} \rightarrow \frac{d}{\sigma} = \sqrt{\frac{\text{SNR}^{\text{left}}}{10}}$$

$$\text{SNR}^{\text{right}} = \frac{\text{Transmit Signal Power}}{\text{Average Noise Power}} = \frac{16d^2}{\sigma^2} \rightarrow \frac{d}{\sigma} = \sqrt{\frac{\text{SNR}^{\text{right}}}{16}}$$

For the same SNR value, the probability of symbol error will be lower for the left constellation because the quantity d/σ will be larger and the Q function is monotonically decreasing with respect to its argument.

Note: Gray coding, which minimizes the number of bit errors when there is a symbol error, is not relevant to the answer here in part (h).

Problem 2.3. Blind Channel Equalization. 28 points.

Blind channel equalization occurs without a training sequence, as shown below.

When the transmitted sequence $x[k]$ is binary phase shift keying (BPSK), i.e. is +1 or -1, an adaptive method can be based on the fact that $x^2[k] = 1$.

Assume a two-tap finite impulse response (FIR) equalizer with its first coefficient fixed at one:

$$w[k] = \delta[k] + w_1 \delta[k-1]$$

(a) Using the objective function

$$J(k) = \frac{1}{4} (1 - r^2[k])^2$$

derive the adaptive update equation for w_1 . 16 points

$$w_1[k+1] = w_1[k] - \mu \left. \frac{dJ(k)}{dw_1} \right|_{w_1=w_1[k]} = w_1[k] - \mu \frac{1}{4} 2(1 - r^2[k]) (-2r[k]) \frac{dr[k]}{dw_1}$$

$$r[k] = y[k] + w_1 y[k-1]$$

$$w_1[k+1] = w_1[k] + \mu (1 - r^2[k]) r[k] y[k-1]$$

This approach minimizes dispersion. It is known as the constant modulus algorithm as well as the Godard algorithm. Please see JSK Section 13.5 on pages 290-292.

(b) Give an initial value for w_1 . Why did you choose that value? 3 points

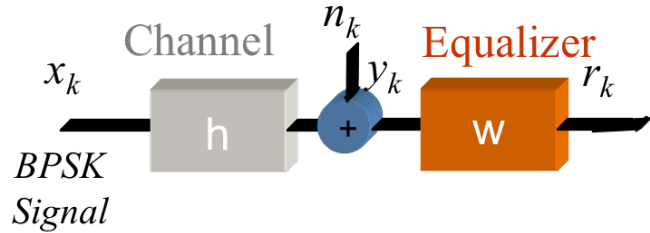
**When $w_1 = 1$, the equalizer is lowpass, which would equalize a highpass channel.
 When $w_1 = -1$, the equalizer is highpass, which would equalize a lowpass channel.
 Since we don't know the channel frequency response, we initialize $w_1 = 0$.**

(c) What range of values would you use for the step size μ ? Why? 3 points

We would like a small, positive value for the step size, e.g. 0.01 or 0.001. The smaller the positive value, the better the smoothing of the derivative of the objective function.

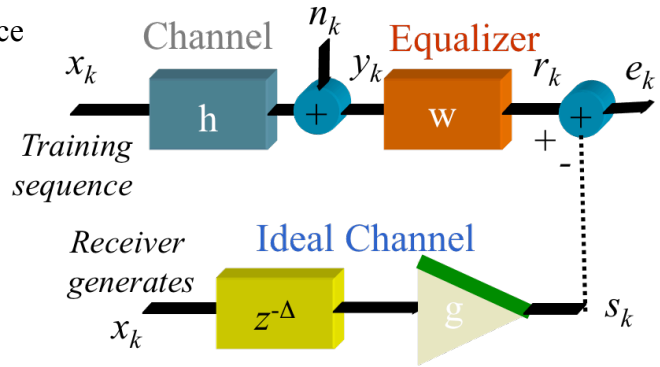
(d) How would you adjust the objective function for 4-level pulse amplitude modulation? 6 points

**BPSK (2-PAM) has symbol amplitude values of -1 and +1; each has a power level of 1.
 4-PAM has symbol amplitude values of -3, -1, +1, and +3, which have power levels of 9, 1, 1, and 9, respectively. Assuming equally likely probability among the four levels, the average power is 5. We can adjust the objective function as follows: $J(k) = \frac{1}{4} (5 - r^2[k])^2$**



Problem 2.4. Channel Equalization With Training. 24 points

For the finite impulse response (FIR) channel equalizer on the right:



(a) Give two reasons why pseudo-noise is a good choice for the training sequence. 3 points.

1. PN sequence has all frequencies in it.
2. PN sequence it is easy to generate in the transmitter and receiver, e.g. by using a feedback shift register.
3. PN sequence, when correlated against the receive signal, gives a peak when finding the same PN sequence and the transmit signal, which can be used to find propagation delay.

(b) Here is the update equation for an adaptive least mean squares FIR filter with coefficients \mathbf{w} :

$$\mathbf{w}[k+1] = \mathbf{w}[k] - \mu e[k] \mathbf{y}[k]$$

where $\mathbf{y}[k] = [y[k] \ y[k-1] \ \dots \ y[k-(N-1)]]$ and $e[k] = r[k] - g x[k - \Delta]$ and $r[k] = \text{FIR}\{y[k]\}$

i. How many multiplications are needed per iteration? How does this compare with an FIR filter? 6 points

$g x[k - \Delta]$ needs 1 multiplication, $r[k] = \text{FIR}\{y[k]\}$ needs N multiplications,

$\mu e[k]$ needs 1 multiplication, and $\mu e[k] \mathbf{y}[k]$ needs N multiplications,

which is a total of $2N + 2$ multiplications per iteration.

An FIR filter requires N multiplications per iteration.

ii. How many words of memory are needed the adaptive FIR filter? How does this compare with an FIR filter? 6 points

$2N + 2$ words— two vectors of N elements each, i.e. $\mathbf{w}[k]$ and $\mathbf{y}[k]$, as well as μ and $e[k]$

$2N$ words for an FIR filter.

iii. What range of values would you use for the step size μ ? Why? 3 points

The step size must be in the interval $(0, 2)$ for convergence.

We would like a small, positive value for the step size, e.g. 0.01 or 0.001. The smaller the positive value, the better the smoothing of the derivative of the objective function.

(c) For a training sequence of length $2N$, would you advocate using a least squares equalizer or an adaptive least mean squares equalizer? 6 points

Because the training sequence length is so short relative to the equalizer length, a least squares equalizer will perform better than an adaptive least mean squares equalizer. That is, the adaptive least mean squares equalizer will not have enough training data to converge to a meaningful solution.

Note: In homework problem 7.2, the training sequence was 250 times the equalizer length.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: December 11, 2017

Course: EE 445S

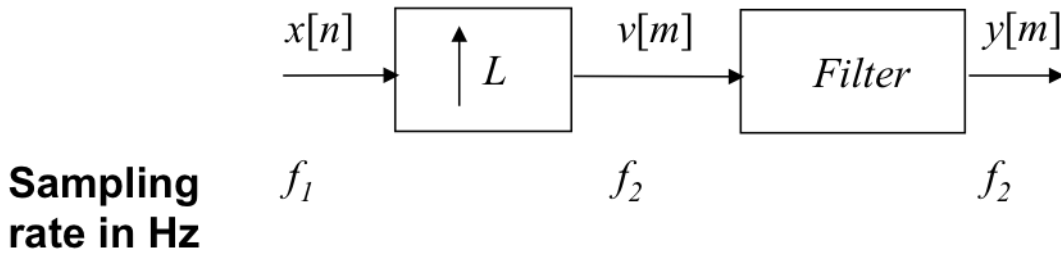
Name: _____ **Pullman** **Auggie**
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Problem	Point Value	Your score	Topic
1	24		Interpolation
2	27		QAM Communication Performance
3	28		Baseband Pulse Amplitude Modulation
4	21		Potpourri
Total	100		

Problem 2.1. Interpolation. 24 points. Lecture Slides 7-14 to 7-15 and 13-6 to 13-10; JSK Sec. 6.4.

Interpolation can change the sampling rate of discrete-time signal $x[n]$ through discrete-time operations of upsampling by L and then filtering:



- (a) Give a formula for $v[m]$ in terms of $x[n]$. 3 points.

For each input sample, the upsampler copies it to the output and then outputs $L-1$ zeros.

$$v[m] = \begin{cases} x\left[\frac{m}{L}\right] & \text{if } m = kL \text{ where } k \text{ is an integer} \\ 0 & \text{otherwise} \end{cases}$$

- (b) Give a formula for f_2 in terms of f_1 . 3 points.

Upsampler has L times more samples on its output than its input: $f_2 = L f_1$

- (c) Specify the filter's passband frequency ω_{pass} and stopband frequency ω_{stop} in rad/sample to pass as many frequencies in $x[n]$ as possible and reduce as many artifacts due to upsampling in $y[m]$ as possible. 6 points. *HW 2.2, 2.3 & 3.1; Lecture Slides 5-16 & 13-8; JSK Sec. 7.2.*

The maximum continuous-time frequency captured in $x[n]$ is $\frac{1}{2} f_1$. Any continuous-time frequencies in $v[m]$ that are at or above $\frac{1}{2} f_1$ are artifacts of the upsampling process.

The lowpass filter attenuates frequencies at or above $\frac{1}{2} f_1$ and operates at sampling rate f_2 .

Answer #1: $\omega_{\text{stop}} = 2\pi \frac{\frac{1}{2} f_1}{f_2} = 2\pi \frac{\frac{1}{2} f_1}{L f_1} = \frac{\pi}{L}$ and $\omega_{\text{pass}} = 0.9 \omega_{\text{stop}}$ to allow a 10% rolloff.

Answer #2: $\omega_{\text{pass}} = 2\pi \frac{\frac{1}{2} f_1}{f_2} = 2\pi \frac{\frac{1}{2} f_1}{L f_1} = \frac{\pi}{L}$ and $\omega_{\text{stop}} = 1.1 \omega_{\text{pass}}$ to allow a 10% rolloff.

Answer #3: $\omega_{\text{cutoff}} = \frac{\pi}{L}$, $\omega_{\text{pass}} = 0.95 \omega_{\text{cutoff}}$, $\omega_{\text{stop}} = 1.05 \omega_{\text{cutoff}}$ to allow a 10% rolloff.

- (d) To ensure that the amplitude values of $x[n]$ remain unchanged after upsampling and filtering, give a constraint on the impulse response of the filter. 6 points. *Lecture Slides 7-10, 7-11, 7-13 & 13-8.*

The impulse response of the filter $h[m]$ must be 1 at one multiple of L for index m and zero at all other multiples of L for index m . Sinc and raised cosine pulses have this property.

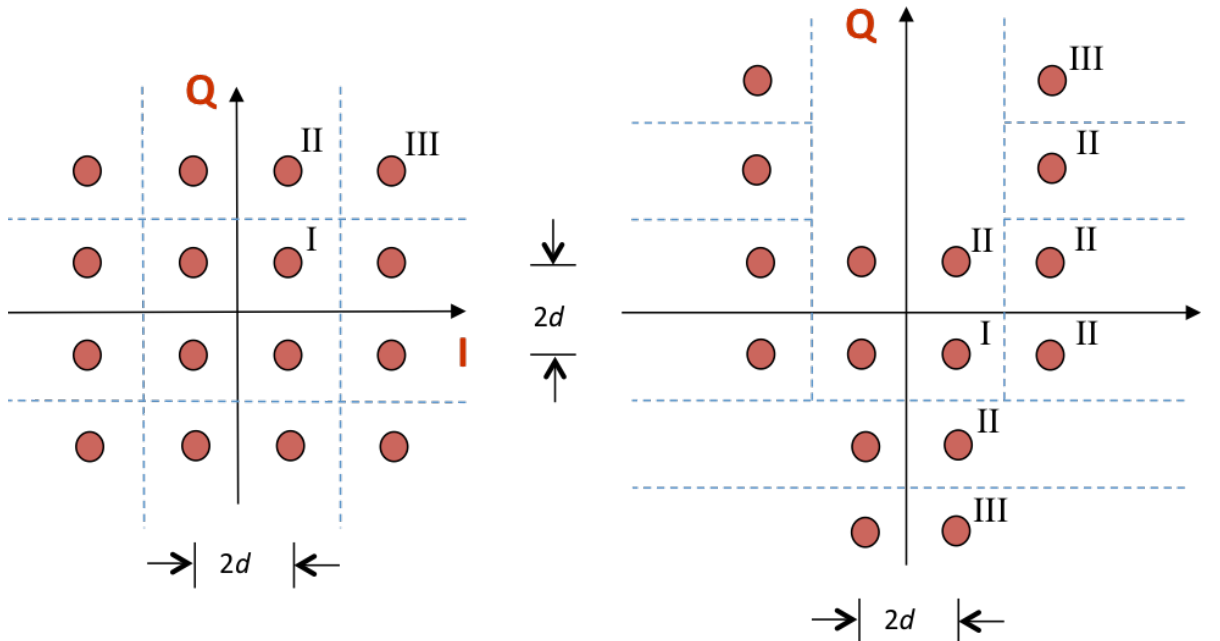
- (e) Give the formula for an infinite impulse response that meets the conditions for (c) and (d) above. 6 points. *Lecture Slides 4-5, 4-7, 4-8, 7-10, 7-11 & 7-13.*

Start with a two-sided continuous-time sinc pulse $h(t) = \frac{\sin(2\pi(\frac{1}{2} f_1)t)}{2\pi(\frac{1}{2} f_1)t} = \frac{\sin(\pi f_1 t)}{\pi f_1 t}$ for

$-\infty < t < \infty$. This is an ideal lowpass filter that passes frequencies up to $\frac{1}{2} f_1$. Sample $h(t)$ at a sampling rate of $f_2 = L f_1$ to obtain $h[m] = \frac{\sin(\omega_0 m)}{\omega_0 m}$ where $\omega_0 = \frac{\pi}{L}$.

Note: Solution would also have worked with a two-sided continuous-time raised cosine pulse.

Problem 2.2 QAM Communication Performance. 27 points. *See also Spring 2017 Midterm 2.2.*
 Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

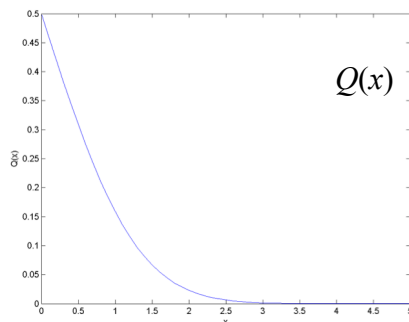
Each part below is worth 3 points. **Please fully justify your answers.** *HW 6.3; Slides 15-12 to 15-16.*

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$34d^2$
(b) Average transmit power	$10d^2$	$14d^2$
(c) Draw the decision regions for the right constellation on top of the right constellation. Yes.		
(d) Number of type I regions	4	2
(e) Number of type II regions	8	10
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$\frac{23}{8}Q\left(\frac{d}{\sigma}\right) - 2Q^2\left(\frac{d}{\sigma}\right)$

$$P_c = \frac{2}{16}P_c^I + \frac{10}{16}P_c^{II} + \frac{4}{16}P_c^{III} = \frac{1}{8}(1 - 2q)^2 + \frac{5}{8}(1 - q)(1 - 2q) + \frac{2}{8}(1 - q)^2 \text{ where } q = Q\left(\frac{d}{\sigma}\right)$$

$$P_c = 1 - \frac{23}{8}q + \frac{16}{8}q^2 \text{ and } P_e = 1 - P_c = \frac{23}{8}q - 2q^2 = \frac{23}{8}Q\left(\frac{d}{\sigma}\right) - 2Q^2\left(\frac{d}{\sigma}\right)$$

(h) Which constellation has lower probability of symbol error vs. signal-to-noise ratio? Why? 6 points.



Left constellation: $\text{SNR} = \frac{10d^2}{\sigma^2}$ which means that $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{10}}$

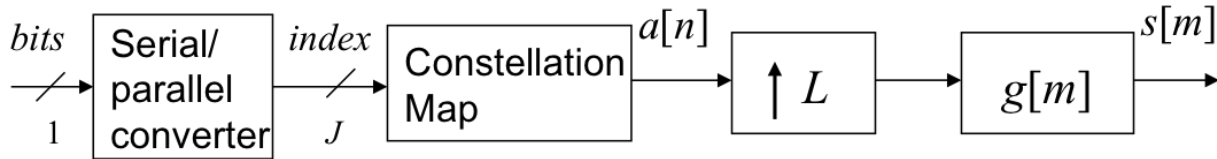
Right constellation: $\text{SNR} = \frac{14d^2}{\sigma^2}$ which means that $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{14}}$

$Q(x)$ falls off faster than exponential for increasing x . *Slide 14-26.*

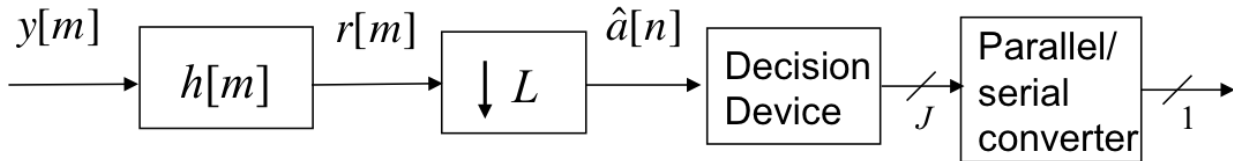
For the same SNR, the left constellation will have a larger d/σ value and hence lower symbol error probability. *Slide 14-29.*

Problem 2.3. Baseband Pulse Amplitude Modulation. 28 points. JSK Ch. 8 & Sec. 9.1.

A baseband pulse amplitude modulation transmitter is described as [Lecture Slides 13-3 to 13-6]



A baseband pulse amplitude modulation receiver is shown below [Lecture Slide 14-7]



where

- | | | |
|-------------------------|---------------------|--------------------|
| $a[n]$ symbol amplitude | f_s sampling rate | $g[m]$ pulse shape |
| J bits/symbol | L samples/symbol | $h[m]$ FIR filter |

(a) What two roles does the finite impulse response (FIR) filter $h[m]$ play? 4 points. **Lect. Slide 16-11.**

Matched filter (i.e. matched to pulse shape $g[m]$) and anti-aliasing filter for downsampler.

(b) For the rest the problem, the FIR filter $h[m]$ will be replaced with an adaptive FIR equalizer.

i. Give initial values for the coefficients of $h[m]$. 6 points. **Lect. Slide 14-15; JSK Sec. 11.5.**

Use the optimum matched filter: $h[m] = g[L - m]$. That is, flip $g[m]$ w/r to m and delay it by enough samples to make it causal. The delay might be different than L .

ii. During training, we will adapt the FIR equalizer coefficients based on the error vector in the decision device, i.e.

$$e[n] = \hat{a}[n] - a[n]$$

Give an objective function $J(e[n])$. 6 points. **Note: Decision-directed FIR equalizer.**

We want to minimize the error vector (i.e. decision error): $J(e[n]) = \frac{1}{2} e^2[n]$. HW 7.2.

iii. Denote the FIR coefficients at iteration k as a vector \vec{h}_k and derive the update equation for \vec{h}_{k+1} , where k is a symbol index. 9 points. **Let N_h be the number of coefficients in \vec{h}_k .**

Because we would like to minimize the objective function, [HW 7.2]

$$\vec{h}_{k+1} = \vec{h}_k - \mu \frac{d}{d\vec{h}_k} J(e[k]) = \vec{h}_k - \mu e[k] \frac{d}{d\vec{h}_k} \hat{a}[k] = \vec{h}_k - \mu e[k] \frac{d}{d\vec{h}_k} r[kL]$$

With $r[kL] = h[0]y[kL] + h[1]y[kL - 1] + \dots$, [Midterm #2 Review Slide 9]

$$\vec{h}_{k+1} = \vec{h}_k - \mu e[k] \vec{y}[k] \text{ where } \vec{y}[k] = [y[kL] \quad y[kL - 1] \quad \dots \quad y[kL - N_h - 1]]$$

iv. What range of values would you recommend for μ ? 3 points.

Small positive value, e.g. $\mu = 0.01$. Must be positive to descend error surface. Cannot be too large; otherwise, iteration will fail to converge. HW 5.1, 6.1, 6.2, 7.2 & 7.3.

Problem 2.4. Potpourri. 21 points

- (a) Give formulas for the system delay and computational complexity vs. the length L_g of a square root raised cosine pulse shape used in a baseband digital pulse amplitude modulation transmitter and receiver. System delay is from the time that the bit stream goes into the transmitter to the time that it would appear at the receiver output. *12 points.*

We can refer to the baseband PAM block diagram in Problem 2.3. There are two finite impulse response (FIR) filters of L_g coefficients each.

Group delay through the raised cosine pulse shaping filter in the transmitter is $\frac{1}{2} L_g$ samples and the group delay through the matched filter in the receiver is $\frac{1}{2} L_g$ samples.

Total delay in the baseband PAM system is L_g samples. Since we're only analyzing the baseband PAM system, we're not including the delay in the analog front ends and channel.

The computational complexity is $2L_g$ multiplication-accumulation operations per sample since there are two FIR filters, each with L_g coefficients.

References: HW 2.3(d) & 5.3; Lecture Slides 5-14, 13-6 & 13-17; JSK Sec. 7.2 & Ch. 8

- (b) Consider a digital pulse amplitude modulation system in which a transmitter sends a training sequence that the receiver uses to adapt a variety of subsystems to compensate for different impairments. For each receiver subsystem below, specify the name of an algorithm (or describe an algorithm) that could be used during data transmission (i.e. when no training data is available) to update each system.

- i. Automatic Gain Control. 3 points. See HW 7.3.*

Adaptive gain control to drive sampler output power to a specified level. JSK Sec. 6.7.

Update gain $c(t)$ using N current/previous A/D output values. Lecture slide 16-5.

$$c(t) = (1 + 2f_0 - f_{-128} - f_{127}) c(t - \tau) \text{ or } c(t) = \frac{2c_0 + \epsilon N}{c_{-128} + c_{127} + \epsilon N} c(t - \tau)$$

- ii. Channel Equalization. 3 points. See HW 7.1 & 7.2.*

Decision-directed equalization. Error prone without training data. JSK Sec. 13.4.

Dispersion-minimizing equalization, a.k.a. constant modulus algorithm. JSK Sec. 13.5.

- iii. Symbol Clock Recovery. 3 points. See HW 6.*

Decision-directed timing recovery. Error prone without training data. JSK Sec. 12.3.

Two single-pole complex bandpass filters plus nonlinearity. Lecture slide 16-10.

Prefilter + squaring block + bandpass filter + phase locked loop. Lab 5 Part 2.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Prof. Brian L. Evans

Date: May 4, 2018

Course: EE 445S

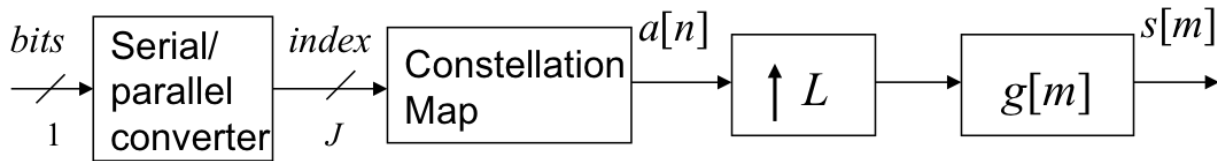
Name: _____ **Time** _____ **Wrinkle In** _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

	Problem	Point Value	Your score	Topic
<i>Meg</i>	1	24		Baseband Pulse Amplitude Modulation
<i>Charles</i>	2	28		QAM Communication Performance
<i>Calvin</i>	3	24		QAM Constellation Derotation
<i>Mrs. Which</i>	4	24		Potpourri
	Total	100		

Problem 2.1. Baseband Pulse Amplitude Modulation. 24 points.

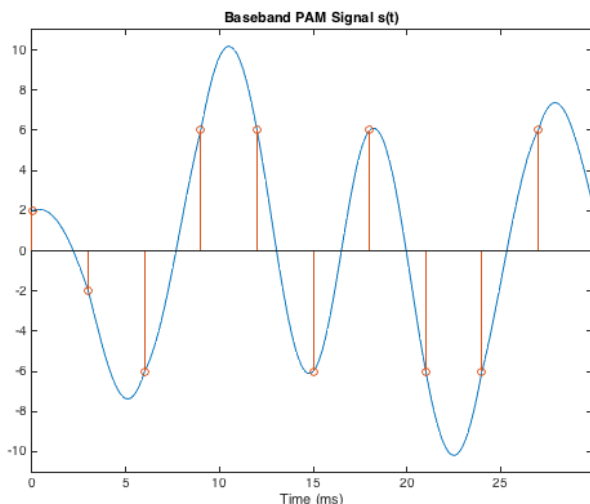
A baseband pulse amplitude modulation transmitter is described as



where

$a[n]$ symbol amplitude f_s sampling rate f_{sym} symbol rate $g[m]$ pulse shape
 J bits/symbol L samples/symbol N_g symbol periods in a pulse shape

For $N_g = 4$ and $L = 20$, a plot is shown below for 10 symbol periods over 0 to 30 ms of $s[m]$ after it had passed through a digital-to-analog converter, and the symbol amplitudes $a[n]$ are shown as a stem plot:



(a) What is the value of J , the number of bits per symbol? Why? 3 points.
From the stem plot, the symbol amplitudes are -6, -2, 2, and 6. Four levels means $J = 2$ bits.

(b) If the spacing between constellation points is $2d$, what is the value of d ? Why? 3 points.
4-PAM has symbol amplitudes of $-3d, -d, d,$ and $3d$. Hence, $d = 2$.

(c) Draw a constellation map with Gray coding. 3 points.
Gray coding means that the bit patterns in two adjacent symbols only differ by one bit to minimize the number of bit errors when a symbol error occurs. →

(d) Accurately compute the symbol time, T_{sym} , in milliseconds. 3 points.
The symbol period is (30 ms) / (10 symbol periods) = 3 ms.

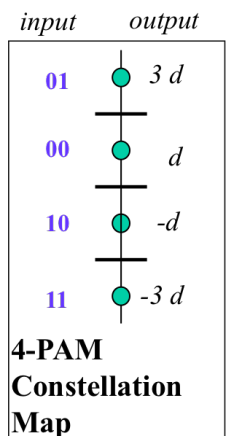
(e) Give a formula for the pulse shape, $g[m]$. How many samples are in $g[m]$? 6 points.

Start with continuous-time sinc pulse $g(t) = \frac{\sin(2\pi(\frac{1}{2}f_{sym})t)}{2\pi(\frac{1}{2}f_{sym})t} = \frac{\sin(\pi f_{sym}t)}{\pi f_{sym}t}$ for

$-\infty < t < \infty$. Sample $h(t)$ at $f_s = L f_{sym}$ to get $h[m] = \frac{\sin(\omega_0 m)}{\omega_0 m}$ where $\omega_0 = \frac{\pi}{L}$.

The pulse shape has $N_g L = 80$ samples in it.

(f) Infer an upper bound on the amplitude of $s[m]$ as a function of d, J and N_g . 6 points.
 $(2^J - 1) d N_g$ because $(2^J - 1) d$ is the maximum symbol amplitude and N_g symbol periods are added in a baseband signal.



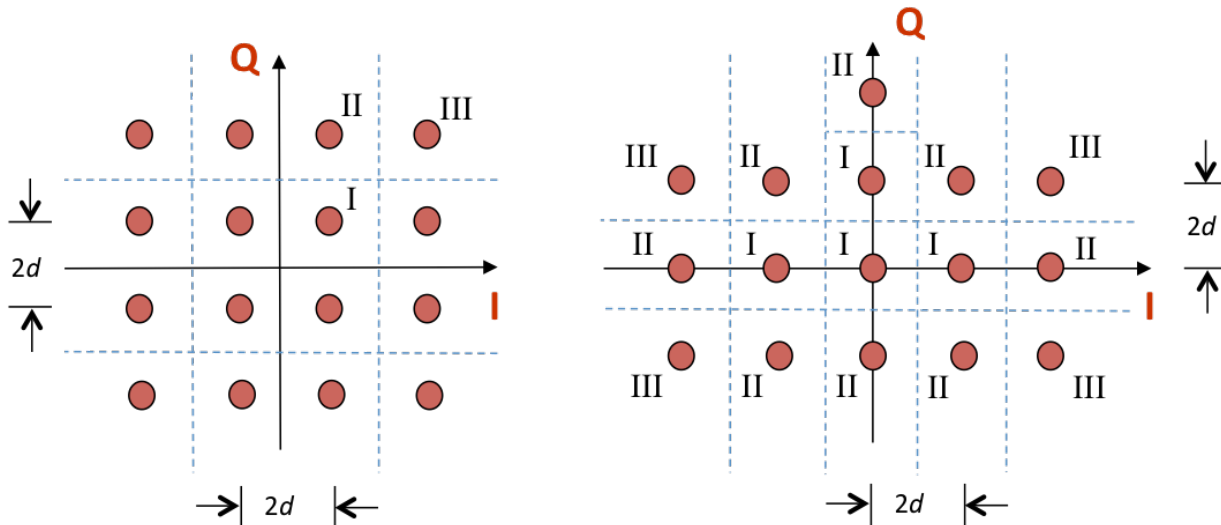
Slide 13-3

MATLAB Code Used to Create the Baseband PAM Signal Plot in Problem 2.1

```
% Spring 2018 Midterm #2
% m is sample index
% n is symbol index
%
% Simulation parameters
N = 12;          % Number symbol periods to generate
% Pulse shape g[m]
Ng = 4;         % Number symbol periods in pulse
L = 20;         % Samples/symbol period in pulse
f0 = 1/L;
midpt = Ng*L/2;
m = (-midpt) : (midpt-1);
g = sinc(f0*m);
% 4-PAM symbol amplitudes
d = 2;
pamLevels = 4;
symAmp = (2*randi(pamLevels,[1,N]) - 5)*d;
symAmp(1) = d;
symAmp(2) = -d;
symAmp(3) = -(pamLevels-1)*d;
symAmp(4) = (pamLevels-1)*d;
symAmp(5) = (pamLevels-1)*d;
symAmp(6) = -(pamLevels-1)*d;
symAmp(7) = (pamLevels-1)*d;
symAmp(8) = -(pamLevels-1)*d;
symAmp(9) = -(pamLevels-1)*d;
symAmp(10) = (pamLevels-1)*d;
% Baseband PAM signal for N symbol periods
mmax = N*L;
v = zeros(1,mmax);
v(1:L:end) = symAmp;          % interpolation
s = conv(v, g);              % pulse shaping
slength = length(s);        % trim result
s = s(midpt+1:slength-midpt+1);
% Plots
Tsym = 3;
fsym = 1/Tsym;
fs = L*fsym;
Ts = 1/fs;
Mmax = length(s);
m = 0 : (Mmax-1);
t = m*Ts;
Nmax = Mmax / L;
n = 0 : (Nmax-1);
figure;
plot(t,s);
hold on;
stem(n*Tsym,symAmp);
hold off;
xlim( [0 (Nmax-2)*Tsym-Ts] ); % Plot N-2 symbol periods
ylim( [-11 11] );
xlabel('Time (ms)');
title('Baseband PAM Signal s(t)');
```


Problem 2.2 QAM Communication Performance. 28 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$20d^2$
(b) Average transmit power	$10d^2$	$11d^2$
(c) Draw the decision regions for the right constellation on top of the right constellation.		
(d) Number of type I regions	4	4
(e) Number of type II regions	8	8
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$

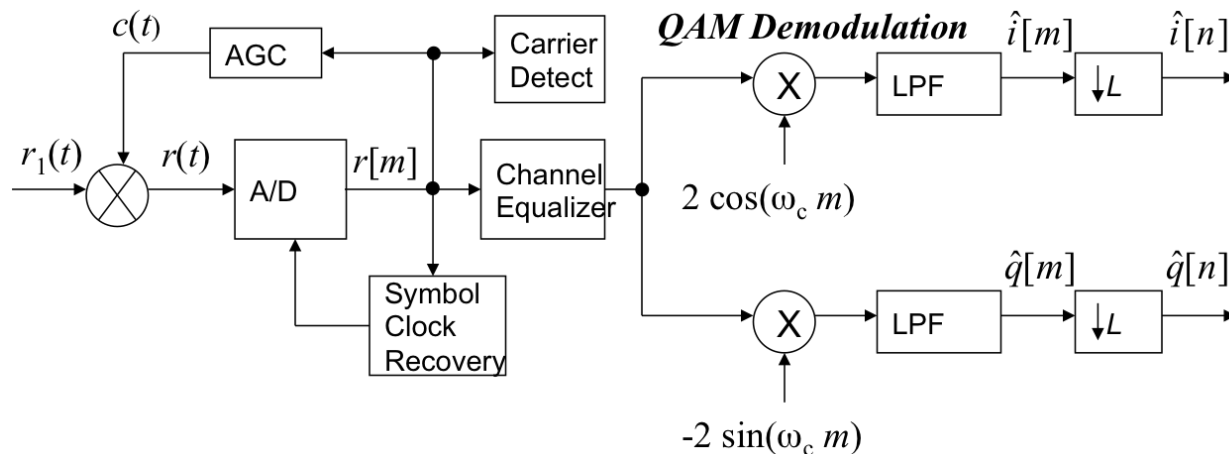
The symbol error probability for the right constellation is the same as the symbol error probability of the left constellation because the right constellation has the same number of type I, II and III decision regions.

(h) Consider using the constellations in upconverted QAM. In the QAM receiver, how would the Costas loop for the phase locked loop perform for the right constellation vs. the left constellation?
 7 points.

The Costas loop is an adaptive method based on steepest descent for tracking the phase offset at in a receiver at the sampling rate. The update in phase offset is the stepsize times the output of in-phase and the quadrature baseband channels (JSK page 208). In the right constellation, either the in-phase component or the quadrature component is zero for half of the constellation points. That is, for these 8 out of 16 constellation points, there is no transmit power in either the in-phase or quadrature channel. During these symbol periods, the Costas loop will not adapt, or if it adapts, it will be solely due to thermal noise and other impairments in the system.

Problem 2.3. QAM Constellation Derotation. 24 points.

A baseband Quadrature Amplitude Modulation (QAM) receiver is given below



where $\hat{i}[n]$ and $\hat{q}[n]$ are the received in-phase and quadrature symbol amplitudes at symbol index n .

At the receiver, the QAM constellation may rotate due to a mismatch in the carrier frequencies.

A phase locked loop running at the sampling rate could track the time-varying phase that is due to the carrier frequency mismatch.

An alternative is to derotate the constellation at the symbol rate by multiplying the complex symbol $\hat{i}[n] + j \hat{q}[n]$ by $e^{j\theta}$, i.e. $i[n] + j q[n] = (\hat{i}[n] + j \hat{q}[n])e^{j\theta} = (\hat{i}[n] + j \hat{q}[n])(\cos \theta + j \sin \theta)$:

$$i[n] = \hat{i}[n] \cos \theta - \hat{q}[n] \sin \theta \quad \text{and} \quad q[n] = \hat{q}[n] \cos \theta + \hat{i}[n] \sin \theta$$

We will adapt the phase offset θ based on the error vector magnitude $e[n]$ in the decision device, i.e.

$$e^2[n] = (i[n] - \hat{i}[n])^2 + (q[n] - \hat{q}[n])^2$$

(a) Give an objective function $J(e[n])$. 6 points.

$$J(e[n]) = \frac{1}{2} e^2[n] = \frac{1}{2} (i[n] - \hat{i}[n])^2 + (q[n] - \hat{q}[n])^2$$

(b) Derive the update equation for θ_{k+1} , where k is a symbol index. 9 points.

We seek to update the phase offset to minimize the mean squared error measure in part (a) and the update would occur at the symbol rate:

$$\theta_{k+1} = \theta_k - \mu \left. \frac{dJ(e[k])}{d\theta} \right|_{\theta=\theta_k} = \theta_k - \mu \left. \frac{d}{d\theta} \left(\frac{1}{2} (i[k] - \hat{i}[k])^2 + (q[k] - \hat{q}[k])^2 \right) \right|_{\theta=\theta_k}$$

Using the chain rule for differentiation,

$$\theta_{k+1} = \theta_k - \mu \left((i[k] - \hat{i}[k]) \frac{di[k]}{d\theta} + (q[k] - \hat{q}[k]) \frac{dq[k]}{d\theta} \right) \Bigg|_{\theta=\theta_k}$$

$$\theta_{k+1} = \theta_k - \mu \left((i[k] - \hat{i}[k]) (-\hat{i}[k] \sin \theta - \hat{q}[k] \cos \theta) + (q[k] - \hat{q}[k]) (-\hat{q}[k] \sin \theta + \hat{i}[k] \cos \theta) \right) \Bigg|_{\theta=\theta_k}$$

Using the fact that $i[k] = \hat{i}[k] \cos \theta - \hat{q}[k] \sin \theta$ and $q[k] = \hat{q}[k] \cos \theta + \hat{i}[k] \sin \theta$,

$$\theta_{k+1} = \theta_k - \mu \left((i[k] - \hat{i}[k])(-q[k]) + (q[k] - \hat{q}[k])(i[k]) \right)_{\theta=\theta_k}$$

And finally, we have the computationally simple update for the phase offset to be

$$\theta_{k+1} = \theta_k - \mu (\hat{i}[k]q[k] - \hat{q}[k]i[k])$$

(c) What range of values would you recommend for μ ? 3 points.

For convergence, use small positive values for μ , e.g. 0.01 or 0.001, should be used.

When μ is zero, the update equation would not be able to update.

When μ is either negative or a large positive number, the update will diverge.

(d) This method can work with or without a training sequence. If you were to use a training sequence, which one would you use? Why? 6 points.

When using a training sequence, we'd know $i[k]$ and $q[k]$ in advance to compute the update

$$\theta_{k+1} = \theta_k - \mu (\hat{i}[k]q[k] - \hat{q}[k]i[k])$$

To generate the symbol amplitudes, one could use a maximal-length pseudo-noise to generate the bit stream and then use the constellation map to generate the equivalent symbol amplitudes.

Please see JSK Section 16.7 Baseband Derotation on page 384. Please note that angle in the above problem θ is actually $-\theta$ in JSK Section 16.7. If one multiplies both sides of the update equation by -1,

$$\theta_{k+1}^{JSK} = \theta_k^{JSK} - \mu (i[k]\hat{q}[k] - q[k]\hat{i}[k])$$

Problem 2.4. Potpourri. 24 points

- (a) What is the primary advantage of using symbol amplitudes of $-3d, -d, d$ and $3d$ for 4-level pulse amplitude modulation instead of $d, 3d, 5d$, and $7d$? 6 points.

First set of symbol amplitudes: peak power is $9d^2$ and average power is $5d^2$. Lecture Slides 13-3, 14-27,
Second set of symbol amplitudes: peak power is $49d^2$ and average power is $21d^2$. 15-10, 15-16
Symbol amplitudes of $-3d, -d, d$ and $3d$ have much lower peak and average power.

- (b) How will fifth-generation (5G) cellular communication systems be able to provide 10 times the average and peak bit rates of fourth-generation (4G) cellular communication systems? 6 points.

Lecture Slides 7-12, 13-3, 16-16;
HW 5.3, 6.3; Lab #5-6; Handout G

Bit rate is proportional to bandwidth.

For QAM, bit rate is Jf_{sym} where J is the number of bits/symbol and f_{sym} is the symbol rate, and the transmission bandwidth is $(1+\alpha)f_{\text{sym}}$ where α is the rolloff parameter for the raised cosine (or square root raised cosine).

Lecture Slide 16-16;
Lecture discussion;
Midterm #2 Spring 2014 Problem 2.3

5G and 4G communication systems divide a wide transmission band in narrow transmission bands, and each narrowband transmission carries a QAM signal. The narrowband QAM signals are transmitted in parallel.

To achieve 10x the average and peak bit rates, 5G will use 10x the transmission bandwidth. The 28 GHz millimeter wave band will one of the bands used to give the larger bandwidth.

- (c) For each communication subsystem below, advocate using either a discrete-time digital implementation or a continuous-time analog implementation.

Lectures 7, 8, 10; HW 5.3; Labs #5-6

- i. Baseband processing. 3 points.

Discrete-time digital implementation. Baseband processing occurs in both the transmitter and receiver. Baseband bandwidth W , which is half of the transmission bandwidth, is generally small enough to enable cost effective analog-to-digital and digital-to-analog converters running at sample rates $f_s > 2W$ and cost effective processing using programmable processors. A discrete-time digital implementation would allow a lot of flexibility in how the baseband signal could be processed.

- ii. Upconversion to carrier frequencies greater than 1 GHz. 3 points.

Continuous-time analog implementation. The transmission band is from $f_c - W$ to $f_c + W$ where f_c is the carrier frequency and W is the baseband bandwidth. For a discrete-time digital implementation, the digital-to-analog converter would have to run at a sampling rate of at least $2(f_c + W)$. When $f_c > 1$ GHz, the digital-to-analog converter, and programmable processors would not be able to keep up.

Lectures 7, 8, 10;
Lecture Slides 15-3 & 15-4;
HW 5.3, 6.1, 6.2; Labs #5-6;
Handout H

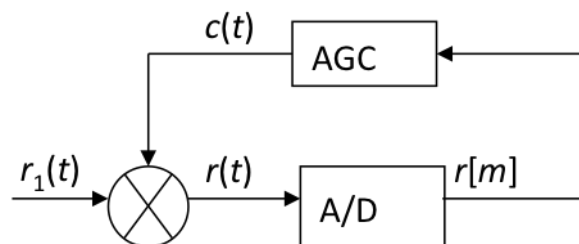
- (d) In the automatic gain control (AGC) block diagram given below, the analog-to-digital converter outputs $r[m]$ which is a signed integer of B bits. Give a formula that uses $r[m]$ and the adapted gain $c(t)$ to create a floating-point approximation of $r_1(t)$. This type of floating-point analog-to-digital conversion is used in practice, e.g. in cellular basestations. 6 points.

$$r(t) = c(t)r_1(t) \text{ and hence } r_1(t) = \frac{r(t)}{c(t)}$$

Gain $c(t)$ should not be 0: *Lecture Slide 16-5;*

$$r_1[m] = \frac{r[m]}{c[m]}$$

JSK Sections 6.7, 6.8, 9.3; HW 7.3



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 *Solutions 3.0*

Prof. Brian L. Evans

Date: December 10, 2018

Course: EE 445S

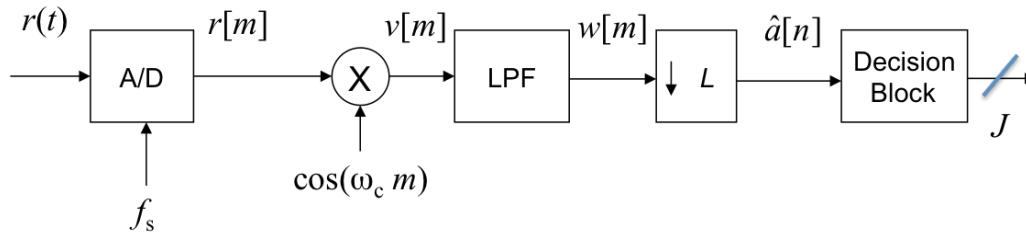
Name: _____ **Molière** _____
Last, First

- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Problem	Point Value	Your score	Topic
1	30		Bandpass PAM Receiver
2	27		QAM Communication Performance
3	25		Impedance Mismatch
4	18		Symbol Timing Recovery
Total	100		

Problem 2.1. Bandpass Pulse Amplitude Modulation Receiver. 30 points.

A bandpass pulse amplitude modulation (PAM) receiver is described as



where m is the sampling index and n is the symbol index, and where

- $\hat{a}[n]$ received symbol amplitude
- f_s sampling rate
- f_{sym} symbol rate
- $g[m]$ raised cosine pulse
- J bits/symbol
- L samples/symbol
- M number of levels, i.e. $M = 2^J$
- ω_c carrier frequency in rad/sample

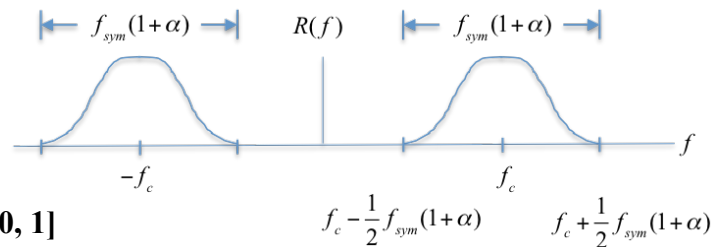
The only impairment being considered is additive thermal noise.

(a) Give a formula for the bit rate. 3 points.

$J f_{sym}$. Units: [bits/symbol] x [symbols/s] = [bits/s]

(b) Draw the spectrum of $r(t)$. See sketch on right.

What is the transmission bandwidth? 6 points.



$f_{sym} (1 + \alpha)$ where α is the raised cosine rolloff in $[0, 1]$

(c) Give a formula for the minimum sampling rate that would prevent aliasing of the frequencies in the transmitted bandpass PAM signal when processed in the receiver. 3 points.

$f_s > 2 f_{max}$ where $f_{max} = 2 f_c + \frac{1}{2} f_{sym} (1 + \alpha)$ where α is the rolloff factor in $[0, 1]$

(d) Describe the three roles that the lowpass filter plays. 6 points.

- Demodulating filter to extract the baseband signal
- Anti-aliasing filter for the downsampling by L operation
- Matched filter to filter out-of-band noise

(e) Give a formula for the optimal choice for the impulse response of the lowpass filter. What measure is being optimized? 6 points.

Impulse response of the optimal matched filter is $k g^* [L - m]$ where k is any non-zero gain.

(f) Give a fast algorithm for the Decision Block to decode the received M -PAM symbol amplitude $\hat{a}[n]$ into the most likely symbol of bits. Your algorithm should work for any finite M . 6 points.

Transmitted symbol amplitude value $a[n]$ is from the set $\{\dots, -3d, -d, d, 3d, \dots\}$. See below.

Received symbol amplitude $\hat{a}[n] = a[n] + v[n]$ where $v[n]$ is random variable due to noise.

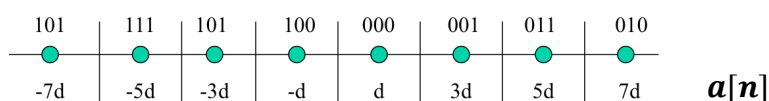
Divide-and-conquer algorithm will eliminate half of possible constellation points each step.

Find first bit: Compare $\hat{a}[n]$ against 0 and negate $\hat{a}[n]$ if negative

Find n th bit: Compare $\hat{a}[n]$ against $2^{J-n} d$ and if true, subtract $2^{J-n} d$ from $\hat{a}[n]$

Use bit pattern as unsigned index into a table of bit patterns corresponding to symbol amplitudes $d, 3d, \dots, 2^{J-1} d, -d, -3d, \dots, -2^{J-1} d$. (Not necessary for constellation map below).

Algorithm takes J comparisons, J subtractions and 1 memory read. $2J$ values are stored.

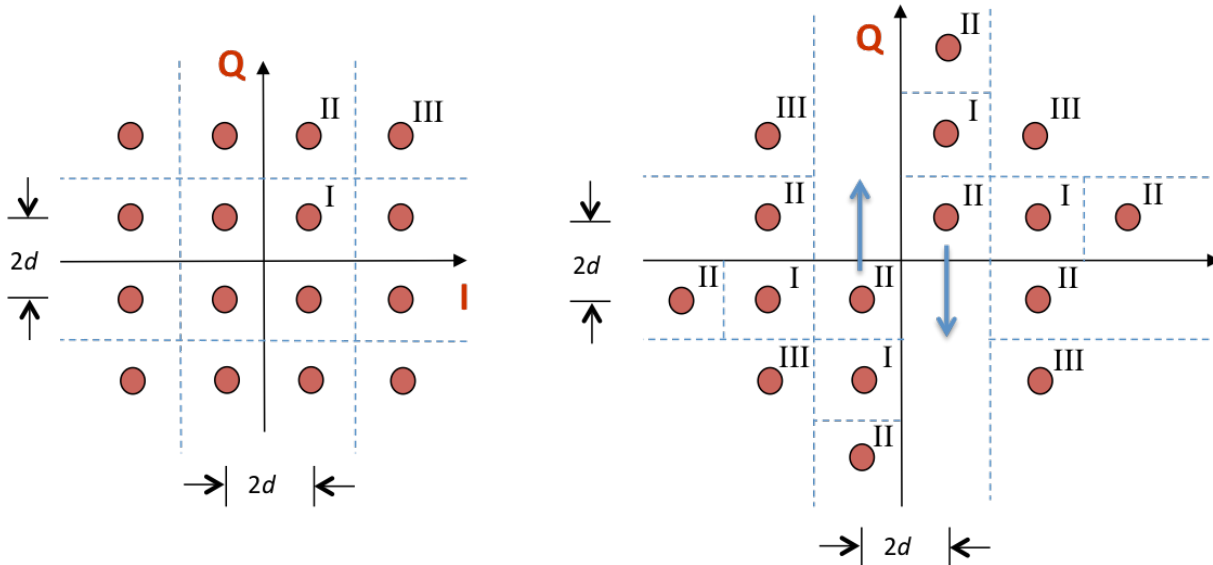


Note: The right constellation below is impractical. It consumes too much power and cannot be Gray coded. For best results in mapping the received symbol amplitude to a symbol of bits, one should find the closest constellation point in Euclidean distance. For rectangular-shaped constellations, such as the left constellation, thresholding would give the same minimum symbol error results as Euclidean distance but would lead to a fast divide-and-conquer algorithm using J comparisons for $M = 2^J$ levels (no multiplications needed).

Lectures 15 & 16
JSK Ch. 16
Lab #6; HW 6.3

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$26d^2$
(b) Average transmit power	$10d^2$	$15d^2$
(c) Draw the decision regions for the right constellation on top of the right constellation.		
(d) Number of type I regions	4	4
(e) Number of type II regions	8	8
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3 Q\left(\frac{d}{\sigma}\right) - \frac{9}{4} Q^2\left(\frac{d}{\sigma}\right)$	$3 Q\left(\frac{d}{\sigma}\right) - \frac{9}{4} Q^2\left(\frac{d}{\sigma}\right)$
(h) Express d/σ as a function of the Signal-to-Noise Ratio (SNR)	$\text{SNR} = \frac{10d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{10}}$	$\text{SNR} = \frac{15d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{15}}$

(i) In simulation, we can test the communication performance for different SNR settings by changing the variance of the additive Gaussian noise model. How would you use different SNR settings in a field test where the amount of noise power is not under our control? 3 points.

Received SNR = Average Transmit Signal Power / Average Noise Power

Answer #1: Adjust d to adjust the average transmit signal power.

Answer #2: Add noise at the receiver.

Problem 2.3. Impedance Mismatch. 25 points.

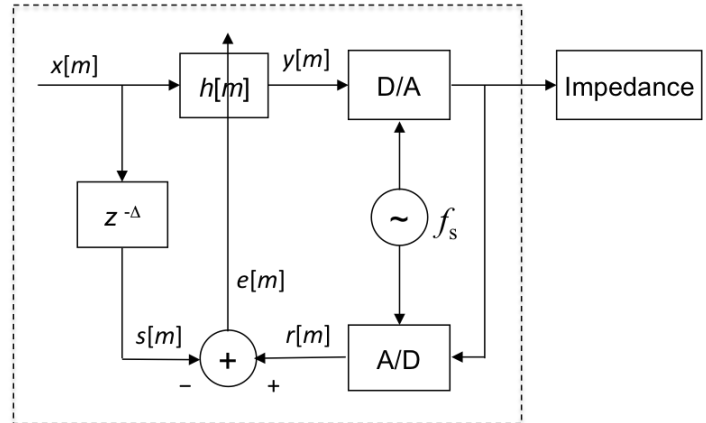
After a system starts up and begins to operate, the temperature inside the system will increase. Power management inside the system can also cause significant changes in the temperature of the system.

These temperature changes cause changes in the resistance, capacitance, and inductance in the system, which can in turn causes changes to the impedance mismatch to any wired connections to the system.

Impedance mismatch can be compensated by means of an adaptive finite impulse response (FIR) filter $h[m]$ that predistorts the signal $x[m]$ prior to digital-to-analog (D/A) conversion.

D/A and analog-to-digital (A/D) converters are synchronized via a common sampling clock.

Let N_h = Number of coefficients in $h[m]$



- (a) How could you determine a fixed value of Δ without the need for training? 6 points.

The delay Δ represents the delay through the adaptive filter, the D/A converter (1 sample) and the A/D converter (1 sample). The adaptive filter won't have symmetry in its impulse response about the midpoint; i.e., the group delay won't be a constant. The worst-case delay is $N_h - 1$ samples. So, $\Delta = (N_h - 1) + 2 = N_h + 1$.

- (b) Give an objective function $J(e[m])$. 3 points.

We would like to minimize the error between $r[m]$ and $s[m]$:

$$J(e[m]) = \frac{1}{2} e^2[m]$$

- (c) Give the update equation for the vector \vec{h} of FIR coefficients. 12 points.

$$\vec{h}[m+1] = \vec{h}[m] - \mu \left. \frac{dJ(e[m])}{d\vec{h}} \right|_{\vec{h}=\vec{h}[m]}$$

$$e[m] = r[m] - s[m] \text{ and } s[m] = x[m - \Delta]$$

Since $s[m]$ does not have any dependence on \vec{h} ,

$$\frac{dJ(e[m])}{d\vec{h}} = e[m] \frac{de[m]}{d\vec{h}} = e[m] \frac{dr[m]}{d\vec{h}}$$

Please see the next page for several ways to complete the derivation.

- (d) What range of values would you recommend for the step size μ ? Why? 4 points.

For convergence, use small positive values for μ , e.g. 0.01 or 0.001, should be used.

When μ is zero, the update equation would not be able to update.

When μ is either negative or a large positive number, the update will diverge.

Solution #1 for (c): Assume there is no impedance mismatch. In this case, $y[m]$ experiences a one sample delay through the D/A converter and a one sample delay through the A/D converter:

$$r[m] = y[m - 2]$$

$$y[m] = h_0 x[m] + h_1 x[m - 1] + h_2 x[m - 2] + \dots + h_{N_h-1} x[m - (N_h - 1)]$$

$$\frac{dJ(e[m])}{d\vec{h}} = e[m] \frac{dr[m]}{d\vec{h}} = e[m] \frac{dy[m - 2]}{d\vec{h}} = e[m] \vec{x}[m - 2]$$

$$\text{where } \vec{x}[m - 2] = [x[m - 2] \quad x[m - 3] \quad x[m - 4] \quad \dots \quad x[m - N_h - 1]]$$

Initial guess for the filter coefficients: $\vec{h} = [0 \ 0 \ \dots \ 0 \ 1]$ to match $\Delta = (N_h - 1) + 2 = N_h + 1$.

Solution #2 for (c): Model a finite number of reflections between A/D converter output and impedance by using an FIR filter $v(t)$. We'll use a discrete-time version of the FIR filter:

$$r[m] = v[m] * y[m] = v[m] * x[m] * h[m] = (v[m] * x[m]) * h[m]$$

We know $x[m]$. Let's assume that we know $v[m]$ for now.

Define $w[m] = v[m] * x[m]$ and hence $r[m] = w[m] * h[m]$:

$$r[m] = h_0 w[m] + h_1 w[m - 1] + \dots + h_{N_v-1} w[m - (N_v - 1)]$$

$$\frac{dJ(e[m])}{d\vec{h}} = e[m] \frac{dr[m]}{d\vec{h}} = e[m] \vec{w}[m]$$

$$\text{where } \vec{w}[m] = [w[m] \quad w[m - 1] \quad \dots \quad w[m - (N_h - 1)]]$$

We can solve for $v[m]$ as a least-squares filter or an adaptive filter.

Solution #2 Algorithm

$$\vec{v} = [0 \ 0 \ 1 \ 0 \ \dots \ 0 \ 0] \text{ which has } N_v \text{ elements and } \bar{\mu} = 0.01$$

Adapt $v[m]$ to learn reflections without pre-distortion filter enabled

$$y[m] = x[m]$$

$$q[m] = v_0 y[m] + v_1 y[m - 1] + \dots + v_{N_v-1} y[m - (N_v - 1)]$$

$$d[m] = r[m] - q[m]$$

$$\vec{y}[m] = [y[m] \quad y[m - 1] \quad \dots \quad y[m - (N_v - 1)]]$$

$$\vec{v}[m + 1] = \vec{v}[m] - \bar{\mu} d[m] \vec{y}[m]$$

$$\Delta = N_v - 1 \text{ and } \mu = 0.01 \text{ and } \vec{h} = [1 \ 0 \ 0 \ 0 \ \dots \ 0 \ 0] \text{ which has } N_h \text{ elements}$$

Adapt both the pre-distortion filter and the reflection model

$$y[m] = h_0 x[m] + h_1 x[m - 1] + h_2 x[m - 2] + \dots + h_{N_h-1} x[m - (N_h - 1)]$$

$$\vec{w}[m] = [w[m] \quad w[m - 1] \quad w[m - 2] \quad \dots \quad w[m - (N_h - 1)]] \text{ and } w[m] = v[m] * x[m]$$

$$e[m] = r[m] - s[m] \text{ and } s[m] = x[m - \Delta]$$

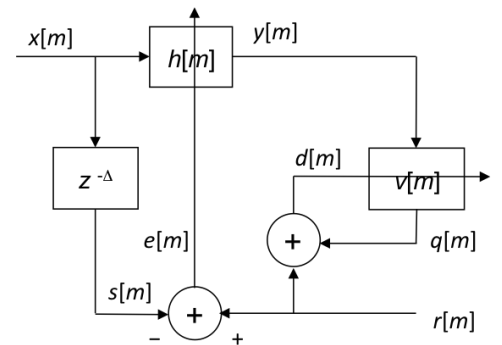
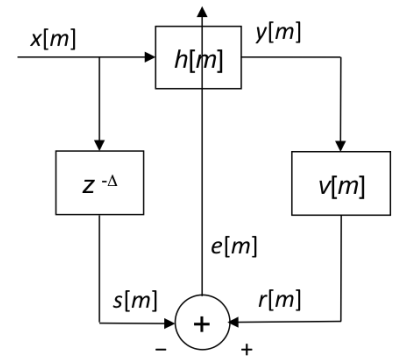
$$\vec{h}[m + 1] = \vec{h}[m] - \mu \left. \frac{dJ(e[m])}{d\vec{h}} \right|_{\vec{h}=\vec{h}[m]} = \vec{h}[m] - \mu e[m] \vec{w}[m]$$

$$q[m] = v_0 y[m] + v_1 y[m - 1] + v_2 y[m - 2] + \dots + v_{N_v-1} y[m - (N_v - 1)]$$

$$d[m] = r[m] - q[m]$$

$$\vec{y}[m] = [y[m] \quad y[m - 1] \quad \dots \quad y[m - (N_v - 1)]]$$

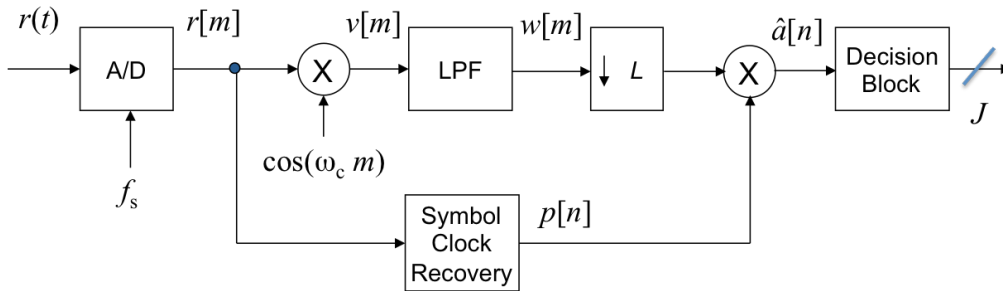
$$\vec{v}[m + 1] = \vec{v}[m] - \bar{\mu} d[m] \vec{y}[m]$$



Problem 2.4. Symbol Timing Recovery. 18 points

We add symbol timing recovery and correction to the bandpass pulse amplitude modulation (PAM) receiver in Problem 2.1:

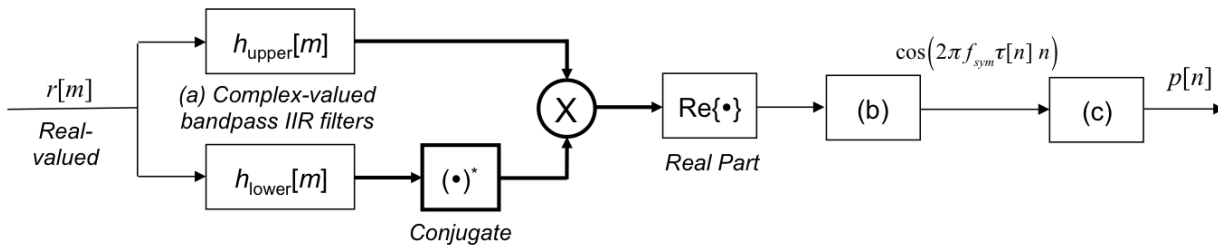
*Slide 16-10
Reader Appendix M
JSK Ch. 12
Lab #5
HW 6.2*



where m is the sampling index and n is the symbol index, and where

- $\hat{a}[n]$ received symbol amplitude
- $g[m]$ raised cosine pulse shape
- M number of levels, i.e. $M = 2^J$
- f_s sampling rate
- J bits/symbol
- ω_c carrier frequency in rad/sample
- f_{sym} symbol rate
- L samples/symbol

The block diagram of the symbol timing recovery and correction subsystem follows:



Upper filter $h_{upper}[m]$ locks onto (passes) continuous-time frequency $f_c + \frac{1}{2}f_{sym}$ and the lower filter $h_{lower}[m]$ locks onto (passes) $f_c - \frac{1}{2}f_{sym}$. **The thicker/bold lines indicate complex-valued signals.**

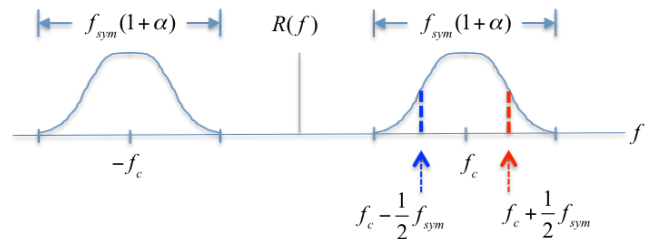
The sample timing offset is τ_s , which accumulates over L samples to give the symbol timing offset τ .

(a) Design a first-order complex-valued infinite impulse response (IIR) filter for $h_{upper}[m]$. 6 points.

Center frequency: $\omega_{upper} = 2\pi \frac{f_c + \frac{1}{2}f_{sym}}{f_s}$

Normally, we would use a pole at radius close to one.

For this case, we'll place the pole on the unit circle.



(b) Design this block to convert from the sampling rate to the symbol rate. 6 points.

Downsampler by L.

(c) Design a filter to smooth the rotation signal $\cos(2\pi f_{sym} \tau[n] n)$. 6 points.

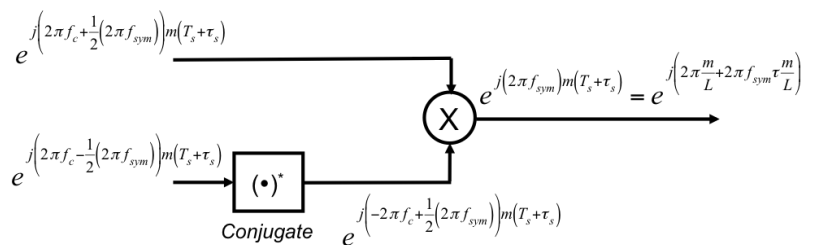
Let $x[n] = \cos(2\pi f_{sym} \tau[n] n)$.

Use a first-order lowpass IIR filter

$p[n] = c p[n - 1] + (1 - c) x[n]$

where the pole location $c = 0.9$.

When $m = L$, output is $\exp(j 2\pi f_{sym} \tau)$:



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 *Solution Set 1.0*

Prof. Brian L. Evans

Date: May 8, 2019

Course: EE 445S

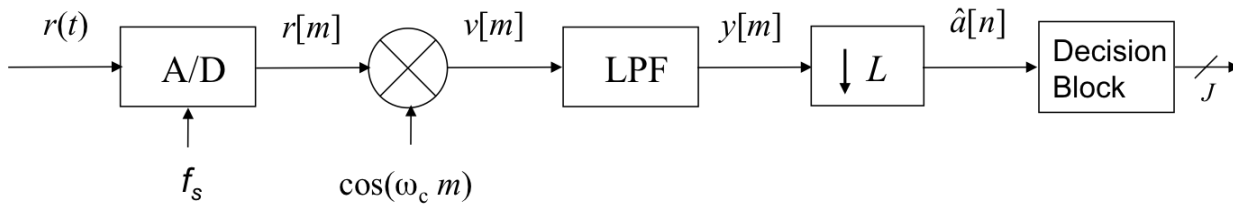
Name: _____
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Problem	Point Value	Your score	Topic
1	24		Bandpass PAM Receiver Tradeoffs
2	30		QAM Communication Performance
3	28		QAM Receiver Design
4	18		Bandpass PAM Receiver Decisions
Total	100		

Problem 2.1. Bandpass Pulse Amplitude Modulation Receiver Tradeoffs. 24 points.

A bandpass pulse amplitude modulation (PAM) receiver is described as



where m is the sampling index and n is the symbol index, and has **system parameters**

$\hat{a}[n]$ received symbol amplitude	f_s sampling rate	f_{sym} symbol rate
$g[m]$ raised cosine pulse	J bits/symbol	L samples/symbol
M number of levels, i.e. $M = 2^J$	ω_c carrier frequency in rad/sample	

The only impairment being considered is additive thermal noise $w(t)$.

Hence, $r(t) = s(t) + w(t)$ where $s(t)$ is the transmitted bandpass PAM signal.

(a) For the additive thermal noise $w(t)$,

- i. What is the probability distribution used to model the amplitude values of $w(t)$? *3 points.*

Gaussian distribution $N(0, \sigma^2)$. Each amplitude value is statistically independent. The mean is zero, and the variance σ^2 indicates the average noise power.

- ii. What is the justification for using that probability distribution? *3 points.*

Thermal noise is due to the random motion of particles due to temperature. For electromagnetic waves, the particles are electrons. As temperature increases, the random motion is faster and the noise power is higher. Each electron received is statistically independent from the transmitted waveform and other electrons. As the number of electrons tends to infinity, the additive effect would converge to a Gaussian distribution due to the Central Limit Theorem (a.k.a. Law of Large Numbers).

(b) If an optimal matched filter is used for the LPF,

- i. Which signal in the receiver is being optimized? *3 points.* **Estimated symbol ampl. $\hat{a}[n]$**
- ii. By what measure is the signal in part (b)i optimal? *3 points.* **Signal-to-noise ratio (SNR)**

(c) Give formulas for communication signal quality measures below in terms of system parameters:

- i. Bit rate. *3 points.* **$J f_{sym}$. Units: [bits/symbol] x [symbols/s] = [bits/s]**
- ii. Probability of symbol error. *3 points.*

$$P_{error}^{PAM} = 2 \frac{M-1}{M} Q\left(\frac{d}{\sigma} \sqrt{T_{sym}}\right) = 2 \frac{M-1}{M} Q\left(\frac{d}{\sigma} \sqrt{L T_s}\right)$$

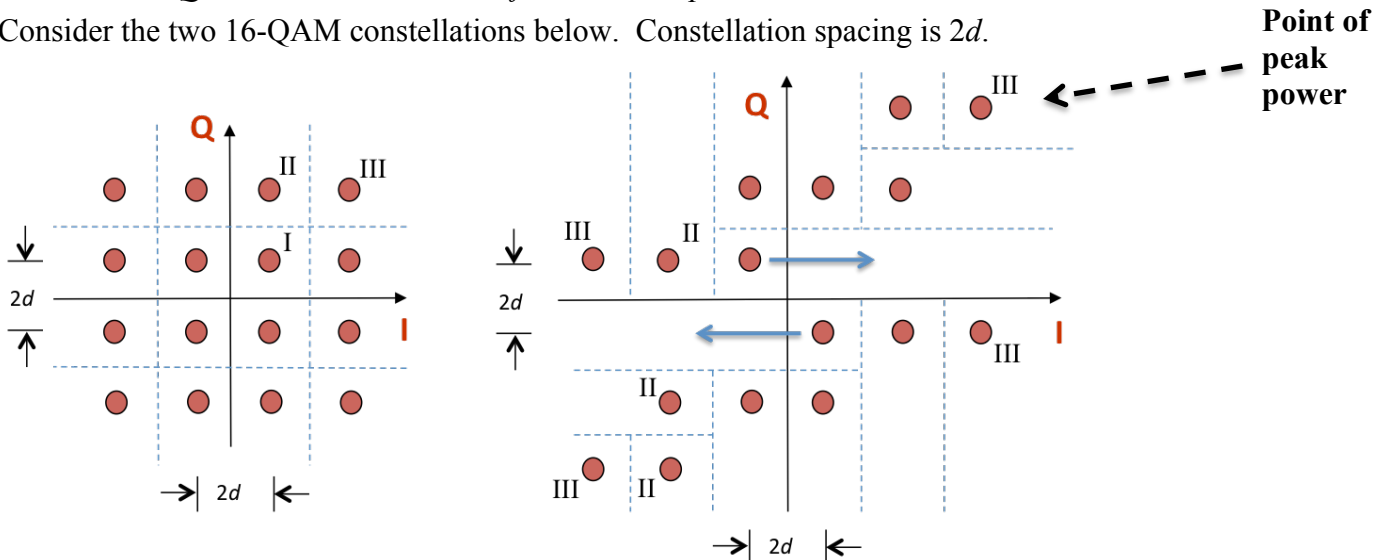
(d) Based on the formulas in (c), what's the impact on bit rate and probability of symbol error if

- i. Transmit power is increased. *3 points.* **On slide 14-27, transmit power is $\frac{1}{3}(M^2 - 1)d^2$.
 (1) M increases, d is same. Increase in bit rate. Mild increase in symbol error prob.
 (2) M is same, d increases. No effect on bit rate. Large decrease in symbol error prob.**
- ii. Number of samples/symbol, L , is increased. *3 points.* **When L increases, $f_{sym} = f_s / L$, decreases, and hence bit rate decreases. Large decrease in symbol error probability.**

Note: The right constellation is impractical. It consumes too much power and cannot be Gray coded. For lowest symbol error probability in mapping a received symbol amplitude to a symbol of bits, find the closest constellation point in Euclidean distance. For rectangular constellations, such as the one on the left, thresholding would give the same minimum symbol error results as Euclidean distance but leads to a fast divide-and-conquer algorithm using J comparisons for $M = 2^J$ levels (no multiplications).

Problem 2.2 QAM Communication Performance. 30 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. *Please fully justify your answers.*

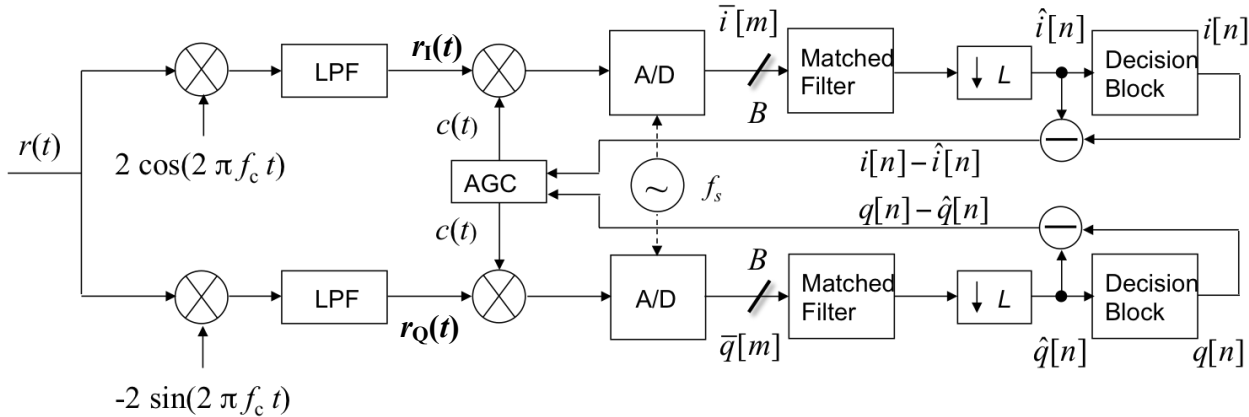
	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$50d^2$
(b) Average transmit power	$10d^2$	$20d^2$
(c) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation <i>that will minimize the probability of symbol error using such decision regions.</i> Avoid Type I regions because they have the highest probability of symbol error. We always have four Type III regions. So, maximize the number of Type II regions.		
(d) Number of type I regions	4	0
(e) Number of type II regions	8	12
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$\frac{11}{4}Q\left(\frac{d}{\sigma}\right) - \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$
(h) Express d/σ as a function of the Signal-to-Noise Ratio (SNR) in linear units	$SNR = \frac{10d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{SNR}{10}}$	$SNR = \frac{20d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{SNR}{20}}$

(i) In a 16-QAM receiver for the right constellation, an estimated symbol amplitude $-3d - j0.5d$. What is the decoded transmitted constellation point using

- Your constellation regions given above. *3 points.* $-3d - j0.5d$ would map to $d - jd$.
- Smallest Euclidean distance. *3 points.* **The closest constellation point to $-3d - j0.5d$ in Euclidean distance is $-3d + jd$.**

Problem 2.3. Quadrature Amplitude Modulation (QAM) Receiver Design. 28 points.

Some QAM receivers have a separate analog-to-digital (A/D) converter for the in-phase component and the quadrature component, as shown below.



System parameters: B bits at A/D output, $2d$ constellation spacing, f_s sampling rate, f_{sym} symbol rate, J bits/symbol, L samples/symbol, and M constellation points (i.e. $M = 2^J$). B is much greater than J .

Assume a rectangular, uniformly spaced, QAM constellation.

(a) If the signal-to-noise ratio (SNR) due to thermal noise in the system increases by 6 dB, and the system is matching the SNR due to thermal noise with the SNR due to quantization noise,

i. How many additional bits are possible for each A/D converter? 3 points.

Each 6 dB gain in SNR yields one extra A/D bit: $SNR_{dB} = C_0 + 6 B$ for B bits.

ii. What is the overall dB/bit increase in the system? 4 points.

2 bits have been added, i.e. 3 dB/bit. For QAM systems, $SNR_{dB} = C_1 + 3 B$ for B bits.

(b) What is the largest value of d that prevents clipping in the A/D converter? 3 points.

In the QAM receiver above, each A/D converter converts a received baseband PAM signal. Rectangular, uniformly spaced, QAM constellations have a PAM in-phase and quadrature constellations. In Problem 2.2, the left 16-QAM constellation has 4-PAM in-phase and 4-PAM quadrature constellations. 4-PAM symbol amplitudes are $-3d, -d, d$ and $3d$. Due to raised cosine pulse shaping, the amplitude of the baseband PAM signal lies in $(-6d, 6d)$.

Each A/D converter produces a signed B -bit integer whose values are from -2^{B-1} to $2^{B-1} - 1$.

Set $-6d = -2^{B-1}$ and solve for d . Or, to make d an integer, set $-8d = -2^{B-1}$ and solve for d .

(c) Receiver supports up to 16-QAM. For a 4-QAM training signal, develop an adaptive automatic gain control (AGC) algorithm. Gain $c(t)$ will be applied to the in-phase and quadrature channels. The gain sampled at the symbol time, $c[n] = c(n T_{sym})$, will be adapted every symbol period.

For the 16-QAM receiver, signals $i[n]$ and $q[n]$ represent transmitted symbol amplitudes from a 4-PAM constellation. Their possible values are $-3d, -d, d$ and $3d$. Their values are independent of the gain $c(t)$. Use the value of d computed in part (b).

When using a pulse shape with zero crossings at multiples of the symbol period other than the current symbol, e.g. a raised cosine pulse shape, and without any symbol timing error, $\hat{i}[n] = \bar{i}[Ln] = Q[r_I(n T_{sym}) c(n T_{sym})]$ and $\hat{q}[n] = \bar{q}[Ln] = Q[r_Q(n T_{sym}) c(n T_{sym})]$. We'll also assume $Q[x] \approx x$ to be able to compute derivatives.

See the explanation in part (c) above, which sets the stage for the following solutions.

- i. Give an objective function $J(n)$. 6 points.

Solution #1: $J(n) = \frac{1}{2}(i[n] - \hat{i}[n])^2 + \frac{1}{2}(q[n] - \hat{q}[n])^2$

Solution #2: $J(n) = \frac{1}{2}e^2[n]$ where $e[n] = (i[n] - \hat{i}[n]) + (q[n] - \hat{q}[n])$

Drawback: positive in-phase error can be offset by negative quadrature error, and vice-versa. **Solution #1 is better. Both are decision-directed.**

Solution #3: During 4-QAM training, each A/D converter will see a 2-PAM baseband signal. We can modify the “naïve” AGC algorithm from Johnson, Sethares & Klein, e.g. $J(n)$ in (6.15) and update in (6.16) on page 123. This “naïve” AGC performed the best of the three AGC algorithms evaluated in homework 7.3 w/r to additive noise.

$$J(n) = \text{Average} \left\{ |c[n]| \left(\left(\frac{\hat{i}^2[n]}{3} - d^2 \right) + \left(\frac{\hat{q}^2[n]}{3} - d^2 \right) \right) \right\}$$

- ii. Derive an update equation for gain $c[n]$. Compute all derivatives. Simplify result. 9 points

Solution #1: Minimize the objective function:

$$c[n+1] = c[n] - \mu \frac{dJ(n)}{dc[n]} = c[n] - \mu \left((i[n] - \hat{i}[n]) \left(-\frac{d\hat{i}[n]}{dc[n]} \right) + (q[n] - \hat{q}[n]) \left(-\frac{d\hat{q}[n]}{dc[n]} \right) \right)$$

$$c[n+1] = c[n] - \mu \left((\hat{i}[n] - i[n]) r_I[n] + (\hat{q}[n] - q[n]) r_Q[n] \right)$$

Since we do not know either $r_I[n] = r_I(n T_{sym})$ or $r_Q[n] = r_Q(n T_{sym})$, we’ll roll them into the step size parameter μ . That is, we’ll treat $r_I[n] = r_Q[n] = \text{constant}$.

$$c[n+1] = c[n] - \mu \left((\hat{i}[n] - i[n]) + (\hat{q}[n] - q[n]) \right)$$

Solution #2: Minimize the objective function:

$$c[n+1] = c[n] - \mu \frac{dJ(n)}{dc[n]} = c[n] - \mu e[n] \frac{de[n]}{dc[n]}$$

$$c[n+1] = c[n] - \mu e[n] (-r_I[n] - r_Q[n])$$

Since we do not know either $r_I[n] = r_I(n T_{sym})$ or $r_Q[n] = r_Q(n T_{sym})$, we’ll roll them into the step size parameter μ . That is, we’ll treat $r_I[n] = r_Q[n] = \text{constant}$.

$$c[n+1] = c[n] - \mu (-e[n]) = c[n] - \mu \left((\hat{i}[n] - i[n]) + (\hat{q}[n] - q[n]) \right)$$

After the approximations in solutions #1 and #2, they’ve become the same solution.

Solution #3: We modify (6.16) on page 123 in JSK:

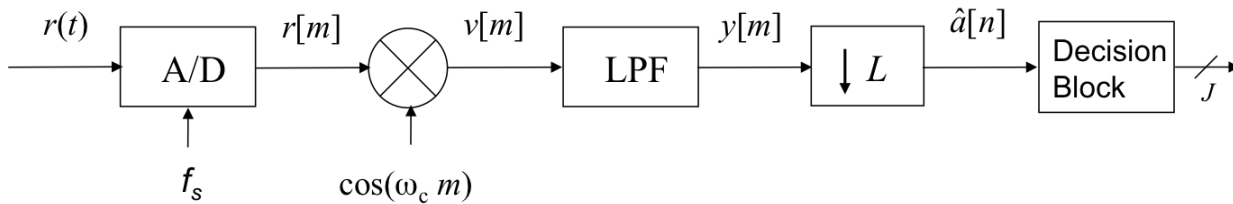
$$c[n+1] = c[n] - \mu \frac{dJ(n)}{dc[n]} = c[n] - \mu \text{Average} \left\{ \text{sign}(c[n]) \left((\hat{i}^2[n] - d^2) + (\hat{q}^2[n] - d^2) \right) \right\}$$

- iii. What range of values would you recommend for the step size μ ? Why? 3 points.

Use a small positive value for the step size μ , such as 0.01 or 0.001, for convergence of the steepest descent algorithm. A step size of zero will prevent any updates. A negative step size and a large positive step size will cause divergence.

Problem 2.4. Bandpass Pulse Amplitude Modulation Receiver Decisions. 18 points

A bandpass pulse amplitude modulation (PAM) receiver is described as



where m is the sampling index and n is the symbol index, and has system parameters

- | | | |
|--|--|-----------------------|
| $\hat{a}[n]$ received symbol amplitude | f_s sampling rate | f_{sym} symbol rate |
| $g[m]$ raised cosine pulse | J bits/symbol | L samples/symbol |
| M number of levels, i.e. $M = 2^J$ | ω_c carrier frequency in rad/sample | |

The only impairment being considered is additive thermal noise $w(t)$.

Hence, $r(t) = s(t) + w(t)$ where $s(t)$ is the transmitted bandpass PAM signal.

(a) Consider an 8-PAM bandpass transmitter.

- i. Draw an 8-PAM constellation map with Gray coding on the right. *6 points* ----->

Gray code means that the symbols of bits between two adjacent constellation points only differ by one bit. The idea is to minimize the number of bit errors when there is a symbol error so that the receiver can use error correcting codes to try to correct for the flipped bit.

- ii. Explain how you would build a lookup table for the constellation map in part (a)i. *3 points*

Use the symbol of bits as an unsigned integer index into a lookup table (array) of symbol amplitudes with entries $\{d, 3d, 7d, 5d, -d, -3d, -7d, -5d\}$.

symbol of bits		symbol amplitude a_n
010	●	$7d$
011	●	$5d$
001	●	$3d$
000	●	d
100	●	$-d$
101	●	$-3d$
111	●	$-5d$
110	●	$-7d$

(b) Consider an M -PAM bandpass receiver. The decision block quantizes the estimated symbol amplitude $\hat{a}[n]$ for M -PAM into a symbol of bits. Give formulas for the computational complexity as a function of M for each decision block quantization algorithm below. *9 points.*

- i. Compare $\hat{a}[n]$ against each constellation point in the transmitter constellation map.

M calculations, $O(M)$

- ii. Divide-and-conquer to discard half of the candidate constellation points each comparison.

$\log_2 M$ calculations, $O(\log_2 M)$

- iii. Determine the index of the closest constellation point using $\text{round}\left(\frac{\hat{a}[n]-d}{2d}\right)$, limit the unsigned index to a value between 0 and $M-1$ inclusive, and then use the unsigned index to find the symbol of bits in the lookup table for the constellation map.

Floating-point subtraction, division, and round operations. Constant time. $O(1)$.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 *Solution Set 2.0*

Prof. Brian L. Evans

Date: December 9, 2019

Course: EE 445S

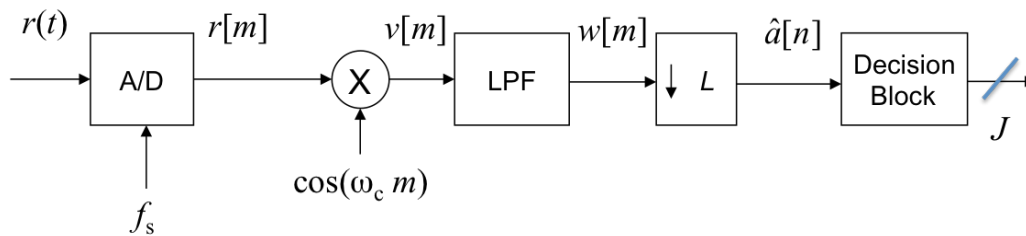
Name: _____
Last, First

- The exam is scheduled to last 75 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. **Disable all wireless access from your standalone computer system.**
- Please turn off all smart phones and other personal communication devices.
- Please remove headphones.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers unless instructed otherwise.** When justifying your answers, you may refer to the Johnson, Sethares & Klein (JSK) textbook, the Welch, Wright and Morrow (WWM) lab book, course reader, and course handouts. Please be sure to reference the page/slide number and quote the particular content in your justification.

Problem	Point Value	Your score	Topic
1	24		Bandpass PAM Receiver Tradeoffs
2	30		QAM Communication Performance
3	28		Channel Equalization
4	18		Total Harmonic Distortion
Total	100		

Problem 2.1. Bandpass Pulse Amplitude Modulation Receiver Tradeoffs. 24 points.

A bandpass pulse amplitude modulation (PAM) receiver is described as



*Lectures 13 & 14
JSK Ch. 8 & 11
Lab #5 & WWM Ch. 17
HW 5.2, 5.3, 6.1, 6.2
Midterm 2.1 F18 & Sp19
Handout P*

where m is the sampling index and n is the symbol index, and has **system parameters**

$a[n]$ transmitted symbol amplitude	$\hat{a}[n]$ received symbol amplitude
$2d$ constellation spacing	f_s sampling rate
$g[m]$ raised cosine pulse with rolloff α	J bits/symbol
M number of levels, i.e. $M = 2^J$	N_g symbol periods in $g[m]$
	f_{sym} symbol rate
	L samples/symbol
	ω_c carrier freq. in rad/sample

The only impairment is additive thermal noise $w(t)$ modeled as zero-mean Gaussian with variance σ^2 .

Hence, $r(t) = s(t) + w(t)$ where $s(t)$ is the transmitted bandpass PAM signal.

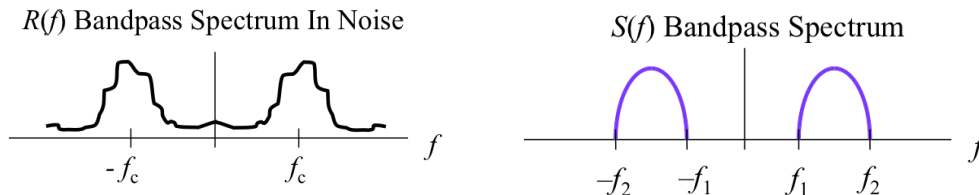
(a) Give formulas for communication signal quality measures below in terms of system parameters:

i. Bit rate. 3 points. $J f_{sym}$. Units: [bits/symbol] x [symbols/s] = [bits/s]

ii. Probability of symbol error. 3 points

$$P_{error}^{PAM} = 2 \frac{M-1}{M} Q\left(\frac{d}{\sigma} \sqrt{T_{sym}}\right) = 2 \frac{M-1}{M} Q\left(\frac{d}{\sigma} \sqrt{L T_s}\right)$$

(b) Draw the spectrum for $r(t)$. What is the transmission bandwidth in Hz? 6 points.



$$\text{Transmission Bandwidth} = f_{sym}(1 + \alpha); f_1 = f_c - \frac{1}{2}f_{sym}(1 + \alpha); f_2 = f_c + \frac{1}{2}f_{sym}(1 + \alpha)$$

(c) For the lowpass filter (LPF),

i. What three roles does it play? 3 points. **Demodulating filter, anti-aliasing filter for downsampling operation, and matched filter to maximize SNR at downsampler output**

ii. If its impulse response is equal to $g[m]$, give a formula for its bandwidth. 3 points

$$\text{Bandwidth} = 2\pi \frac{\frac{1}{2}f_{sym}(1+\alpha)}{f_s} = \frac{\pi}{L}(1 + \alpha) \text{ by using } f_s = L f_{sym}$$

(d) For the cascade of the lowpass filter (LPF) and downsampling by L ,

i. How many multiplications per second are required? 3 points

$L^2 N_g f_{sym}$, because the FIR filter has $N_g L$ coefficients and runs at rate $L f_{sym}$.

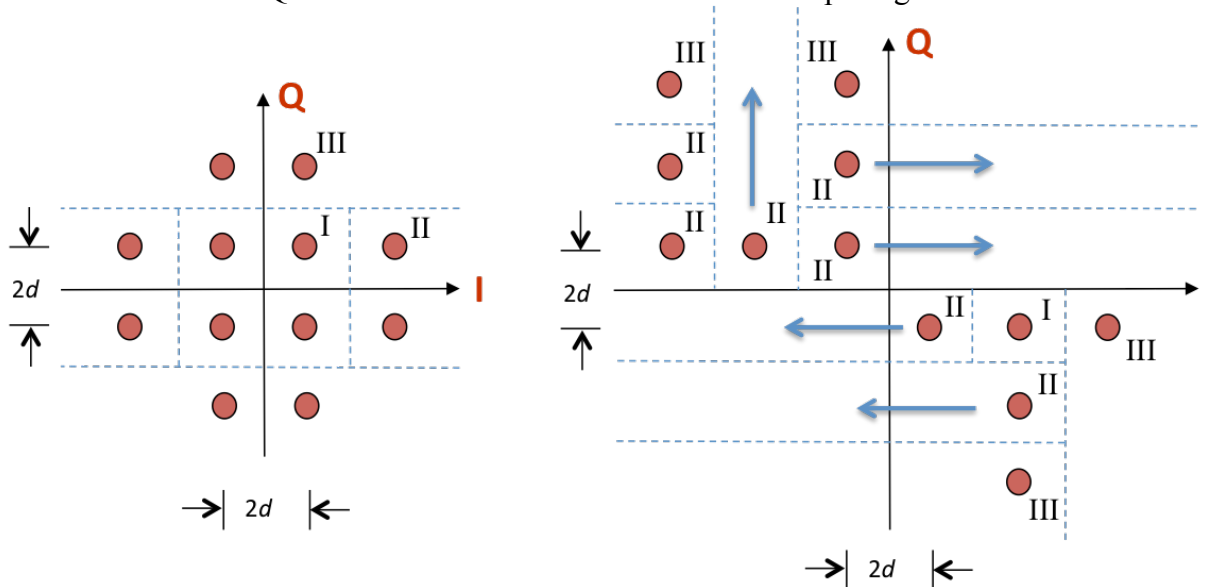
ii. What would the savings be if the cascade were realized in polyphase form? Why? 3 points.

The downsampler only keeps 1 sample of every block of L samples produced by the FIR filter. A polyphase form would only compute the 1 sample by using a bank of L filters, each with N_g coefficients, running at the symbol rate. Savings by a factor of L .

Note: The constellation on the right is impractical. It consumes too much power. For the lowest symbol error probability in mapping a received symbol amplitude to a symbol of bits, find the closest constellation point in Euclidean distance. For rectangular constellations, such as for 16-QAM, using rectangular decision regions would give the same minimum symbol error results as Euclidean distance but has a fast divide-and-conquer algorithm using J comparisons for $M = 2^J$ levels (no multiplications).

Problem 2.2 QAM Communication Performance. 30 points.

Consider the two 12-QAM constellations below. Constellation spacing is $2d$.



Lecture 15
Lecture 16
JSK Ch. 16
Lab #6
HW 6.3
Midterm 2.2
Problems in
Sp18, F18
and Sp19
Handout P

Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$10d^2$	$50d^2$
(b) Average transmit power	$\frac{22}{3}d^2 \approx 7.33d^2$	$\frac{248}{12}d^2 \approx 20.67d^2$
(c) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation that will minimize the probability of symbol error using such decision regions.		
(d) Number of type I regions	4	1
(e) Number of type II regions	4	7
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3 Q\left(\frac{d}{\sigma}\right) - Q^2\left(\frac{d}{\sigma}\right)$	$\frac{11}{4} Q\left(\frac{d}{\sigma}\right) - \frac{11}{6} Q^2\left(\frac{d}{\sigma}\right)$
(h) Express d/σ as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{22}{3} \left(\frac{d^2}{\sigma^2}\right)$ $\frac{d}{\sigma} = \sqrt{\frac{3}{22} \text{SNR}} \approx 0.369\sqrt{\text{SNR}}$	$\text{SNR} = \frac{62}{3} \left(\frac{d^2}{\sigma^2}\right)$ $\frac{d}{\sigma} = \sqrt{\frac{3}{62} \text{SNR}} \approx 0.0484\sqrt{\text{SNR}}$

(i) In a 12-QAM receiver for the right constellation, an estimated symbol amplitude is $-5d - jd$. What is the decoded transmitted constellation point using

- Your constellation regions given above. 3 points. $-d - jd$
- Smallest Euclidean distance. 3 points. $-5d + jd$

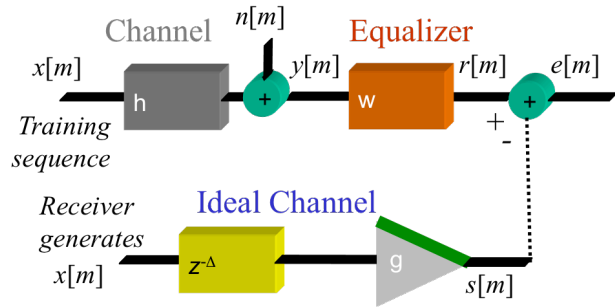
Problem 2.3. Channel Equalization. 28 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate.

The equalizer is a finite impulse response (FIR) filter with N real coefficients w_0, w_1, \dots, w_{N-1} :

$$r[m] = w_0 y[m] + w_1 y[m-1] + \dots + w_{N-1} y[m - (N-1)]$$

Channel model is an FIR filter with impulse response $h[m]$ in cascade with additive noise $n[m]$.



- (a) What two training sequences for $x[m]$ could you use? Why? 6 points.

Pseudo-noise sequences and chirp sequences have all discrete-time frequencies present in them. Either can be independently generated by the receiver. A pseudo-noise sequence can be generated using only logical operations and memory.

- (b) For one of the training sequences in part (a), describe how you would estimate the delay parameter Δ in the ideal channel model. 3 points.

The receiver can correlate the received signal $y[m]$ against the anticipated training sequence $x[m]$, and we can take the location of the first peak to be Δ samples.

- (c) For an adaptive FIR equalizer, **derive the update equation** for the vector of FIR coefficients \vec{w} for the objective function $J(e[m]) = |e[m]|$. Here, $\vec{w} = [w_0 \ w_1 \ \dots \ w_{N-1}]$. Please use the fact that $\frac{d}{dx}|x| = \text{sign}(x)$ except at $x = 0$ which we will extend to include $x = 0$. What value should $\text{sign}(x)$ take at $x = 0$ for the adaptive update? Let $\vec{w}[m] = [w_0[m] \ w_1[m] \ \dots \ w_{N-1}[m]]$. 12 points.

We would like to drive the error $e[m]$ to zero and hence minimize $J(e[m]) = |e[m]|$.

$$\vec{w}[m+1] = \vec{w}[m] - \mu \left. \frac{dJ(e[m])}{d\vec{w}} \right|_{\vec{w}=\vec{w}[m]} = \vec{w}[m] - \mu \text{sign}(e[m]) \vec{y}[m]$$

where $\vec{y}[m] = [y[m] \ y[m-1] \ \dots \ y[m-(N-1)]]$.

If error $e[m]$ reaches zero, then we'd stop the update, so we'll need $\text{sign}(0) = 0$.

- (d) Compare your answer in (c) with an adaptive least mean squares (LMS) equalizer. For the LMS approach, use $J(e[m]) = \frac{1}{2} e^2[m]$ which leads to the update equation

$$\vec{w}[m+1] = \vec{w}[m] - \mu e[m] \vec{y}[m]$$

where $\vec{y}[m] = [y[m] \ y[m-1] \ \dots \ y[m-(N-1)]]$. Would you use (c) or (d)? 4 points.

The update equation in (c) will scale $\vec{y}[m]$ by either $+\mu$, $-\mu$, or 0 . Large errors are treated the same way as non-zero small errors. In part (d), the offset is proportional to the error value. The update will initially make rapid progress and then slow down as it approaches the optimal answer. Parts (c) and (d) have similar complexity: they would compute either $\mu \text{sign}(e[m])$ or $\mu e[m]$ before multiplying that scalar value by the vector $\vec{y}[m]$.

- (e) For your answer in (c), what values of the step size (learning rate) μ would you use? 3 points.

Use a small positive value for the step size μ , such as 0.01 or 0.001, for convergence of the steepest descent algorithm. A step size of zero will prevent any updates. A negative step size and a large positive step size will cause divergence.

Harmonic Distortion: JSK Sec. 3.5; Slides 8-13 & 8-14

Comb Filters: Lab #7; WWM Ch. 10; In-Lecture Prob. 4; https://en.wikipedia.org/wiki/Comb_filter

Notch Filters: Lecture Slide 6-6; HW 3.1; Midterm Problems 1.1 Sp06, 1.3 F19, 2.3 F15

Problem 2.4. Total Harmonic Distortion. 18 points

Total harmonic distortion is a measure of the power in the harmonics of a fundamental frequency.

Design a discrete-time, linear time-invariant (LTI), infinite impulse response (IIR) comb filter to

- **Pass** harmonics of a 1 kHz tone, i.e. 2 kHz, 3 kHz, etc.
- **Not pass** 0 kHz or 1 kHz frequencies.

Assume a sampling rate of 48 kHz.

Please give the transfer function of your design and explain your reasoning to get there.

We'll use a cascade of an IIR comb filter, IIR DC notch filter, and IIR filter to notch 1 kHz.

- **IIR comb filter:** $y[n] = x[n] - \alpha y[n - K]$. Zero initial conditions to ensure LTI properties.

Transfer function: $H_{comb}(z) = \frac{1}{1 - \alpha z^{-K}}$ which has K poles with radius α^K and uniformly spaced in phase, where $|\alpha| < 1$ for bounded-input bounded-output (BIBO) stability.

Magnitude response: Peaks at integer multiples of f_s / K between $[-\frac{1}{2}f_s, \frac{1}{2}f_s]$.

- **IIR DC notch filter.** $y[n] = x[n] - x[n - 1] + r y[n - 1]$. Zero initial conditions for LTI.

Transfer function: $H_0(z) = \frac{1 - z^{-1}}{1 - r z^{-1}}$ which has a zero at $z = 1$ and pole at $z = r$.

- **IIR notch filter to remove $f_1 = 1$ kHz** which is a discrete-time frequency of $\omega_1 = 2\pi \frac{f_1}{f_s}$.

Second-order IIR filter with zeros on the unit circle at $-\omega_1$ and ω_1 ; poles at same angle.

Transfer function: $H_1(z) = \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})}$ with $z_1 = e^{j\omega_1}$, $z_2 = e^{-j\omega_1}$, $p_1 = r e^{j\omega_1}$, and $p_2 = r e^{-j\omega_1}$.

Overall transfer function: $H(z) = H_{comb}(z) H_0(z) H_1(z)$

Parameters: $K = 48$, $\alpha = 0.9$, and $r = 0.95$.

See the next page for MATLAB code and plots (not required to answer the question).

```

% UT Austin    EE 445S Real-Time DSP Lab
% Fall 2019    Prof. Brian L. Evans

```

```

% Midterm Question 2.4
% Design an IIR comb filter that
% (1) passes harmonics on 1 kHz
% (2) does not pass 0 or 1 kHz

```

```

fs = 48000;    % Sampling rate
N = 48000;    % 1 Hz accuracy in plots

```

```

% IIR Comb Filter Design
% Input-Output relationships in the time
% domain is  $y[n] = x[n] - \alpha * y[n-\text{delay}]$ 
delay = 48;    % IIR filter order
alpha = 0.9;
numerComb = 1;
denomComb = zeros(1, delay+1);
denomComb(1) = 1;
denomComb(delay+1) = -alpha;

```

```

% Design DC IIR notch filter to remove 0 kHz
z0 = 1;
p0 = 0.95;
numer0 = [1 -z0];
denom0 = [1 -p0];

```

```

% Design IIR notch filter to remove 1 kHz
f1 = 1000;
z1 = exp(-j*2*pi*f1/fs);
z2 = conj(z1);
p1 = 0.95*z1;
p2 = conj(p1);
numer1 = [1 -(z1+z2) z1*z2];
denom1 = [1 -(p1+p2) p1*p2];

```

```

% Combine the three filters into 1 section
% IIR Comb + DC Notch + 1 kHz notch
% Convolution implements polynomial multiplication
numer = conv(numerComb, numer0);
numer = conv(numer, numer1);
denom = conv(denomComb, denom0);
denom = conv(denom, denom1);

```

```

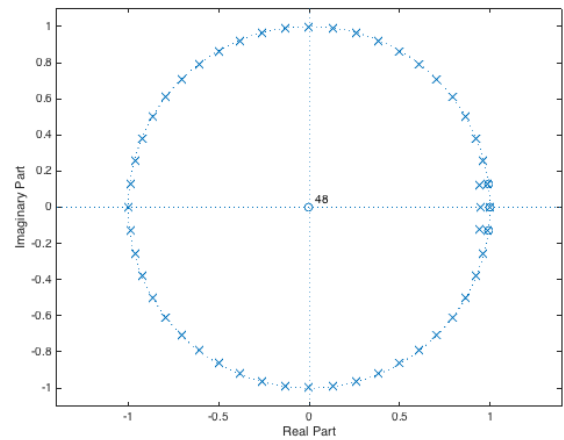
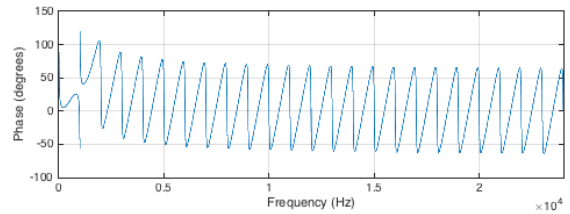
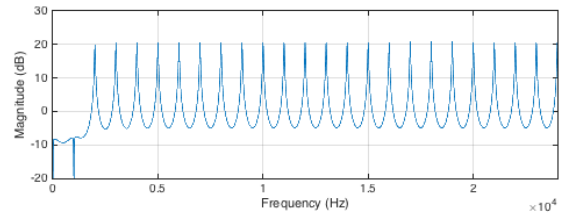
% Frequency response
figure;
freqz(numer, denom, N, fs);
ylim([-20 30]);

```

```

% Pole-zero diagram.
figure;
zplane(numer, denom);

```



The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 **Take-Home Exam** *Solution Set 1.0*

Prof. Brian L. Evans

Date: May 6, 2020

Course: EE 445S

Name: _____
Last, First

Please sign your name below to certify that you did not receive any help, directly or indirectly, on this test from another human other your instructor, Prof. Brian L. Evans, and to certify that you did not provide help, directly or indirectly, to another student taking this exam.

(please sign here) _____

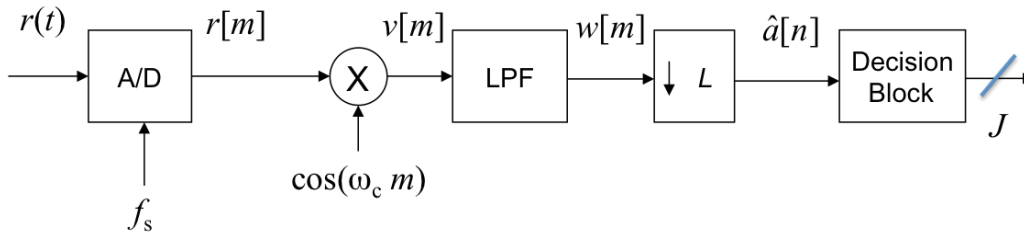
- **Take-home exam** is scheduled for Wednesday, May 6, 2020, from noon to 11:59pm.
 - The exam will be available on the course Canvas page at noon on May 6, 2020.
 - Please upload your solution to the course Canvas page by 11:59pm on May 6, 2020.
- **Perform all work on test.** All work should be performed on the exam. If more space is needed, then use the backs of the pages or scan in the extra page(s) with each problem.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification.
- **Internet access.** Yes, you may fully access the Internet when answering exam questions provided that you comply with the other instructions on this page.
- **Academic integrity.** You shall not receive help directly or indirectly on this test from another human except your instructor, Prof. Evans. You shall not provide help, directly or indirectly, to another student taking this exam.
- **Send questions to Prof. Evans.** You may send any questions or concerns about midterm #2 to Prof. Evans by e-mail at bevans@ece.utexas.edu.
- **Contact by Prof. Evans.** Prof. Evans might contact all students in the class during the exam through Canvas announcements. Please periodically monitor those announcements.

Problem	Point Value	Your score	Topic
1	24		Bandpass PAM Receiver Tradeoffs
2	30		QAM Communication Performance
3	28		Nonlinear Channel Equalization
4	18		Potpourri
Total	100		

Problem 2.1. Bandpass Pulse Amplitude Modulation Receiver Tradeoffs. 24 points.

A bandpass pulse amplitude modulation (PAM) receiver is described as

*Lectures 13 & 14
JSK Ch. 8 & 11
Lab #5 & WWM Ch. 17
HW 5.2, 5.3, 6.1, 6.2
Midterm 2.1 F19
Handout P*



where m is the sampling index and n is the symbol index, and has **system parameters**

$a[n]$ transmitted symbol amplitude	$\hat{a}[n]$ received symbol amplitude
$2d$ constellation spacing	f_s sampling rate
$g[m]$ raised cosine pulse with rolloff α	J bits/symbol
M number of levels, i.e. $M = 2^J$	N_g symbol periods in $g[m]$
	ω_c carrier freq. in rad/sample
	f_{sym} symbol rate
	L samples/symbol

The only impairment is additive thermal noise $w(t)$ modeled as zero-mean Gaussian with variance σ^2 .

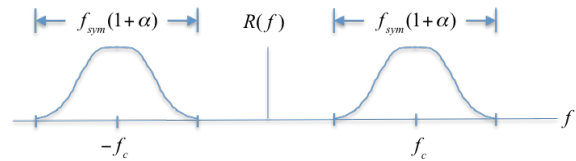
Hence, $r(t) = s(t) + w(t)$ where $s(t)$ is the transmitted bandpass PAM signal.

Give a formula for each quantity below in terms of the symbol rate f_{sym} and describe how much the quantity changes when the symbol rate increases.

(a) Bit rate in bits/s. 4 points. **Bit rate $J f_{sym}$ increases linearly when f_{sym} increases.**

(b) Transmission bandwidth in Hz. 4 points.

Transmission bandwidth $f_{sym} (1 + \alpha)$ increases linearly when f_{sym} increases.



(c) Sampling rate f_s . 4 points.

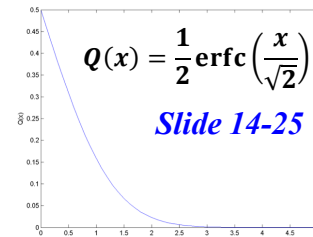
Sampling rate $L f_{sym}$ increases linearly when f_{sym} increases. (Alternately, $f_s > 2 f_{max}$ where $f_{max} = f_c + 1/2 f_{sym}$ and $f_c > f_{sym}$. The product $2 f_{max}$ increases linearly when f_{sym} increases.)

(d) Probability of symbol error. 4 points. **Recalling that $T_{sym} = 1 / f_{sym}$,**

$P_{error}^{PAM} = 2 \frac{M-1}{M} Q\left(\frac{d}{\sigma} \sqrt{T_{sym}}\right) = 2 \frac{M-1}{M} Q\left(\frac{d}{\sigma \sqrt{f_{sym}}}\right)$ increases when f_{sym} increases because $Q(x)$ is monotonically decreasing vs. $x \rightarrow$

Increase in P_{error}^{PAM} is proportional to $\sqrt{T_{sym}}$ in low SNR (small positive x) and faster than exponential in T_{sym} for high SNR

(large positive x). Slide 14-26: $Q(\sqrt{\rho}) \leq 1/\sqrt{2\pi} e^{-(\rho/2)}/\sqrt{\rho}$ for large positive ρ .



(e) Implementation complexity in multiplications per second. 8 points. **(r) means r multiplications/s.**

Analog-to-digital (A/D) converter would have at least an analog filter, sampler, quantizer (0).

Generate $\cos(\omega_c m)$ via lookup table (0), diff. equ. ($2 L f_{sym}$) or math library call ($30 L f_{sym}$).

Multiply $r[m]$ and $\cos(\omega_c m)$ to produce $v[m]$ ($L f_{sym}$)

LPF plus downsampler takes ($L^2 N_g f_{sym}$) in direct form or ($L N_g f_{sym}$) in polyphase form

Decision block algorithm: Euclidean distance ($2 M f_{sym}$) or using comparisons (0)

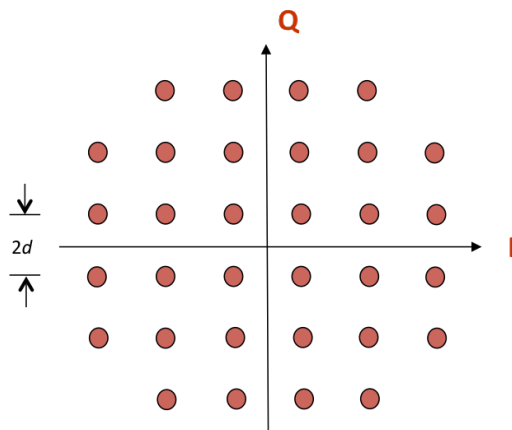
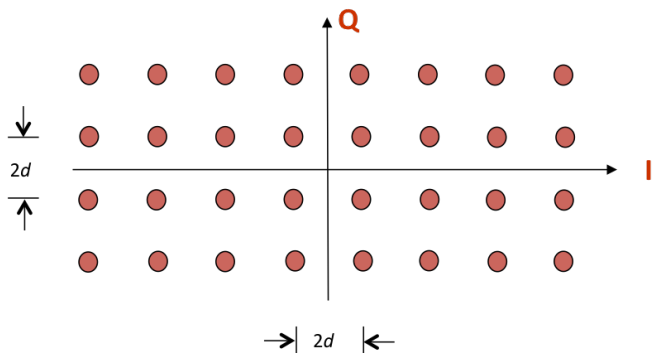
Implementation complexity increases linearly when f_{sym} increases

Problem 2.2 QAM Communication Performance. 30 points. Problem continues onto the next page.

Part I. Consider choosing a 32-QAM constellation.

Constellation #2: Power Efficient

Constellation #1: 8x4 rectangular



Constellation spacing is $2d$. Pulse shape energy is 1. Symbol time T_{sym} is 1s. $\rightarrow | 2d | \leftarrow$

Compute the peak and average power for constellation #2. 3 points each.

	Constellation #1	Constellation #2
(a) Peak transmit power	$58d^2$	$34d^2$
(b) Average transmit power	$26d^2$	$20d^2$

3 points

3 points

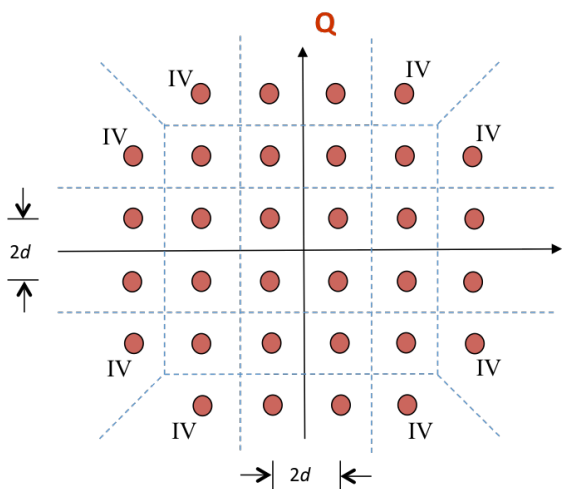
Constellation #2 has the lowest peak and average power possible, and is commonly used in practice.

Part II. For constellation #2, we're going to introduce a type IV constellation region. We'll also use it in Part III.

On the right, decision region boundaries are shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Type IV region has a diagonal line separating two nearest neighbors on a corner. It's a union of a type II region (finite in one dimension and infinite in the other) and half of a type III region (quarter plane).

We now have eight type IV regions instead of four type III regions (quarter planes at corner points).



	Constellation #1	Constellation #2
(c) Number of type I regions	12	16
(d) Number of type II regions	16	8
(e) Number of type III regions	4	0
(f) Number of type IV regions	0	8
(g) Symbol error probability for additive Gaussian noise, zero mean & variance σ^2	$\frac{13}{4} Q\left(\frac{d}{\sigma}\right) - \frac{21}{8} Q^2\left(\frac{d}{\sigma}\right)$	see next page
(h) Express d/σ as a function of Signal-to-Noise Ratio (SNR) in linear units	$SNR = 26 \left(\frac{d^2}{\sigma^2}\right)$ $\frac{d}{\sigma} = \sqrt{\frac{SNR}{26}} \approx 0.196 \sqrt{SNR}$	$SNR = 20 \left(\frac{d^2}{\sigma^2}\right)$ $\frac{d}{\sigma} = \sqrt{\frac{SNR}{20}} \approx 0.224 \sqrt{SNR}$

3 points

3 points

6 points

3 points

(a) We find the constellation points with the largest radii. There are 8 such points. The two in the upper right quadrant are $3d + j5d$ and $5d + j3d$. Their radius is $\sqrt{34d^2}$. Peak power is $34d^2$.

(b) We can use the fact that the constellation has quadrant symmetry to compute the average power in the upper right quadrant. The instantaneous power calculations for the constellation points are $2d^2, 10d^2, 10d^2, 18d^2, 26d^2, 26d^2, 34d^2, 34d^2$, which has an average of $20d^2$.

(g) As a shorthand notation, we'll use $q = Q\left(\frac{d}{\sigma}\right)$. The probabilities of correct detection for the type I, II, and III constellation regions, respectively, follow from [Lecture Slides 15-13 and 15-14](#):

$$\text{Type I. } P_{\text{correct}}^{\text{type I}} = (1 - 2q)^2$$

$$\text{Type II. } P_{\text{correct}}^{\text{type II}} = (1 - 2q)(1 - q)$$

$$\text{Type III. } P_{\text{correct}}^{\text{type III}} = (1 - q)^2$$

For the type IV constellation region, we'll use the constellation point $5d + j3d$:

$$P_{\text{correct}}^{\text{type IV}} = P\left(\left(v_I(nT_{\text{sym}}) > -d\right) \& \left(-d < v_Q(nT_{\text{sym}}) < v_I(nT_{\text{sym}}) + 2d\right)\right)$$

$$P_{\text{correct}}^{\text{type IV}} = P\left(\left(v_I(nT_{\text{sym}}) > -d\right) \& \left(v_Q(nT_{\text{sym}}) > -d\right) \& \left(v_Q(nT_{\text{sym}}) - v_I(nT_{\text{sym}}) < 2d\right)\right)$$

Let $z = v_Q(nT_{\text{sym}}) - v_I(nT_{\text{sym}})$ be a random variable. Assuming that $v_Q(nT_{\text{sym}})$ and $v_I(nT_{\text{sym}})$ are statistically independent, z has zero mean and its variance is equal to the variance of $v_Q(nT_{\text{sym}})$ plus the variance of $v_I(nT_{\text{sym}})$, per [Spring 2016 Midterm #2 Problem 2.3](#).

$$P_{\text{correct}}^{\text{type IV}} = P\left(\left(v_I(nT_{\text{sym}}) > -d\right) \& \left(v_Q(nT_{\text{sym}}) > -d\right) \& \left(z < 2d\right)\right)$$

$$P_{\text{error}}^{\text{total}} = 1 - P_{\text{correct}}^{\text{total}}$$

Given the difficulty in finding a closed-form solution for the symbol error probability using type I, II and IV constellation regions, we'll find the lower and upper bounds instead.

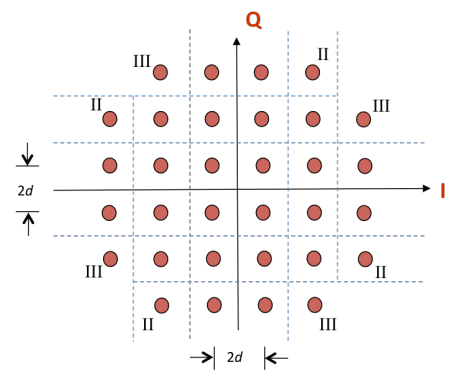
Lower bound: Using only type I, II, III regions, we have $\frac{16}{32}, \frac{12}{32}$ and $\frac{4}{32}$ probabilities for each region:

$$P_{\text{correct}}^{\text{total}} = \frac{1}{2}(1 - 2q)^2 + \frac{3}{8}(1 - 2q)(1 - q) + \frac{1}{8}(1 - q)^2$$

$$P_{\text{correct}}^{\text{total}} = \frac{1}{2}(1 - 4q + 4q^2) + \frac{3}{8}(1 - 3q + 2q^2) + \frac{1}{8}(1 - 2q + q^2)$$

$$P_{\text{correct}}^{\text{total}} = 1 - \frac{27}{8}q + \frac{23}{8}q^2$$

$$P_{\text{error}}^{\text{total}} = 1 - P_{\text{correct}}^{\text{total}} = \frac{27}{8}q - \frac{23}{8}q^2 = \frac{27}{8}Q\left(\frac{d}{\sigma}\right) - \frac{23}{8}Q^2\left(\frac{d}{\sigma}\right)$$



Upper bound: Using only type I and II regions, which would leave quarter plane gaps in each quadrant at $5d + j5d, -5d + j5d$, etc.,

$$P_{\text{correct}}^{\text{total}} = \frac{1}{2}(1 - 2q)^2 + \frac{1}{2}(1 - 2q)(1 - q) = 1 - \frac{7}{2}q + 3q^2$$

$$P_{\text{error}}^{\text{total}} = 1 - P_{\text{correct}}^{\text{total}} = \frac{7}{2}q - 3q^2 = \frac{7}{2}Q\left(\frac{d}{\sigma}\right) - 3Q^2\left(\frac{d}{\sigma}\right) = \frac{28}{8}Q\left(\frac{d}{\sigma}\right) - \frac{24}{8}Q^2\left(\frac{d}{\sigma}\right)$$

Summary: There is very little difference between the lower and upper bounds. Almost all of the symbol error probability is in the type I and type II regions.

Note: The symbol error probability expression has the form $c_0 Q\left(\frac{d}{\sigma}\right) - c_1 Q^2\left(\frac{d}{\sigma}\right)$. As $\sigma \rightarrow \infty$, knowing that d is positive, the symbol error probability goes to $c_0 Q(0) - c_1 Q^2(0)$ where $Q(0) = 0.5$. Generally, $c_0 > c_1$. As $\sigma \rightarrow 0$, the symbol error probability goes to 0.

Randomly guessing a M -PAM symbol amplitude gives an symbol error probability of $\frac{M-1}{M}$.

Part III. In the receiver, finding the nearest constellation point to the received QAM symbol amplitude using Euclidean distance provides high-accuracy in the symbol detection. Complexity is proportional to the number of levels $M = 2^J = 32$ where $J=5$ is the number of bits in a symbol: 64 multiplications, 96 additions, and 64 memory reads in words for each (I, Q) symbol amplitude.

We introduced the type IV region in part II to unlock a low-complexity divide-and-conquer method that is just as accurate as using Euclidean distance but only needs to use comparison operations. Describe the method and compare its complexity with the Euclidean distance method. *9 points.*

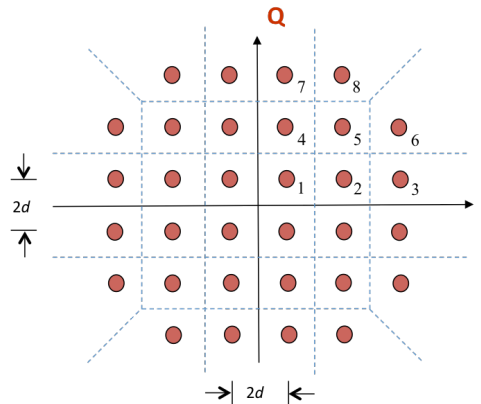
Let (i, q) be the received symbol amplitude.

Midterm 2.2(i) Sp16
Midterm 2.1(f) F18

1. If $i > 0$, then in right-half plane; otherwise, in left-half plane
2. If $q > 0$, then in upper half plane; otherwise, in bottom half plane

Remaining steps are for the upper right quarter plane (i.e. for $i > 0$ and $q > 0$):

3. If $i > 4d$,
 - If $q < 2d$, then point 3
 - Else If $q > i$, then point 8
 - Else point 6
- Else If $i < 2d$,
 - If $q > 4d$, then point 7
 - Else If $q < 2d$, then point 1
 - Else point 4
- Else If $q > 4d$, then point 8
- Else If $q < 2d$, then point 2
- Else point 5



The above divide-and-conquer algorithm uses 4, 5 or 6 comparisons (5.375 on average) and takes $J+1$ comparisons and memory reads.

Euclidean distance method computes the Euclidean distance squared from the received symbol amplitude (i, q) to each symbol amplitude in the constellation map (I, Q) : $(i - I)^2 + (q - Q)^2$. The square root is an unnecessary calculation—it doesn't change the ordering among the distances.

Its complexity is 2×2^J multiplications, 3×2^J additions, 2×2^J memory reads and 2^J comparisons.

In practice, QAM constellations are as large as $J = 8$ in cellular and Wi-Fi, and $J = 15$ for ADSL. The difference between linear and exponential complexity can be significant.

Problem 2.3. Nonlinear Channel Equalization. 28 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate.

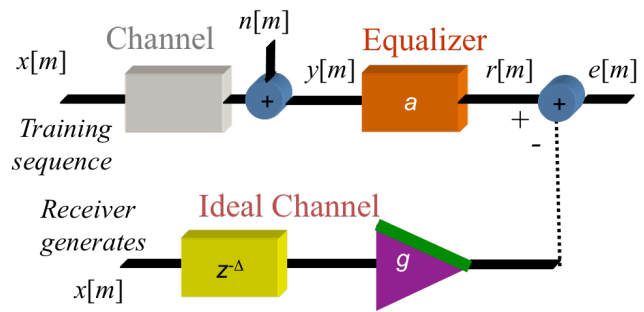
The channel has significant **nonlinear distortion**.

We're going to use a **nonlinear equalizer** of the form

$$r[m] = a_0 + a_1 y[m] + a_2 y^2[m] + \dots + a_N y^N[m]$$

where $a_0, a_1, a_2, \dots, a_N$ are real-valued coefficients.

The channel model includes additive noise $n[m]$ that has a Gaussian distribution with zero mean and variance σ^2 .



- (a) Give a training sequence for $x[m]$ that you would use? Why? 3 points.

Pseudo-noise sequences and chirp sequences have all discrete-time frequencies present in them. Either can be independently generated by the receiver. A pseudo-noise sequence can be generated using only logical operations and memory.

- (b) For one of the training sequences in part (a), describe how you would estimate the transmission delay parameter Δ in the ideal channel model. 3 points.

The receiver can correlate the received signal $y[m]$ against the anticipated training sequence $x[m]$, and we can take the location of the first peak to be Δ samples.

- (c) What objective function would you use? Why? 6 points.

We would like to minimize the error between $r[m]$ and $s[m]$: $J(e[m]) = \frac{1}{2} e^2[m]$

By driving $e^2[m]$ to zero, we also drive $e[m]$ to zero.

Another advantage is that the amount of the offset in the parameter update at each iteration will be proportional to the error—the update will rapidly converge for large errors and slowly converge for small errors, provided that the step size has been chosen correctly.

Another advantage is relatively low computational complexity.

- (d) For an adaptive nonlinear equalizer, **derive the update equation** for the vector of coefficients \vec{a} for the objective function in part (c). Here, $\vec{a} = [a_0 \ a_1 \ a_2 \ \dots \ a_N]$. 12 points.

$$e[m] = r[m] - s[m]$$

$$\vec{a}[m+1] = \vec{a}[m] - \mu \left. \frac{dJ(e[m])}{d\vec{a}} \right]_{\vec{a}=\vec{a}[m]} = \vec{a}[m] - \mu e[m] \vec{y}[m]$$

$$\text{where } \vec{y}[m] = [1 \ y[m] \ y^2[m] \ \dots \ y^N[m]]$$

- (e) For your answer in (c), what values of the step size (learning rate) μ would you use? 4 points.

Use a small positive value for the step size μ , such as 0.01 or 0.001, for convergence of the steepest descent algorithm. A step size of zero will prevent any updates. A negative step size and a large positive step size will cause divergence.

Problem 2.4. Potpourri. 18 points

- (a) A handheld garage door opener transmits a binary request to an automatic garage door control unit. The binary request is to open the garage door if closed, and close the garage door if open.

Please describe the signals based on PN sequences that the garage door opener would transmit to indicate that it is sending its binary request to the automatic garage door control unit (receiver)?

The transmission/reception would have to be

- Reliable -- a very high probability of correct detection
- Secure -- nearly impossible for an opener meant for another unit to work on your garage door.

Lab #4; HW 4.1, 4.2, 4.3, 5.2; Wikipedia "garage door opener"

Initialization. At manufacturing, the opener and control unit are initialized with the same random 32-bit number that will be the value of the pseudo-noise (PN) shift register generator in both. This will make it highly unlikely for openers of the same brand to open your garage.

Transmission/Reception: The PN generator would be used to create a self-synchronizing data scrambler in the opener and descrambler in the control unit. The opener would scramble a text message (converted to bits) that contains a 16-bit number of how many times the opener has been pressed in its lifetime. The number must be equal to or higher than that tracked by the control unit. This prevents an eavesdropper from playing back a recorded transmission to open your garage. An opener will repeat the transmission 10 times to improve reliability.

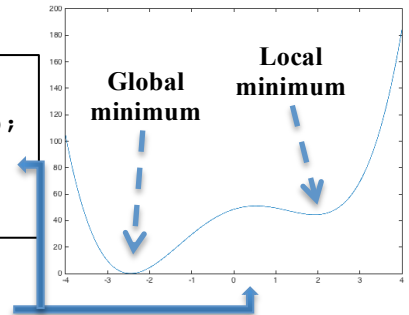
- (b) Steepest descent algorithm to minimize an objective (cost) function.

Steepest descent algorithm for parameter x is

$$x[m + 1] = x[m] - \mu \left. \frac{dJ(x)}{dx} \right|_{x=x[m]}$$

to minimize the objective function $J(x)$ given a positive value for step size (learning rate) μ .

```
x = -4 : 0.001 : 4;
% p = [1 0 -10 10 48.5];
p = poly( [-3, -1, 1, 3] );
p(4) = 10;
p(5) = p(5) + 39.5;
Jx = polyval(p, x);
plot(x, Jx);
```



- Draw an objective function that has at least one global minimum value and at least one local minimum that is not a global minimum. 3 points.
- Give a way to determine if a steepest descent algorithm converged to an answer. 3 points.
Method #1: Since $J'(x) = 0$ at a minimum, $|J'(x)| \leq 10^{-7}$ for 10 iterations in a row.
Method #2: For 10 iterations in a row, the relative change in $x[m]$ is below a threshold $|x[m + 1] - x[m]| \leq 10^{-4} |x[m]| + \epsilon$ where small $\epsilon > 0$ is added in case $x[m] \approx 0$
- How would you use multiple steepest descent algorithms running in parallel to reach a solution with a lower objective (cost) function? 3 points.
Start each steepest descent algorithm with a different randomized initial guess and then choose the answer with smallest objective function value.
- If you could only run one steepest descent algorithm, how could you modify it to get out of a possible local minimum? 3 points.

Use the test in part (b)ii to determine if the algorithm might be a local minimum.

Answer #1: Negate step size μ to head toward a maximum value and then switch back. Downside is that it will take a long time to get out of a local minimum because near a local minimum, $J'(x)$ is close to zero, and at a local minimum, $J'(x)$ is zero.

Answer #2: Temporarily switch to a large step size μ and then switch back.

Answer #3: Set the parameter to a random number and continue updating. Once the iteration converges, compare the objective function value with the previous solution.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 **Take-Home Exam Solutions 3.0**

Prof. Brian L. Evans

Date: December 7, 2020

Course: EE 445S

Name: _____ **Scallon,** _____ **Rob** _____
Last, First

Please sign your name below to certify that you did not receive any help, directly or indirectly, on this test from another human other your instructor, Prof. Brian L. Evans, and to certify that you did not provide help, directly or indirectly, to another student taking this exam.

(please sign here) _____

- **Take-home exam** is scheduled for Monday, Dec. 7, 2020, 10:30am to 11:59pm.
 - The exam will be available on the course Canvas page at 10:30am on Dec. 7, 2020.
 - Your solutions can be on notebook paper, or on the test and your own paper, or whatever. This means that you won't have to print the test to complete the test.
 - Please upload your solution as a single PDF file to the course Canvas page by 11:59pm on Oct 14, 2020.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content for your justification. You may reference homework solutions, exam solutions, lecture slides, textbooks, Internet resources, etc.
- **Internet access.** Yes, you may fully access the Internet when answering exam questions provided that you comply with the other instructions on this page.
- **Academic integrity.** You shall not receive help directly or indirectly on this test from another human except your instructor, Prof. Evans. You shall not provide help, directly or indirectly, to another student taking this exam.
- **Send questions to Prof. Evans.** You may send questions or concerns about this midterm exam during the test to Prof. Evans via Canvas or by e-mail at bevans@ece.utexas.edu.
- **Contact by Prof. Evans.** Prof. Evans might contact all students in the class during the exam through Canvas announcements. Please periodically monitor those announcements.

Problem	Point Value	Your score	Topic
1	24		PAM Symbol Error Probability
2	27		QAM Communication Performance
3	24		Interference Cancellation
4	25		Communication System Design Tradeoffs
Total	100		

Problem 2.1. PAM Symbol Error Probability. 24 points.

Consider a three-level baseband pulse amplitude modulation (PAM) system:

- $2d$ constellation spacing f_s sampling rate
- f_{sym} symbol rate $g[m]$ raised cosine pulse with rolloff α
- J bits/symbol L samples/symbol
- M number of levels, i.e. $M = 2^J$ N_g symbol periods in $g[m]$
- T_s sampling time T_{sym} symbol time.

The 3-PAM constellation is shown on the right (i.e. $M = 3$).

This problem is asking you to analyze the 3-PAM system and enter your results in the table below to compare against 2-PAM and 4-PAM systems.

For this problem, assume that $T_{sym} = 1$ s and hence $f_{sym} = 1$ Hz.

(a) What is the bit rate for the 3-PAM system? Why? 4 points.

Bit rate = $J f_{sym}$. With $f_{sym} = 1$ Hz and $J = \log_2 M = \log_2 3 \approx 1.585$, the bit rate is 1.585 bits/s.

(b) Give a symbol of bits for each symbol amplitude on the 3-PAM constellation. Make sure that the bit encoding for adjacent symbols only differs by one bit (Gray coding). 6 points.

Shown in bold on the constellation map above.

(c) Derive 3-PAM symbol error probability. Thresholds are at $-d$ and d . 8 points.

The constellation regions are $(-\infty, -d)$, $(-d, d)$, and (d, ∞) .

The region $(-d, d)$ is an inner region, and the others are outer regions. (Lecture Slide 15-11).

Assuming each symbol is equally likely to be transmitted, we use Lecture Slide 15-11 to find

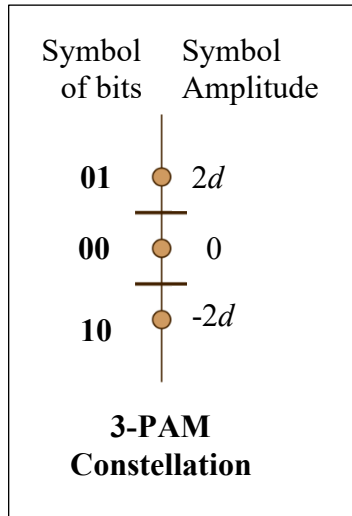
$$P(e) = \frac{1}{3} P_I(e) + \frac{2}{3} P_O(e) = \frac{1}{3} \left(2Q \left(\frac{d}{\sigma} \right) \right) + \frac{2}{3} \left(Q \left(\frac{d}{\sigma} \right) \right) = \frac{4}{3} Q \left(\frac{d}{\sigma} \right)$$

(d) Derive the transmitted maximum power, average power, and peak-to-average power ratio. 6 points.

The maximum symbol amplitude is $2d$ Volts.

Maximum (peak) power is proportional to the maximum amplitude squared., which is $4d^2$.

The average power, assuming each symbol is equally likely, is $(4d^2 + 0 + 4d^2) / 3 = (8/3) d^2$.

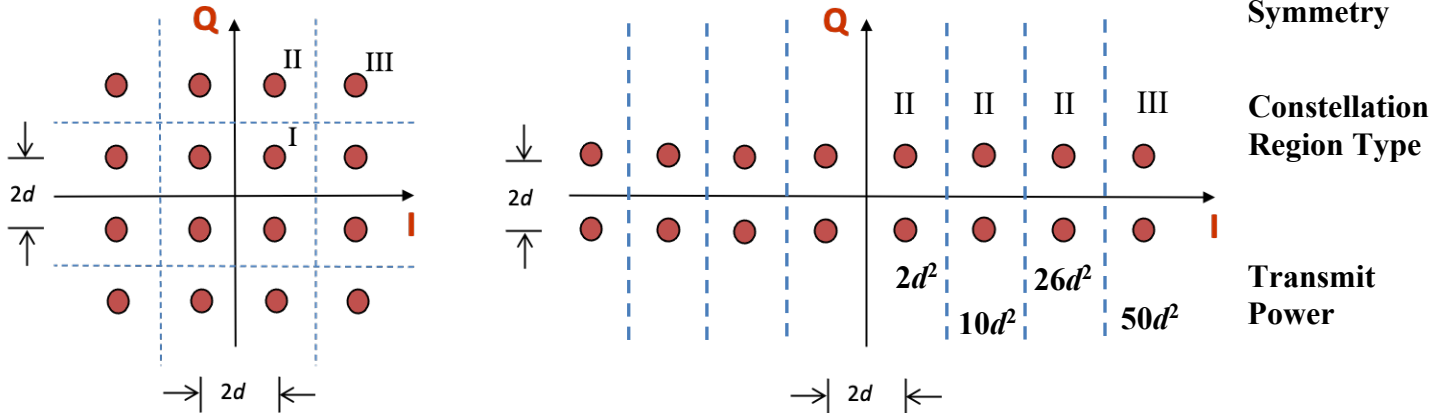


M	Bit rate	Symbol Error Probability	Peak Power	Average Power	Peak-to-Average Power Ratio
2	1 bit/s	$Q \left(\frac{d}{\sigma} \right)$	d^2	d^2	1.0
3	1.585 bits/s	$\frac{4}{3} Q \left(\frac{d}{\sigma} \right)$	$4 d^2$	$\frac{8}{3} d^2$	1.5
4	2 bits/s	$\frac{3}{2} Q \left(\frac{d}{\sigma} \right)$	$9 d^2$	$5 d^2$	1.8

Note: All entries in the above table assume that $T_{sym} = 1$ s and hence $f_{sym} = 1$ Hz.

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$50d^2$
(b) Average transmit power	$10d^2$	$((2+10+26+50)/4) d^2 = 22d^2$
(c) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation <i>that will minimize the probability of symbol error using such decision regions.</i>		
(d) Number of type I regions	4	0
(e) Number of type II regions	8	12
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$\frac{11}{4}Q\left(\frac{d}{\sigma}\right) - \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$
(h) Express d/σ as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{10d^2}{\sigma^2} \Rightarrow \frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{10}}$	$\text{SNR} = \frac{22d^2}{\sigma^2} \Rightarrow \frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{22}}$

See below for work

(g) *Approach #1:* Same number of types 1-3 regions as Sp19 2.2 gives same symbol error prob.

Approach #2: Lecture slides 15-13 to 15-15. $P(e) = 1 - P(c)$. Let $q = Q\left(\frac{d}{\sigma}\right)$.

$$P(c) = \frac{3}{4}P_2(c) + \frac{1}{4}P_3(c) = \frac{3}{4}(1 - q)(1 - 2q) + \frac{1}{4}(1 - q)^2$$

$$P(c) = \frac{3}{4}(1 - 3q + 2q^2) + \frac{1}{4}(1 - 2q + q^2) = 1 - \frac{11}{4}q + \frac{7}{4}q^2 \text{ and } P(e) = 1 - P(c) = \frac{11}{4}q - \frac{7}{4}q^2$$

(i) For an application, four of the 16 symbols have the most significance and we would like to transmit them with the lowest symbol error probability. Using the **left constellation**, which constellation points would you use to represent these four high-significance symbols? Why? 3 points.

Type III regions have the lowest symbol error probability among the three types.

Use the four corner points in the left constellation for the four symbols of highest significance.

Epilogue. Application for (i). Consider an image in a progressive format as a low-resolution plus a medium-resolution plus a high-resolution image. Since the low-resolution image is the most important part because it is needed by the other two resolutions, the low-res data would be carried by the four highest-significance symbols. The medium- and high-resolution images would be carried by type 2 and 3 symbols, respectively.

Problem 2.3. Interference Cancellation. 24 points.

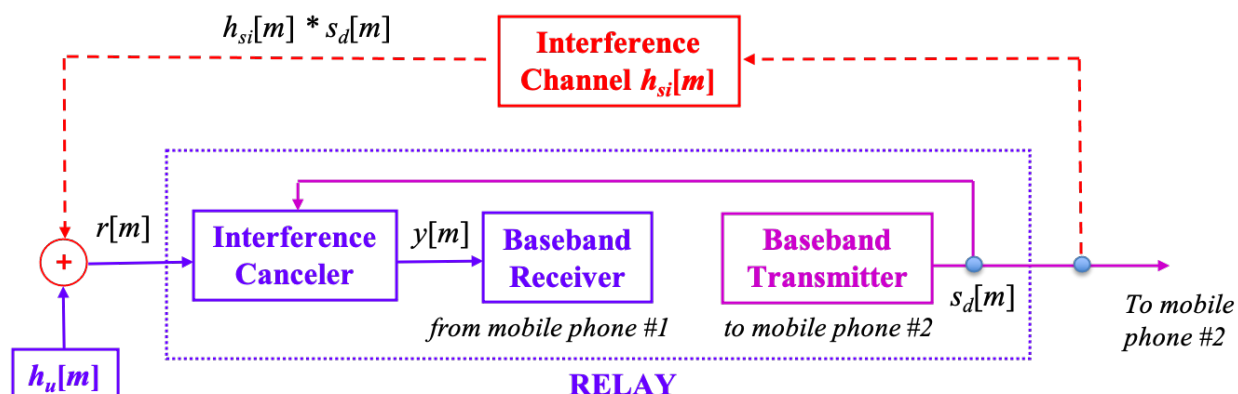
In wireless communication systems, transmitted signals experience a wide variety of impairments by the time they reach a receiver, including attenuation over distance traveled.

A relay can be deployed to receive and decode a basestation transmission and then re-encode and retransmit the signal so that it arrives at a mobile phone with higher signal power.

A relay can also receive and decode a mobile phone transmission and then re-encode and retransmit the signal so that it arrives at a basestation with higher signal power.

When a relay receives and transmits signals at the same time and in the same frequency band, the propagation of the relay transmission to the relay receiver causes **interference**.

The diagram below gives a baseband model for a relay system serving two mobile phones. Mobile phone #1 is transmitting at the same time that the relay is receiving and transmitting.



The interference canceler does not know $h_{si}[m]$, $h_u[m]$, or $s_u[m]$, and uses knowledge of $s_d[m]$ and $r[m]$ to adapt a finite impulse response (FIR) filter $c[m]$ to subtract interference from $r[m]$:

$$r[m] = \underbrace{h_u[m] * s_u[m]}_{\text{from mobile phone \#1}} + \underbrace{h_{si}[m] * s_d[m]}_{\text{interference}}$$

$$y[m] = r[m] - c[m] * s_d[m]$$

$$y[m] = h_u[m] * s_u[m] + h_{si}[m] * s_d[m] - c[m] * s_d[m] = h_u[m] * s_u[m] + (h_{si}[m] - c[m]) * s_d[m]$$

(a) What objective function would you use? Why? 8 points. **Adapt canceler to remove interference in $r[m]$ and hence reduce its power: $J(y[m]) = \frac{1}{2} y^2[m]$. Best removal when $c[m] = h_{si}[m]$.**

(b) For an adaptive FIR canceler, **derive the update equation** for the vector of FIR coefficients \vec{c} for the objective function in part (a). Here, $\vec{c} = [c_0 \ c_1 \ c_2 \ \dots \ c_{N-1}]$. 12 points.

Transmitted baseband relay signal $s_d[m]$ also serves as a reference signal to adapt the interference canceler. A training sequence is not needed. Using steepest descent,

$$y[m] = r[m] - c_0 s_d[m] - c_1 s_d[m - 1] - \dots - c_{N-1} s_d[m - (N - 1)]$$

$$\text{Let } \vec{s}_d[m] = [s_d[m] \ s_d[m - 1] \ \dots \ s_d[m - (N - 1)]]$$

For steepest descent algorithm, let $\vec{c}[m] = [c_0[m] \ c_1[m] \ \dots \ c_{N-1}[m]]$:

$$\vec{c}[m + 1] = \vec{c}[m] - \mu \left. \frac{d}{d\vec{c}} J(y[m]) \right|_{\vec{c}=\vec{c}[m]} = \vec{c}[m] + \mu y[m] \vec{s}_d[m]$$

(c) For your answer in (b), what values of the step size (learning rate) μ would you use? 4 points. **Use a small positive value for the step size μ , such as 0.01 or 0.001, for convergence of the steepest descent algorithm. A step size of zero will prevent any updates. A negative step size and a large positive step size will cause divergence.**

Epilogue
 Transmitting and receiving at the same time in the same frequency band is known as **full duplex**.
 Potential spectral efficiency gain **2x**.
 Subscripts are **d** downlink, **si** self-interference and **u** uplink.

Problem 2.4. *Communication System Design Tradeoffs. 25 points*

Describe the effect of increasing each pulse amplitude modulation system design parameter in the first column on the quantity in the other columns: **increase**, **decrease** or leave it **blank** to mean no effect.

When considering the impact of increasing the parameter in the first column, assume that the values of the other parameters are held constant.

After the table, justify your answers for each row. The row for J bits/symbol is given as an example. For the entry, “ $2d$ constellation spacing in Volts”, consider what happens when d increases.

Parameter	Transmit Power Consumption	Transmission Bandwidth	Bit Rate	Symbol Error Rate	Run-Time Complexity
B bits in A/D output & D/A input			?	?	increase
$2d$ constellation spacing in Volts	increase			decrease	
f_{sym} symbol rate in Hz		increase	increase	increase	increase
J bits/symbol	increase		increase	increase	increase
L samples/symbol				decrease	increase
N_g symbol periods in pulse shape				decrease	increase

J bits/symbol. Increasing J increases number of constellation points, $M = 2^J$. Peak transmit power is proportional to square of max amplitude $(M-1)d$ and hence increases exponentially in J . Bit rate Jf_{sym} increases linearly with J . For symbol error probability $\frac{2(M-1)}{M} Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$, the constant in front is on the interval $[1, 2)$ and increases slightly with J because $M = 2^J$. In the transmitter, the constellation map has $M = 2^J$ entries and fast symbol decoding in the receiver takes J comparisons.

B bits in A/D output & D/A input. Power consumption by a data converter is exponential in B .

- **Transmit power consumption** (i.e. transmitted signal power). As B increases, the power consumed in the transmitter increases but does not affect the transmit power consumption.
- **Bit rate.** Increasing B has no effect unless $B < J$. In that case, the number of levels of resolution in the data converters 2^B would not be enough to represent the 2^J different symbol amplitudes uniquely and increasing B until $B = J$ would increase the symbol rate.
- **Symbol error rate** would not be affected because the parameters in the symbol error probability expression (J , d , σ , and T_{sym}) remain the same per the problem statement except
 - when $B < J$, the number of levels of resolution in the data converters 2^B would not be enough to represent the 2^J different symbol amplitudes uniquely and in that case, increasing B until $B = J$ would decrease the symbol error rate.

- when the SNR w/r to thermal noise in the receiver analog/RF front end is less than the A/D SNR w/ quantization noise power, and in the case, adding more bits to the A/D converter would decrease the symbol error rate until the two SNRs are equal.
- **Run-time complexity** in memory and computation in a software receiver will increase as B increases from 8 to 9 bit due to switching from bytes to 16-bit integers to represent the A/D output and from 16 to 17 bits due to switching from 16-bit integers to 32-bit integers or floating-point numbers. Multiplying two 32-bit IEEE floating-point numbers takes much longer than multiplying two 16-bit integers (4x as long on the TI TMS320C6748 DSP processor).

2d constellation spacing in Volts.

- **Transmit power.** Peak and average transmit power increases with d^2 .
- **Symbol error rate** $\frac{2(M-1)}{M} Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$ decreases with d because $Q(x)$ is monotonically decreasing vs. x .

f_{sym} symbol rate in Hz. (Sp20 Midterm #2 Problem 2.1)

- **Transmission bandwidth** $f_{sym}(1 + \alpha)$ increases linearly with f_{sym} where α is the rolloff factor. **Bit rate** $J f_{sym}$ increases linearly with f_{sym} .
- **Symbol error rate** $\frac{2(M-1)}{M} Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$ increases with f_{sym} because $Q(x)$ is decreasing vs. x and $T_{sym} = 1/f_{sym}$.
- **Run-time complexity** in transmitter and receiver increases linearly with sampling rate $L f_{sym}$.

L samples/symbol.

- **Symbol error rate/probability.** In the receiver, the matched filter correlates the received signal against a known pulse shape of $L N_g$ samples, and the increase in the SNR is proportional to $L N_g$, which in turn decreases the symbol error probability.
- **Run-time complexity** in the transmitter and receiver increases linearly with the sampling rate $L f_{sym}$. The pulse shape has $L N$ samples. It takes $L N_g f_{sym}$ multiplications/s for the transmitter pulse shaping filter and receiver matched filter when using an efficient polyphase filter bank implementation.

N_g symbol periods in pulse shape.

- **Symbol error rate/probability.** In the receiver, the matched filter correlates the received signal against a known pulse shape of $L N_g$ samples to improve SNR. Most of the improvement in SNR is due to the lowpass filtering out-of-band additive noise, and the stopband attenuation can increase with the FIR filter length. When N_g increases, SNR improves and in turn the symbol error probability decreases.
- **Run-time complexity** increases with N_g because the transmitter pulse shaping filter and receiver matched filter take $L N_g f_{sym}$ multiplications/s when using an efficient polyphase filter bank implementation

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 **Take-Home Exam Solutions 3.0**

Prof. Brian L. Evans

Date: May 5, 2021

Course: EE 445S

Name: _____ **Space,** _____ **Lost In**
Last, First

Please sign your name below to certify that you did not receive any help, directly or indirectly, on this test from another human other your instructor, Prof. Brian L. Evans, and to certify that you did not provide help, directly or indirectly, to another student taking this exam.

(please sign here) _____

- **Take-home exam** is scheduled for Wednesday, May 5, 2021, 10:30am to 11:59pm.
 - The exam will be available on the course Canvas page at 10:30am on May 5, 2021.
 - Your solutions can be on notebook paper, or on the test and your own paper, or whatever. This means that you won't have to print the test to complete the test.
 - **Please include this cover page** signed by you with your solution and upload your solution as a single PDF file to the course Canvas page by 11:59pm on May 5, 2021.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. Sources can include course lecture slides, handouts, homework solutions, books, Web pages, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Internet access.** Yes, you may fully access the Internet when answering exam questions provided that you comply with the other instructions on this page.
- **Academic integrity.** You shall not receive help directly or indirectly on this test from another human except your instructor, Prof. Evans. You shall not provide help, directly or indirectly, to another student taking this exam.
- **Send questions to Prof. Evans.** You may send questions or concerns about this midterm exam during the test to Prof. Evans via Canvas or by e-mail at bevans@ece.utexas.edu.
- **Contact by Prof. Evans.** Prof. Evans might contact all students in the class during the exam through Canvas announcements. Please periodically monitor those announcements.

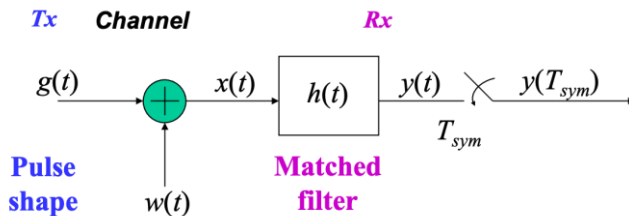
	Problem	Point Value	Your score	Topic
Will Robinson	1	24		Baseband PAM System
Penny Robinson	2	27		QAM Communication Performance
Judy Robinson	3	28		QAM Receiver Architecture Tradeoffs
The Robot	4	21		Potpourri
	Total	100		

Problem 2.1. Baseband PAM System. 24 points.

Consider a baseband pulse amplitude modulation (PAM) system with the parameters on the right.

The PAM system does not have A/D or D/A converters.

The problem focuses on the following part of the baseband PAM system:



PAM System Parameters

- $2d$ constellation spacing
- f_{sym} symbol rate
- $g(t)$ pulse shape
- $h(t)$ matched filter impulse response
- J bits/symbol
- k constant
- M levels, i.e. $M = 2^J$
- n symbol index
- N_g symbol periods in $g(t)$
- T_{sym} symbol time

where $w(t)$ is a Gaussian random signal with zero mean and variance σ^2 .

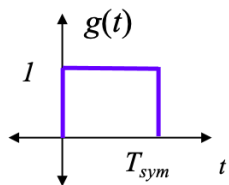
Assume the receiver is synchronized with the transmitter in parts (a), (b) and (c).

- (a) Using the PAM System Parameters, give a formula for $h(t)$ that maximizes the SNR at the estimated symbol amplitude $y(T_{sym})$? 4 points.

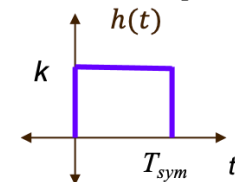
Lecture Slide 14-14

The optimal matched filter, per lecture slide 14-15, maximizes the SNR at the estimated symbol amplitude $y(T_{sym})$ which in turn minimizes the symbol error probability because it is a monotonically decreasing function vs. SNR. The optimal matched filter impulse response is $h(t) = k g^*(T_{sym} - t)$ where k is a real-valued constant ($k \neq 0$ in practice).

- (b) Using your answer in part (a), plot $h(t)$ when $g(t)$ is the rectangular pulse shown below. 4 points.



Since $g(t)$ is real-valued, complex conjugation has no effect. $g(-t)$ flips it with respect to the vertical axis, and $g(T_{sym} - t)$ delays the flipped version by T_{sym} and we're back to where we started with $g(t)$. Gain k is real-valued ($k \neq 0$).



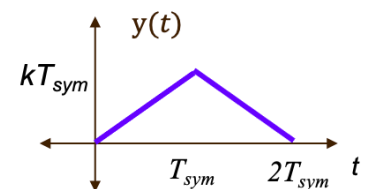
Lecture Slide 14-17

- (c) Using your answer in part (b), plot $y(t)$ assuming there is no noise, i.e. $w(t) = 0$. 4 points.

$$y(t) = g(t) * h(t) + w(t) * h(t)$$

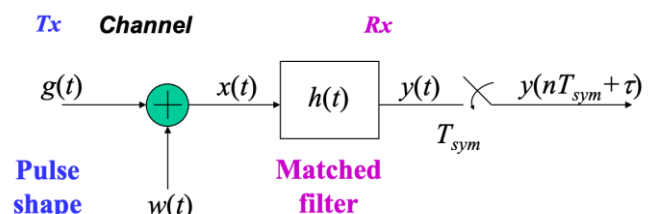
$$\text{With } w(t) = 0, y(t) = g(t) * h(t)$$

Convolution gives a triangular pulse whose peak value occurs at the symbol time, T_{sym} . Note that k can be negative.



- (d) Assume the receiver has an accurate T_{sym} but needs to find a symbol timing offset τ to synchronize with the transmitter as shown below. Develop an adaptive method to update τ in the n th symbol period using analog continuous-time signal processing; e.g., a differentiator circuit will compute a derivative of an analog continuous-time signal. Use $g(t)$ and $h(t)$ from part (b)

- i. Give an objective function. 6 points.
- ii. Give the update for $\tau[n+1]$ given $\tau[n]$. 3 points.
- iii. How would you determine the value of the step size μ ? 3 points.



See next two pages for two different solutions for (d).

Solution #1 for 2.1(d): (A student's solution is below *with additional information in blue*):

- i. Give an objective function. 6 points.

Define the error $e[n]$ between what we have and what we would like to have:

$$e[n] = y[n] - y(T_{sym}) = y[n] - k T_{sym} \text{ where } y[n] = y(n T_{sym} + \tau).$$

Define the objective function to be used to drive the error to zero:

$$J(e[n]) = \frac{1}{2} e^2[n]$$

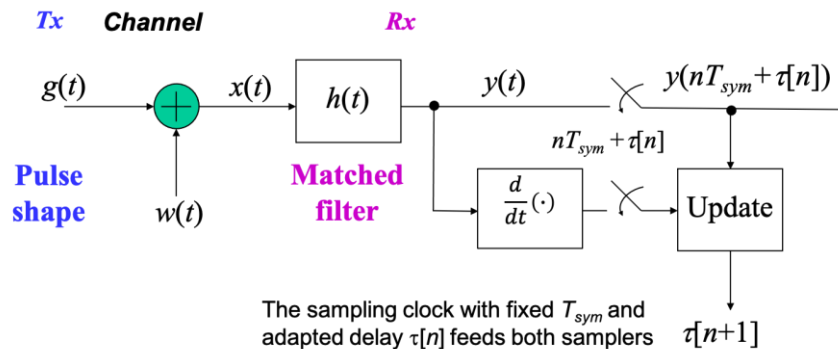
- ii. Give the update for $\tau[n+1]$ given $\tau[n]$. 3 points.

Seek to minimize the objective function to drive the error to zero:

$$\tau[n + 1] = \tau[n] - \mu \left. \frac{d}{d\tau} J(e[n]) \right|_{\tau=\tau[n]}$$

$$\tau[n + 1] = \tau[n] - \mu e[n] \left. \frac{d}{d\tau} y[n] \right|_{\tau=\tau[n]} = \tau[n] - \mu e[n] y'(n T_{sym} + \tau[n])$$

Although not asked, which is why the text is in blue here, here's the system diagram with the differentiator and adaptive element to perform the update:



The approach can be extended to any pulse shape when using $e[n] = y[n] - y(T_{sym})$.

- iii. How would you determine the value of the step size μ ? 3 points.

Choose a small positive value by using trial-and-error in simulation:

- A negative value for μ would cause the objective function to be minimize which would lead to minimizing the signal power.
- A value of zero for μ means that the update will not change the initial value of τ .
- A large positive μ value will lead to divergence of the adaptive algorithm.

Solution #2 for 2.1(d):

- i. Give an objective function. *6 points.*

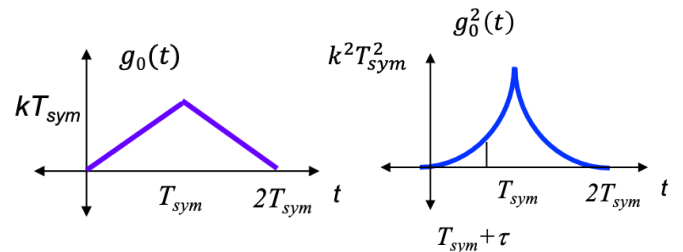
Seek to maximize the power at the estimated symbol amplitude $y(nT_{sym} + \tau)$:

$J(y(t)) = \frac{1}{2}y^2(t)$ where $y(t) = g(t) * h(t) + w(t) * h(t)$. In $y(t)$, the noise term $n(t) = w(t) * h(t)$ is a Gaussian random signal with zero mean and variance $\frac{\sigma^2}{T_{sym}}$ because $h(t)$ is a lowpass filter with bandwidth $\frac{1}{2T_{sym}}$ (Sec. 4.12 Gaussian Processes, Haykin's *Communication Systems*) and $w(t)$ is a Gaussian random signal with zero mean and variance σ^2 . Any sample of $n(t)$ is a Gaussian random variable $N\left(0, \frac{\sigma^2}{T_{sym}}\right)$.

Average noise power is $\frac{\sigma^2}{T_{sym}}$ regardless of sampling time.

In $y(t)$, the deterministic signal term $g_0(t) = g(t) * h(t)$ has its instantaneous power change with the sampling time.

Recall that k can be negative.



- ii. Give the update for $\tau[n+1]$ given $\tau[n]$. *3 points.*

$$\tau[n+1] = \tau[n] + \mu \left. \frac{d}{d\tau} J(y(nT_{sym} + \tau)) \right|_{\tau=\tau[n]}$$

$$\tau[n+1] = \tau[n] + \mu y(nT_{sym} + \tau[n]) \left. \frac{d}{dt} y(t) \right|_{t=nT_{sym} + \tau[n]}$$

This update is also represented by the block diagram in solution #1.

A differentiator circuit can be as simple as an RC and or RL circuit where output is tapped across resistor in the RC circuit or inductor in the RL circuit. The above is for a general pulse shape and its matched filter. For $h(t)$ in part (b), we can simplify the derivative of $y(t)$ by using a linear time-invariant (LTI) model for differentiation with impulse response $v(t)$ per JSK Appendix G.2 Derivatives and Filters:

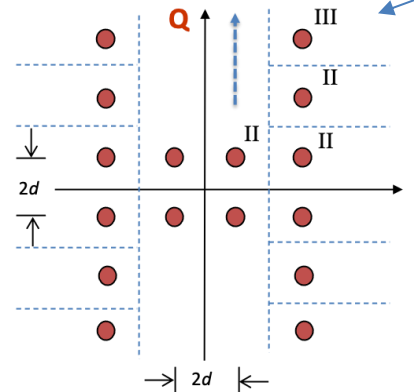
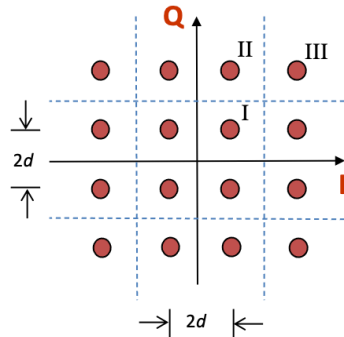
$$\frac{d}{dt}y(t) = v(t) * (h(t) * x(t)) = (v(t) * h(t)) * x(t) = (\delta(t) - \delta(t - T_{sym})) * x(t) = x(t) - x(t - T_{sym})$$

- iii. How would you determine the value of the step size μ ? *3 points.*

Same as the answer in solution #1 for 2.1(d)iii on the previous page.

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Peak power of $34d^2$ at symbol ampl. $3d + j5d$

Constellation Region Type I, II or III

Average power is average of $2d^2, 10d^2, 18d^2$ and $34d^2$

Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	$34d^2$
(b) Average transmit power	$10d^2$	$16d^2$
(c) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation <i>that will minimize the probability of symbol error using such decision regions.</i>		
(d) Number of type I regions	4	0
(e) Number of type II regions	8	12
(f) Number of type III regions	4	4
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	$\frac{11}{4}Q\left(\frac{d}{\sigma}\right) - \frac{7}{4}Q^2\left(\frac{d}{\sigma}\right)$
(h) Express d/σ as a function of the Signal-to-Noise Ratio (SNR) in linear units	$SNR = \frac{10d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{SNR}{10}}$	$SNR = \frac{16d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{SNR}{16}}$

(g) **Approach #1:** Same number of type 1-3 regions as F2020 2.2 gives same symbol error prob.

Approach #2: Lecture slides 15-12 to 15-14. $P(e) = 1 - P(c)$. Let $q = Q\left(\frac{d}{\sigma}\right)$.

$$P(c) = \frac{3}{4}P_2(c) + \frac{1}{4}P_3(c) = \frac{3}{4}(1 - q)(1 - 2q) + \frac{1}{4}(1 - q)^2$$

$$P(c) = \frac{3}{4}(1 - 3q + 2q^2) + \frac{1}{4}(1 - 2q + q^2) = 1 - \frac{11}{4}q + \frac{7}{4}q^2 \text{ and } P(e) = 1 - P(c) = \frac{11}{4}q - \frac{7}{4}q^2$$

(i) For the right constellation, will using the type I, II, and III rectangular decision regions lead to Gray coding for symbols? Either give a Gray coding for the right constellation, or show that it is not possible. *3 points.* **Gray coding means that any two symbols in adjacent decision regions can only differ by one bit. The constellation point at $d + jd$ has five adjacent decision regions, and hence, we cannot encode each pair to differ by one bit because a 16-QAM symbol only has 4 bits.**

Epilogue: The right constellation wouldn't be used in practice. The right constellation has higher peak power, average power, and peak-to-average power ratio than the left one. Unlike the left constellation, the right constellation cannot be Gray coded and its rectangular decision regions do not give the same result as Euclidean distance. Both constellations have a fast binary search decoding algorithm.

Problem 2.3. QAM Receiver Architecture Tradeoffs. 28 points.

In this problem, you evaluate tradeoffs in the two QAM receiver architectures on the right:

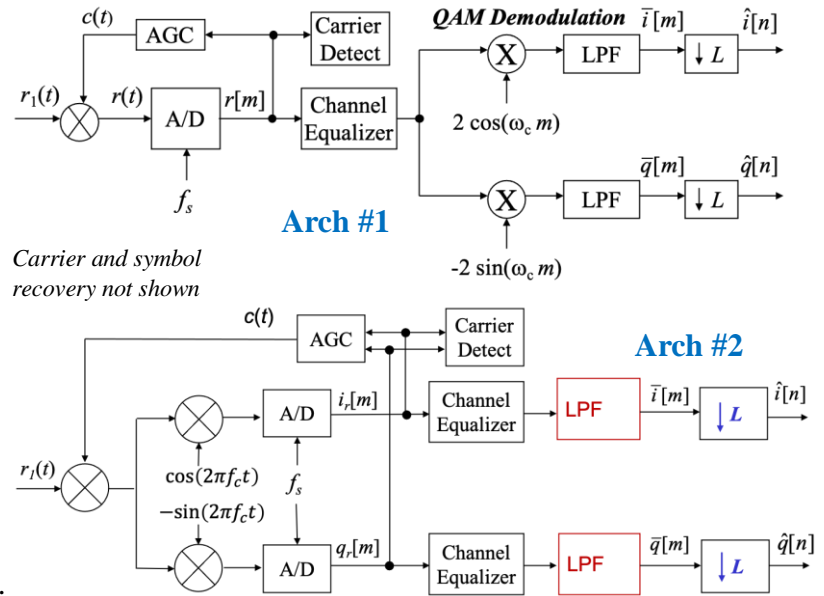
- (1) Single analog-to-digital (A/D) converter
- (2) Two A/D converters, one for the in-phase channel and one for the quadrature channel

In both architectures,

- $r_1(t)$ is the baseband QAM signal
- i is the in-phase component
- q is the quadrature component
- J bits per symbol (assume J is even)
- M constellation points where $M = 2^J$

Please evaluate the following tradeoffs.

- (a) Which architecture consumes less power in its A/D converters? How much less? 9 points.



In an A/D converter, power consumption is proportional to the sampling rate f_s and 2^B where B is the number of bits of amplitude resolution at the A/D output.

Assume the sampling rate is the same for all A/D converters. Difference is in number of bits.

Arch #1: The single A/D converter has to support all possible symbol amplitudes, so $B \geq J$. Power consumption is proportional to 2^J .

Arch #2: Each A/D converter supports a PAM constellation of $J/2$ bits, i.e. $B \geq J/2$. Power consumption is proportional to $2^{2^{J/2}} = 2^{(2^{0.5J})} = 2^{(2^{0.5})^J} = 2^{\sqrt{2}^J}$.

For minimum number of bits for all converters, Arch #1 consumes $\frac{2^J}{2^{\sqrt{2}^J}} = \frac{\sqrt{2}^J}{2}$ more power. This ratio is {1, 2, 4, 8} for $J = \{2, 4, 6, 8\}$.

Lecture Slides 15-6 to 15-8

If we use the minimum sampling rates for all A/D converters, Arch #1 will consume another factor of $\frac{f_c+W}{W}$ more power than Arch #2. Note that $f_c > W$ for sinusoidal modulation.

Arch #1 uses a sampling rate of $f_s > 2(f_c + W)$. Here, the baseband PAM bandwidth is $W = \frac{1}{2}f_{sym}(1+\alpha)$ where $(1+\alpha)$ is the bandwidth expansion factor. For a raised cosine pulse shape, α is the rolloff factor in $[0, 1]$. For a rectangular pulse, $\alpha = 1$. See [lecture slide 7-10](#).

Arch #2 uses a sampling rate of $f_s > 2W$ because it is sampling a baseband PAM signal. Demodulation filters are the analog continuous-time lowpass filters in the A/D converters.

- (b) Describe an automatic gain control (AGC) algorithm for architecture #2 including equations. The algorithm has access to both A/D converter outputs. Give the computational complexity. 6 points.

We modify AGC algorithm for Arch #1 on [lecture slide 16-5](#) for signed 8-bit A/D converters:

c_{-128}, c_0, c_{127} are counts for the number of times -128, 0, or 127, respectively, occurs in last $N/2$ samples in the first A/D converter and last $N/2$ samples in the second A/D converter.

f_{-128}, f_0, f_{127} represent how frequently outputs -128, 0, 127 occur where $f_i = \frac{c_i}{N}$.

Update gain $c(t)$ every τ seconds using $c(t) = A c(t - \tau)$ where $A = 1 + 2(f_0 - f_{-128} - f_{127})$.

Computational complexity (*same as that of arch. #1 algorithm*)

Substituting $f_i = \frac{c_i}{N}$, $A = \frac{N + 2c_0 - c_{-128} - c_{127}}{N}$. Computing the numerator takes 3 additions and 1 left shift by one bit to implement multiplication by two. We would like a floating-point value for A. Numerator takes integer values between 0 and $2N$, inclusive. We could create a lookup table of $2N+1$ entries to store all precomputed floating-point values for A and use the numerator as index into the lookup table. A is computed every τ seconds.

For each sample, we need 6 comparisons to update the 3 counters according to the values of $i_r[m]$ and $q_r[m]$. We update the 3 counters based on values of $i_r[m-N/2]$ and $q_r[m-N/2]$ that will be discarded from the circular buffers of $i_r[m]$ and $q_r[m]$ values. (For reduced storage, we would store the values of the changes to the counters for each sample in a circular buffers instead of the $i_r[m]$ and $q_r[m]$ values themselves.) Assume N is even.

- (c) Describe a carrier detection algorithm for architecture #2 including equations. The algorithm has access to both A/D converter outputs. Give the computational complexity. 6 points.

We modify the carrier detection algorithm for Arch #1 on lecture slide 16-9 as follows:

Let $x[m] = i_r^2[m] + q_r^2[m]$ be the instantaneous power of the in-phase and quadrature baseband PAM channels combined. (We assume the in-phase and quadrature baseband PAM channels are orthogonal, i.e. 90 degrees or 270 degrees out of phase. In practice, the two channels are close enough to orthogonal for the purposes of a carrier detection algorithm. The loss of orthogonality is called IQ imbalance.)

Compute average power using first-order IIR filter $p[m] = c p[m - 1] + (1 - c) x[m]$ where $0 < c < 1$. The pole location is c . The closer c is to 1, the more selective the filter (i.e. the more narrow the passband and the larger the stopband attenuation in dB).

- If there is no transmission being received, assume there is transmission if $p[m]$ is larger than a large threshold.
- If there is transmission being received, assume that the transmission has stopped if $p[m]$ is smaller than a small threshold.

Lecture
Slide 16-9

Computational complexity (*2x of that of arch #1 algorithm*):

$x[m] = i_r^2[m] + q_r^2[m]$ takes 2 multiplications and 1 addition per sample.

$p[m] = c p[m - 1] + (1 - c) x[m]$ takes 2 multiplications and 1 addition per sample.

Total run-time computational complexity: 4 multiplications and 2 additions per sample plus one threshold operation applied periodically.

- (d) Which architecture would you advocate using? Why? Describe the tradeoffs considered. 7 points.

Arch #1 A/D converter consumes more power than the combined power consumption of the A/D converters in Arch #2 by a factor of $\left(\frac{f_c+W}{W}\right)\left(\frac{\sqrt{2}^J}{2}\right)$.

In comparing baseband discrete-time signal processing, Arch #1 has one channel equalizer as well as pointwise multiplication and generation of cosine and sine signals. Arch #2 has two channel equalizers, but this is offset because Arch #2 runs at less than half the sampling rate.

Arch #1 advantages: fewer components.

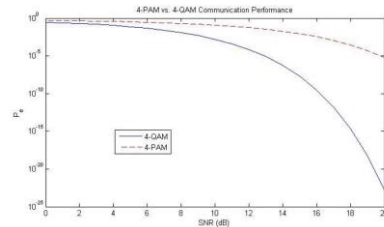
Arch #2 advantages: lower power consumption in the A/D conversion (which dominates power consumption in an analog/RF frontend) and lower baseband discrete-time complexity.

Problem 2.4. Potpourri. 21 points.

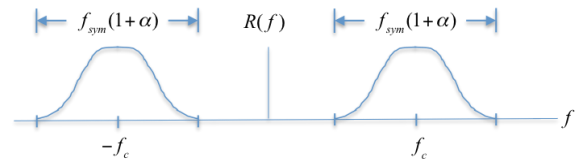
Please determine whether the following claims are true or false and **support each answer with a brief justification**. A true or false answer without any justification will not earn any points.

- (a) PAM and QAM transmission using the same constellation size and symbol rate will always have the same symbol error rate when both receivers are operating at the same received SNR. *3 points.*

False. For same symbol rate, the symbol error rate (symbol error probability) for 4-QAM is much lower than that of 4-PAM for received SNR greater than 0 dB. The plot on the right of symbol error rate vs. SNR in dB is from [Handout P: Communication Performance of PAM vs. QAM Handout](#). As SNR $\rightarrow -\infty$ dB, curves converge to $3/4$; see part (g).



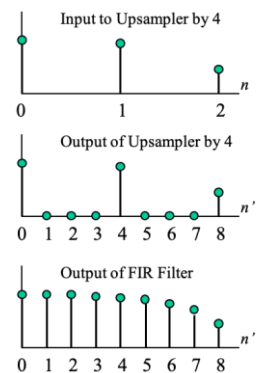
- (b) Pulse shaping filters are designed to contain the spectrum of a transmitted signal in a communication system. In a communication system, the pulse shape should be zero at non-zero integer multiples of the symbol duration and have its maximum value at the origin. *3 points.*



First Claim is True. In a PAM transmitter, the pulse shaping filter determines the baseband bandwidth, which is $1/2$

$f_{sym} (1+\alpha)$ where $(1+\alpha)$ is the bandwidth expansion factor over the ideal lowpass filter. After upconversion in the analog/RF front end, the PAM transmission bandwidth would be $f_{sym} (1+\alpha)$ as shown above (the plot is from [Spring 2020 Midterm 2.1](#)). For a QAM transmitter, the baseband signal in the frequency domain would be centered at frequency f_c with bandwidth $f_{sym} (1+\alpha)$ as shown above, and the transmission bandwidth after upconversion in the analog/RF front end would be $2f_c + f_{sym} (1+\alpha)$.

Second Claim is True. The pulse shape is used as the impulse response of an FIR filter that interpolates the output of upsampling by L , where L is the number of samples in a symbol period. For a non-causal pulse shape centered at the origin, the pulse shape should be zero at non-zero integer multiples of L so that the symbol amplitudes pass through unchanged. That is, the FIR filter implements convolution; as the impulse response (pulse shape) is flipped and slid across the input signal, the zero crossings at a non-zero integer multiples of the symbol duration (L samples) ensure that the symbol amplitudes remain unchanged. See the plot on the right for $L = 4$ from [Lecture Slide 13-8](#).



FIR fills in zero values

- (c) The LTI components of wired and wireless channels have impulse responses of infinite duration, and each can be modeled as an FIR filter. Wired channel impulse responses do not change over time, whereas wireless channel impulse responses change over time. *3 points.*

First claim is true. Wired channels have impulse responses that resemble RLC circuits. From the transmitter to receiver in wireless channels, there can be a direct path, paths involving one reflection (bounce), paths involving two reflections (bounces), etc. We can truncate the impulse responses to model the impulse responses as FIR. See [Lecture 12 on Wireless Impairments, slides 12-5 to 12-8](#).

Second claim is false. Impulse responses in wired channels change with temperature because resistance, capacitance and inductance depend on temperature across the wire.

- (d) A receiver in a digital communication system employs a variety of adaptive subsystems, including automatic gain control, carrier recovery, and symbol timing recovery. A transmitter in a digital communication system does not employ any adaptive systems. *3 points.*

First claim is true. Adaptive methods based on steepest descent for carrier recovery and symbol timing recovery are subjects of [homework problems 6.1 and 6.2](#). The *JSK textbook* discusses adaptive methods automatic gain control ([pp. 120-128](#)), carrier recovery ([pp. 198-220](#)) and symbol timing recovery ([pp. 250-269](#)), with many based on steepest descent.

Second claim is false. Several examples of adaptive systems in the baseband transmitter:

1. Use feedback from the receiver to adapt the pause time (guard interval) after each symbol transmission to reduce inter-symbol interference in the receiver ([see Lecture Slide 14-6](#)).
 2. Compensate for impedance mismatches using adaptive predistortion ([Midterm 2.3 Fall 2018](#)) or echo cancellation ([Midterm 2.3 Fall 2014](#)). An impedance mismatch can occur between the baseband output and analog/RF front end input, and the mismatch can vary with time due to temperature. Impedance mismatches can also occur between the analog/RF front end and the wired channel as well as at each junction in the wired channel.
 3. [Nonlinear pre-distorter](#) to improve the linearity of [radio transmitter amplifiers](#).
- (e) When designing an FIR channel equalizer for a communication system using same amount of training data and the same filter length, an adaptive least mean-squares [Midterm 2.4\(c\)](#) method should always be used over a least-squares method. *3 points.* [Fall 2015](#)
- False.* From [Fall 2017 Midterm 2.4\(c\)](#), when the training sequence length is short relative to the equalizer length, a LS equalizer will perform better because the adaptive LMS equalizer will not have enough training data to converge to a meaningful solution. In [homework 7.2](#) on the adaptive LMS method, the training sequence was 250 times the equalizer length.
- False.* Another reason is that an adaptive LMS method can have an issue with stability. Stability requires a small enough positive value of the step size (learning rate). Adaptive LMS method would have much lower complexity than the LS method in this case.
- (f) In a communications system using a rectangular QAM constellation, the fastest and most accurate way for the receiver to find the constellation point closest to the received symbol amplitude is to use Euclidean distance. *3 points.* [Midterm 2.2\(i\)](#)
[Spring 2016](#)

False. It is true that using Euclidean distance to find the constellation point closest to a received symbol amplitude is the most accurate but not the fastest way. Euclidean distance requires 2 multiplications per constellation point and a square root operation for each of the $M = 2^J$ points constellation points for a J -bit symbol. (To reduce complexity, we use Euclidean distance squared to remove the square root.) When using rectangular decision regions for a rectangular QAM constellation, such as in the left constellation in problem 2.2, we can use binary search. Binary search eliminates half of the remaining constellation points each step, which requires J comparisons vs. $2 \cdot 2^J$ multiplications. For rectangular QAM constellations, rectangular decision regions match those from using Euclidean distance.

- (g) When the received SNR is $-\infty$ dB, the symbol error rate is 100%. That is, there is no chance that any symbol will be decoded correctly. *3 points*

False. As received SNR goes to $-\infty$ dB, noise swamps the signal. Receiver can only randomly guess the symbol among a constellation of M symbols with an error probability of $\frac{M-1}{M}$.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 *Solutions 2.0*

Date: Dec. 6, 2021

Course: EE 445S Evans

Name: _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

Problem	Point Value	Your score	Topic
1	21		Baseband PAM System
2	33		PAM vs. QAM Communication Performance
3	28		Decision-Directed Equalization
4	18		Potpourri
Total	100		

Problem 2.1. Baseband PAM System. 21 points.

Consider a binary phase shift keying (BPSK) system, a.k.a. a two-level pulse amplitude modulation (2-PAM) system.

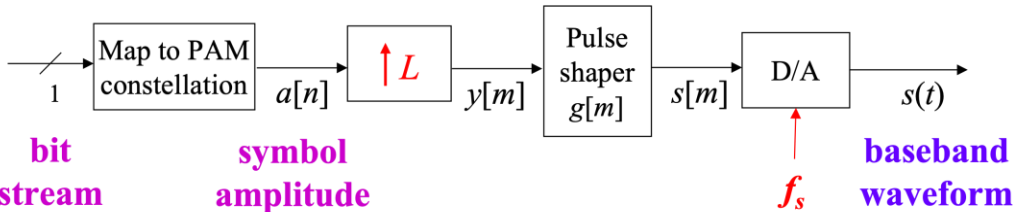
The system parameters are described on the right:

- $J = 1$ bit/symbol
- $L = 4$ samples per symbol period
- Pulse shape $g[m]$ is a rectangular pulse of $L = 4$ samples in duration with amplitude $1/L$.
- A bit of value 0 is mapped to symbol amplitude $-d$, and a bit of value 1 is mapped to symbol amplitude d .

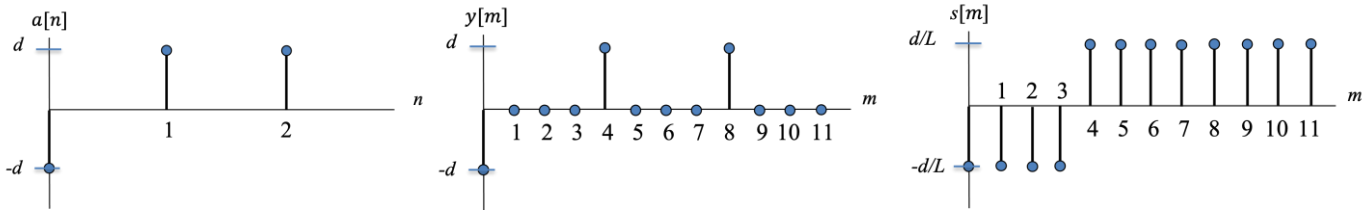
PAM System Parameters

$a[n]$	symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse resp.
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index

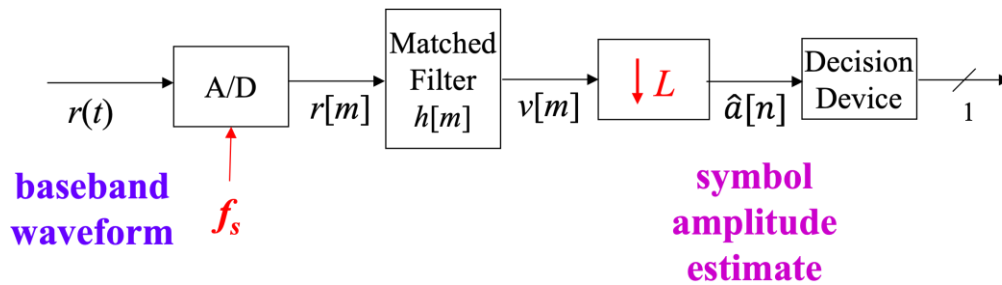
- (a) For the **BPSK transmitter** below, the input bit stream is 011. Plot the discrete-time signals $a[n]$, $y[m]$ and $s[m]$. 9 points.



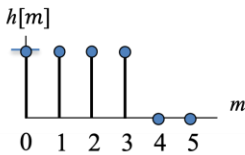
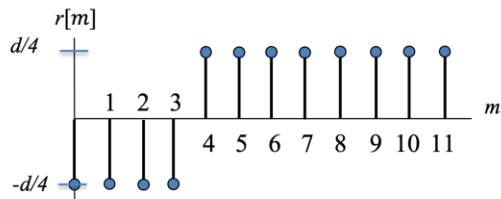
Lecture
Slide 13-8



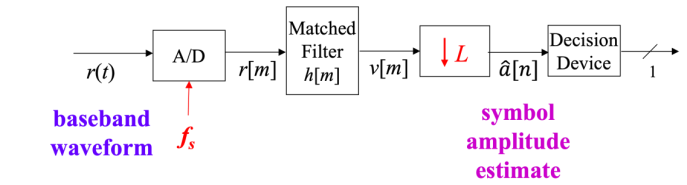
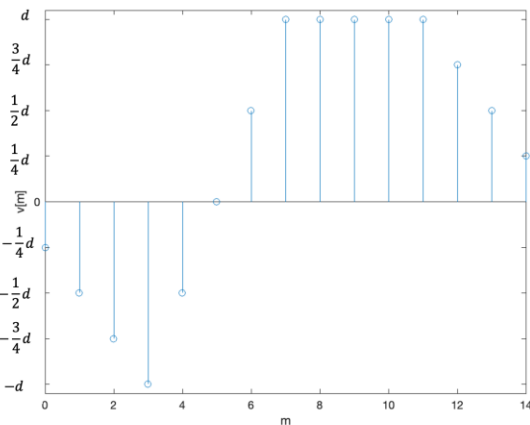
- (b) For the **BPSK receiver** below, assume there is no channel distortion or additive noise and assume that $r[m] = s[m]$. Plot the discrete-time signals $r[m]$, $h[m]$, $v[m]$ and $\hat{a}[n]$ based on the BPSK transmitter in (a). 12 points.



See the next page.



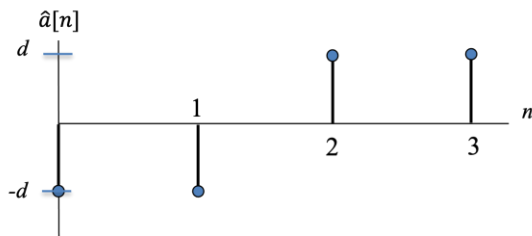
Matched filter
 $h[m] = k g^*[L - m]$



```

r = [ -1 -1 -1 -1 1 1 1 1 1 1 1 1 ] / 4;
h = [ 1 1 1 1 ];
v = conv(h, r);
M = length(v);
m = 0 : M-1;
stem(m, v);
ylim( [-1.05 1.05] );
xlabel('m');
ylabel('v[m]');

```



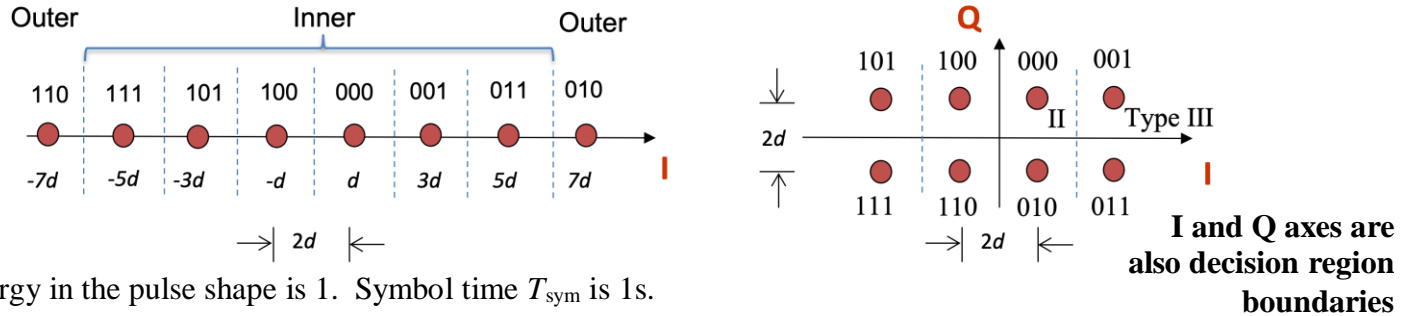
Epilogue: Four bits are output by the receiver (0011) even though only three bits were input to the transmitter (011).

The reason is that we haven't accounted for the group delay through the pulse shaping filter of $\frac{L-1}{2}$ samples or the group delay through the matched filter also of $\frac{L-1}{2}$ samples.

Prior to the downsampler in the receiver, we should discard the first $L - 1$ samples.

Problem 2.2 PAM vs. QAM Communication Performance. 33 points.

Consider the 8-PAM (left) and 8-QAM (right) constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s.

Each part below is worth 3 points. **Please fully justify your answers.**

	8-PAM Constellation	8-QAM Constellation
(a) Peak transmit power	$49d^2$	$10d^2$
(b) Average transmit power	$21d^2$	$6d^2$
(c) Peak-to-average power ratio	$\frac{49d^2}{21d^2} = \frac{7}{3} \approx 2.33$	$\frac{10d^2}{6d^2} = \frac{5}{3} \approx 1.67$
(d) Draw the type I, II and/or III decision regions for the right constellation <i>that will minimize the probability of symbol error using such decision regions.</i>		
8-PAM inner constellation points	6	
8-PAM outer constellation points	2	
(e) Number of type I QAM regions		0
(f) Number of type II QAM regions		4
(g) Number of type III QAM regions		4
(h) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2 . For QAM, the variance is σ^2 in the in-phase component and σ^2 in the quadrature component. For PAM, the variance is $2\sigma^2$ to keep the total noise power the same as in QAM.	$\frac{7}{4} Q\left(\frac{d}{\sqrt{2}\sigma}\right)$	$P_c = \frac{1}{2}P_c^{II} + \frac{1}{2}P_c^{III}$ $P_c^{II} = (1-q)(1-2q)$ $P_c^{III} = (1-q)^2$ $P_e = 1 - P_c$ $P_e = \frac{5}{2}Q\left(\frac{d}{\sigma}\right) - \frac{3}{2}Q^2\left(\frac{d}{\sigma}\right)$
(i) Express the argument of the Q function as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{21d^2}{2\sigma^2}$ $\frac{d}{\sqrt{2}\sigma} = \sqrt{\frac{\text{SNR}}{21}}$	$\text{SNR} = \frac{6d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{6}}$

(j) Give a Gray coding for the 8-QAM constellation points on the constellation above. 3 points.

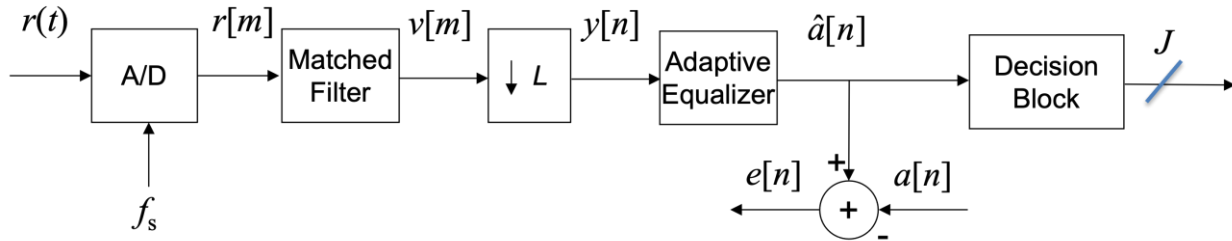
Gray coding means the bit pattern only differs by one bit in adjacent decision regions to minimize the number of bit errors when a symbol error occurs. One approach is to use the first two bits to encode the quadrant, and third bit to specify which point in the quadrant.

(k) Would you recommend using 8-PAM or 8-QAM? Give two reasons. 3 points.

Both 8-PAM and 8-QAM would have the same bit rate. Both can be Gray coded. Choose 8-QAM because it has a lower probability of symbol error for the same SNR, and lower maximum power, average power, and peak-to-average-power ratio for the same value of d . (8-QAM baseband Tx has at least 2x implementation complexity as 8-PAM; same for Rx.)

Problem 2.3. Decision-Directed Equalization. 28 points.

Consider the following baseband pulse amplitude modulation (PAM) receiver with an adaptive finite impulse response (FIR) equalizer placed immediately before the decision device:



The adaptive FIR equalizer runs at the symbol rate and has N coefficients. We place the coefficients in a vector

$$\vec{w} = [w_0 \ w_1 \ \dots \ w_{N-1}]$$

Consider adapting the decision-directed equalizer during training.

The error signal is $e[n] = \hat{a}[n] - a[n]$ which is the difference between the estimated and transmitted symbol amplitudes.

This is similar to the adaptive least-mean squares (LMS) FIR filter in HW 7.2 except it runs at the symbol rate instead of the sampling rate and uses a different error measure.

For the adaptive FIR equalizer,

- (a) Give an objective function $J(n)$. 6 points.

We want drive the error to zero by driving the square of the error to zero: $J(n) = \frac{1}{2} e^2[n]$

- (b) What is the initial value of the adaptive FIR equalizer coefficients you would use? Why? 3 points.

$\vec{w}[0] = [1 \ 0 \ 0 \ \dots \ 0]$ is equivalent to the original receiver block diagram that does not have an adaptive filter. So, if it's not needed, it shouldn't adapt. Using all zeros means equalizer output $\hat{a}[n]$ will initially be zero, but equalizer will still adapt. Any initialization works.

- (c) Derive an update equation for the adaptive FIR equalizer coefficients vector at iteration n

$$\vec{w}[n] = [w_0[n] \ w_1[n] \ \dots \ w_{N-1}[n]]$$

Compute all derivatives. Simplify the result. 9 points

$$\vec{w}[n+1] = \vec{w}[n] - \mu \left. \frac{dJ(n)}{d\vec{w}} \right|_{\vec{w}=\vec{w}[n]} \quad \text{where} \quad \frac{dJ(n)}{d\vec{w}} = e[n] \frac{de[n]}{d\vec{w}} = e[n] \frac{d\hat{a}[n]}{d\vec{w}} = e[n] \vec{y}[n]$$

where $\hat{a}[n] = w_0 y[n] + w_1 y[n-1] + w_2 y[n-2] + \dots + w_{N-1} y[n-(N-1)]$ and

$$\frac{d\hat{a}[n]}{dw_m} = y[n-m] \quad \text{so} \quad \frac{d\hat{a}[n]}{d\vec{w}} = \vec{y}[n] \quad \text{where} \quad \vec{y}[n] = [y[n] \ y[n-1] \ \dots \ y[n-(N-1)]]$$

$$\vec{w}[n+1] = \vec{w}[n] - \mu e[n] \vec{y}[n]$$

PAM System Parameters

$a[n]$	symbol amplitude
$\hat{a}[n]$	estimated symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse response
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index
T_{sym}	symbol time

(d) What range of values would you recommend for the step size μ ? Why? 3 points.

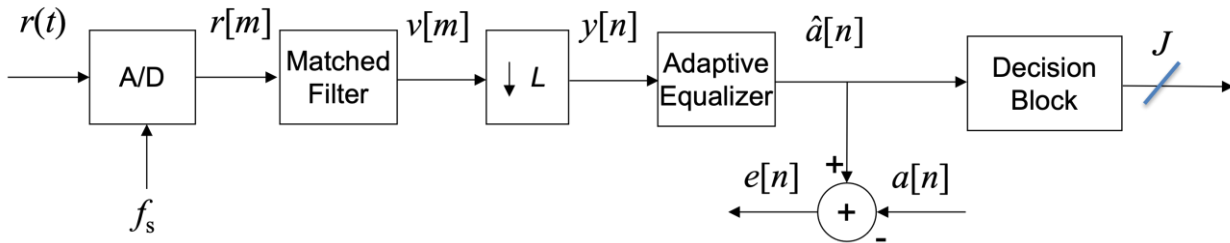
Use small positive values for the step size such as 0.01 or 0.001 for convergence as seen on HW 7.2 on adaptive equalizers. Using a large positive value can lead to instability. Using value of zero would not update the value of the coefficients. Using a negative value would maximize the objective function instead of minimizing it.

(e) Is it possible for the adaptive equalizer to compensate symbol timing error? Why or why not? 7 points.

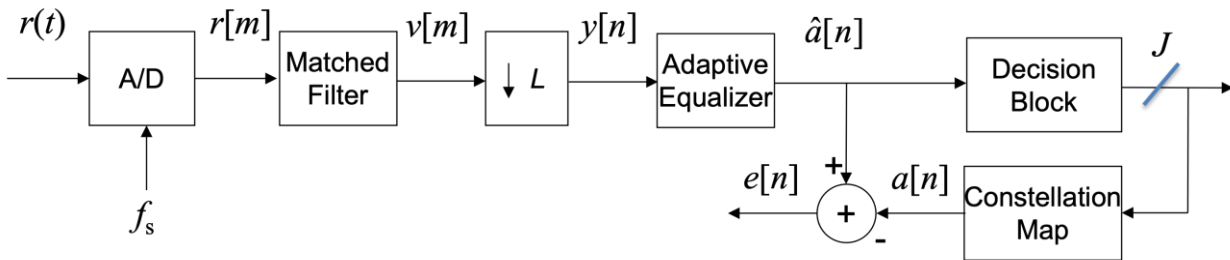
During training, the error (vector) $e[n]$ is a measurement of all the remaining impairments, including symbol timing error.

By driving the error (vector) $e[n]$ to zero, the adaptive equalizer can compensate linear magnitude and phase distortion that varies over the training period. Compensating for phase distortion would include compensating for symbol timing error because compensating symbol timing error can be expressed as a phase shift. (See Fall 2018 Midterm Problem 2.4.)

Epilogue: We can also use decision-directed methods when we're not training. That is, when an unknown message is being received, we can adjust the equalizer from



to the following:



The adjusted version can be “fooled” when there is a symbol error. This version of the decision-directed FIR equalizer described in JSK Section 13.4.

Problem 2.4. Potpourri. 18 points.

Please determine whether the following claims are true or false and **support each answer with a brief justification**. A true or false answer without any justification will not earn any points.

- (a) An increase in thermal noise power always causes a decrease in the signal-to-noise ratio (SNR) in the receiver after the analog-to-digital (A/D) converter. 3 points.

False. An A/D converter is shown on the right. The analog lowpass filter attenuates noise at frequencies above $\frac{1}{2}f_s$ which will alias after sampling. At the quantizer output, noise power will be the greater of the thermal noise power at the quantizer input and the quantization noise power introduced by the quantizer. If the quantization noise power were greater than the thermal noise power, noise power at the quantizer output will be the quantization noise power, until the thermal noise power increases to exceed the quantization noise power. So, an increase in the thermal noise power at the A/D input does not always cause a decrease in SNR at the A/D output. (Matched filter after the A/D converter will increase SNR by filtering out-of-baseband noise.)



$$\text{SNR} = \frac{\text{Signal Power}}{\text{Noise Power}}$$

- (b) For an M -level pulse amplitude modulation (PAM) transmitter using a raised cosine pulse shape and a B -bit digital-to-analog (D/A) converter, setting $B = \log_2 M$ will avoid any clipping in amplitude of the input of the D/A converter. 3 points.

False. The number of bits B needed to avoid clipping in the transmitter depends on J which is the number of bits in the PAM constellation, d which is half of the distance between constellation points, and $g[m]$ which is the pulse shape. PAM symbol amplitudes are in $[-(M-1)d, (M-1)d]$. For a raised cosine pulse shape with max amplitude of 1, pulse shaping filter output values are in $[-2(M-1)d, 2(M-1)d]$ which requires $\log_2(4(M-1)d)$ bits. See lecture slides 13-6 and 15-8, and Spring 2019 Midterm 2.3(b).

- (c) PAM and QAM transmission using the same constellation size and symbol rate will always have the same bit rate. 3 points. **True.** For PAM and QAM, the bit rate in bits/s is $J f_{sym}$ where J is the number of bits/symbol and f_{sym} is the symbol rate in symbols/s. Moreover, $J = \log_2 M$ where M is the constellation size. Hence, the bit rate is $f_{sym} \log_2 M$. See lecture slide 13-3.

- (d) The number of constellation points for a PAM system must always be a power of 2. 3 points.

False. The High-Speed Digital Subscriber Line 2 standard supports PAM constellation sizes that are not powers of two. Also, Fall 2020 Midterm Problem 2.1 concerned 3-PAM systems.

- (e) In a QAM system, the only way to reduce the symbol error rate is to reduce the symbol rate. 3 points. **False.** Symbol error rate for 16-QAM is $3Q \left(\frac{d}{\sigma} \sqrt{T_{sym}} \right) - \frac{9}{4} Q^2 \left(\frac{d}{\sigma} \sqrt{T_{sym}} \right)$ per lecture slide 15-15. Q is monotonically decreasing (lecture slide 14-22). In addition to reducing the symbol error rate by reducing the symbol rate f_{sym} or equivalently increasing the symbol time T_{sym} , one can increase d , which is half of the distance between constellation points. Increasing d increases transmit power, which in turn increases SNR.

- (f) In a communication system, the sampling rate used in the receiver must always be equal to the sampling rate used in the transmitter. 3 points. **False.** Consider a transmitter that implements cosine modulation by a carrier frequency f_c in discrete time. Sampling rate must be greater than $2(f_c + W)$ where W is the baseband bandwidth. The receiver will demodulate by first modulating the bandpass signal and the sampling rate must exceed $2(2f_c + W)$. Receiver could use 2x sampling rate, and the number of samples per symbol period L would double. Also, transmitter and receiver usually have different circuits. Symbol timing recovery is used to synchronize the receiver sampling clock with the transmitter sampling clock.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 *Solutions 2.0*

Date: May 4, 2022

Course: EE 445S Evans

Name: _____ **Breakfast** _____ **Japanese** _____ **(Band)**
Last, First

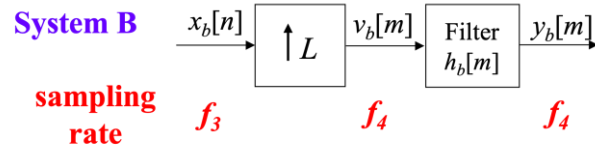
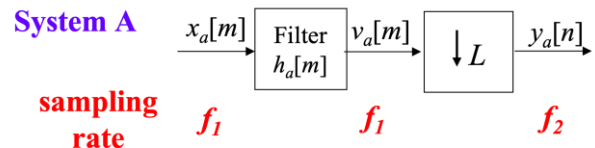
- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

	Problem	Point Value	Your score	Topic
<i>Peter Bradley</i>	1	19		Changing Sampling Rates
<i>Michelle Zauner</i>	2	33		QAM Communication Performance
<i>Craig Hendrix</i>	3	24		Adaptive Spatial Filter
<i>Deven Craige</i>	4	24		Potpourri
	Total	100		

Problem 2.1. Changing Sampling Rates. 19 points.

Consider the two systems to change the sampling rate:

- System A consists of linear time-invariant (LTI) filtering followed by downsampling by L .
- System B consists of upsampling by L followed by LTI filtering.



(a) Give a formula for f_2 in terms of f_1 . 2 points.

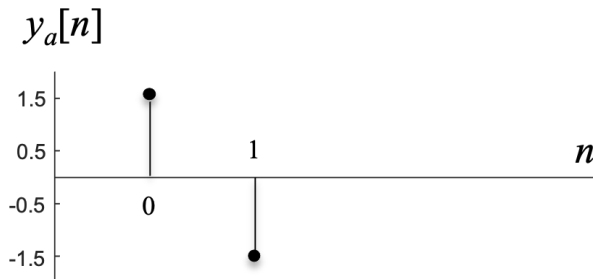
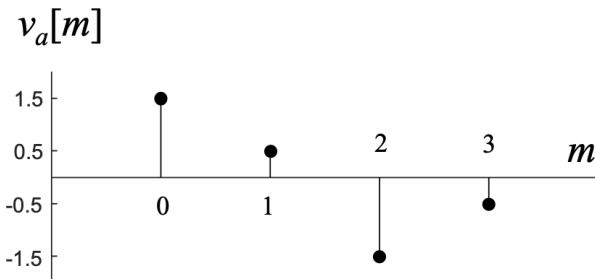
$$f_2 = \frac{1}{L} f_1$$

(b) Give a formula for f_4 in terms of f_3 . 2 points.

$$f_4 = L f_3$$

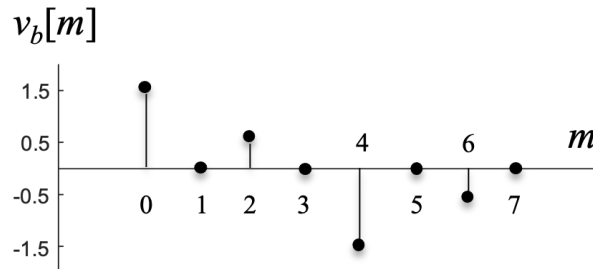
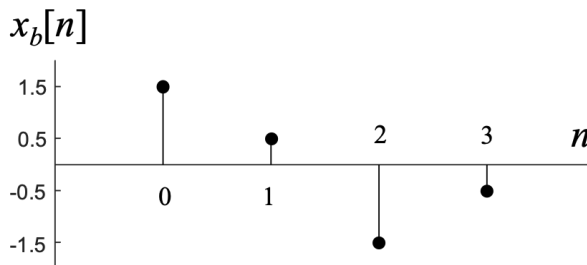
(c) Assuming $L = 2$, draw $y_a[n]$ corresponding to the input $v_a[m]$ shown below. 4 points.

Lecture Slide 26-9; HW 2.2(d)



(d) Assuming that $L = 2$, draw $v_b[m]$ corresponding to the input $x_b[n]$ shown below. 4 points.

Lecture Slides 13-8 and 26-7; HW 2.2(e)



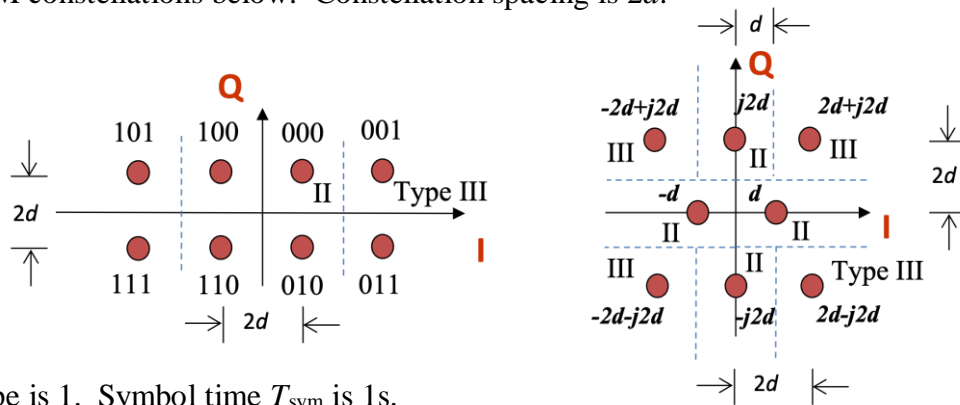
(e) Assume the filter in System B is a finite impulse response (FIR) filter with N coefficients.

1. How many multiplication operations per second does System B use in the block diagram above? 3 points. **Upsampling does not require any multiplications. An FIR filter with N coefficients requires N multiplications to compute one output sample in response to an input sample. FIR filter processes f_4 samples per second. Total: $N f_4$ mults/sec.**
2. How many multiplication operations per second would System B use if implemented as a polyphase filter bank? 4 points. **A polyphase filterbank uses a bank of L polyphase FIR filters, each with N/L coefficients, followed by a commutator to take the L parallel outputs and put them in sequential order. Each polyphase FIR filter runs at the lower rate, f_3 . Total: $N f_3$ mults/sec. Savings in mults/sec by a factor of L because $f_4 = L f_3$.**

Lecture Slides 13-9, 13-14 to 13-16, and 26-8; HW 5.3

Problem 2.2 QAM Communication Performance. 33 points.

Consider the two 8-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s.

Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$10d^2$	$8d^2$
(b) Average transmit power	$6d^2$	$\frac{4(8d^2) + 2(4d^2) + 2(d^2)}{8} = 5.25d^2$
(c) Peak-to-average power ratio	$\frac{10d^2}{6d^2} = \frac{5}{3} \approx 1.67$	$\frac{8d^2}{5.25d^2} = \frac{32}{21} \approx 1.52$
(d) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation that will minimize the probability of symbol error using such decision regions.		
(e) Number of type I QAM regions	0	0
(f) Number of type II QAM regions	4	4
(g) Number of type III QAM regions	4	4
(h) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2 .	$P_e = \frac{5}{2}Q\left(\frac{d}{\sigma}\right) - \frac{3}{2}Q^2\left(\frac{d}{\sigma}\right)$	Same as left constellation due to same number of type I, II and III regions
(i) Express the argument of the Q function as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{6d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{6}}$	$\text{SNR} = \frac{5.25d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{5.25}}$

(j) Give one advantage of the left constellation vs. the right constellation. 3 points.

- **Gray coding** minimizes the number of bit errors when there is a symbol error. The bit pattern for each symbol only differs in one bit with that of the nearest neighbor. Left constellation can be gray coded as shown above; right constellation cannot be gray coded because we only have two bits of freedom at each constellation point and the constellation points at locations d and $-d$ have four nearest neighbors.
- **Binary search** fast algorithm can be used for decoding the received symbol amplitude by finding the nearest constellation point. Each iteration rules out half the points with a single

real-valued comparison. Binary search achieves same accuracy as selecting the constellation point that is closest in Euclidean distance to the received symbol amplitude. For the right constellation, there is no vertical or horizontal line through the complex plane that will eliminate half the points; however, decoding the received symbol constellation using the decision regions will be as accurate as using Euclidean distance.

- **Rectangular constellation** has 4-PAM (QPSK) in the in-phase component and 2-PAM (BPSK) in the quadrature component. The QAM receiver separates into a 4-PAM receiver in the in-phase direction and 2-PAM receiver in the quadrature direction, which can simplify the design. For example, one could apply a single Costas Loop to only the in-phase component to determine the phase offset for the baseband QAM carrier, instead of having to run eight Costas loops in parallel per homework problem 7.3. The right constellation is not a rectangular constellation.

(k) Give one advantage of the right constellation vs. the left constellation. *3 points.*

- **Lower peak transmit power** for same value of d
- **Lower average transmit power** for same value of d
- **Lower average peak-to-average power ratio** which makes it easier to design the power amplifier for the analog/RF chains in the transmitter and receiver
- **Lower symbol error rate vs. SNR.** The Q function is a non-negative monotonically decreasing function of its non-negative argument. The probability of error expressions are identical in terms of $\frac{d}{\sigma}$. Let SNR = 6 in linear units. The value of $\frac{d}{\sigma}$ is 1 for the left constellation and 1.2 for the right constellation, which means the right constellation has lower symbol error rate.

Problem 2.3 Adaptive Spatial Filter. 24 points.

A single sound source is recorded by several microphones simultaneously as shown on the right.

The microphones are arranged in a line and separated by distance d . Sound arrives at an unknown angle θ .

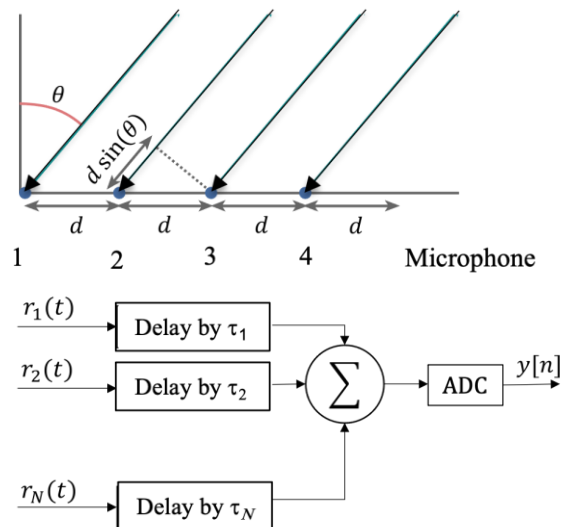
Sound arrives at the i th microphone with a different delay t_i based on the distance to the source. The source is far enough away for the propagation to be a plane wave.

The signal recorded by the i th microphone $r_i(t)$ is delayed by τ seconds, and all the signals are added together before being sampled by an analog-to-digital converter:

We would like to design an adaptive spatial filter that amplifies a signal located at an unknown angle θ by adapting the delays $\tau_1, \tau_2, \dots, \tau_N$.

A known training signal $x[n]$ is sent by the source so that we can find the best values for $\tau_1, \tau_2, \dots, \tau_N$.

Hence $r_i(t) = x(t - t_i)$ where $x(t)$ is the continuous-time version of the training signal $x[n]$.



(a) **Training.** What training signal $x[n]$ would you send? Why? Describe its parameters. 6 points.

We want $x[n]$ to be easy to generate at the receiver, have good correlation properties and contain all discrete-time frequencies because the channel will attenuate/reject some frequencies. After digital-to-analog conversion (DAC) of $x[n]$ at rate f_s , continuous-time signal $x(t)$ would have all frequencies $-\frac{1}{2}f_s$ to $\frac{1}{2}f_s$.

Option #1: Long maximal-length pseudo-noise sequence. Length is $2^r - 1$ bits where r is the number of states in the PN generator. Map 1 bit to +1 and 0 bit to -1 in amplitude. As an audio signal, a PN sequence sounds like noise.

Option #2: Chirp signal that sweeps from 0 Hz to 20 kHz (upper end of audible range) over 0 to t_{max} seconds. $x(t) = \cos(2\pi \gamma t^2)$ where $\gamma = \frac{10 \text{ kHz}}{t_{max}}$. For the DAC, use $f_s > 44 \text{ kHz}$.

Since chirp sounds can be startling, we could use low-volume chirps, or chirps in 15-20 kHz range, where human hearing is less sensitive.

(b) **Propagation Delay.** Consider the signal received by the i th microphone $r_i(t)$. Propose and explain an algorithm to find the delay t_i from the source to the i th microphone. 6 points.

Correlate output of i th microphone $r_i(t)$ against the training sequence, and the location of the first peak will provide an estimate of the delay t_i from the source to the i th microphone.

This method works because the correlation function provides a measure of similarity between $r_i(t)$ and delayed versions of $x(t)$. We had seen this in HW 4.2, 4.3, and 5.2; lab #4 on PN sequences; and labs 5-6 on matched (correlation) filtering.

With the estimates of the t_i values, we can determine the best τ_i values as follows without the need for an adaptive algorithm:

$$t_{max} = \max([t_1 \ t_2 \ \dots \ t_N])$$

$$\tau_i = t_{max} - t_i \text{ for } i = 1, \dots, N$$

This will allow all the versions of the received training signal to be aligned in time and add constructively. This assumes the sound source is not moving. One of the advantages of an adaptive algorithm is to be able to track a moving sound source. We can use the above calculation as the initial values for τ_i for the adaptive algorithm.

- (c) **Adaptive Spatial Filter.** Develop a discrete-time adaptive algorithm to apply to $y[n]$ to determine the best set of delays

$$\vec{\tau} = [\tau_1 \quad \tau_2 \quad \cdots \quad \tau_N]$$

for the microphone array to amplify the sound coming from the source at an unknown angle θ .

1. Give an objective function and explain why you have chosen it. *3 points.*

$$J(e[n]) = \frac{1}{2} e^2[n] \text{ where } e[n] = y[n] - x[n - n_0] \text{ and } n_0 \text{ is a constant average delay.}$$

We can choose $n_0 = 0$ for simplicity. Driving $e^2[n]$ to zero will drive $e[n]$ to zero.

2. Give an adaptive steepest descent/ascent algorithm for $\vec{\tau}[i + 1]$ in terms of $\vec{\tau}[i]$. *6 points.*

The adaptive steepest descent algorithm to minimize $J(e[n])$ is

$$\vec{\tau}[n + 1] = \vec{\tau}[n] - \mu \left. \frac{d}{d\vec{\tau}} J(e[n]) \right|_{\vec{\tau}=\vec{\tau}[n]} = \vec{\tau}[n] - \mu e[n] \left. \frac{d}{d\vec{\tau}} y[n] \right|_{\vec{\tau}=\vec{\tau}[n]}$$

since $x[n]$ does not depend on $\vec{\tau}$. This would be a sufficient answer on the exam.

To keep working the problem,

$$y(t) = \sum_{i=1}^N r_i(t - \tau_i) = \sum_{i=1}^N x(t - t_i - \tau_i)$$

$$y[n] = y(n T_s) = \sum_{i=1}^N x(n T_s - t_i - \tau_i)$$

$$\frac{d}{d\tau_i} y[n] = \frac{d}{d\tau_i} x(n T_s - t_i - \tau_i)$$

At this point, we have not included knowledge of the microphone array geometry or the training signal. If a chirp training signal is being used, then $x(t) = \cos(2\pi \gamma t^2)$ and

$$\frac{d}{d\tau_i} x(n T_s - t_i - \tau_i) = \frac{d}{d\tau_i} \cos(2\pi \gamma (n T_s - t_i - \tau_i)^2)$$

$$\frac{d}{d\tau_i} x(n T_s - t_i - \tau_i) = 4\pi \gamma (n T_s - t_i - \tau_i) \sin(2\pi \gamma (n T_s - t_i - \tau_i)^2)$$

3. What values of the step size would you use? Why? *3 points.*

We would like to small positive values of the step size μ such as 0.01 to ensure convergence of the algorithm. Using $\mu = 0$ would not allow the iterative algorithm to update. Using a negative μ would convert the steepest descent algorithm to minimize $e[n]$ into a steepest ascent algorithm to maximize $e[n]$. A large positive value would cause the steepest descent algorithm to diverge.

Problem 2.4. Potpourri. 24 points.

(a) Consider 16-QAM system transmitting at a 1200 bps (bits per second) using a 12-sample discrete-time raised cosine pulse shaping filter. What are the possible sampling rates in Hz? 12 points.

Bit rate is $J f_{sym}$ where J is number of bits/symbol and f_{sym} is symbol rate in symbols/s or Hz. With $J = 4$ bits/symbol due to 16-level QAM and a bit rate of 1200 bps, $f_{sym} = 300$ Hz. Pulse shape has $N = N_g L$ samples where N_g is number of symbol periods in the pulse shape and L is number of samples in a symbol period. With $N = 12$ given in the question, possible factorizations are $L = 1$ and $N_g = 12$; $L = 2$ and $N_g = 6$; $L = 3$ and $N_g = 4$; $L = 4$ and $N_g = 3$; $L = 6$ and $N_g = 2$; $L = 12$ and $N_g = 1$.

With the sampling rate $f_s = L f_{sym}$, the possible sampling rates are 300, 600, 900, 1200, 1800, and 3600 Hz.

Lecture slide 13-3 for bit rate; slides 13-9 to 13-16 for pulse shape length; slide 13-7 for sampling rate; Labs 5 & 6

(b) Consider a wireless communication system that uses two transmit antennas and two receive antennas. This allows two signals $x_1[n]$ and $x_2[n]$ to be sent at the same time and over the same frequency band as shown below: 12 points

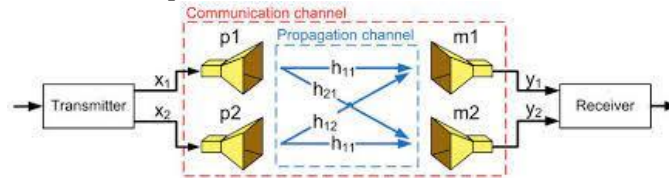


Figure from Lars Reichardt, Juan Pontes, Yoke Leen Sit, and Thomas Zwick, “[Antenna Optimization for Time-Variant MIMO Systems](#)”, EuCap, 2011.

Each antenna at the receiver receives both transmitted signals.

The communication channel has a **complex-valued** scalar gain between the i th transmit antenna and j th receive antenna. No other impairments are being modeled.

The received signal is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

which can be written as

$$\vec{y} = \mathbf{H} \vec{x}$$

Lecture 12; JSK Sec. 2.1, 2.10 & 2.12; JSK Ch. 8 & 11; Labs 5 & 6; HW 4.2, 4.3, 5.2, 5.3, 6.1, 6.2, 7.1, 7.2 & 7.3

In the receiver, assume \mathbf{H} is known.

Find **two possible values** for matrix \mathbf{G} in terms of \mathbf{H} that can allow us to estimate \vec{x} from \vec{y} via

$$\vec{x}_{estimated} = \mathbf{G} \vec{y}$$

Hints: One way could equalize (invert) the channel. Other could use a matched filtering approach.

We want to find $\vec{x}_{estimated} = \mathbf{G} \vec{y} = \mathbf{G} \mathbf{H} \vec{x}$ so that we can recover \vec{x} from $\vec{x}_{estimated}$.

We can equalize (invert) the channel by using $\mathbf{G} = \mathbf{H}^{-1}$ which would give $\vec{x}_{estimated} = \mathbf{G} \mathbf{H} \vec{x} = \mathbf{H}^{-1} \mathbf{H} \vec{x} = \vec{x}$. The closed-form formula for \mathbf{H}^{-1} is

$$\mathbf{H}^{-1} = \frac{1}{\det(\mathbf{H})} \begin{bmatrix} h_{22} & -h_{12} \\ -h_{21} & h_{11} \end{bmatrix}$$

In general, $\mathbf{G} = \alpha \mathbf{H}^{-1}$ for any non-zero scalar α .

The idea of a matched filter is phase reversal. For example, in pulse amplitude modulation, the matched filter impulse response is $h[m] = k g^*[L - m]$ where $g[m]$ is the pulse shape. For this problem, we let $\mathbf{G} = \mathbf{H}^*$ where \mathbf{H}^* means the conjugate transpose of \mathbf{H} :

$$\mathbf{H}^* = \begin{bmatrix} h_{11}^* & h_{21}^* \\ h_{12}^* & h_{22}^* \end{bmatrix} \text{ so that } \mathbf{H}^* \mathbf{H} = \begin{bmatrix} h_{11}^2 + h_{21}^2 & h_{11}^* h_{12} + h_{21}^* h_{22} \\ h_{12}^* h_{11} + h_{22}^* h_{21} & h_{12}^2 + h_{22}^2 \end{bmatrix}$$

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Date: December 5, 2022

Course: EE 445S Evans

Name: _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

Problem	Point Value	Your score	Topic
1	22		Baseband PAM System
2	30		QAM Communication Performance
3	24		Channel Equalization
4	24		Potpourri
Total	100		

Problem 2.1. Baseband PAM System. 22 points.

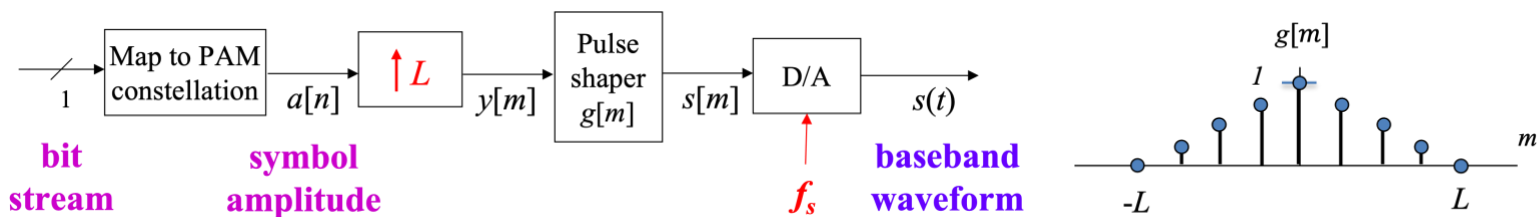
Consider a binary phase shift keying (BPSK) system, a.k.a. a two-level pulse amplitude modulation (2-PAM) system.

The system parameters are described on the right:

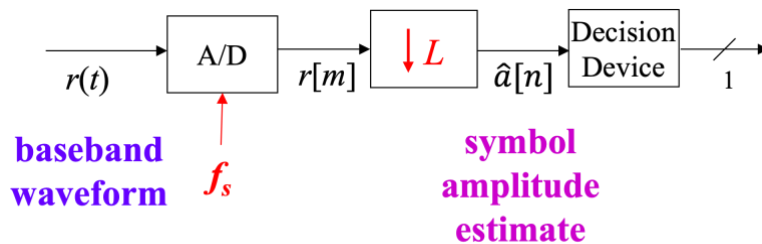
- $J = 1$ bit/symbol
- $L = 4$ samples per symbol period
- Pulse shape $g[m]$ is a **triangular** pulse of 8 samples in duration plotted below on the right.
- A bit of value 0 is mapped to symbol amplitude $-I$, and a bit of value 1 is mapped to symbol amplitude I .

(a) For the **BPSK transmitter** below, the input bit stream is 101. Plot the discrete-time signals $a[n]$, $y[m]$ and $s[m]$. 12 points.

PAM System Parameters	
$a[n]$	symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse resp.
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index

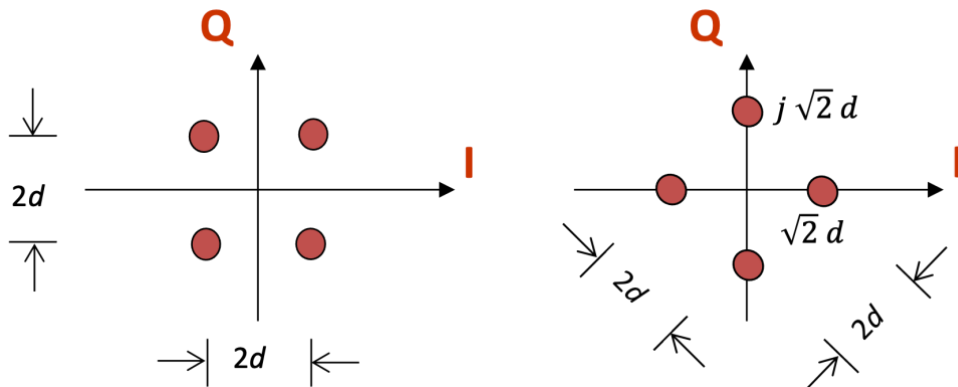


(b) For the **BPSK receiver** to the right, assume there is no channel distortion or additive noise and assume that $r[m] = s[m]$. **There is no matched filter.** Plot the discrete-time signal $\hat{a}[n]$ and give the received bit stream based on the BPSK transmitter in (a). 10 points.



Problem 2.2 QAM Communication Performance. 30 points.

In transmitting synchronization sequences from its low-earth orbit satellites to ground terminals, the SpaceX Starlink system uses both 4-QAM constellations below [1]. The constellation on the right is the constellation on the left rotated by 90 degrees. Constellation spacing for both constellations is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s.

Each part below is worth 3 points. **Please fully justify your answers. Show intermediate steps.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$2d^2$	
(b) Average transmit power	$2d^2$	
(c) Peak-to-average power ratio	$\frac{2d^2}{2d^2} = 1$	
(d) For the left constellation, the decision regions are the four quadrants and the IQ axes are the boundaries. For the right constellation, draw rotated type I, II and/or III decision regions <i>that will minimize the probability of symbol error using such decision regions.</i>		
(e) Number of type I QAM regions	0	
(f) Number of type II QAM regions	0	
(g) Number of type III QAM regions	4	
(h) Probability of symbol error in presence of additive Gaussian noise with zero mean & variance σ^2 .	$P_e = 2Q\left(\frac{d}{\sigma}\right) - Q^2\left(\frac{d}{\sigma}\right)$	
(i) Express the argument of the Q function as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{2d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{2}}$	

(j) Give a Gray coding for right constellation or show that one does not exist. 3 points.

[1] T.E. Humphreys, P.A Iannucci, Z. Komodromos and A.M. Graff, “[Signal Structure of the Starlink Ku-Band Downlink](#)”, submitted Oct. 20, 2022. The primary and secondary synchronization sequences use the 4-QAM constellation on the right, and the coda synchronization sequence uses the 4-QAM constellation on the left. The primary synchronization sequence can also use 16-QAM in high SNR scenarios.

Problem 2.3. Channel Equalization. 24 points.

In the discrete-time system on the right, the equalizer operates at the sampling rate. The equalizer is a finite impulse response (FIR) filter with N real coefficients w_0, w_1, \dots, w_{N-1} :

$$r[m] = w_0 y[m] + w_1 y[m-1] + \dots + w_{N-1} y[m - (N-1)]$$

Channel model is an FIR filter with impulse response $h[m]$ in cascade with additive noise $n[m]$.

When using a least mean squares (LMS) cost function

$$J_{LMS}(e[m]) = \frac{1}{2} e^2[m]$$

the adaptive update equation using a steepest descent algorithm becomes

$$\vec{w}[m + 1] = \vec{w}[m] - \mu e[m] \vec{y}[m]$$

$$\text{where } \vec{w}[m] = [w_0[m] \quad w_1[m] \quad \dots \quad w_{N-1}[m]]$$

$$\text{and } \vec{y}[m] = [y[m] \quad y[m - 1] \quad \dots \quad y[m - (N - 1)]]$$

With channels that have deep nulls in their frequency response, and a high SNR in the received signal, the minimization of the LMS cost function results in a large spike in the equalizer frequency response to compensate for the nulls. The large spikes also amplify additive noise.

To inhibit large values of FIR filter coefficients w_i needed to create a frequency response with bands of high gain, consider a cost function that penalizes the sum of the squares of the FIR coefficients:

$$J_{LMSnorm}(e[m]) = \frac{1}{2} e^2[m] + \frac{1}{2} \lambda \sum_{i=0}^{N-1} w_i^2[m]$$

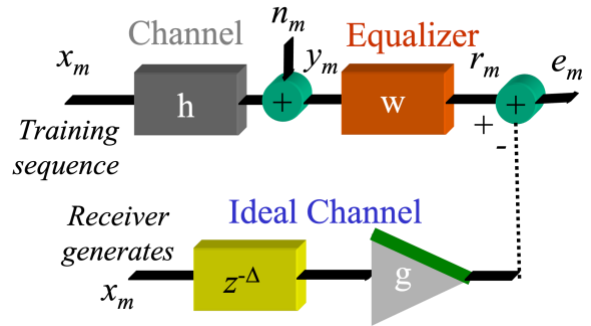
where λ is a positive constant.

(a) What training sequence for $x[m]$ would you use? Why? 3 points.

(b) How you would estimate the delay parameter Δ in the ideal channel model. 3 points.

(c) Derive the adaptive update equation for the FIR equalizer coefficients using the $J_{LMSnorm}(e[m])$ cost function. 15 points.

(d) What values of the step size (learning rate) μ would you use? 3 points.



Problem 2.4. Potpourri. 24 points.

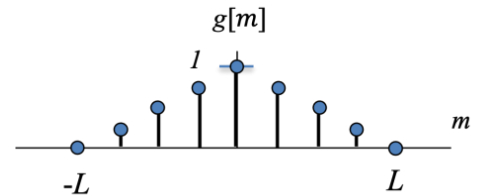
(a) **Duplexing.** A cellular communication system is deployed in one of two ways at present:

- *Frequency division duplexing* means the basestation sends data to the smart phone on one frequency band and the smart phone sends data to the basestation on another frequency band.
- *Time division duplexing* means that in a timeslot, the basestation sends data to the smart phone in the first 40% of the timeslot and the smart phone sends data to the basestation in the last 40% of the timeslot. The 20% gap is needed to toggle each side between transmit and receive.

In a full duplex system, the basestation is sending/receiving data to/from the smart phone at the same time and in the same frequency band.

- What is the maximum spectral efficiency improvement of full duplex vs. frequency division duplexing? Why? 6 points.
- What is the maximum spectral efficiency improvement of full duplex vs. time division duplexing? Why? 6 points.

(b) For the baseband PAM receiver in problem 2.1(b), we'll include additive noise in the channel model and add a matched filter in the receiver to compensate for additive noise. The pulse shape in the transmitter (where $L = 4$ samples) is shown on the right:



- Give a general formula for the impulse response $h[m]$ of the matched filter that optimizes the SNR in the symbol amplitude estimate $\hat{a}[n]$ and that is expressed in terms of $g[m]$, L , and possibly other system parameters. 4 points.
- Plot the impulse response $h[m]$ of the matched filter that optimizes the SNR in $\hat{a}[n]$. 4 points.
- What is the reduction in the symbol error rate by using the matched filter? 4 points.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Date: April 24, 2023

Course: EE 445S Evans

Name: _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

Problem	Point Value	Your score	Topic
1	22		Baseband PAM System
2	30		QAM Communication Performance
3	24		Symbol Timing Recovery
4	24		Potpourri
Total	100		

Problem 2.1. Baseband PAM System. 22 points.

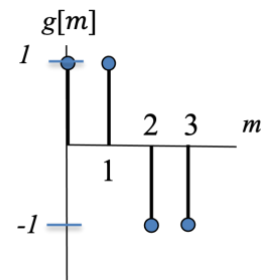
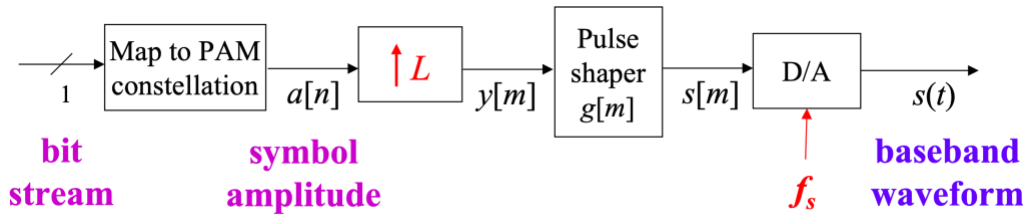
Consider a binary phase shift keying (BPSK) system, a.k.a. a two-level pulse amplitude modulation (2-PAM) system.

The system parameters are described on the right:

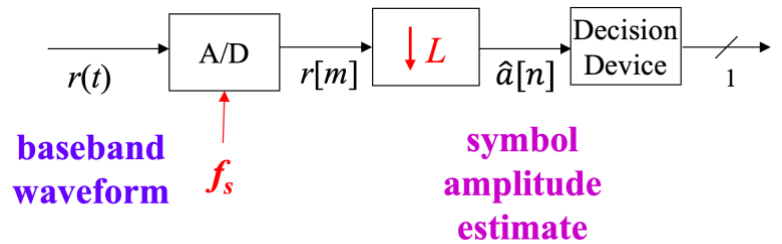
- $J = 1$ bit/symbol
- $L = 4$ samples per symbol period
- Pulse shape $g[m]$ is a **Manchester non-return-to-zero** pulse of 4 samples in duration plotted below on the right.
- A bit of value 0 is mapped to symbol amplitude $-I$, and a bit of value 1 is mapped to symbol amplitude I .

(a) For the **BPSK transmitter** below, the input bit stream is 101. Plot the discrete-time signals $a[n]$, $y[m]$ and $s[m]$. 12 points.

PAM System Parameters	
$a[n]$	symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse resp.
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index

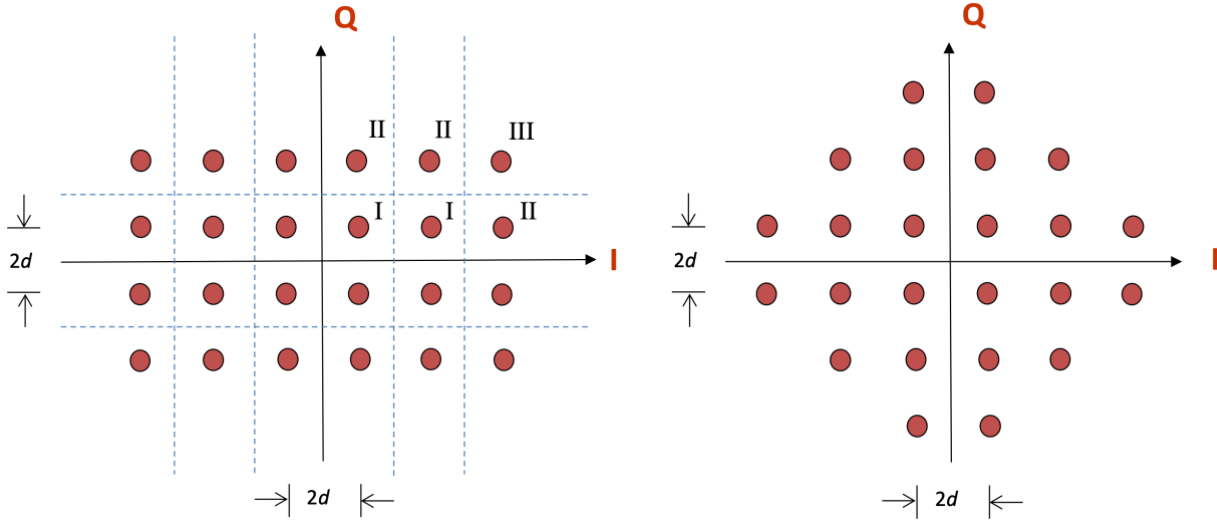


(b) For the **BPSK receiver** to the right, assume there is no channel distortion or additive noise and assume that $r[m] = s[m]$. **There is no matched filter.** Plot the discrete-time signal $\hat{a}[n]$ and give the received bit stream based on the BPSK transmitter in (a). 10 points.



Problem 2.2 QAM Communication Performance. 30 points.

Consider the two 24-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s.

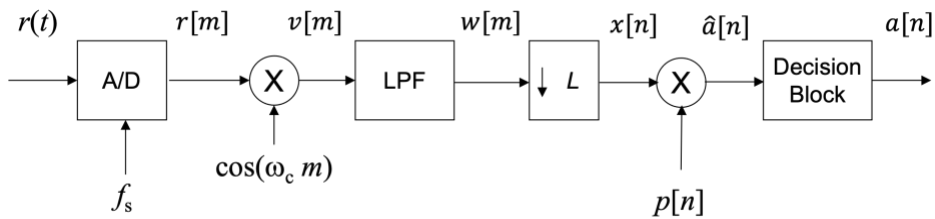
Each part below is worth 3 points. **Please fully justify your answers. Show intermediate steps.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$34 d^2$	
(b) Average transmit power	$\frac{50}{3} d^2$	
(c) Peak-to-average power ratio	$\frac{34 d^2}{\frac{50}{3} d^2} = \frac{51}{25} = 2.04$	
(d) For the left constellation, IQ axes are also boundaries of decision regions. For the right constellation, draw type I, II and/or III decision regions.		
(e) Number of type I QAM regions	8	
(f) Number of type II QAM regions	12	
(g) Number of type III QAM regions	4	
(h) Probability of symbol error in presence of additive Gaussian noise with zero mean & variance σ^2 .	$P_e = \frac{19}{6} Q\left(\frac{d}{\sigma}\right) - \frac{5}{2} Q^2\left(\frac{d}{\sigma}\right)$	
(i) Express the argument of the Q function as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{\left(\frac{50}{3}\right) d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{3}{50} \text{SNR}}$	

(j) Give a Gray coding for right constellation or show that one does not exist. 3 points.

Problem 2.3. Symbol Timing Recovery. 24 points.

For the upconverted pulse amplitude modulation (PAM) receiver,



you'll develop an adaptive algorithm to generate the symbol timing offset τ to be used in the symbol timing offset correction signal

$$p[n] = \cos(2 \pi f_{sym} \tau n)$$

The lowpass filter (LPF) is the matched filter.

(a) What training sequence for the symbol amplitude values $a[n]$ would you use? Why? 3 points.

(b) Derive the steepest descent algorithm to update $\tau[n]$ using the objective function

$$J_{DD}[n] = e^2[n]$$

where $e[n]$ is the decision-directed (DD) error signal

$$e[n] = \hat{a}[n] - a[n]$$

and

$$\hat{a}[n] = p[n] x[n]$$

15 points.

(c) What is initial value of $\tau[n]$ would you use? 3 points.

(d) What value of the step size (learning rate) μ would you use? 3 points.

PAM System Parameters

$a[n]$	Tx symbol amplitude
$\hat{a}[n]$	Rx symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index
$p[n]$	symbol timing offset correction signal
ω_c	discrete-time carrier freq.

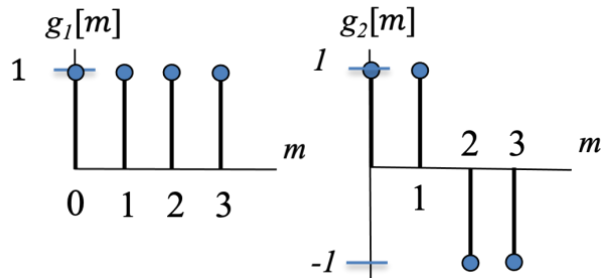
Problem 2.4. Potpourri. 24 points.

(a) One version of the MP3 (MPEG layer III) audio compression standard divides an audio signal into 32 equally spaced frequency bands. The direct approach uses a set of 32 bandpass FIR filters in parallel and each FIR filter output is downsampled by 32. Each FIR filter has 512 coefficients.

- i. How many multiplication operations are needed for the direct approach? *4 points.*
- ii. How many multiplication operations are needed if each of the 32 cascades of a bandpass FIR filter and downsampling by 32 are implemented as a polyphase filter bank? *4 points.*
- iii. What is the amount of savings for the polyphase approach? *4 points.*

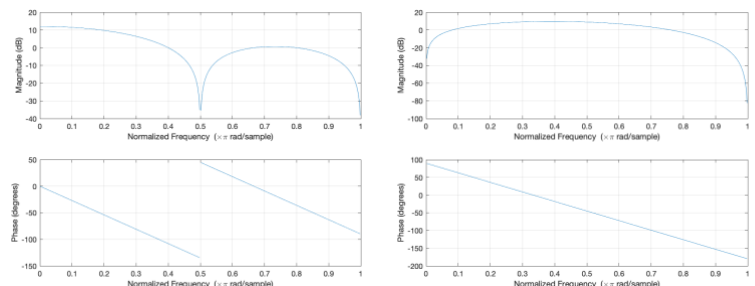
(b) You're evaluating two candidate pulse shapes for a baseband pulse amplitude modulation (PAM) system – rectangular pulse $g_1[m]$ and Manchester non-return-to-zero pulse $g_2[m]$. The system is like problem 2.1 but with a matched filter after the A/D converter. Symbol period is $L = 4$ samples.

- i. The pulse shape is the impulse response of the pulse shaping filter. How many multiplications does each pulse shaping filter for $g_1[m]$ and $g_2[m]$ require? *4 points.*



- ii. For a pulse shape $g[m]$, the optimal choice of the impulse response of the matched filter is $h[m] = C g^*[L - m]$ where C is a constant and $*$ is complex conjugation. Plot the optimal matched filter impulse responses $h_1[m]$ and $h_2[m]$ for pulse shapes $g_1[m]$ and $g_2[m]$, respectively. *4 points.*

- iii. Here are the magnitude and phase plots in dB and degrees for $g_1[m]$ and $g_2[m]$ by using the freqz command in Matlab. Which pulse shape would you choose for your system and why? *4 points.*



Rectangular Pulse

Manchester Pulse

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Date: December 4, 2023

Course: EE 445S Evans

Name: _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

Problem	Point Value	Your score	Topic
1	22		Baseband PAM System
2	30		QAM Communication Performance
3	27		Decision-Directed Channel Equalization
4	21		Communication System Tradeoffs
Total	100		

Problem 2.1. Baseband PAM System. 22 points.

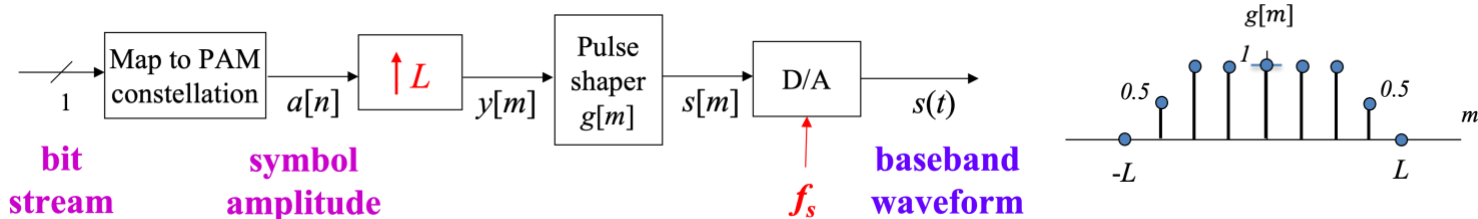
Consider a binary phase shift keying (BPSK) system, a.k.a. a two-level pulse amplitude modulation (2-PAM) system.

The system parameters are described on the right:

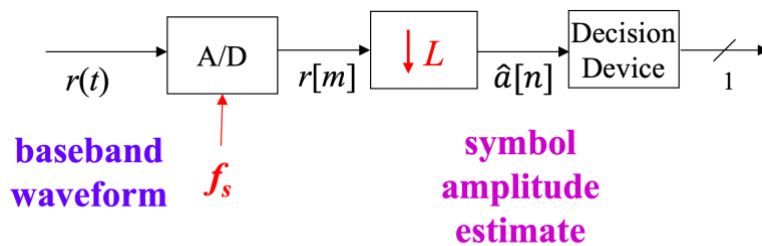
- $J = 1$ bit/symbol
- $L = 4$ samples per symbol period
- Pulse shape $g[m]$ is a trapezoidal pulse of 8 samples in duration from $m = -4$ to $m = 3$ inclusive of both and plotted below.
- A bit of value 0 is mapped to symbol amplitude -1, and a bit of value 1 is mapped to symbol amplitude 1.

(a) For the **BPSK transmitter** below, the input bit stream is 01. Plot the discrete-time signals $a[n]$, $y[m]$ and $s[m]$. 12 points.

PAM System Parameters	
$a[n]$	symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse resp.
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index

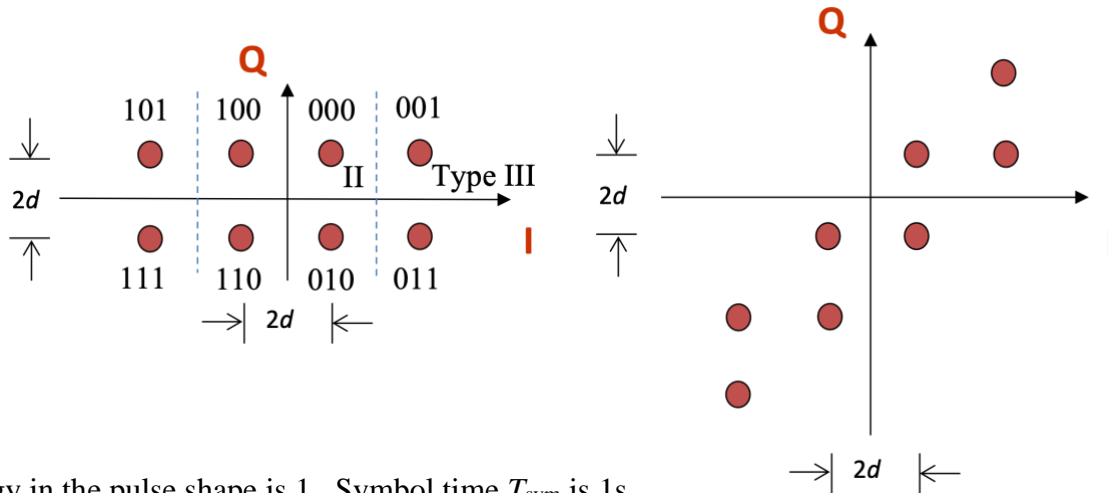


(b) For the **BPSK receiver** to the right, assume there is no channel distortion or additive noise and assume that $r[m] = s[m]$. **There is no matched filter.** Plot the discrete-time signal $\hat{a}[n]$ and give the received bit stream based on the BPSK transmitter in (a). 10 points.



Problem 2.2 QAM Communication Performance. 30 points.

Consider the two 8-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s.

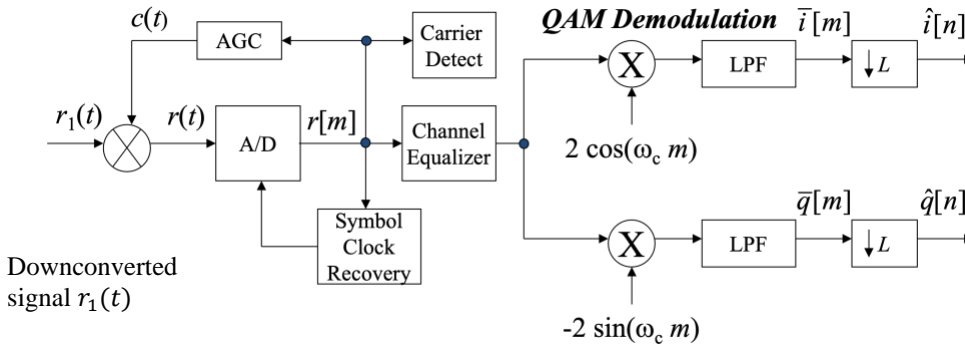
Each part below is worth 3 points. **Please fully justify your answers. Show intermediate steps.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$10d^2$	
(b) Average transmit power	$6d^2$	
(c) Peak-to-average power ratio	$\frac{10d^2}{6d^2} = \frac{5}{3} \approx 1.67$	
(d) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation that will minimize the probability of symbol error using such decision regions.		
(e) Number of type I QAM regions	0	
(f) Number of type II QAM regions	4	
(g) Number of type III QAM regions	4	
(h) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2 .	$P_e = \frac{5}{2}Q\left(\frac{d}{\sigma}\right) - \frac{3}{2}Q^2\left(\frac{d}{\sigma}\right)$	
(i) Express the argument of the Q function as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{6d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{6}}$	

(j) Give a Gray coding for right constellation or show that one does not exist. 3 points.

Problem 2.3. Decision-Directed Channel Equalization. 27 points.

For a quadrature amplitude modulation (QAM) receiver,



The adaptive FIR channel equalizer has N coefficients $\vec{w} = [w_0 \ w_1 \ \dots \ w_{N-1}]$.

The channel equalizer runs at the sampling rate but updates at the symbol rate.

Decision error $e[n]$ is the Euclidean distance between the transmitted symbol amplitude $i[n] + j q[n]$ and the received symbol amplitude $\hat{i}[n] + j \hat{q}[n]$ at symbol index n .

Consider adapting the decision-directed channel equalizer during training.

(a) What training sequence for the symbol amplitude values would you use? Why? 3 points.

(b) Give an objective function $J(n)$. Why did you choose it? 6 points.

(c) What is the initial value of the adaptive FIR equalizer coefficients you would use? Why? 3 points.

(d) Derive an update equation for the adaptive FIR equalizer coefficients vector at iteration n

$$\vec{w}[n] = [w_0[n] \ w_1[n] \ \dots \ w_{N-1}[n]]$$

Compute all derivatives. Simplify the result. 12 points

(e) What range of values would you recommend for the step size μ ? Why? 3 points.

QAM System Parameters

$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$i[n]$	Tx symbol amplitude in-phase component
$\hat{i}[n]$	Rx symbol amplitude in-phase component
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index
$q[n]$	Tx symbol amplitude quadrature component
$\hat{q}[n]$	Rx symbol amplitude quadrature component

Problem 2.4. Communication System Tradeoffs. 21 points.

Two-way communication systems have a data channel and a control channel in each direction.

The data channel conveys high speed data such as streaming audio or video whereas the control channel conveys configuration and feedback information. For example, a phone will send the received signal strength to the basestation on a control channel.

The control channel provides lower bit rates at lower symbol error rates with higher delay relative to the data channel.

- (a) For an increase in each parameter below, indicate whether the pulse amplitude modulation (PAM) communication system performance measure in each column increases, decreases, or stays the same. As you consider increasing one of the parameters in the table, assume the other two parameters in the table are not changing in value. Assume the symbol rate, f_{sym} , does not change.
- (b) Justify your answers.

Increasing Parameter	Effect on the Bit Rate	Effect on Symbol Error Rate	Effect on Delay
J bits/symbol			
L samples/symbol			
N_g symbol periods in pulse shape			

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2

Date: April 29, 2024

Course: EE 445S Evans

Name: _____
Last, First

- **Exam duration.** The exam is scheduled to last 75 minutes.
- **Materials allowed.** You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks.** Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **No AI tools allowed.** As mentioned on the course syllabus, you may not use GPT or other AI tools during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers.** When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab.** No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test.** All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

Problem	Point Value	Your score	Topic
1	25		Baseband PAM System
2	30		QAM Communication Performance
3	27		Channel Estimation
4	18		Communication System Tradeoffs
Total	100		

Problem 2.1. Baseband PAM System. 25 points.

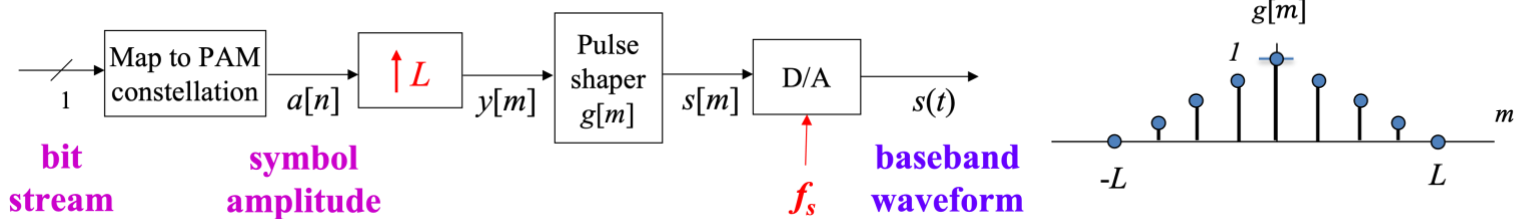
Consider a binary phase shift keying (BPSK) system, a.k.a. a two-level pulse amplitude modulation (2-PAM) system.

The system parameters are described on the right:

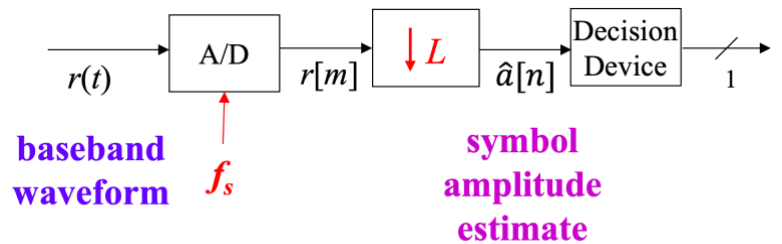
- $J = 1$ bit/symbol
- $L = 4$ samples per symbol period
- Pulse shape $g[m]$ is a triangular pulse of 8 samples in duration from $m = -4$ to $m = 3$ inclusive of both and plotted below.
- A bit of value 0 is mapped to symbol amplitude -1, and a bit of value 1 is mapped to symbol amplitude 1.

(a) For the **BPSK transmitter** below, the input bit stream is 10. Plot the discrete-time signals $a[n]$, $y[m]$ and $s[m]$. 12 points.

<u>PAM System Parameters</u>	
$a[n]$	symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse resp.
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index



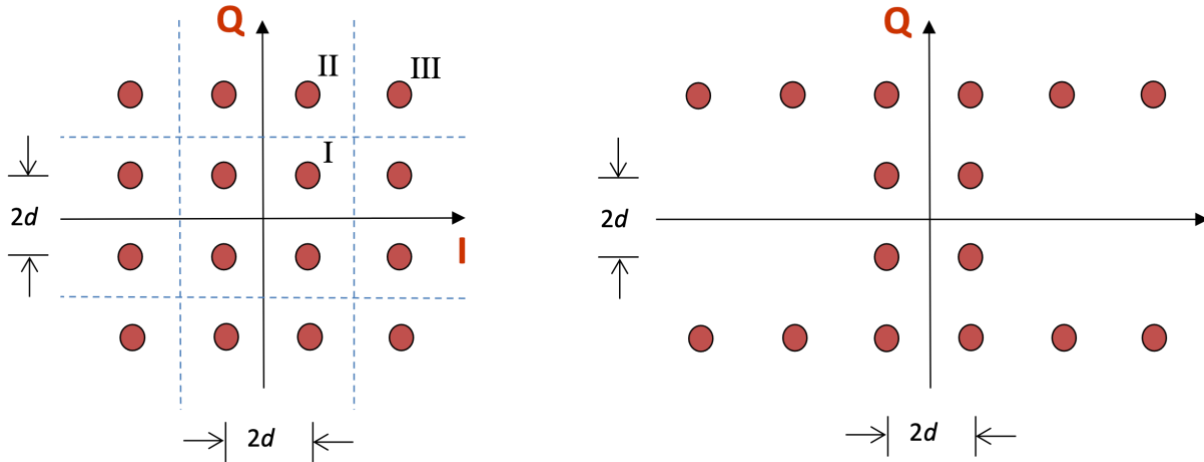
(b) For the **BPSK receiver** to the right, assume there is no channel distortion or additive noise and assume that $r[m] = s[m]$. **There is no matched filter.** Plot the discrete-time signal $\hat{a}[n]$ and give the received bit stream based on the BPSK transmitter in (a). 7 points.



(c) What is the formula for the optimal matched filter $h[m]$? Plot the optimal matched filter impulse response $h[m]$ given the pulse shape $g[m]$ in part (a). 6 points.

Problem 2.2 QAM Communication Performance. 27 points.

Consider the two 16-QAM constellations below. Constellation spacing is $2d$.



Energy in the pulse shape is 1. Symbol time T_{sym} is 1s. The constellation on the left includes the decision regions with boundaries shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

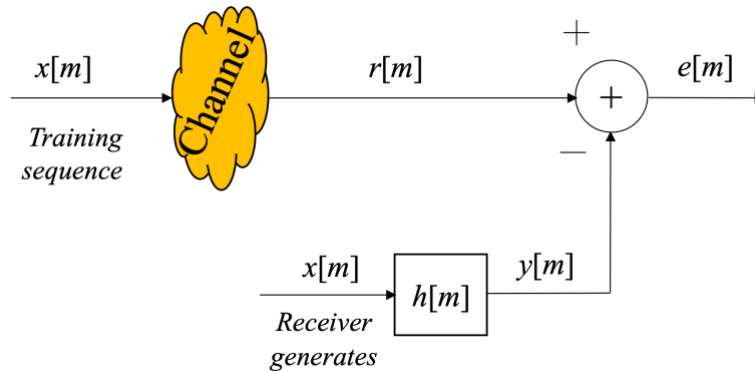
Each part below is worth 3 points. **Please fully justify your answers.**

	Left Constellation	Right Constellation
(a) Peak transmit power	$18d^2$	
(b) Average transmit power	$10d^2$	
(c) Draw the type I, II and/or III decision regions for the right constellation on top of the right constellation <i>that will minimize the probability of symbol error using such decision regions.</i>		
(d) Number of type I regions	4	
(e) Number of type II regions	8	
(f) Number of type III regions	4	
(g) Probability of symbol error for additive Gaussian noise with zero mean & variance σ^2	$3Q\left(\frac{d}{\sigma}\right) - \frac{9}{4}Q^2\left(\frac{d}{\sigma}\right)$	
(h) Express d/σ as a function of the Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = \frac{10d^2}{\sigma^2}$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{10}}$	

(i) For the right constellation, will using your type I, II, and III rectangular decision regions lead to Gray coding for symbols? Either give a Gray coding for the right constellation, or show that it is not possible. 3 points.

Problem 2.3. Channel Estimation. 27 points.

In a communication system, we model impairments between the baseband transmitter output $x[m]$ and baseband receiver input $r[m]$ as the baseband equivalent channel shown as “Channel” below:



The baseband equivalent channel is a nonlinear, time-varying system.

A common step in many communication systems is to model the baseband equivalent channel as a linear-time invariant (LTI) system and estimate its impulse response $h[m]$.

We will adapt the baseband equivalent channel impulse response $h[m]$ estimate during training.

We will model the LTI system $h[m]$ as an FIR filter with M coefficients: $\vec{h} = [h_0 \ h_1 \ \dots \ h_{M-1}]$

(a) What training sequence $x[m]$ would you use? Why? 3 points.

(b) Give an objective function $J[m]$. Why did you choose it? 6 points.

(c) What is the initial value of the adaptive FIR coefficients you would use? Why? 3 points.

(d) Derive an update equation for the adaptive FIR coefficients vector at iteration m

$$\vec{h}[m] = [h_0[m] \ h_1[m] \ \dots \ h_{M-1}[m]]$$

Compute all derivatives. Simplify the result. 12 points

(e) What range of values would you recommend for the step size μ ? Why? 3 points.

Problem 2.4. Communication System Tradeoffs. 18 points.

Two-way communication systems have a data channel and a control channel in each direction.

The data channel conveys high speed data such as streaming audio or video whereas the control channel conveys configuration and feedback information. For example, a phone will send the received signal strength to the basestation on a control channel.

The control channel provides lower bit rates at lower symbol error rates and higher delays relative to the data channel.

- (a) For an increase in each parameter below, indicate whether the pulse amplitude modulation (PAM) transmitter performance measure in each column increases, decreases, or stays the same. As you consider increasing one of the parameters in the table, assume the other two parameters in the table are not changing in value. Assume the bits/symbol, J , and the number of samples in a symbol period, L , do not change.
- (b) Justify your answers.

<u>PAM System Parameters</u>	
$a[n]$	symbol amplitude
$2d$	constellation spacing
f_s	sampling rate
f_{sym}	symbol rate
$g[m]$	pulse shape
$h[m]$	matched filter impulse resp.
J	bits/symbol
L	samples/symbol period
M	levels, i.e. $M = 2^J$
m	sample index
n	symbol index

Increasing Parameter	Transmitter Total Power Consumption	Transmitter Run-Time Complexity	Transmitter Bit Rate
B bits in D/A converter			
$2d$ constellation spacing in Volts			
f_{sym} symbol rate in Hz			

Direct Sequence Spreading

Gene W. Marsh

I. A General Description of Direct Sequence Spreading

A. The standard view of a communication system

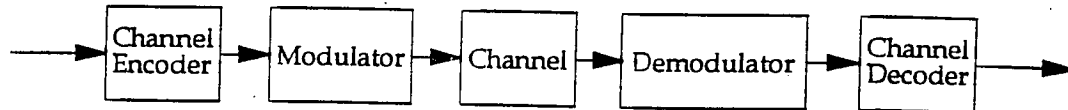


Fig. 1. A block diagram for a Standard Communication System.

B. A PN-Spread Communication System

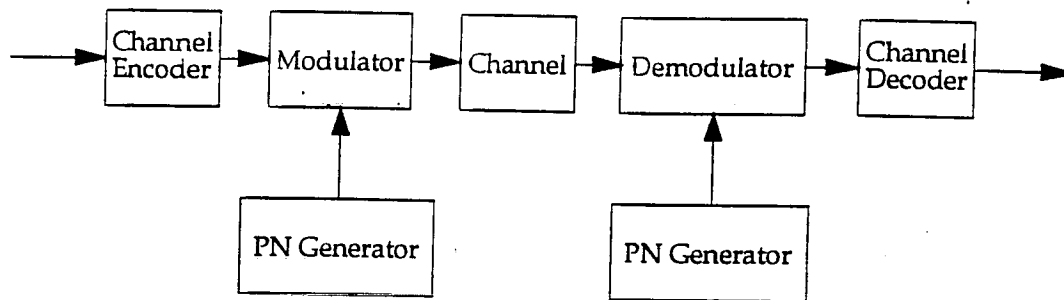


Fig. 2. A block diagram for a PN-spread Communication System.

1. To approach channel capacity, it is desirable to make the signal more noiselike. Therefore, we introduce a random number generator, to make the modulation appear random

C. A closer look at the modulator/demodulator.¹

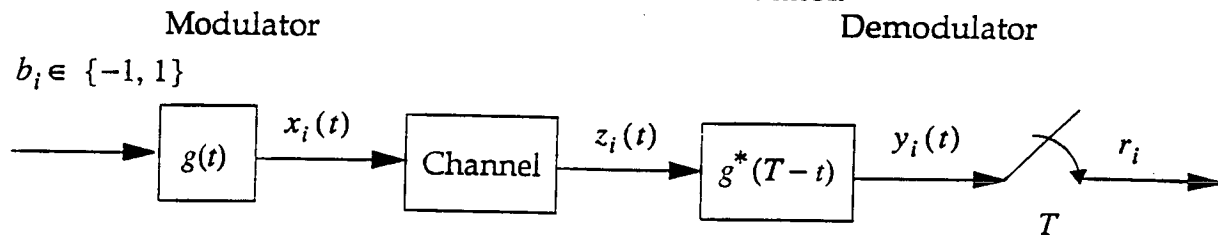


Fig. 3. The Standard Model

1. In the standard modulator pair shown in Figure 3, a bit determines whether the transmit filter or its inverse is emitted every T seconds.

1. If you like your bits to be from the set $\{0, 1\}$, then you can replace b_i and c_i with $2b_i - 1$ and $2c_i - 1$, below.

Direct Sequence Spreading

The waveform then passes through the channel, and is corrupted by noise.

The resulting signal is passed through a matched filter, and sampled every T seconds.

a) T is known as the **bit time** and $R = \frac{1}{T}$ is known as the **bit rate**.

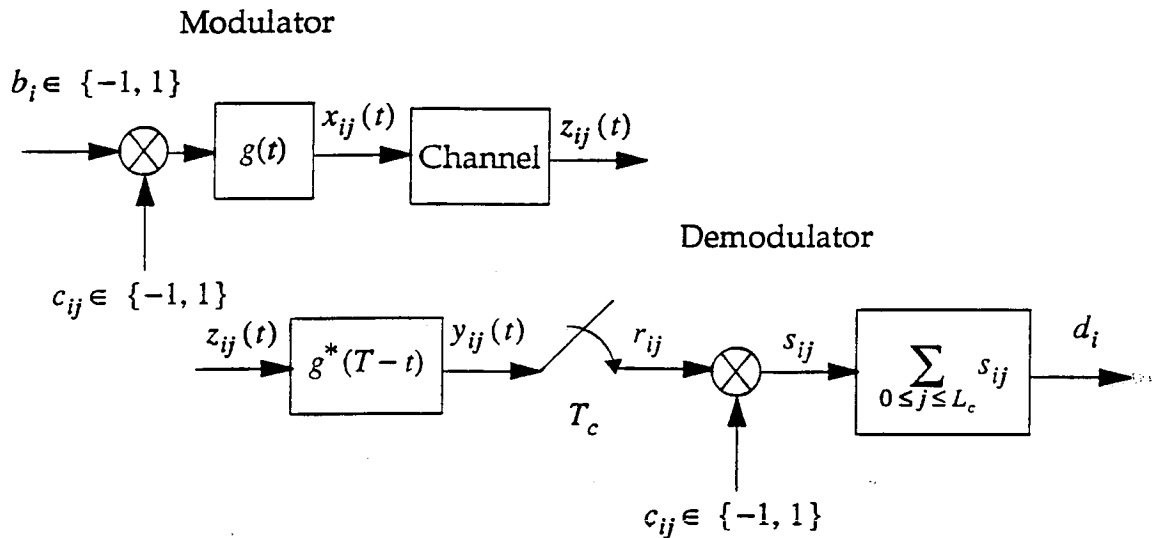


Fig. 4. The CDMA Model

2. In the CDMA system shown in Figure 3, a slightly different thing happens.

A bit still enters the system every T seconds, but it is now multiplied by a faster moving, random sequence every T_c .

The result is sent through the channel, sampled, and the sampled signal is multiplied by the corresponding random bit. The result is then summed.

a) Clearly things that move faster in time are wider in frequency. Hence the name "spreading".

b) Each of the short transmitted symbols is known as a **chip**.

c) The time T_c is then known as a **chip time**, while $R_c = \frac{1}{T_c}$ is known as the **chip rate** or **spreading rate**.

d) The spreading bits c_{ij} are assumed to be i.i.d Bernoulli random variables over $\{-1, 1\}$ with parameter $\frac{1}{2}$.

- e) Clearly, $L_c = \frac{T}{T_c} = \frac{R_c}{R}$ is the number of chips per bit. This is often referred to as the **bandwidth expansion factor**.

II. Performance issues

A. Does this little stunt cost us anything?

- Let us examine the performance of each system assuming the channel simply corrupts the signal using zero mean, white, additive Gaussian noise $n(t)$ with variance $\sigma^2 = \frac{N_0}{2}$, which is independent of the data symbols.

- In the standard system, we have

$$y_i(t) = b_i g(t) * g^*(T-t) + n(t) * g^*(T-t)$$

After sampling, we find that $E[d_i | b_i] = 2b_i E_b$ and

$$\sigma^2(d_i | b_i) = \frac{N_0}{2} \int_{-\infty}^{\infty} |G(f)|^2 df = \frac{N_0}{2} E_b \text{ where } E_b = \int_0^T |g(t)|^2 dt.$$

If we assume that there is no ISI, then successive samples of the noise are independent. The threshold for decisions is placed at 0. Assume zeros and ones are equally likely.

We can then calculate the probability of error as follows:

$$P(e | b_i = -1) = P(d_i > 0 | b_i = -1) = Q\left(\frac{E_b}{\sqrt{\frac{N_0}{2} E_b}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P(e | b_i = 1) = P(d_i < 0 | b_i = 1) = 1 - Q\left(\frac{-E_b}{\sqrt{\frac{N_0}{2} E_b}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P(e) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{1}$$

- In the CDMA system, for each chip, we have

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + n(t) * g^*(T-t)$$

Direct Sequence Spreading

The r_{ij} are then independent, with a Gaussian distribution such that $E[r_{ij}|b_i, c_j] = b_i c_j E_c$ and

$$\sigma^2(r_{ij}|b_i, c_j) = \frac{N_0}{2} \int_{-\infty}^{\infty} |G(f)|^2 df = \frac{N_0}{2} E_c \text{ where } E_c = \int_0^T |g(t)|^2 dt.$$

After the multiplier, the s_{ij} clearly have mean $E[s_{ij}|b_i, c_j] = b_i E_c$ and variance $\sigma^2(s_{ij}|b_i, c_j) = \sigma^2(r_{ij}|b_i, c_j)$. In particular, we should note that the s_{ij} do not depend on the c_{ij} at all.

Now, if we sum the L_c chips corresponding to a particular bit, we find that d_i is again Normal, with conditional mean and variance given by $E[d_i|b_i] = b_i L_c E_c$ and $\sigma^2(d_i|b_i) = \frac{N_0}{2} L_c E_c$. The probability of error is therefore still

$$P(e) = Q\left(\sqrt{\frac{2L_c E_c}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (2)$$

where $E_b = L_c E_c = \frac{R_c}{R} E_c$.

- c) Therefore, spreading has not cost us anything, except some complexity in the modulator/demodulator.
- (1) In general, this is only true for direct spreading in coherent communications systems.
 - (2) It should surprise no one that this is true, since, if you look carefully at our picture, all we have done is modify the transmit filter, and then made a matched filter for the receiver.

III. Pseudo-Noise (PN) Sequences

A. M-Sequences (Maximal Length Shift Register Sequences)

1. From Galois field theory, we have the notion of an M-sequence.
 - a) Let us consider Galois Field 2. This is composed of the elements 0 and 1 with addition defined by "exclusive-or" and multiplication defined by "and".
 - b) Let n be any positive integer. Then we can define a Galois Field with 2^n elements by considering all possible polynomials of degree $n - 1$ or less. Addition is defined by polynomial addition.

modulo $x^n + 1$, and multiplication is polynomial multiplication modulo x^n .

- c) For any such $GF(2^n)$, it is possible to find an element, α , such that, if $\beta \neq 0 \in GF(2^n)$, then $\beta = \alpha^k$ for some $0 \leq k < 2^n$. Thus, you can cycle through all of the elements of the field by multiplying repeated multiplication or division by α . These elements are the primitive elements of the field, and are represented by the primitive polynomials.
- d) There are tables of primitive polynomials.
- e) This sequence of n -bit numbers can be used to generate a sequence of $2^n - 1$ bits with useful properties (to be discussed later).
- f) For any primitive polynomial, there are two ways to generate this bit sequence. I will illustrate this by example. Let $n = 5$. The primitive polynomial for $GF(32)$ is $p(x) = x^5 + x^2 + 1$.
 - (1) The "xor into the middle" method is shown in Figure 5. In

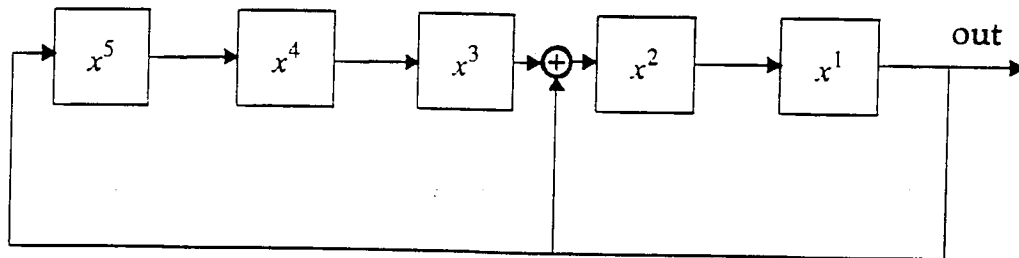


Fig. 5. A Galois Configuration

general, this is the easiest way to implement an M-sequence in software. The sequence produced by this machine is shown in Table 1.

Direct Sequence Spreading

Table 1: Sequence Generated by the Galois Configuration^a

State (Hex)	Output Bit	State (Hex)	Output Bit	State (Hex)	Output Bit	State (Hex)	Output Bit
1f	1	14	0	12	0	15	1
1d	1	a	0	9	1	18	0
1c	0	5	1	16	0	c	0
e	0	10	0	b	1	6	0
7	1	8	0	17	1	3	1
11	1	4	0	19	1	13	1
1a	0	2	0	1e	0	1b	1
d	1	1	1	f	1	1f	1

a. States are listed sequentially down the columns

- (2) The "shift around" method is shown in Figure 6. This is the preferred method for implementing an M-sequence in hardware. The sequence produced by this implementation is shown in Table 2. Note that the output sequence shown here is the reverse of the one in Table 1.

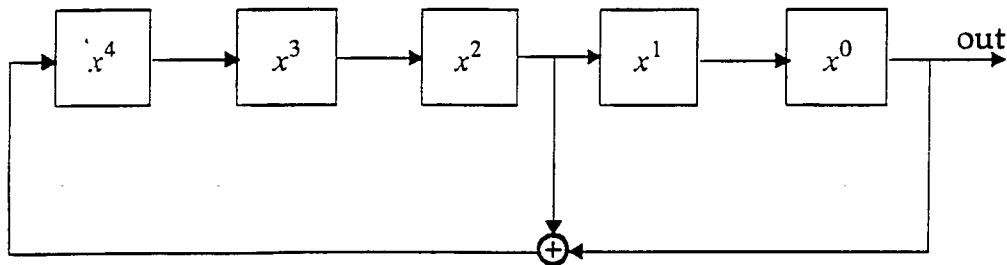


Fig. 6. A Fibonacci Configuration

Table 2: Sequence Generated by the Fibonacci Configuration^a

State (Hex)	Output Bit	State (Hex)	Output Bit	State (Hex)	Output Bit	State (Hex)	Output Bit
1f	1	1b	1	2	0	1a	0
f	1	1d	1	1	1	d	1
7	1	e	0	10	0	6	0
3	1	17	1	8	0	13	1
11	1	b	1	4	0	19	1
18	0	15	1	12	0	1c	0
c	0	a	0	9	1	1e	0
16	0	5	1	14	0	1f	1

a. States are listed sequentially down the columns.

2. Some nice properties of M-sequences.

- a) Even numbers of ones and zeros are produced.
 - b) The sequence produced is relatively uncorrelated with shifts of itself.
 - c) It is possible to generate the output bit in a different way.
 - (1) Associate with each bit in the register a 0 or a one i.e., create a mask for the register.
 - (2) To compute a bit, "and" the mask with the register. The parity of the result is the next output bit. Note: Do not destroy the contents of the register, as you need it to compute the next state.
 - (3) The sequences generated in this way are shifted versions of the original sequence. Indeed, all shifts of the sequence can be generated in this fashion.
 - d) Given a starting time and a clock rate, there are fast algorithms for computing the current state of the shift register.
3. Thus, M-sequences work well for generating our spreading bits. Indeed, this is the most common way to do it.

B. Acquisition of a Spread Spectrum Signal

1. Suppose we are generating our spreading codes using M-sequences. How do we get the sequences synchronized in the transmitter and the

receiver?

- a) Technically, this is known in the industry as acquiring the signal.
 - b) At Qualcomm, we referred to this as making the jump to hyperspace.
2. This is the hardest problem in any spread spectrum system.
 3. Here is how it is done.
 - a) First, tie the state of the shift register to absolute time. For instance, you can declare it to have the all one's state at midnight on January 1, 1992, and that you will clock it at a 1 MHz rate.
 - b) Now, suppose someone is sending you a signal, and you know what mask he is using. You need a clock. Check it, and determine what the state of the shift register is at this point in time, and initialize your shift register to that value.
 - c) Now, you need to accumulate a sufficient number of chips. If you accumulate N chips, the average value of the signal will be NE_c , and the variance will be $NE_c \frac{N_0}{2}$.
 - d) After accumulating N chips, assume that your clock is off by one register position, update it accordingly, and accumulate N new chips at this new time hypothesis.
 - e) In the end, if you have an n -bit shift register, you should accumulate $2^n - 1$ different hypothesis, one for each time offset. The hypothesis with the highest absolute value is declared to be the correct offset, the shift register is initialized to that value, and demodulation can begin.
 - f) Using the statistics of the signal, it is possible to compute the probability that you select the wrong time offset. By choosing a large enough value of N , it is possible to make this acceptably small.
 - g) Just in case, after declaring a hypothesis to be correct, we wait for a bit to see if the tracking loops take hold. This is a good way of detecting a bad acquisition.

IV. Applications of Spread Spectrum

A. Low probability of Intercept (LPI) Communications

1. If we simply take the original waveform and scale it down in time by L_c , then we decrease its energy by a factor of L_c .

$$E_c = \int_{-\infty}^{\infty} |g(L_c t)|^2 dt = \frac{1}{L_c} \int_{-\infty}^{\infty} |g(t')|^2 dt' = \frac{1}{L_c} E_b$$

After despreading, the energy has increased by a factor of L_c , as we saw above. Therefore, we have spread the same energy over a wider spectrum. This makes the signal harder to distinguish from the noise background. It is used in aircraft communications so make the signal hard to detect. It would be a shame for a stealth bomber to be given away by a radio transmission.

- a) This is the root of my earlier comment about making the signal more noise like.
2. This same thing can be seen by looking at the signal in time, where we have divided the signal energy up over a sequence of smaller chips.
3. In most spread systems, the signal is buried in noise.

B. Code Division Multiple Access (CDMA)

1. Consider a system where we have 2 signals. These signals use the same code bits, but one uses a time shifted version of the other, i.e. the second user is shifted in time by n chips. Let $k = \left(i + \left\lfloor \frac{j+n}{L_c} \right\rfloor \right)$ and $l = (j+n) \bmod L_c$. The undesired user adds some interference. We are going to assume that interference is Gaussian. We then have

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + b_k c_{kl} g(t) * g^*(T-t) + n(t) * g^*(T-t) \quad (3)$$

This will in turn give:

$$E[r_{ij} | b_i, c_j] = b_i c_{ij} E_c$$

$$\sigma^2(r_{ij} | b_i, c_{ij}) = \frac{N_0}{2} E_c + E_c^2$$

$$E[s_{ij} | b_i, c_{ij}] = b_i E_c$$

$$\sigma^2(s_{ij} | b_i, c_{ij}) = \sigma^2(r_{ij} | b_i, c_{ij})$$

$$E[d_i | b_i] = b_i L_c E_c$$

$$\sigma^2(d_i | b_i) = \frac{N_0}{2} L_c E_c + L_c E_c^2$$

Direct Sequence Spreading

$$P(e) = Q \left(\sqrt{\frac{(L_c E_c)^2}{\frac{N_0}{2} L_c E_c + L_c E_c^2}} \right) = Q \left(\sqrt{\frac{E_b}{\frac{N_0}{2} + E_c}} \right) \quad (4)$$

- a) Note that we assumed that the two users were lined up in time. In general, this is not true, but it is the worst case. In that sense, the above is an upper bound on what happens.
- b) We also assumed that the other user's interference was Gaussian. For a long enough spreading sequence, this is not unreasonable.
2. Other user looks like white noise to the current bit, contributing an energy equal to their chip energy. Thus, his interference is controlled.
3. It is possible to demodulate both users. We can use this to make a multiple access scheme called Code Division Multiple Access (CDMA).
 - a) Let each user use either
 - (1) a unique spreading sequence, different from and uncorrelated with all other users;
 - (2) the same spreading pattern, delayed by some number of chips from his fellows.
 - b) Every user in the system can demodulate his own signal. The other users simply raise the noise floor. For N users, we can bound the probability of error as

$$P(e) \leq Q \left(\sqrt{\frac{E_b}{\frac{N_0}{2} + (N-1) E_c}} \right) \quad (5)$$

- c) It is theoretically possible to approach Shannon capacity using this approach.
 - (1) If I remember right, "theoretically" is the correct word, because the demodulator needs to demodulate each signal, and subtract off its effect before it demodulates the next signal.

C. Anti-Jam (AJ) Communications

1. Jamming is the transmission of a signal in order to degrade a communication system.
2. Design of a communication system which is robust against jamming can be viewed as a game, where the transmitter tries to minimize the

- bit error rate, and the jammer tries to maximize the bit error rate.
3. Often, the jammer is power limited. In this case, what the transmitter wants to do is to force him to spread his power over the widest possible bandwidth, so that he wastes as much of his power as possible.
 4. Now, consider a channel with no noise, over which we send a BPSK signal. Suppose the jammer transmits a single loud tone at the carrier frequency. In a normal system, we can imagine the tone being loud enough that it would dominate our own signal, and we would demodulate that rather than the received signal. However, passing it through the despreader "modulates" that signal, making it look like wide band noise. Our own signal, on the other hand, becomes despread, making it look like a tone. The spread system performs better than the standard system in such a situation.
 5. Direct sequence spreading is not the best or most common technique used for AJ modems. Most of them use non-coherent FSK and hopping rather than spreading.

D. Position Location

1. To do time tracking for the system above, one takes 2 samples, one at $\frac{T}{4}$ and one at $\frac{3T}{4}$. Call them s_{ij}^e and s_{ij}^l . The time tracking metric is then computed as $TM = \left(\sum_{0 \leq j < L_c} s_{ij}^e \right)^2 - \left(\sum_{0 \leq j < L_c} s_{ij}^l \right)^2$. If we are sampling too late, then this metric will be positive, and if we are sampling too early, then this metric will be negative.
2. If we are off by a fraction of a chip, this metric drops appreciably. Therefore, it is possible to do time tracking down to a fraction of a chip.
 - a) For example, in CDMA cellular, we track to $\frac{1}{8}$ of a chip. We could do better, but this is deemed sufficient.
3. If the spreading sequence is tied to absolute time (more on this below), it is possible to use this feature to measure the time for a signal to travel from source to destination. Given several independent transmitters, one can do triangulation.
4. GPS works on a scheme similar to this.

E. Multi-path Mitigation

1. First, realize that we only really need concern ourselves with fading on the chip level.

Direct Sequence Spreading

- a) The wider your signal bandwidth, the less serious the effect of fading. This is because fades only have a certain bandwidth in the frequency domain. If your signal is wider than that bandwidth, it is unlikely that a fade will occur which will crush all of the signal energy.
 - b) The accepted way to combat fading is with diversity. In this case, direct sequence spreading gives you time diversity. Since a fade lasts for only a finite amount of time, it is possible that only a fraction of the chips that make up one bit will be faded.
2. Consider a case where we have 2 signals arriving with different delays, and suppose that the difference in the delays, τ is such that $T_c < \tau$. Let $\tau = nT_c + \tau_c$, $k = \left(i + \left\lfloor \frac{j+n}{L_c} \right\rfloor \right)$ and $l = (j+n) \bmod L_c$. We then have

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + b_k c_{kl} g(t+\tau) * g^*(T-t) + n(t) * g^*(T-t) \quad (6)$$

This will in turn give:

$$E[r_{ij} | b_i, c_j] = b_i c_{ij} E_c$$

$$\sigma^2(r_{ij} | b_i, c_{ij}) = \frac{N_0}{2} E_c + E'^2$$

$$E[s_{ij} | b_i, c_{ij}] = b_i E_c$$

$$\sigma^2(s_{ij} | b_i, c_{ij}) = \sigma^2(r_{ij} | b_i, c_{ij})$$

$$E[d_i | b_i] = b_i L_c E_c$$

$$\sigma^2(d_i | b_i) = \frac{N_0}{2} L_c E_c + L_c E'^2$$

$$P(e) = Q \left(\sqrt{\frac{(L_c E_c)^2}{\frac{N_0}{2} L_c E_c + L_c E'^2}} \right)$$

Now, we know that $E' \leq E_c$, so that

$$P(e) \leq Q \left(\sqrt{\frac{L_c E_c}{N_0/2 + E_c}} \right) = Q \left(\sqrt{\frac{E_b}{N_0/2 + E_c}} \right) \quad (7)$$

3. What is the significance of this?
 - a) Any signal arriving at a time offset greater than one chip looks like white noise to the current bit. Thus, ISI is controlled.
 - b) It is possible to demodulate the late path as well. This helps improve our performance. Thus, the ISI can help you. This is a benefit a narrowband system cannot provide.

V. CDMA Cellular

A. Capacity of the CDMA System²

1. The capacity of the system is determined by the capacity of the mobile-to-cell link.
 - a) This is principally because the mobiles must use smaller transmitters, and cannot synchronize their signals.
2. Let us begin by realizing that the capacity of the system is controlled by the signal to noise ratio needed to achieve an acceptable link.
 - a) Since we have a wideband system, it is possible to use powerful coding techniques with little penalty. Because of this, the CDMA system only requires $\left(\frac{E_b}{N_0} \right)_{\text{desired}} = 7 \text{ dB}$.
3. Assume the following:
 - a) The system is interference limited, i.e. that the noise from other users in our own cell is much greater than the background thermal noise.
 - b) Assume that we only care about users in our own cell.
 - c) All signals are power controlled so that they reach the cell with equal power.³
4. Under these assumptions, we see from (5) that the signal-to-noise ratio is just $\frac{E_b}{(N-1)E_c} = \frac{R_c}{R(N-1)}$.

2. This is a "back-of-the-envelope" approximation. It is relatively easy to understand, and gives approximate results. Actual capacity is determined through simulation and testing.

3. This is crucial for achieving good capacity in the CDMA system.

Direct Sequence Spreading

- a) We need this to be equal to $\left(\frac{E_b}{N_0}\right)_{\text{desired}}$.
- b) Solving gives $N \approx \frac{R_c}{R} \cdot \frac{1}{\left(\frac{E_b}{N_0}\right)_{\text{desired}}}$.
5. The assumption that we are limited only by the users in our own cell is too generous. We can mitigate this by appropriate scaling.
- a) With an omnidirectional antenna, the number of users is decreased by about a factor of $F = 0.6$.
- b) In the actual system, the cells use sectorized antennas. Each sector has a field of view of 120° . Due to overlap between the sectors, this only buys back a factor of $G = 2.55$.
- c) Note that both of these numbers are empirical.
6. In a CDMA system, it is possible to have a variable rate transmission, so that one does not transmit when there is no data.
- a) Speech does not occur 100% of the time in a conversation.
- b) The amount of dead time varies with the language spoken. For English, the empirical number is $d = 0.4$.
- c) By using a variable rate vocoder, and not transmitting during the silent times, we get to take advantage of this, and our signal to noise ratio increases appropriately.
7. Therefore, the total capacity of the system is approximately

$$N \approx \frac{R_c}{R} \cdot \frac{1}{\left(\frac{E_b}{N_0}\right)_{\text{desired}}} \cdot \frac{1}{d} \cdot F \cdot G \quad (8)$$

8. For the CDMA system, $R = 9600$ Hz and $R_c = 1.2288$ MHz.
9. Using the numbers above, this gives us about 98 CDMA channels in a 1.25MHz bandwidth.
10. After blocking is considered, this yields about a factor of 20 increase in the number of calls per cell.
- a) This estimate varies considerably depending on who is doing it, as most of the data is empirical.

B. Power Control and the Near/Far Problem

1. One of the principle assumptions made in the capacity calculation

above was that all mobiles reach the cell at the same power level. This assumption is absolutely essential to operation of the system.

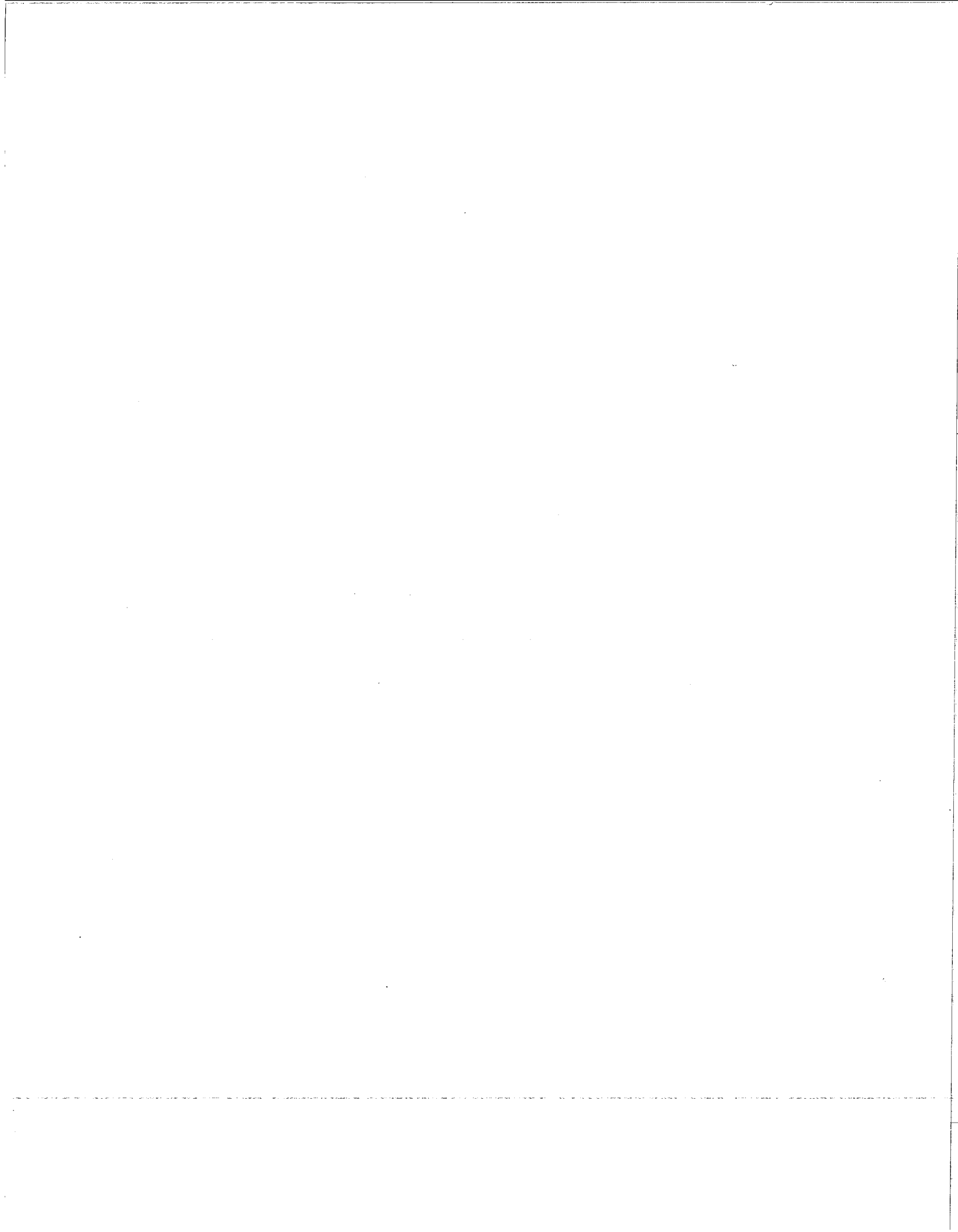
2. As mobiles are spread out over the entire cell, this is somewhat problematical, to say the least.
3. In order to combat this, we had to institute a method of power control. Very simply, it works as follows.
 - a) the mobile and the cell both monitor their average received signal-to-noise ratio (SNR). The mobile communicates his received SNR to the cell.
 - b) The cell adjusts his output to provide an acceptable SNR to the cell.
 - c) If the SNR received at the cell is too low, the cell tells the mobile to increase his power.
 - d) If all cells controlling the mobile tell him to increase his power, he does. Otherwise, he decreases his power.
 - e) In this way, as a mobile leaves one cell, he will begin to pick up another. Both cells will tell him to increase his power, so that both can hear.
 - f) As he moves into the new cell, it will tell him to decrease his output, and he will. Eventually, the old cell can no longer hear him.
 - g) This is the basic mechanism for a handoff, but the reality is much more complicated.

C. A Final Note

1. There is a lot more that can be said about CDMA cellular. Due to time pressure, I will stop here. [4] provides a good general description of what CDMA cellular is and how it works.

VI. References

- [1] John G. Proakis. *Digital Communications*. McGraw-Hill Book Company, New York, 1983.
- [2] Marvin K. Simon, Jim K. Omura, Robert A. Sholtz, Barry K. Levitt. *Spread Spectrum Communications, Volume 1*. Computer Science Press, Rockville, Maryland, 1985.
- [3] Richard E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1983
- [4] *An Overview of the Application of Code Division Multiple Access (CDMA) to Digital Cellular Systems and Personal Cellular Networks*. Document Number EX60-10010. Qualcomm, Inc., San Diego, CA, May 21, 1992.



Symbol Recovery Simplified

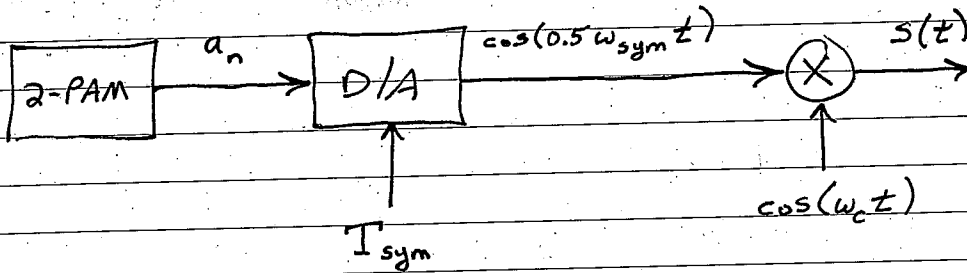
EE345S
Evans



Transmit 2-PAM Symbols

$$a_n = (-1)^n = \cos(n\pi) = \cos(0.5 \omega_{sym} T_{sym} n)$$

Receiver



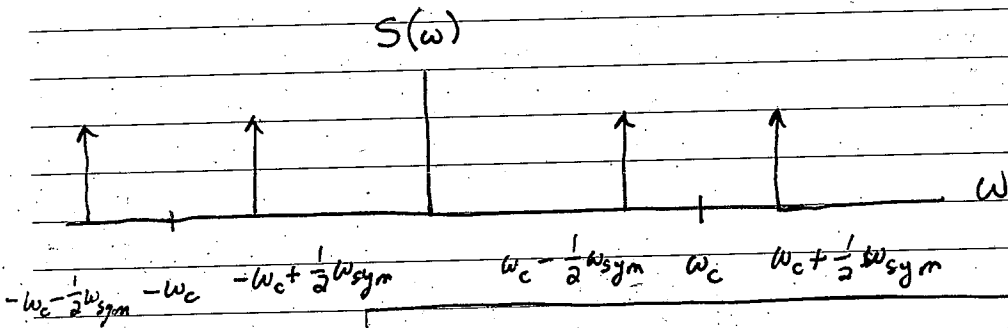
$$f[n] = e^{j\omega_{sym}(nT_{sym} + T)}$$

$$= e^{j(\omega_{sym}T_{sym}n + \omega_{sym}T)}$$

$$= e^{j(2\pi n + \omega_{sym}T)}$$

$$f[n] = e^{j\omega_{sym}T}$$

$$s(t) = \cos(0.5 \omega_{sym} t) \cos(\omega_c t)$$



$$p[n] = \alpha v[n] + \beta y[n]$$

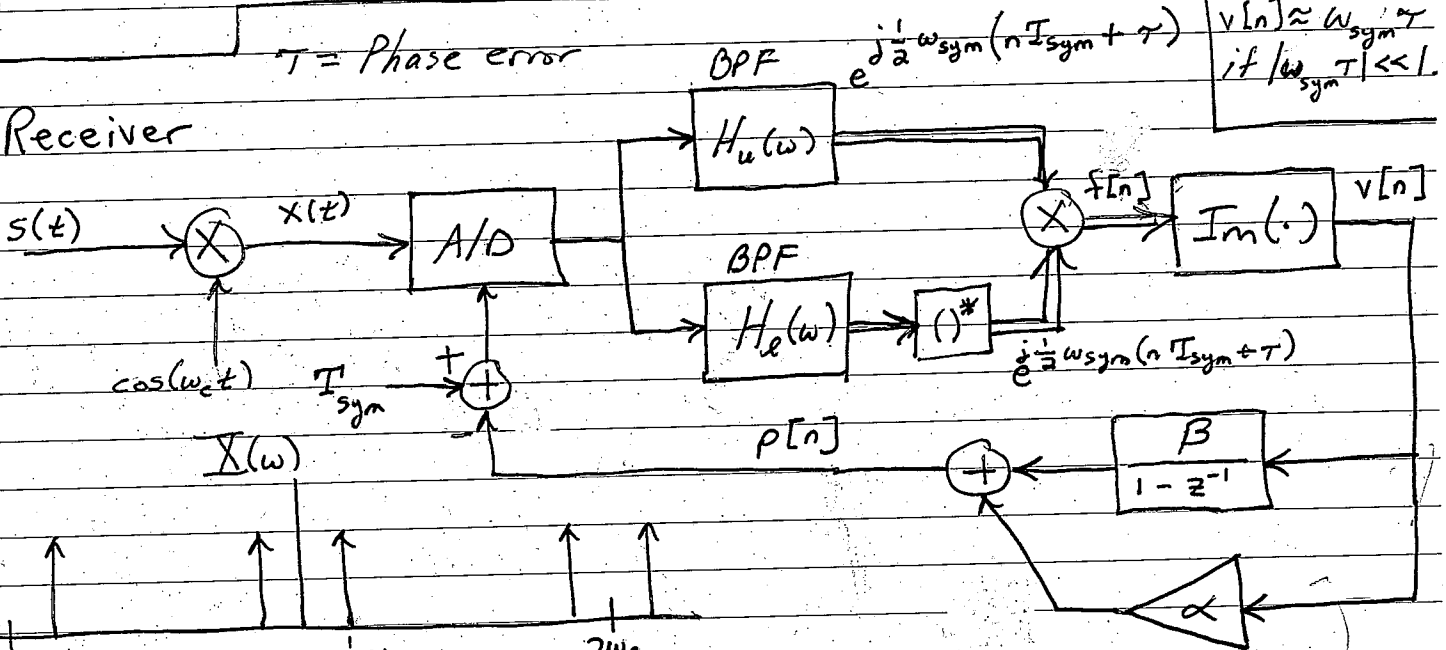
$$\delta[n] = v[n] + \delta[n-1]$$

with $\delta[-1] = 0$.

$$v[n] = \text{Im}(e^{j\omega_{sym}nT}) = \sin(\omega_{sym}nT)$$

$T = \text{Phase error}$

Receiver



$-2\omega_c$ $-\frac{1}{2}\omega_{sym}$ $\frac{1}{2}\omega_{sym}$ $2\omega_c$

"DEFINING MEETING EXCELLENCE"
 500 North IH - 35 • Austin, Tx 78701
 Business Center Phone: 512-457-7907
 Business Center Fax: 512-457-7995
www.crownplazaatx.com

Ask about our other locations.

Discussion on YouTube at <https://www.youtube.com/watch?v=usu5Yp6EQfQ> (32:30 to 47:39)

Computational Complexity of Implementing a Tapped Delay Line on the C6700 DSP

To compute one output sample $y[n]$ of a finite impulse response filter of N coefficients (h_0, h_1, \dots, h_{N-1}) given one input sample $x[n]$ takes N multiplication and $N-1$ addition operations:

$$y[n] = h_0 x[n] + h_1 x[n-1] + \dots + h_{N-1} x[n - (N-1)]$$

Two bottlenecks arise when using single-precision floating-point (32-bit) coefficients and data on the C6700 DSP. First, only one data value and one coefficient can be read from internal memory by the CPU registers during the same instruction cycle, as there are only two 32-bit data busses. The load command has 4 cycles of delay and 1 cycle of throughput. Second, accumulation of multiplication results must be done by four different registers because the floating-point addition instruction has 3 cycles of delay and 1 cycle of throughput. Once all of the multiplications have been accumulated, the four accumulators would be added together to produce one result. The code below does not use looping, and does not contain some of the necessary setup code (e.g. to initiate modulo addressing for the circular buffer of past input data).

<u>Cycle</u>	<u>Instruction</u>
1	LDW $x[n]$ LDW h_0
2	LDW $x[n-1]$ LDW h_1 ZERO accumulator0
3	LDW $x[n-2]$ LDW h_2 ZERO accumulator1
4	LDW $x[n-3]$ LDW h_3 ZERO accumulator2
5	LDW $x[n-4]$ LDW h_4 ZERO accumulator3
6	LDW $x[n-5]$ LDW h_5 MPYSP $x[n], h_0, \text{product0}$
7	LDW $x[n-6]$ LDW h_6 MPYSP $x[n-1], h_1, \text{product1}$
8	LDW $x[n-7]$ LDW h_7 MPYSP $x[n-2], h_2, \text{product2}$
9	LDW $x[n-8]$ LDW h_8 MPYSP $x[n-3], h_3, \text{product3}$
10	LDW $x[n-9]$ LDW h_9 MPYSP $x[n-4], h_4, \text{product4}$ ADDSP product0, accumulator0, accumulator0
11	LDW $x[n-10]$ LDW h_{10} MPYSP $x[n-5], h_5, \text{product5}$ ADDSP product1, accumulator1, accumulator1
12	LDW $x[n-11]$ LDW h_{11} MPYSP $x[n-6], h_6, \text{product6}$ ADDSP product2, accumulator2, accumulator2
13	LDW $x[n-12]$ LDW h_{12} MPYSP $x[n-7], h_7, \text{product7}$ ADDSP product3, accumulator3, accumulator3
14	LDW $x[n-13]$ LDW h_{13} MPYSP $x[n-8], h_8, \text{product8}$ ADDSP product4, accumulator0, accumulator0
15	...

The total number of execute cycles to compute a tapped delay line of N coefficients is the delay line length (N) + LDW throughput (1) + LDW delay (4) + MPYSP throughput (1) + MPYSP delay (3) + ADDSP throughput (1) + ADDSP delay (3) + adding four accumulators together (8) + STW throughput (1) + STW delay (4) = $N + 26$ cycles. If we were to include two instructions to set up the modulo addressing for the circular buffer, then the total number of execute cycles would be $N + 28$ cycles.

EE 3455 Real-Time Digital Signal Processing Lab

Prof. Brian L. Evans

All-Pass Filters

- An all-pass filter has a magnitude response that is constant for all frequencies. The phase response may or may not be linear.
- A simple all-pass filter is the gain, i.e. $y(t) = g x(t)$ or $y[n] = g x[n]$ where x is the input and y is the output. Impulse response is $h(t) = g \delta(t)$ or $h[n] = g \delta[n]$. The frequency response is simply equal to g .
- Another simple all-pass filter is the ideal delay, i.e. $y(t) = x(t - t_0)$ or $y[n] = x[n - n_0]$ where t_0 and n_0 are constants. Impulse response is $h(t) = \delta(t - t_0)$ or $h[n] = \delta[n - n_0]$. The frequency response is $H_{\text{freq}}(f) = e^{-j2\pi f t_0}$ or $H_{\text{freq}}(\omega) = e^{-j\omega n_0}$. The magnitude response is equal to one in either case. Phase response is linear.
- A cascade of a gain and an ideal delay also has an all-pass response.
- A first-order IIR filter with one real-valued pole and one real-valued zero is all-pass if the zero location is equal to the reciprocal of the pole location:

$$H(z) = \frac{z - \frac{1}{r}}{z - r} \longrightarrow H_{\text{freq}}(\omega) = \frac{e^{j\omega} - \frac{1}{r}}{e^{j\omega} - r}$$

assuming that $|r| < 1$ for asymptotic stability. Magnitude response is

$$|H_{\text{freq}}(\omega)| = \left| \frac{e^{j\omega} - \frac{1}{r}}{e^{j\omega} - r} \right| = \frac{|e^{j\omega} - \frac{1}{r}|}{|e^{j\omega} - r|}$$

$$\begin{aligned} \text{Here, } |e^{j\omega} - a| &= |(\cos \omega - a) + j \sin \omega| = \sqrt{(\cos \omega - a)^2 + \sin^2 \omega} \\ &= \sqrt{\cos^2 \omega - 2a \cos \omega + a^2 + \sin^2 \omega} = \sqrt{a^2 - 2a \cos \omega + 1} \end{aligned}$$

$$|H_{\text{freq}}(\omega)| = \frac{\sqrt{\frac{1}{r^2} - \frac{2}{r} \cos \omega + 1}}{\sqrt{r^2 - 2r \cos \omega + 1}} = \frac{\sqrt{\frac{1}{r^2} (r^2 - 2r \cos \omega + 1)}}{\sqrt{r^2 - 2r \cos \omega + 1}} = \frac{1}{|r|}$$

- A first-order IIR filter with one complex-valued pole and one complex-valued zero is all-pass if the zero radius is the reciprocal to the pole radius and if the angles are the same:

$$H(z) = \frac{z - \frac{1}{r_0} e^{j\omega_0}}{z - r_0 e^{j\omega_0}} \Rightarrow H_{\text{freq}}(\omega) = \frac{e^{j\omega} - \frac{1}{r_0} e^{j\omega_0}}{e^{j\omega} - r_0 e^{j\omega_0}}$$

assuming that $r_0 < 1$ for asymptotic stability. The magnitude response is

$$|H_{\text{freq}}(\omega)| = \frac{|e^{j\omega} - \frac{1}{r_0} e^{j\omega_0}|}{|e^{j\omega} - r_0 e^{j\omega_0}|}$$

$$\begin{aligned} \text{Here, } |e^{j\omega} - (a + jb)| &= |(\cos\omega - a) + j(\sin\omega - b)| \\ &= \sqrt{(\cos\omega - a)^2 + (\sin\omega - b)^2} \\ &= \sqrt{\cos^2\omega - 2a\cos\omega + a^2 + \sin^2\omega - 2b\sin\omega + b^2} \\ &= \sqrt{(a^2 + b^2) - 2\sqrt{a^2 + b^2} \cos(\omega + \theta) + 1} \end{aligned}$$

where $\theta = \arctan\left(-\frac{b}{a}\right)$.

$$|H_{\text{freq}}(\omega)| = \frac{\sqrt{\frac{1}{r_0^2} - \frac{2}{r_0} \cos(\omega + \theta) + 1}}{\sqrt{r_0^2 - 2r_0 \cos(\omega + \phi) + 1}}$$

where $\theta = \arctan\left(-\frac{\frac{1}{r_0} \sin\omega_0}{\frac{1}{r_0} \cos\omega_0}\right) = -\omega_0$

$\phi = \arctan\left(-\frac{r_0 \sin\omega_0}{r_0 \cos\omega_0}\right) = -\omega_0$

Therefore,

$$\begin{aligned} |H_{\text{freq}}(\omega)| &= \frac{\sqrt{\frac{1}{r_0^2} - \frac{2}{r_0} \cos(\omega - \omega_0) + 1}}{\sqrt{r_0^2 - 2r_0 \cos(\omega - \omega_0) + 1}} \\ &= \frac{\sqrt{\frac{1}{r_0^2} (r_0^2 - 2r_0 \cos(\omega - \omega_0) + 1)}}{\sqrt{r_0^2 - 2r_0 \cos(\omega - \omega_0) + 1}} = \frac{1}{r_0} \end{aligned}$$

Handout P: Communication Performance of PAM vs. QAM Handout

Prof. Brian L. Evans

In the transmitter,

- Assume the bit stream on the transmitter side 0's and 1's appear with equal probability.
- Assume that the symbol period T is equal to 1.

In the channel,

- Assume that the noise is additive white Gaussian noise with zero mean. For QAM, the variance is σ^2 in each of the in-phase and quadrature components. For PAM, the variance is $2\sigma^2$. The difference is the variance is to keep the total noise power the same in QAM and PAM.
- Assume that there is no nonlinear distortion
- Assume there is no linear distortion

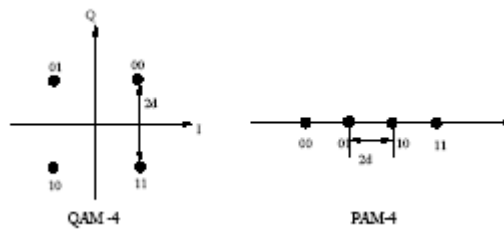
In the receiver,

- Assume that all subsystems (e.g. automatic gain control and symbol timing recovery) prior to matched filtering and sampling at the symbol rate are working perfectly
- Hence, assume that reception is synchronized with transmission

Given these mostly ideal conditions, the lower bound on symbol error probability for 4-PAM when the additive white Gaussian noise in the channel has variance $2\sigma^2$ is

$$P_e = \frac{3}{2} Q\left(\frac{d}{\sqrt{2}\sigma}\right)$$

Given the 4-QAM and 4-PAM constellations below,



- Derive the symbol error probability formula for 4-QAM, also known as Quadrature Phase Shift Keying (QPSK), shown in Figure 1.
- Calculate the average power of the QPSK signal given d .
- Write the probability of symbol error for 4-PAM and 4-QAM as functions of the signal-to-noise ratio (SNR). Superimposed on the same plot, plot the probability of symbol error for 4-PAM and 4-QAM as a function of SNR. For the horizontal axis, let the SNR take on values from 0 dB to 20 dB. Comment on the differences in the symbol error rate vs. SNR curves.
- Are the bit assignments for the PAM or QAM optimal with respect to bit error rate in Figure 1? If not, then please suggest another bit assignment to achieve a lower bit error rate given the same scenario, i.e., the same SNR. The optimal bit assignment (in terms of bit error probability) is commonly referred to as Gray coding.

- a) Based on lecture notes on slides 15-13 through 15-15, the case of 4-QAM corresponds to having the four corner points in the 16-QAM constellation. So, the probability of correct detection is given by **type 3 correct detection** given on page 15-4 in the lecture notes. Since $T=1$, then the formula for the probability of correct detection is given

$$\text{by } P_3(c) = \left(1 - Q\left(\frac{d}{\sigma}\right)\right)^2. \text{ Thus the probability of error is given}$$

$$\text{by } P_e = 1 - P_3(c) = 1 - \left(1 - Q\left(\frac{d}{\sigma}\right)\right)^2 = 2Q\left(\frac{d}{\sigma}\right) - Q^2\left(\frac{d}{\sigma}\right).$$

- b) To obtain the energy of s_i , we notice that the sum of the squared coordinates will give you the energy of the signal s_i . To see this, notice that s_i is represented by the following vector

$$\left(\sqrt{E} \cos\left[(2i-1)\frac{\pi}{4}\right], \sqrt{E} \sin\left[(2i-1)\frac{\pi}{4}\right]\right) \text{ in the } (\phi_1(t) - \phi_2(t)) \text{ coordinate system. Thus, it is}$$

immediate that $E\left(\cos^2\left[(2i-1)\frac{\pi}{4}\right] + \sin^2\left[(2i-1)\frac{\pi}{4}\right]\right) = E$. This implies that $P = \frac{E}{T}; T=1 \Rightarrow P = E$.

$$P_{AVG} = \frac{1}{4}(4 \times 2d^2) = 2d^2.$$

- c) SNR is defined as $SNR = \frac{P_{Signal}}{P_{Noise}} = \frac{E/T}{2\sigma^2} = \frac{E}{2\sigma^2} = \frac{2d^2}{2\sigma^2} = \frac{d^2}{\sigma^2}$ for the 4-QAM. For the 4-

$$\text{PAM, } SNR = \frac{P_{Signal}}{P_{Noise}} = \frac{E/T}{2\sigma^2} = \frac{E}{2\sigma^2} = \frac{\frac{1}{4}(2 + 2 \times 9)d^2}{2\sigma^2} = \frac{5d^2}{2\sigma^2}.$$

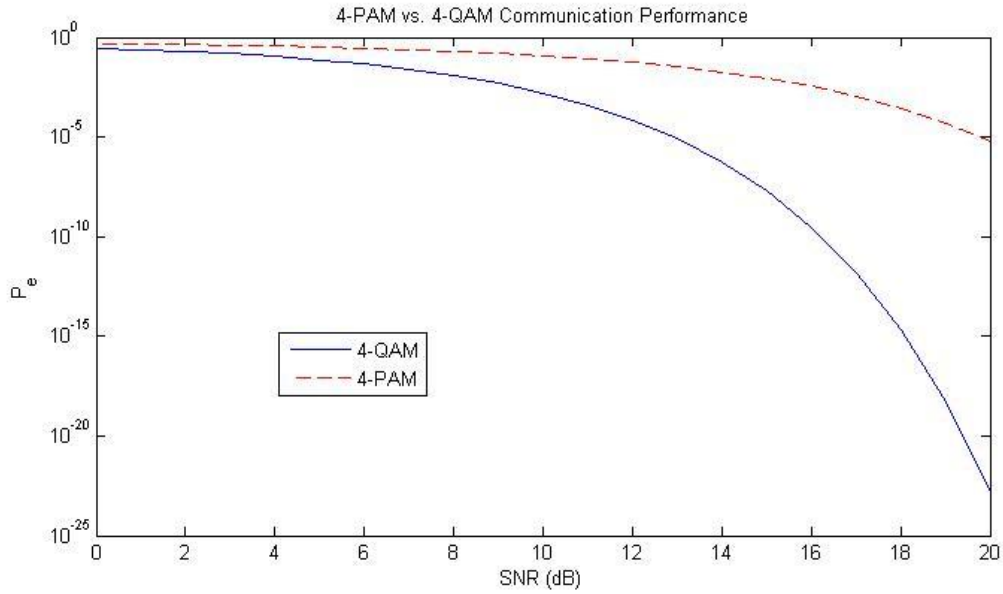
Substituting this into the P_e

formula we obtain the following formulas:

$$P_{e-QAM} = 2Q\left(\frac{d}{\sigma}\right) - Q^2\left(\frac{d}{\sigma}\right) = 2Q(\sqrt{SNR}) - Q^2(\sqrt{SNR})$$

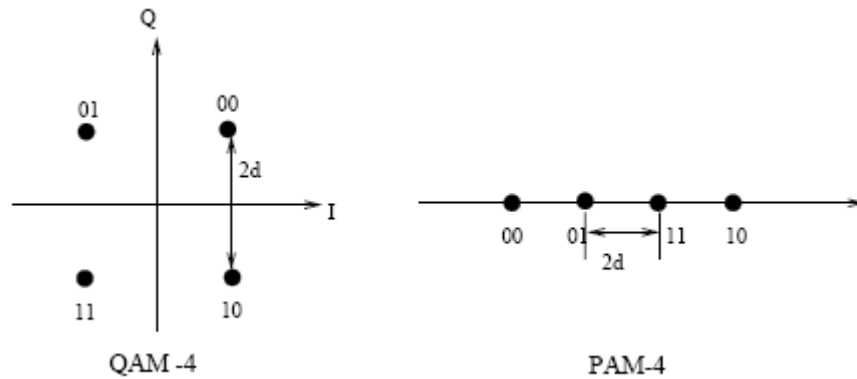
$$P_{e-PAM} = \frac{3}{2}Q\left(\sqrt{\frac{SNR}{5}}\right)$$

```
SNR = 0:20; % dB scale SNR
SNR_lin = 10.^(SNR/10); % linear scale SNR
Pq = 2*qfunc(sqrt(SNR_lin)) - (qfunc(sqrt(SNR_lin)).)^2; % QAM error
Probability
Pp = 3/2 * qfunc(sqrt(SNR_lin/5)); % PAM error Probability
semilogy(SNR, Pq, 'Displayname', '4-QAM');
hold on;
semilogy(SNR, Pp, 'r', 'Displayname', '4-PAM');
title('4-PAM vs. 4-QAM Communication Performance');
ylabel('P_e'); xlabel('SNR (dB)');
legend('show');
```



QAM performs much better than the PAM system due to the following reasons: first the noise variance in the PAM system is higher so we expect its error rate to be higher; on the other hand the PAM system is not fully utilizing the bandwidth as opposed to QAM.

d) The bit assignments are not optimal because the difference between the bits across the decision regions are more than one bit while they can be made one by using Gray Coding since each decision region has only two neighbors. The following bit assignment is optimal.



Handout Q: Four Ways to Filter a Signal

Problem: Evaluate four ways to filter an input signal. Run `waystofilt.m` on page 143 (Section 7.2.1) of Johnson, Sethares & Klein using

- $h[n]$ that is a four-symbol raised cosine pulse with $\beta = 0.75$ (4 samples/symbol, i.e. 16 samples)
- $x[n]$ that is an upsampled 8-PAM symbol amplitude signal with $d = 1$ and 4 samples/symbol and that is defined as the following 32-length vector (where each number is a sample value) as

$$\mathbf{x} = [-7000 \quad -5000 \quad -3000 \quad -1000 \quad 1000 \quad 3000 \quad 5000 \quad 7000]$$

In the code provided by Johnson, Sethares & Klein, please replace `plot` with `stem` so that the discrete-time signals are plotted in discrete time instead of continuous time.

Please comment on the different outputs. Please state whether each method implements linear convolution or circular convolution or something else. Please see the online homework hints.

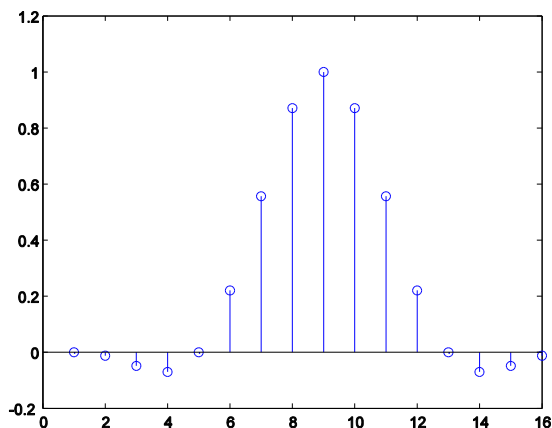
Hints: To compute the values of h , please use the "rcosine" command in Matlab and not the "SRRC" command. The length of h should be 16. The syntax of the "rcosine" command is

```
rcosine(Fd, Fs, TYPE_FLAG, beta)
```

The ratio F_s/F_d must be a positive integer. Since the the number of samples per symbol is 4, F_s/F_d must be 4. The `rcosine` function is defined in the Matlab communications toolbox.

Running the `rcosine` function with these parameters gives a pulse shape of 25 samples. We want to keep four symbol periods of the pulse shape. That is, we want to keep two symbol periods to the left of the maximum value, the symbol period containing the maximum value as the first sample, and the symbol period immediately following that:

```
rcosinelen25 = rcosine(1, 4, 'fir', 0.75);
h = rcosinelen25(5:20);
stem(h)
```



Some of the methods yield linear convolution, and some do not. With an input signal of 32 samples in length and a pulse shaping filter with an impulse response of 16 samples in length, linear convolution would produce a result that is 47 samples in length (i.e., $32 + 16 - 1$).

For the FFT-based method, the length of the FFT determines the length of the filtered result. An FFT length of less than 47 would yield circular convolution, but it wouldn't be linear convolution. When the FFT length is long enough, the answer computed by circular convolution is the same as by linear convolution.

Consider when the filter is a block in a block diagram, as would be found in Simulink or LabVIEW. When executing, the filter block would take in one sample from the input and produce one sample on the output. How many times to execute the block? As many times as there are samples on the input. How many samples would be produced? As many times as the block would be executed.

In particular, pay attention to the use of the FFT to implement linear filtering. A similar trick is used in multicarrier communication systems, such as DSL, WiFi (IEEE 802.11a/g), WiMax (IEEE 802.16e-2005), next-generation cellular data transmission (LTE), terrestrial digital audio broadcast, and handheld and terrestrial digital video broadcast.

Solution: The filter is given by its impulse response $h[n]$ that has a length of L_h samples. The signal is given by $x[n]$ and it has a length of L_x samples. Both the impulse response and input signal are causal. In this problem, L_h is 16 samples and L_x is 32 samples.

The ***first way*** of filtering computes the output signal as the linear convolution of $x[n]$ and $h[n]$:

$$y_{linear}[n] = x[n] * h[n] = \sum_{m=0}^{L_h-1} h[m] x[n-m]$$

Linear convolution yields a signal of length $L_x + L_h - 1 = 47$ samples.

The ***second way*** is to use the filter command in Matlab/Mathscript. The filter command produces one output sample for each input sample. This is a common behavior for a filter block in a block diagram simulation framework, e.g. Simulink or LabVIEW. When executing, the filter block would take in one sample from the input and produce one sample on the output. The scheduler will execute the block as many times as there are samples on the input. So, the length of the filtered signal would be $L_x = 32$ samples. To obtain an output of length of $L_x + L_h - 1$ samples, one would append $L_h - 1$ zeros to $x[n]$.

The ***third way*** is compute the output by using a Fourier-domain approach. For linear convolution, the discrete-time Fourier transform of the linear convolution of $x[n]$ and $h[n]$ is simply the product of their individual discrete-time Fourier transforms. The product could then be inverse transformed to find the filtered signal in the discrete-time domain. That approach, however, is difficult to automate using only numeric calculations. An alternative is to use the Fast Fourier Transform (FFT).

The FFT of the *circular convolution* of $x[n]$ and $h[n]$ is the product of their individual FFTs. In circular convolution, the signals $x[n]$ and $h[n]$ are considered to be periodic with period N . One period of N samples of the circular convolution is defined as

$$y_{\text{circular}}[n] = x[n] \otimes_N h[n] = \sum_{m=0}^{N-1} h[(m)_N] x[(n-m)_N]$$

where $((\bullet))_N$ means that the argument is taken modulo N . We will henceforth refer to the circular convolution between periodic signals of length N as circular convolution of length N . On a programmable digital signal processor, we would use the modulo addressing mode to accelerate the computation of circular convolution.

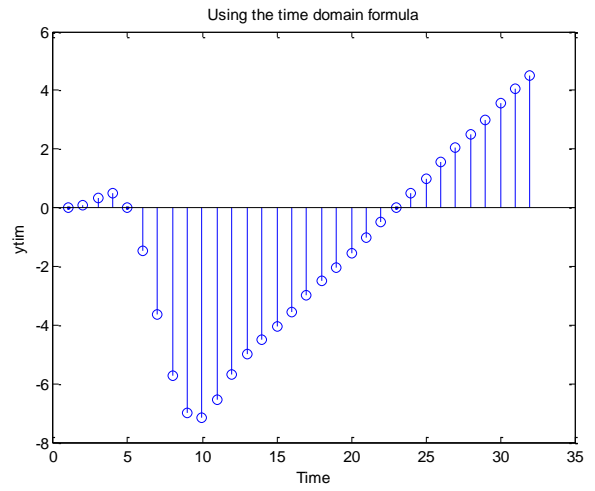
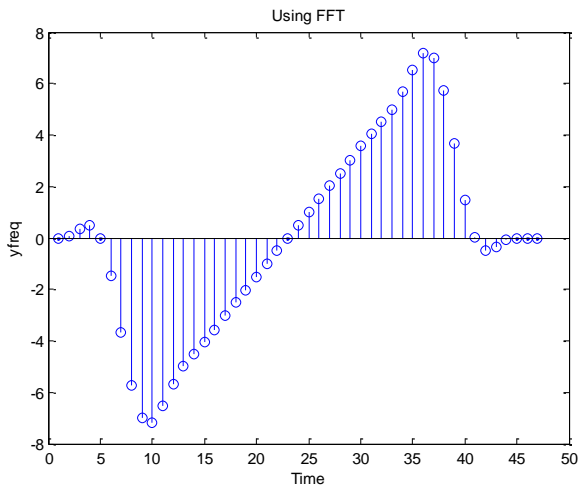
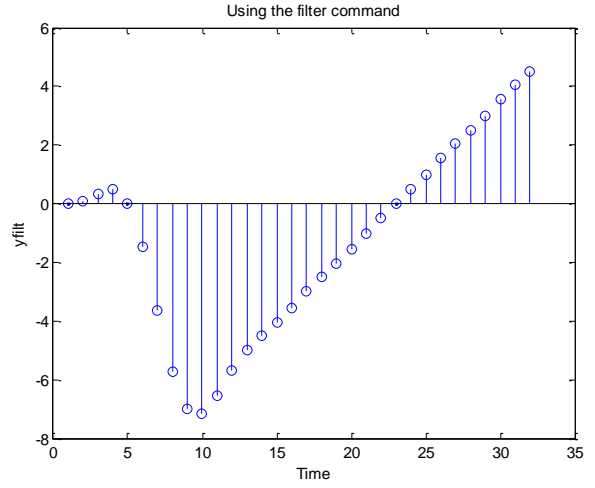
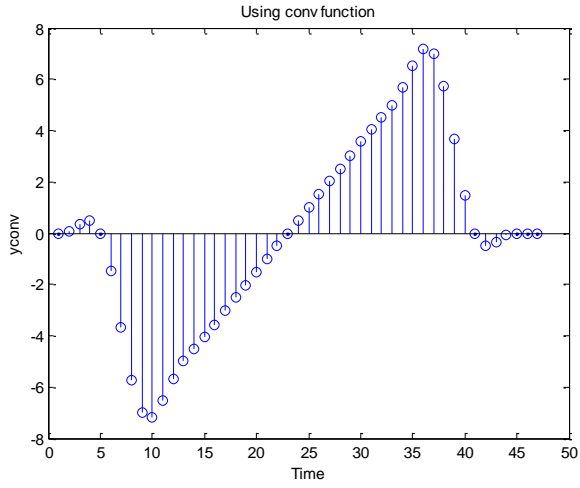
Circular convolution of two finite-length sequences $x[n]$ and $h[n]$ is equivalent to linear convolution of those sequences by padding (appending) L_x-1 zeros to $h[n]$ and L_h-1 zeros to $x[n]$ so that both of them are of the same length and using a circular convolution length of L_x+L_h-1 samples. This is the approach used in the FFT-based method in this problem.

The FFT-based method to compute the linear convolution uses an FFT length of N of L_x+L_h-1 . First, the FFT of length N of the zero-padded $x[n]$ is computed to give $X[k]$, and the FFT of length N of the zero-padded $h[n]$ is computed to give $H[k]$. Second, the product $Y_{\text{circular}}[k] = X[k]H[k]$ for $k = 0, \dots, N-1$ is computed. Then, the inverse FFT of length N of $Y_{\text{circular}}[k]$ is computed to find $y_{\text{circular}}[n]$. This *third way* results in an output signal of $L_x+L_h-1 = 47$ samples.

The *fourth way* to filter a signal uses a time-domain formula. It is an alternate implementation of the same approach used by the filter command. Hence, this approach gives an output of length $L_x = 32$ samples.

```
% waystofilt.m "conv" vs. "filter" vs. "freq domain" vs. "time domain"
over=4; % 4 samples/symbol
r=0.75; % roll-off
rcosinelen25 = rcosine(1, 4, 'fir', 0.75);
h = rcosinelen25(5:20);
x = [-7 0 0 0 -5 0 0 0 -3 0 0 0 -1 0 0 0 1 0 0 0 3 0 0 0 5 0 0 0 7 0 0 0];
yconv=conv(h,x) ; % (a) convolve x[n] * h[n]
n=1:length(yconv);stem(n,yconv)
xlabel('Time');ylabel('yconv');title('Using conv function');figure
yfilt=filter(h,1,x) ; % (b) filter x[n] with h[n]
n=1:length(yfilt);stem(n,yfilt)
xlabel('Time');ylabel('yfilt');title('Using the filter command');figure
N=length(h)+length(x)-1; % pad length for FFT
ffth=fft([h zeros(1,N-length(h))]); % FFT of impulse response = H[k]
fftx=fft([x, zeros(1,N-length(x))]); % FFT of input = X[k]
ffty=ffth .* fftx; % product of H[k] and X[k]
yfreq=real(ifft(ffty)); % (c) IFFT of product gives y[n]
% it's complex due to roundoff

n=1:length(yfreq); stem(n,yfreq)
xlabel('Time');ylabel('yfreq');title('Using FFT');figure
z=zeros(1,length(h)-1,x); % initial state in filter = 0
for k=1:length(x) % (d) time domain method
    ytim(k)=fliplr(h)*z(k:k+length(h)-1)'; % iterates once for each x[k]
end % to directly calculate y[k]
n=1:length(ytim); stem(n,ytim)
xlabel('Time');ylabel('ytim');title('Using the time domain formula');
%end of function
```



Quick Introduction to the Fourier Transform

by Prof. Brian Evans
and Mr. Jean Faria's
The University of Texas
at Austin Spring 2013

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

Fourier Transform / $\frac{\text{radian}}{\text{Hz}}$ / $\frac{\text{seconds}}{\text{seconds}}$
 real variable f Amplitude real or complex Real variable t

Amplitude is usually complex

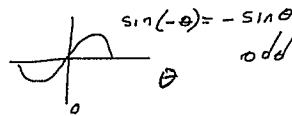
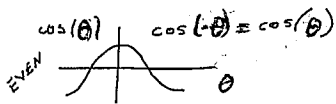
real imag

$$e^{-j2\pi ft} = \cos(-2\pi ft) + j \sin(-2\pi ft)$$

complex-valued sinusoid = $\cos(2\pi ft) - j \sin(2\pi ft)$

$$x(t) \xrightarrow{\mathcal{F}} X(f)$$

$$x_1(t) + x_2(t) \xrightarrow{\mathcal{F}} X_1(f) + X_2(f)$$



$$\bar{X}(s) = \int \{ x(t) \}$$

Laplace Transform $\bar{X}(s) = \int_{-\infty}^{\infty} x(t) e^{-st} dt$

region of convergence (ROC)

If the ROC includes the imaginary axis

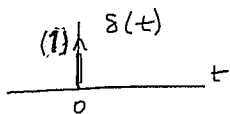
$$x(t) = \bar{X}(s) \Big|_{s=j2\pi f}$$

Example

Dirac delta functional — Continuous-Time Impulse

unit area

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad \text{Unit Area}$$



Mathematically models impulsive event at origin

Amplitude undefined at origin

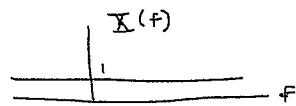
$$\int_{-\infty}^{\infty} g(t) \delta(t) dt = g(0)$$

Unit area concentrated at origin — area denoted by parentheses

sifting property

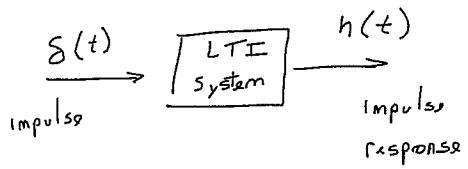
IF $g(s)$ is defined at origin

$$X(f) = \int_{-\infty}^{\infty} \delta(t) \underbrace{e^{-j2\pi ft}}_{g(t)} dt = 1 \quad \text{when } t=0$$



contains all frequencies

How to get direct impulse response

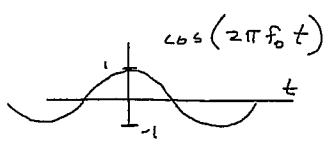


$$\delta(t) \xrightarrow{\mathcal{F}} 1$$

Example

two sided cosine

this does not have laplace
it have fourier transform
can't build this in lab



Euler's Formula

$$e^{j\theta} = \cos \theta + j \sin \theta$$

$$\cos(2\pi f_0 t) = \frac{1}{2} e^{-j2\pi f_0 t} + \frac{1}{2} e^{j2\pi f_0 t}$$

$$= \frac{1}{2} (\cos(-2\pi f_0 t) + j \sin(-2\pi f_0 t)) + \frac{1}{2} (\cos(2\pi f_0 t) + j \sin(2\pi f_0 t))$$

$$= \cos(2\pi f_0 t)$$

Inverse Fourier Transform

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$$

For $X(f) = \delta(f)$,

$$x(t) = \int_{-\infty}^{\infty} \delta(f) e^{j2\pi ft} df$$

$$= 1$$

$$1 \xrightarrow{\mathcal{F}} \delta(f)$$

$$\int_{-\infty}^{\infty} e^{-j2\pi f_0 t} e^{-j2\pi ft} dt$$

$$= \int_{-\infty}^{\infty} e^{-j2\pi(f+f_0)t} dt = \delta(f+f_0)$$



example continued on next page

$$X(f+f_0) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi(f+f_0)t} dt$$

$$= \int_{-\infty}^{\infty} x(t) e^{-j2\pi f_0 t} e^{-j2\pi f t} dt$$

$$x(t) e^{-j2\pi f_0 t} \xrightarrow{\mathcal{F}} X(f+f_0)$$

$$1 \xrightarrow{\mathcal{F}} \delta(f)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} df \quad \text{Inverse Fourier}$$

$$X(f) = \delta(f), \quad x(t) = \int_{-\infty}^{\infty} \delta(f) \underbrace{e^{j2\pi f t}}_{g(f)} df = g(0) = 1$$

always evaluate this at the origin in the above case is when $f=0$

$$x(t) e^{-j2\pi f_0 t} \xrightarrow{\mathcal{F}} X(f+f_0)$$

$$1 \xrightarrow{\mathcal{F}} \delta(f)$$

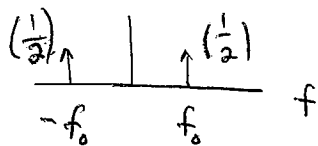
$$1 \cdot e^{-j2\pi f_0 t} \xrightarrow{\mathcal{F}} \delta(f+f_0)$$

$$e^{j2\pi f_0 t} \xrightarrow{\mathcal{F}} \delta(f-f_0)$$

$$\mathcal{F}\{\cos(2\pi f_0 t)\} = \mathcal{F}\left\{\frac{1}{2}e^{-j2\pi f_0 t} + \frac{1}{2}e^{j2\pi f_0 t}\right\}$$

$$= \frac{1}{2} \delta(f+f_0) + \frac{1}{2} \delta(f-f_0)$$

$$\underbrace{\hspace{1.5cm}}_{\text{area of } \frac{1}{2}} \quad \underbrace{\hspace{1.5cm}}_{\text{area of } \frac{1}{2}}$$

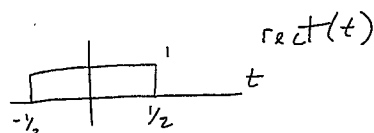


1st find the origin

$$f+f_0=0 \quad f-f_0=0$$

$$f=-f_0 \quad f=f_0$$

Example using the rectangular pulse



Has unit area.

Taking the Fourier transform,

$$\int_{-1/2}^{1/2} x(t) e^{-j2\pi ft} dt$$

$$\frac{1}{-j2\pi f} e^{-j2\pi ft} \Big|_{-1/2}^{1/2}$$

$$\frac{1}{-j2\pi f} e^{-j\pi f} + \frac{1}{j2\pi f} e^{j\pi f}$$

$$\frac{j}{j(-j)2\pi f} e^{-j\pi f} + \frac{j}{j(j)2\pi f} e^{j\pi f}$$

$$\frac{j}{2\pi f} e^{-j\pi f} - \frac{j}{2\pi f} e^{j\pi f}$$

$$\frac{j}{2\pi f} (e^{-j\pi f} - e^{j\pi f})$$

$$\frac{j}{2\pi f} (-2j \sin(\pi f)) = \frac{\sin(\pi f)}{\pi f} = \text{sinc}(f)$$

Adding Random Variables and Connections with the Signals and Systems Pre-requisite

Problem

A key connection between a Linear Systems and Signals course and a Probability course is that when two independent random variables are added together, the resulting random variable has a probability density function (pdf) that is the convolution of the pdfs of the random variables being added together. That is, if X and Y are independent random variables and $Z = X + Y$, then $f_Z(z) = f_X(z) * f_Y(z)$ where $f_R(r)$ is the probability density function for random variable R and $*$ is the convolution operation. This is true for continuous random variables and discrete random variables. (An alternative to a probability density function is a probability mass function. They represent the same information but in different formats.)

- a) Consider two fair six-sided dice. Each die, when rolled, generates a number in the range of 1 to 6, inclusive, with each outcome having an equal probability. That is, each outcome is uniformly distributed. When adding the outcomes of a roll of these two six-sided dice, one would have a number between 2 and 12, inclusive.
 - 1) Tabulate the likelihood for each outcome from 2 to 12, inclusive.
 - 2) Compute the pdf of Z by convolving the pdfs of X and Y . Compare the result to the first part of this sub-problem (a)-(1).

- b) Compute the pdf of continuous random variable Z where $Z = X + Y$ and X is a continuous random variable uniformly distributed on $[0, 2]$ and Y is a continuous random variable uniformly distributed on $[0, 4]$. Assume that X and Y are independent.

- c) A constant value C can be modeled as a pdf with only one non-zero entry. Recall that the pdf can only contain non-negative values and that the area under a continuous pdf (or equivalently the sum of a discrete pdf) must be 1.
 - 1) Plot the pdf of a discrete random variable X that is a constant of value C .
 - 2) Plot the pdf of a continuous random variable Y that is a constant of value C .
 - 3) Using convolution, determine the pdf of a continuous random variable Z where $Z = X + Y$. Here, X has a uniform distribution on $[0, 3]$ and Y is a constant of value 2. Assume that X and Y are independent.

Solution

(a) (1) Likelihood for each outcome from 2 to 12

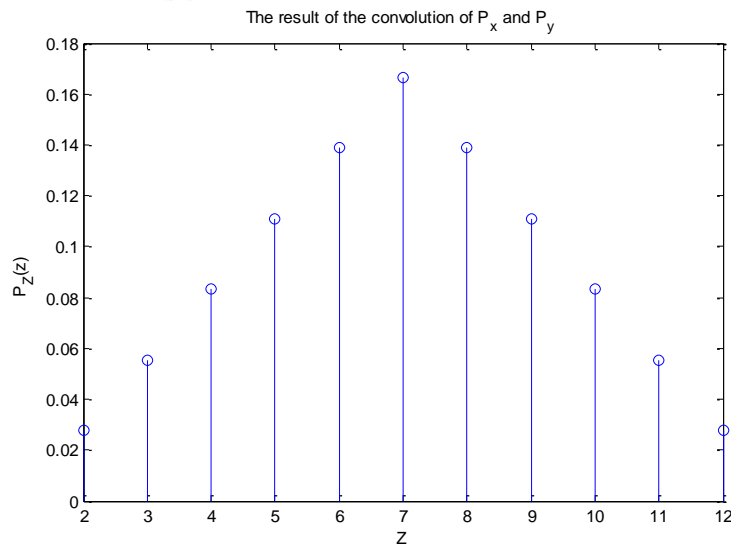
Let X be the number generated when the first die is rolled and Y be the number generated when the second die is rolled. Since each outcome is uniformly distributed for each die, $P(X = x) = 1/6$ where $x \in \{1, 2, 3, 4, 5, 6\}$ and $P(Y = y) = 1/6$ where $y \in \{1, 2, 3, 4, 5, 6\}$:

Z	$P(z)$
2	1/36
3	2/36

4	3/36
5	4/36
6	5/36
7	6/36
8	5/36
9	4/36
10	3/36
11	2/36
12	1/36

(2) Adding the two random variables results in another random variable $Z = X + Y$ which takes on values between 2 and 12, inclusive. Since the dice are rolled independently, the numbers generated are independent.

$$p_z(z) = p_{X+Y}(z) = p_x(z) * p_y(z) = \sum_{k=1}^6 p_x(k)p_y(z-k).$$



The convolution of two rectangular pulses of the same length N samples gives a triangular pulse of length $2N - 1$ samples. Example calculations:

$$p_z(2) = p_x(1)p_y(1) = \frac{1}{36}$$

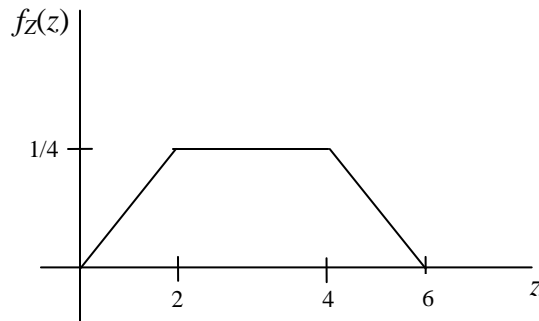
$$p_z(3) = p_x(1)p_y(2) + p_x(2)p_y(1) = \frac{2}{36}$$

$$p_z(4) = p_x(1)p_y(3) + p_x(2)p_y(2) + p_x(3)p_y(1) = \frac{3}{36}$$

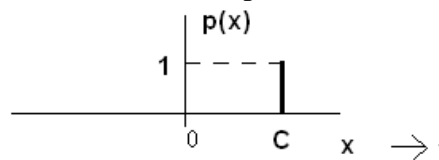
Evaluating the above convolution, we get the same pdf as obtained in the table. The output of the Matlab simulation of the convolution is displayed in the above graph. The *conv* method was used for the convolution. The *stem* method was used for plotting.

(b) X is uniformly distributed on $[0, 2]$. Therefore $f_x(x) = \frac{1}{2}$ for all $x \in [0, 2]$. Similarly, since Y is uniformly distributed on $[0, 4]$, $f_Y(y) = \frac{1}{4}$ for all $y \in [0, 4]$.

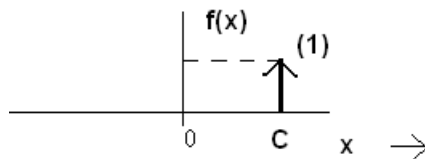
$$f_{X+Y}(z) = f_X(z) * f_Y(z) = \begin{cases} \int_0^z f_X(\lambda) - f_Y(z-\lambda) d\lambda = \frac{z}{8} & 0 \leq z \leq 2 \\ \int_0^2 f_X(\lambda) - f_Y(z-\lambda) d\lambda = \frac{1}{4} & 2 \leq z \leq 4 \\ \int_{z-4}^2 f_X(\lambda) - f_Y(z-\lambda) d\lambda = \frac{6}{8} - \frac{z}{8} & 4 \leq z \leq 6 \end{cases}$$



(c) (1) The answer is a Kronecker (discrete-time) impulse located at $x = C$.



(2) For a continuous random variable we require that $\int_{-\infty}^{\infty} f_X(x) dx = 1$ and this is satisfied by an continuous impulse (Dirac delta functional) at C . Mathematically, $\int_{-\infty}^{\infty} \delta(x - C) dx = 1$



(3) X is uniformly distributed on $[0, 3]$. Therefore $f_X(x) = \frac{1}{3}$ for all $x \in [0, 3]$. Y has a constant value of 2 and hence $f_Y(y) = \delta(y - 2)$. Since X and Y are independent, $Z = X + Y$ implies that

$$f_{X+Y}(z) = f_X(z) * \delta(z - 2) = \begin{cases} \frac{1}{3} & 2 \leq z \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

This follows from the fact that convolution by $\delta(z - 2)$ shifts $f_X(z)$ by 2.

$$x(t) y(t) \xrightarrow{\mathcal{F}} \frac{1}{2\pi} X(\omega) * Y(\omega)$$

$$Z(\omega) = X(\omega) * Y(\omega) = \int_{-\infty}^{\infty} X(\Omega) Y(\omega - \Omega) d\Omega$$

$$z(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Z(\omega) e^{j\omega t} d\omega$$

$$z(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} X(\Omega) Y(\omega - \Omega) d\Omega \right] e^{j\omega t} d\omega$$

$$= \int_{-\infty}^{\infty} X(\Omega) \underbrace{\left[\frac{1}{2\pi} \int_{-\infty}^{\infty} Y(\omega - \Omega) e^{j\omega t} d\omega \right]}_{\mathcal{F}_\omega^{-1}\{Y(\omega - \Omega)\}} d\Omega$$

$$= \int_{-\infty}^{\infty} X(\Omega) e^{j\Omega t} y(t) d\Omega$$

$$= 2\pi \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega) e^{j\Omega t} d\Omega \right) y(t)$$

$$= 2\pi x(t) \cdot y(t)$$

Prof. Brian L. Evans Fall 2017
The University of Texas at Austin EE445S

Property of Time-Invariance (Shift-Invariance) for a System Under Observation

Prof. Brian L. Evans and Mr. Jaehong Moon

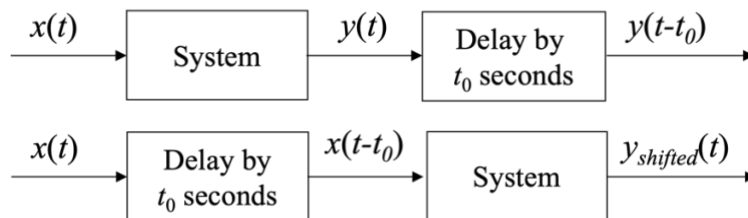
The University of Texas at Austin, bevans@ece.utexas.edu

September 20, 2018 and updated January 2, 2023, and January 23, 2023

When evaluating system properties, we treat a system as a closed box and analyze the relationships between input signals and their corresponding output signals. This process assumes that after inputting a signal, we can return the system to its original state.

A continuous-time system with input signal $x(t)$ and output signal $y(t)$ is time-invariant (shift-invariant) if whenever the input signal is delayed by t_0 seconds, then the output signal will always be delayed by t_0 seconds as well for all real values of t_0 .

A way to visualize the time-invariance property is to show the equivalence between



That is, does $y_{shifted}(t) = y(t - t_0)$ for all possible real constant values of t_0 ?

One-sided infinite observation. Let's consider the system under observation for $t \geq 0^-$. Time 0^- means a time of 0 seconds before occurrence of a Dirac delta occurring at the origin. We can only observe $x(t)$ for $t \geq 0^-$ and $y(t)$ for $t \geq 0^-$. This means that we can only observe $x(t - t_0)$ for $t \geq t_0$ and $y(t - t_0)$ for $t \geq t_0$.

Example. Consider a delay system that delays the input by T seconds, and we can only observe the input signal and the output signal for $t \geq 0^-$.

Conceptually, the delay block can be thought as a long wire that conducts electricity from the input to the output. Assuming electrons travel at $2/3$ the speed of light, the length of the wire would be $(2/3) c T$ where c is the speed of light (3×10^8 m/s). Such an implementation would be impractical, but nonetheless helpful in analyzing the system.

The first observed output value $y(0)$ would be due to the initial conditions in the delay system. In fact, the first T seconds of the output would due solely to the initial conditions in the system. For input $x(t)$ and output $y(t)$, once the initial conditions have been output, $y(T) = x(0)$. That is, it takes T seconds for an input value (voltage) to arrive at the output.

The initial conditions for the delay system consist of the voltage values at different points in the wire at $t = 0$. Let's denote these voltage values $v(t)$ for $-T < t \leq 0$. That is, $v(0)$ will be first, and $v(-T)$ will be the last, value among the initial conditions to be output. The spatial location for the voltage $v(t)$ for $-T < t \leq 0$ is $(-2/3) c t$ meters from the output location.

Let $x(t) = 0$ for $0 \leq t < T$ and 1 for $t \geq T$. For input $x(t)$, the output is

$$y(t) = \begin{cases} v(-t) & \text{for } 0 \leq t < T \\ x(t-T) & \text{for } t \geq T \end{cases} = \begin{cases} v(-t) & \text{for } 0 \leq t < T \\ 0 & \text{for } T \leq t < 2T \\ 1 & \text{for } t \geq 2T \end{cases}$$

Let's keep the same initial conditions, i.e. $v(t)$ for $-T < t \leq 0$, and the same definition for signal $x(t)$. Now, we input $x(t - t_0)$ into the delay system

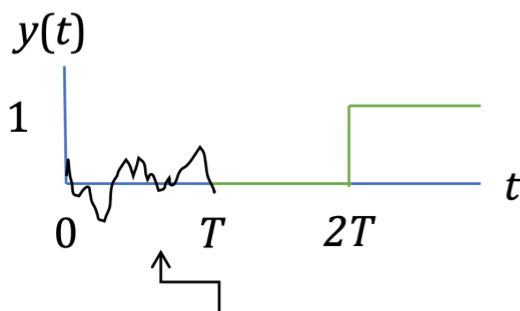
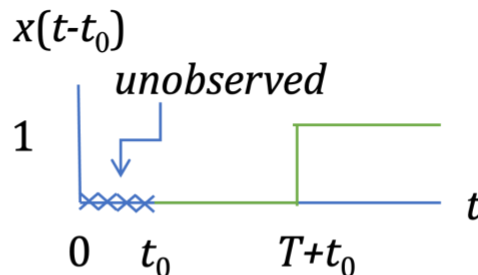
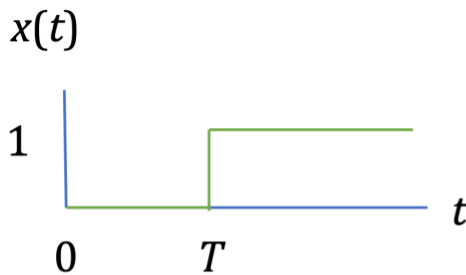
$$x(t - t_0) = \begin{cases} \text{unobserved} & \text{for } 0 \leq t < t_0 \\ 0 & \text{for } t_0 \leq t < T + t_0 \\ 1 & \text{for } t \geq T + t_0 \end{cases}$$

and the output is

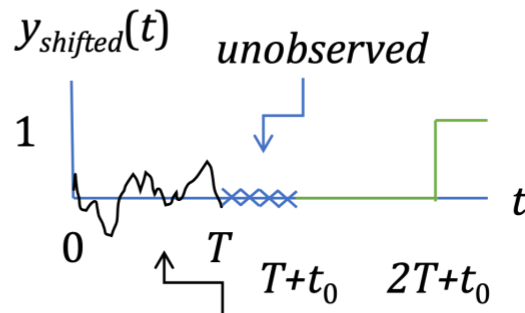
$$y_{\text{shifted}}(t) = \begin{cases} v(-t) & \text{for } 0 \leq t < T \\ \text{unobserved} & \text{for } T \leq t < T + t_0 \\ 0 & \text{for } T + t_0 \leq t < 2T + t_0 \\ 1 & \text{for } t \geq 2T + t_0 \end{cases} = \begin{cases} v(-t) & \text{for } 0 \leq t < T \\ \text{unobserved} & \text{for } T \leq t < T + t_0 \\ y(t - t_0) & \text{for } t \geq T + t_0 \end{cases}$$

$y_{\text{shifted}}(t)$ only equals $y(t - t_0)$ for $t \geq T + t_0$ because the initial conditions did not shift in time even though the input did.

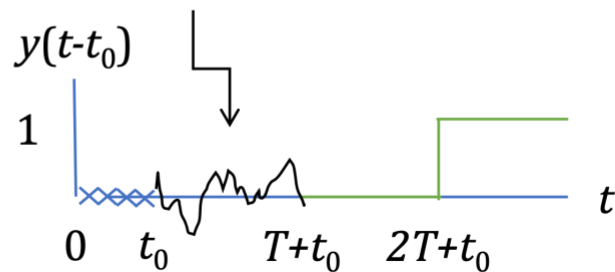
Plots of the signals are given next followed by an analysis of initial conditions:



Due to initial conditions



Due to initial conditions



If the system were time-invariant, then $y_{\text{shift}}(t) = y(t - t_0)$ for all real t_0 and $t \geq 0^-$. This holds for $t \geq T + t_0$. For $0 \leq t < T + t_0$, all unobserved values and initial conditions would have to be equal to a constant value.

LTI filters and frequency selectivity

Linear time-invariant (LTI) systems have many [useful properties](#). We can utilize these properties to design [frequency selective](#) filters.

Mathematical definitions

Additivity

Let $x_1(t)$ and $x_2(t)$ be arbitrary input signals to a system \mathcal{S} . The system satisfies additivity if

$$\mathcal{S}\{x_1(t) + x_2(t)\} = \mathcal{S}\{x_1(t)\} + \mathcal{S}\{x_2(t)\}$$

Homogeneity

Let $x(t)$ be an arbitrary input to a system \mathcal{S} . The system satisfies homogeneity if, for any constant a ,

$$\mathcal{S}\{ax(t)\} = a\mathcal{S}\{x(t)\}$$

Time-invariance

Let $x(t)$ be an arbitrary input to a system \mathcal{S} and let $y(t) = \mathcal{S}\{x(t)\}$ be the corresponding output. The system \mathcal{S} is time-invariant if, for any time shift τ ,

$$\mathcal{S}\{x(t - \tau)\} = y(t - \tau)$$

Linear time-invariant (LTI)

A system is linear time-invariant (LTI) if it satisfies the additivity, homogeneity, and time-invariance properties. A common way for a system to fail to violate these properties is if the system has nonzero initial conditions.

Properties of LTI systems

An LTI system is [uniquely characterized by its impulse response](#). The [Frequency response](#) of an LTI system is the Fourier transform of its impulse response.

Impulse response and convolution

Continuous time

A continuous time impulse, (also known as the Dirac delta) can be defined as a unit area pulse in the limit that its duration approaches zero.

$$\delta(t) = \lim_{\epsilon \rightarrow 0} \frac{\text{rect}(t/\epsilon)}{\epsilon}$$

If a system is LTI, then its impulse response $h(t) = \mathcal{S}\{\delta(t)\}$ uniquely characterizes the system. The output $y(t)$ of an LTI system is the convolution between the input $x(t)$ and the system's impulse response $h(t)$.

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(t - \tau)h(\tau)d\tau$$

Discrete time

A discrete time impulse, (also known as the Kronecker delta) can be defined as a piecewise function.

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

If a system is LTI, then its impulse response $h[n] = \mathcal{S}\{\delta[n]\}$ uniquely characterizes the system. The output $y[n]$ of an LTI system is the convolution between the input $x[n]$ and the system's impulse response $h[n]$.

$$y[n] = x[n] * h[n] = \sum_{-\infty}^{\infty} x[n - m]h[m]$$

Frequency response

Complex exponentials are eigenfunctions of LTI systems. Combined with the previous property, This allows us to uniquely characterize a system by its frequency response.

Eigenfunctions:

If application of the system \mathcal{S} to the signal $x(t)$ results in scaling only (i.e. $\mathcal{S}\{x(t)\} = \lambda x(t)$ for some constant λ) then we say that $x(t)$ is an eigenfunction of the system and λ is the corresponding eigenvalue.

Continuous time

If the input to an LTI system is a complex exponential $x(t) = e^{j\omega t}$, then the corresponding output is

$$y(t) = H(j\omega)e^{j\omega t}$$

where $H(j\omega)$ (called the frequency response) is the Fourier transform of the impulse response or, equivalently, the Laplace transform of the impulse response evaluated as $s = j\omega$.

$$H(j\omega) = \mathcal{F}\{h(t)\} = \mathcal{L}\{h(t)\}|_{s=j\omega}$$

Discrete time

If the input to an LTI system is a complex exponential $x[n] = e^{j\omega n}$, then the corresponding output is

$$y[n] = H(e^{j\omega})e^{j\omega n}$$

where $H(e^{j\omega})$ (called the frequency response) is the Discrete-time Fourier transform of the impulse response or, equivalently, the Z transform of the impulse response evaluated at $z = e^{j\omega}$.

$$H(e^{j\omega}) = \text{DTFT}\{h[n]\} = \mathcal{Z}\{h[n]\}|_{z=e^{j\omega}}$$

We often write $H(\omega)$ instead of $H(j\omega)$ or $H(e^{j\omega})$.

Magnitude and phase response

The frequency response is, in general, complex valued. Typically, we represent it in terms of its magnitude and phase.

$$\text{Magnitude response} = |H(\omega)| = \sqrt{\text{Re}\{H(\omega)\}^2 + \text{Im}\{H(\omega)\}^2}$$

$$\text{Phase response} = \angle H(\omega) = \text{atan2}(\text{Im}\{H(\omega)\}, \text{Re}\{H(\omega)\})$$

where `atan2` is the [two argument arctangent](#).

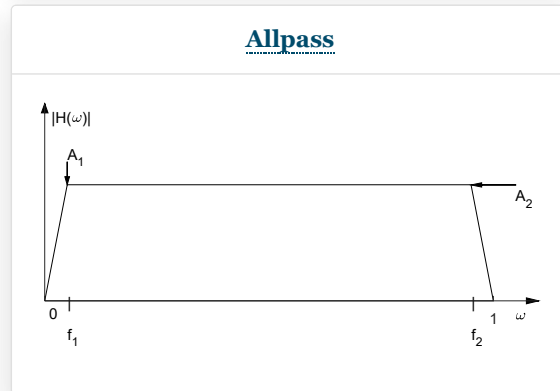
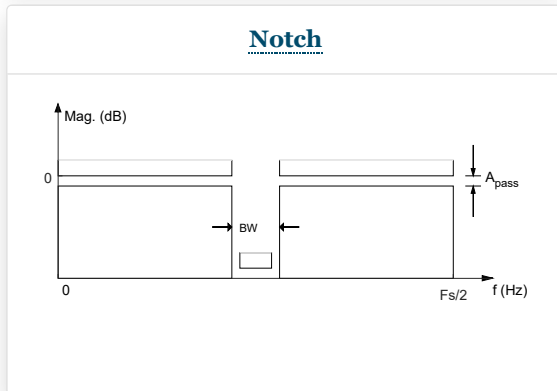
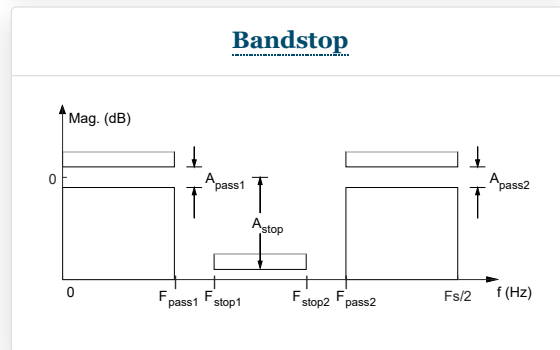
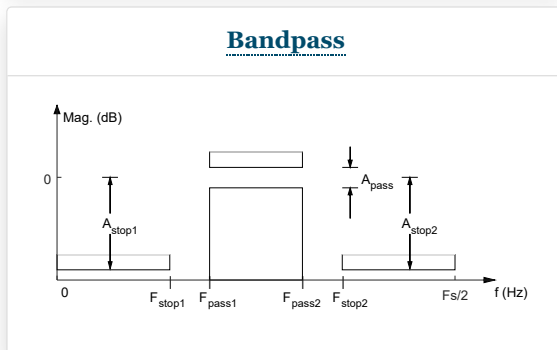
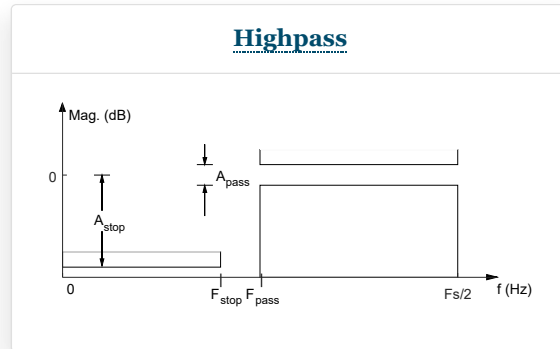
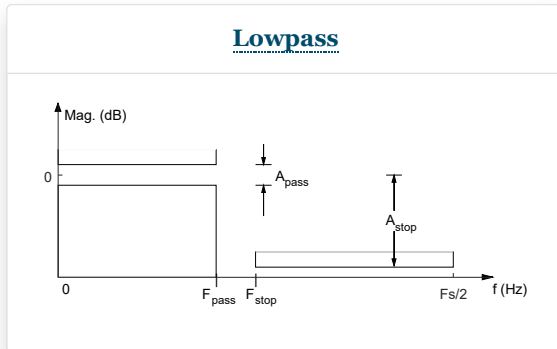
It is common to use the magnitude/phase representation when measuring and plotting the frequency response of a system. Typically, the magnitude response is expressed in decibels

$$\text{Magnitude response in decibels} = 10 \log_{10} |H(\omega)|^2 = 20 \log_{10} |H(\omega)|$$

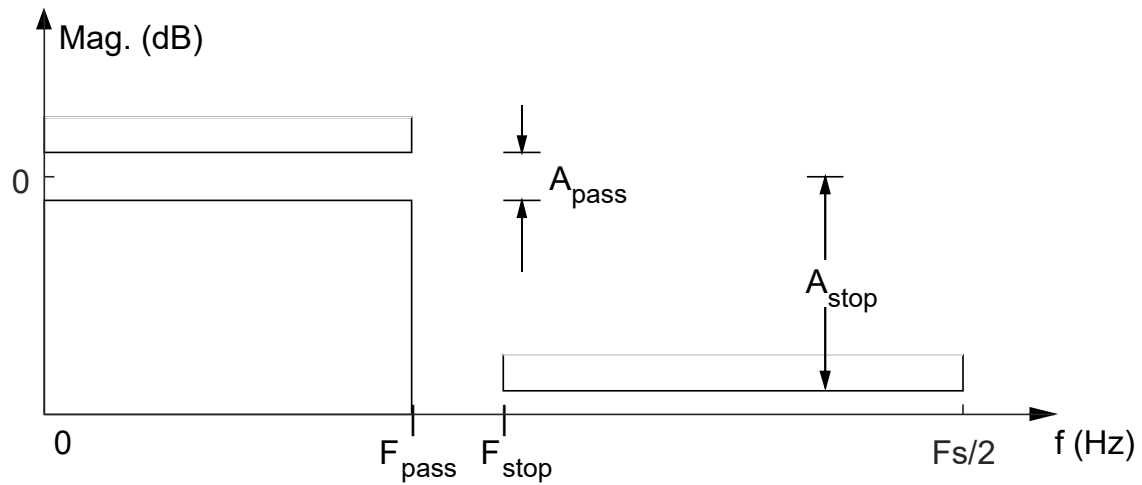
In MATLAB, the `freqz` function will calculate and plot the magnitude and phase response of a discrete-time LTI system.

Frequency selectivity

The goal of a filter is to suppress or attenuate some signal components while retaining or boosting others. We often group LTI filters into six categories based on their frequency selectivity, i.e. the arrangement of frequency bands that are boosted relative to the bands which are attenuated.



Lowpass



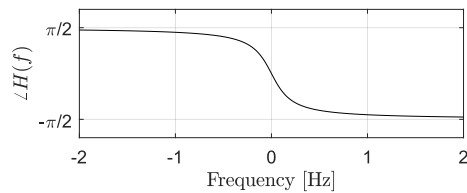
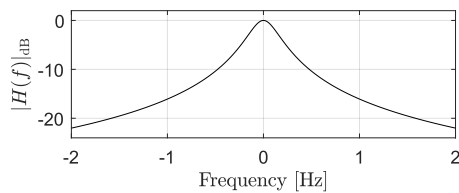
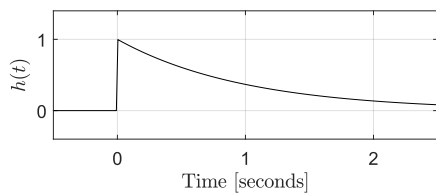
Continuous time lowpass example

Impulse response:

$$h(t) = e^{-t}u(t)$$

Frequency response:

$$H(f) = \frac{1}{1 + 2\pi jf}$$



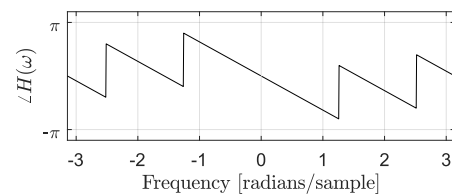
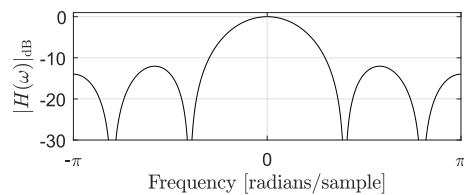
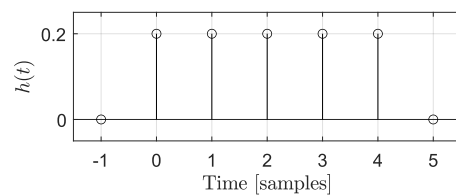
Discrete time moving average

Impulse response:

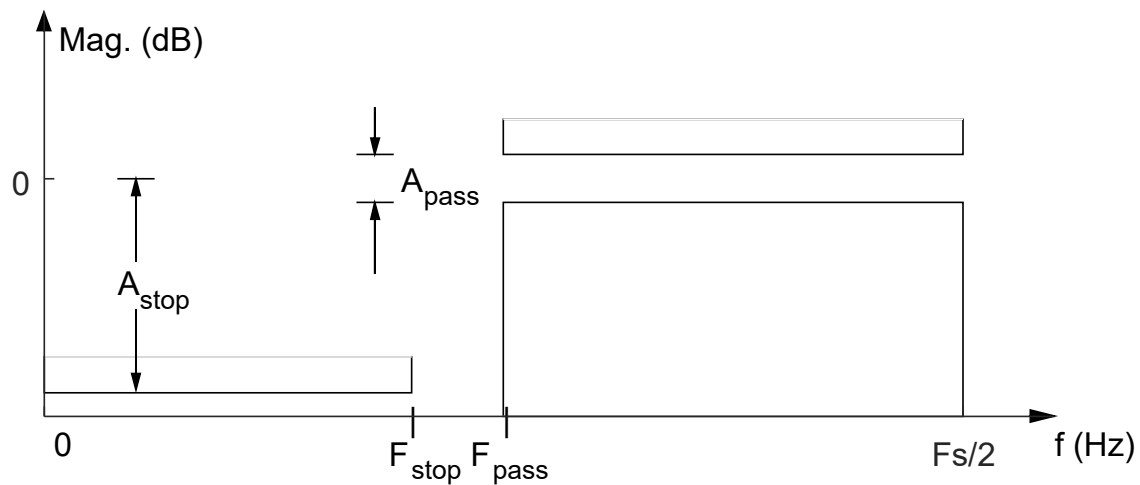
$$h[n] = \delta[n] + \delta[n-1] + \dots + \delta[n-5]$$

Frequency response:

$$H(\omega) = e^{-2j\omega} \frac{\sin(5\omega/2)}{5 \sin(\omega/2)}$$



Highpass



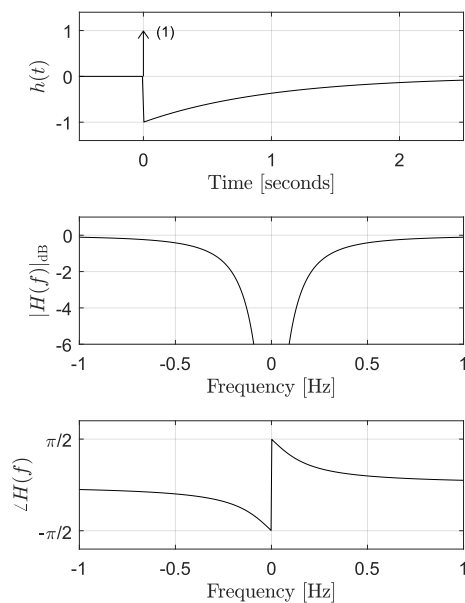
Continuous time highpass example

Impulse response:

$$h(t) = \delta(t) - e^{-t}u(t)$$

Frequency response:

$$H(f) = \frac{2\pi j f}{1 + 2\pi j f}$$



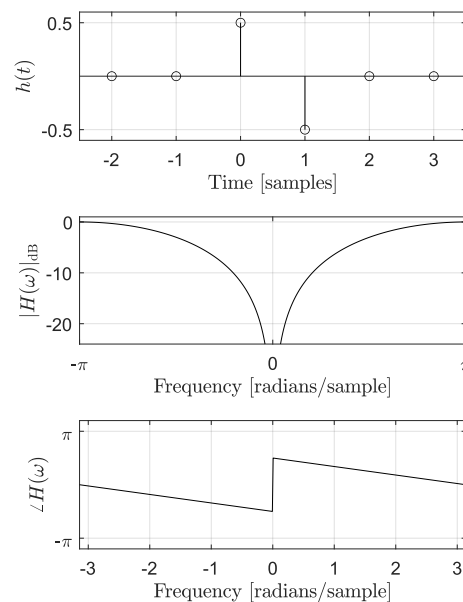
Discrete time first order difference

Impulse response:

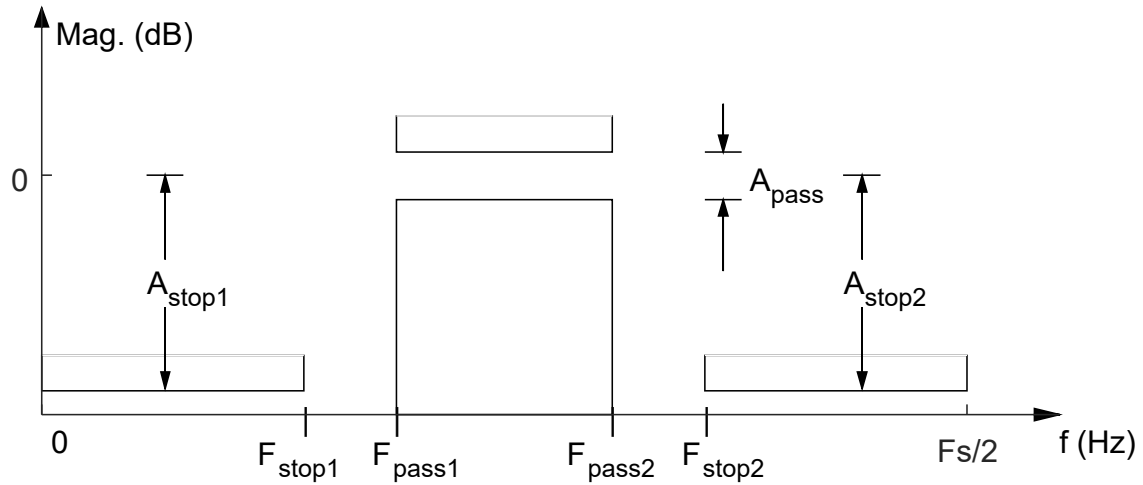
$$h[n] = \frac{1}{2}(\delta[n] - \delta[n - 1])$$

Frequency response:

$$H(\omega) = \frac{1}{2}(1 - e^{-j\omega})$$



Bandpass



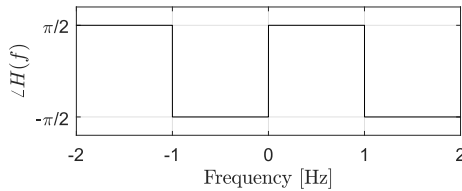
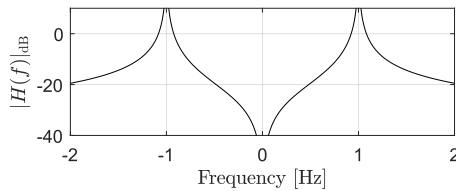
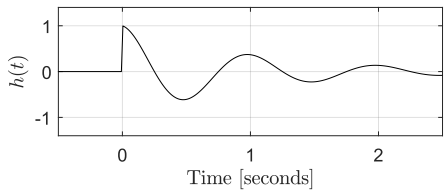
Continuous time bandpass example

Impulse response:

$$h(t) = e^{-t} \cos(2\pi t) u(t)$$

Frequency response:

$$H(f) = \frac{2\pi j f}{(2\pi)^2 + (2\pi j f)^2}$$



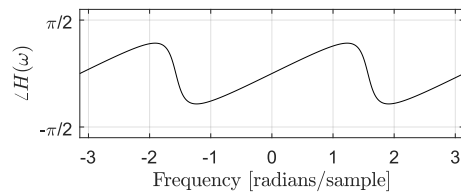
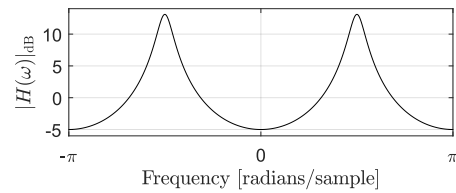
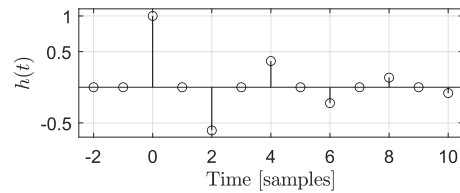
Discrete time bandpass example

Impulse response:

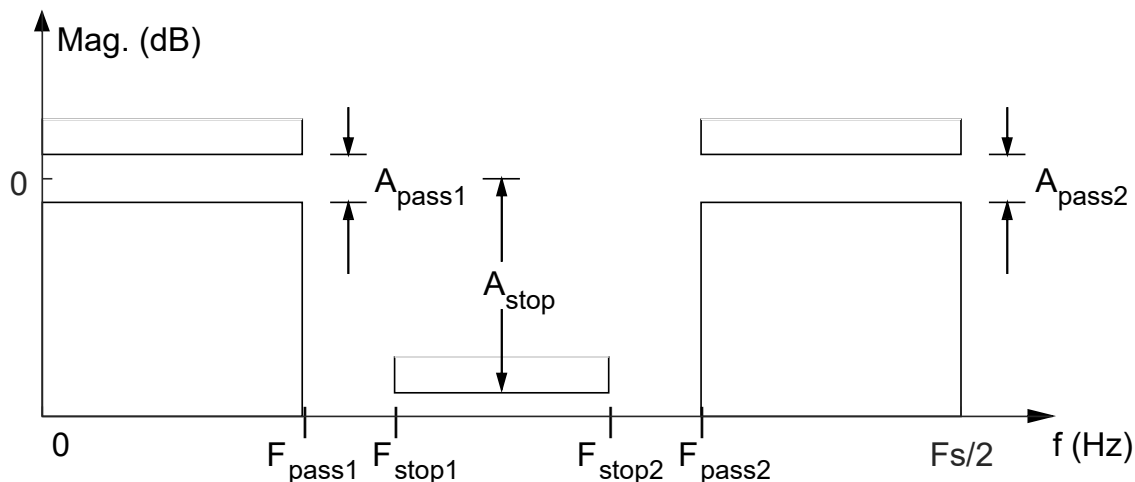
$$h[n] = e^{-n/4} \cos\left(\frac{\pi}{2}n\right) u[n]$$

Frequency response:

$$H(\omega) = \frac{1}{1 + e^{-(2j\omega + 1/4)}}$$



Bandstop and notch



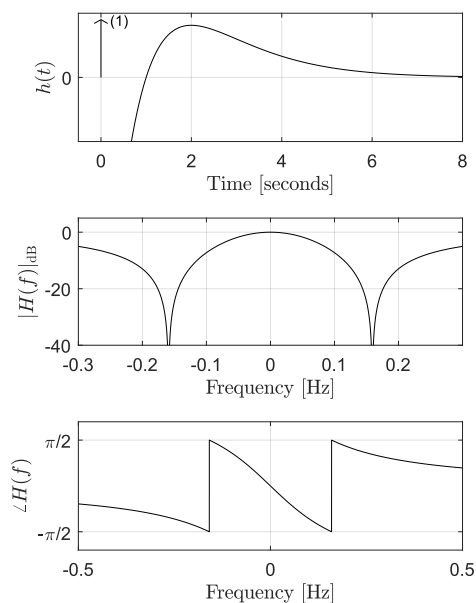
Continuous time bandstop example

Impulse response:

$$h(t) = \delta(t) - 2e^{-t}(1-t)u(t)$$

Frequency response:

$$H(f) = \frac{(2\pi jf)^2 + 1}{(2\pi jf + 1)^2}$$



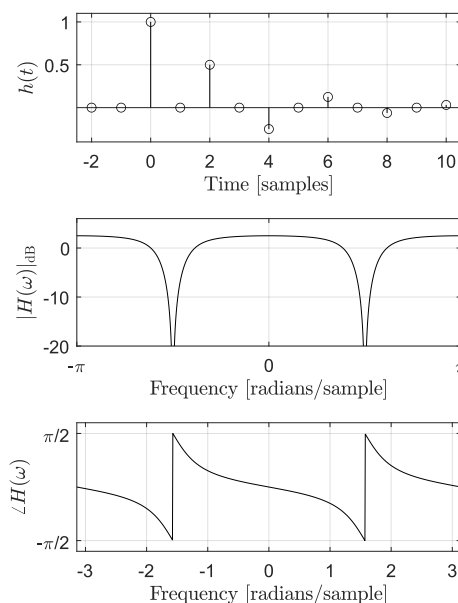
Discrete time notch example

Impulse response:

$$h[n] = 2\delta[n] - 2^{-n/2} \cos\left(\frac{\pi}{2}n\right)u[n]$$

Frequency response:

$$H(\omega) = \frac{1 + e^{-2j\omega}}{1 + \frac{1}{2}e^{-2j\omega}}$$



Allpass

Allpass filters have a flat magnitude response but affect the signal's phase. Two common examples are the ideal delay and the [Hilbert transform](#).

Time Invariance for an Integrator

Prof. Brian L. Evans and Mr. Jaehong Moon

Dept. of Electrical and Computer Engineering

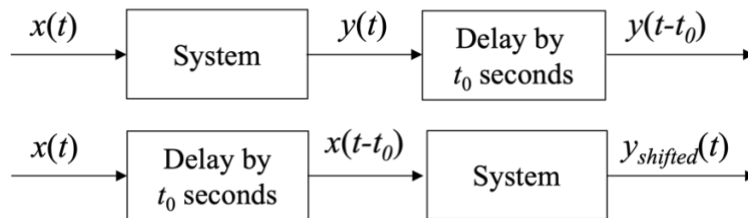
The University of Texas at Austin

February 17, 2022, and updated January 2, 2023

When evaluating system properties, we treat a system as a closed box and analyze the relationships between input signals and their corresponding output signals. This process assumes that after inputting a signal, we can return the system to its original state.

A continuous-time system with input signal $x(t)$ and output signal $y(t)$ is time-invariant (shift-invariant) if whenever the input signal is delayed by t_0 seconds, then the output signal will always be delayed by t_0 seconds as well for all real values of t_0 .

A way to visualize the time-invariance property is to show the equivalence between



That is, does $y_{shifted}(t) = y(t - t_0)$ for all possible real constant values for t_0 ?

Integrator. An integrator captures the idea of conservation or storage [1]. “Capacitors can be modeled as integrators (capacitors are reservoirs for electric charge).” [1] The integrator integrates the input signal $x(t)$; i.e., the output of the integrator is

$$y(t) = \int_{-\infty}^t x(\lambda) d\lambda$$

Two-sided infinite observation. For this case, we observe the input and output signals for all time, i.e. $-\infty < t < \infty$. We can determine if the system is time-invariant as follows. We delay the input signal $x(t - t_0)$ and analyze the resulting output signal

$$y_{shifted}(t) = \int_{-\infty}^t x(\lambda - t_0) d\lambda$$

and see if it is equal to $y(t - t_0)$. We can use a change of variables $\xi = \lambda - t_0$ which means that $d\xi = d\lambda$, $\xi_{upper} = t - t_0$ and $\xi_{lower} = -\infty$ to obtain

$$y_{shifted}(t) = \int_{-\infty}^{t-t_0} x(\xi) d\xi = y(t - t_0)$$

Conclusion: An integrator under two-sided observation in time is time-invariant.

One-sided infinite observation. In this case, we observe the input and output signals of the integrator for $t \geq 0^-$. Time 0^- means a time of 0 seconds before occurrence of a Dirac delta occurring at the origin. We will only allow positive values of t_0 so that the delay for the input and/or output signals will be at times being observed.

Under what conditions will the integrator be time-invariant? [2]

The integrator output for $t \geq 0^-$ is

$$y(t) = \int_{-\infty}^t x(\lambda) d\lambda = \int_{-\infty}^{0^-} x(\lambda) d\lambda + \int_{0^-}^t x(\lambda) d\lambda = C_0 + \int_{0^-}^t x(\lambda) d\lambda$$

The scalar constant C_0 captures the result of the integration of the input signal over time not observed from $-\infty < t < 0$. Delaying the output signal by t_0 seconds gives

$$y(t - t_0) = C_0 + \int_{0^-}^{t-t_0} x(\lambda) d\lambda \quad (1)$$

for $t \geq 0^-$. For $0^- < t < t_0$, the integral accesses values of the input signal $x(t)$ from $-t_0 < t < 0$ which we are not able to observe.

When delaying the input signal $x(t - t_0)$, the resulting output signal for $t \geq 0^-$ is

$$y_{shifted}(t) = C_0 + \int_{0^-}^t x(\lambda - t_0) d\lambda$$

That is, the integrator has its own time reference, and shifting the input or output signal in time does not affect the clock internal to the integrator.

To help compare $y(t - t_0)$ and $y_{shifted}(t)$, we perform algebraic manipulations on

$$y_{shifted}(t) = C_0 + \int_{0^-}^t x(\lambda - t_0) d\lambda$$

We use a substitution of variables $\xi = \lambda - t_0$ which means that $d\xi = d\lambda$, $\xi_{upper} = t - t_0$ and $\xi_{lower} = -t_0$ to obtain

$$\begin{aligned} y_{shifted}(t) &= C_0 + \int_{-t_0}^{t-t_0} x(\xi) d\xi \\ y_{shifted}(t) &= C_0 + \int_{-t_0}^{0^-} x(\xi) d\xi + \int_{0^-}^{t-t_0} x(\xi) d\xi \\ y_{shifted}(t) &= C_0 + C_1(t_0) + \int_{0^-}^{t-t_0} x(\xi) d\xi \end{aligned} \quad (2)$$

For (1) and (2) to be equal for $t \geq t_0$, $C_1(t_0)$ has to be zero for all values of t_0 . That is, when integrating $x(t)$ with respect to t from $-t_0$ to 0^- , the answer is always 0 regardless of the value of t_0 . The only way for this to happen is if $x(t)$ has amplitude of zero for $-t_0 < t < 0^-$. This means that as $t_0 \rightarrow \infty$, $C_0 \rightarrow 0$.

Alternately, as $t_0 \rightarrow \infty$, $C_1(t_0) \rightarrow C_0$. In the limit, a necessary condition for (1) and (2) to be equal for $t \geq t_0$ is for $C_0 = 0$ to solve $C_0 = 2 C_0$.

Conclusion: An integrator observed for $t \geq 0^-$ is time-invariant if the initial condition is 0.

References

- [1] Pedro Albertos and Iven Mareels, [Feedback and Control for Everyone](#), Springer, 2010.
- [2] Stack Exchange, ["Consider the integrator and check for time invariance"](#), April 14, 2019.