EE 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 2014

Table of Contents

Handouts

Italics means available on Web only at http://www.ece.utexas.edu/~bevans/courses/rtdsp

EE445S Real-Time DSP Lab: Lecture & Lab

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

For spring 2014, lecture will be held in ETC 5.148 on Mondays, Wednesdays, and Fridays from 11:00 am to 12:00 pm, beginning Jan. 13th and ending May 2nd. The laboratory sessions will be held in ENS 252B on Mondays, Tuesdays, Wednesdays, and Fridays from Jan. 21st to May 2nd.

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

All lecture slides (13 MB) and the course reader (18 MB) are available for Spring 2014.

For the first half of the semester, the weekly schedule of lecture and lab topics follows. Reading assignments are also given, where JSK means Johnson, Sethares and Klein, *Software Receiver Design*, and WWM means Welch, Wright and Morrow, *Real-Time Digital Signal Processing*.

Week	*Monday Lecture*	*Wednesday Lecture*	*Friday Lecture*	*Lab*	*Reading*
Jan. 13th	Introduction	Introduction	Sinusoidal Generation	NONE	Wednesday: JSK ch. 1 Friday: JSK 2.1-2.7 Reader handouts A-D & R
Jan. 20th	DR. MARTIN LUTHER KING DAY	Sinusoidal Generation	*Discussion of homework #0 solutions*	Introduction - Tools	**Tuesday: Pre-lab Reading** Wednesday: JSK 2.8-2.16 Friday: JSK 3.1-3.4
Jan. 27th	Signals and Systems	Signals and Systems	*Discussion of homework #1 solutions*	Sine Wave Generation	**Monday: Pre-lab Quiz** Wednesday: JSK 3.5-3.8 and app. A.2, A.4, G.1 & G.2 Friday: JSK 4.1-4.6, and Reader handouts E & F
Feb. 3rd	Finite Impulse Response Filters	Finite Impulse Response Filters	*Discussion of homework #2 solutions*	Sine Wave Generation	Monday: JSK 7.1-7.2 Wednesday: JSK app. F
Feb. 10th	Finite Impulse Response Filters	Finite Impulse Response Filters	Introduction to Digital Signal Processors (DSPs)	Digital Filters	**Monday: Pre-lab Quiz** Friday: Reader handout N
Feb. 17th	Introduction to DSPs	Infinite Impulse Response Filters	Infinite Impulse Response Filters	Digital Filters	Wednesday: Reader handout O
Feb. 24th	*Discussion of homework #3*	Infinite Impulse Response Filters	Infinite Impulse Response Filters	Digital Filters	Friday: JSK 5.1-5.2, 6.1-6.3 and A.3; Reader handout H
Mar. 3rd	Sampling and Aliasing	Sampling and Aliasing	Midterm #1	Digital Filters	Monday: JSK 6.4

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

Week	*Monday Lecture*	*Wednesday Lecture*	*Friday Lecture*	*Lab*	*Reading*
Mar. 17th	Interpolation and Pulse Shaping	Interpolation and Pulse Shaping	*Discussion of midterm #1 solutions*	Data Scramblers	**Monday: Pre-lab Quiz** Wednesday: JSK 5.3-5.4, 8.1-8.5, and Reader handout I Friday: JSK 9.1-9.4 and app. B & E
Mar. 24th	Channel Impairments	Digital Pulse Amplitude Modulation	*Discussion of homework #4 solutions*	Pulse Amplitude Modulation	**Monday: Pre-lab Quiz** Wednesday: JSK 11.1-11.6 and app. C Friday: JSK 10.1-10.4, and Reader handout M & S
Mar. 31st	Digital Pulse Amplitude Modulation	Matched Filtering	*Discussion of homework #5 solutions*	Pulse Amplitude Modulation	*Monday: JSK 13.1-13.3* Wednesday: JSK 12.1-12.4 Friday: JSK 16.1-16.2
Apr. 7th	Matched Filtering	Matched Filtering	Matched Filtering	Pulse Amplitude Modulation	Monday: JSK 16.3-16.6 Wednesday: JSK 16.7-16.11 Friday: Reader handout P
Apr. 14th	QAM Transmitter	QAM Transmitter	QAM Receiver	Quadrature Amplitude Modulation	**Monday: Pre-lab Quiz**
Apr. 21st	Quantization	Quantization	Data Conversion	Quadrature Amplitude Modulation	*Monday: Reader handout J* Friday: JSK ch. 7 & app. D
Apr. 28th	Data Conversion	Review	Midterm #2	Guitar Special Effects	**Monday**: **Pre-lab Quiz**

The following lectures are not scheduled to be presented this semester:

- TMS320C6000 DSP
- Advanced Data Conversion
- Fast Fourier Transform
- 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

EE445S Real-Time Digital Signal Processing Laboratory - Overview

Prof. Brian L. Evans

This undergraduate elective is an introduction to the analysis, design, and implementation of embedded real-time digital signal processing systems. "Real-time" means guaranteed delivery of data by a certain time. "Embedded" means that the subsystem performs behind-the-scenes tasks within a larger system. These tasks are often tailored to an application, e.g. speech compression/decompression for a cell phone.

Traditionally, users do not directly interact with the embedded systems in the product. For example, a modern cell phone contains several embedded systems, including processors, memory systems, and input/output systems, but the user interacts with the cell phone through the touchscreen and/or voice commands. As another example, a PC contains several embedded digital systems, including the disk drive, CD/DVD player, video processor, and wireless LAN system. Embedded systems range from a system-on-chip to a board to a rack of computing engines to a distributed network of computing engines.

The application space for embedded systems includes control, communication, networking, signal processing, and instrumentation. High-volume products in units shipped worldwide in 2012 include

- 1750M cell phones
- 350M PCs/laptops
- 115M DVD/Blu-ray players
- 100M digital still cameras
- 75M DSL/VDSL modems
- 70M cars/light trucks
- 34M video game consoles

High-end cars now have more than 150 embedded processors in them. More than two billion products are sold each year with multiple embedded digital signal processing systems in them. In fact, there are more embedded programmable processors in the world than people.

Texas is a worldwide epicenter for microprocessors for control, signal processing, and communication systems. In 2007, Texas Instruments (Dallas, TX) and Freescale (Austin, TX) have 64% and 12%, respectively, of the $8B embedded programmable digital signal processor market. Their digital signal processors were developed, and are still being developed, in Texas. Near three-fourths of all digital signal processors are used in wireless systems for both cellular and data networks. Texas Instruments and Freescale are also market leaders in the embedded programmable microcontroller market, esp. for the automotive sector. In addition, Qualcomm and Cirrus Logic have developed several generations of programmable digital signal processors in Austin, Texas, for cell phones and audio systems, respectively. To boot, Austin is a worldwide leader in ARM-based digital VLSI design centers.

Through this undergraduate elective, I hope that students gain an intuitive feel for basic discrete-time signal processing concepts and how to translate these concepts into real-time software using digital signal processor technology. The course will review some of the mathematical foundations of the course material, but emphasize the qualitative concepts. The qualitative concepts are reinforced by hands-on laboratory work and homework assignments.

In the laboratory 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, real-time programming, and modern digital signal processor architectures.

In particular, we will discuss design tradeoffs between implementation complexity and signal quality/communication performance.

In the laboratory component, students implement transceiver subsystems in C on a Texas Instruments TMS320C6748 floating-point dual-core programmable digital signal processor. The C6000 family is used in DSL modems, wireless LAN modems, mobile wireless 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 rack equipment, Texas Instruments Code Composer Studio software, and National Instruments LabVIEW software. A voiceband transceiver reference design and simulation is available in LabVIEW.

In addition to learning about voiceband modem 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, esp. asymmetric digital subscriber line (ADSL) and wireless LAN systems. In particular, we discuss the data transmission subsystems in ADSL and wireless LAN transceivers. 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).

A UT undergraduate ECE student who took the real-time DSP laboratory course in Fall 1999 and graduated in May of 2000 wrote the following about the course in August 2000:

"... keep that real-time DSP lab as good as it was when I took it. I have to say, that lab was the best class I took at UT. It is close enough to the cutting edge of technology that you can hold a conversation with someone from industry and actually contribute useful ideas. 345L is a close second. Good work."

A UT undergraduate BME student who took the real-time DSP laboratory course in Spring 2009 wrote the following about the course in June 2009:

"I wanted to thank you for teaching the EE 445S course this last spring semester (Spring 09). I got my summer internship based on my experience in the EE 445S lab and the whole course. I am told that [Company X] has never hired any Biomedical Engineering student before, but because of this course I got the opportunity to be the first BME student in this company."

EE 445S Real-Time Digital Signal Processing Lab *Spring 2014*

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/rtdsp*

Outline

- Instructional staff
- Real-time digital signal processing
- Course overview
- Communication systems
- Single carrier transceiver
- Multicarrier transceivers
- Conclusion

0-2

Instructional Staff

- **Prof. Brian L. Evans**

  Conducts research in digital communication, digital image processing & embedded systems

  Past and current projects on next two slides

  Office hours: M 12:00-12:30pm, W 12:00-12:30pm, TH 12:30-2:30pm (ENS 433B)

  Coffee hours F 12:00-2:00pm starting Jan. 17th

- **Teaching assistants (lab sections/office hours below)**

Mr. Chao Jia
W & F lab sections
TH 3:30-5:30pm
F 9:30-10:30am

Ms. Zeina Sinno
M & T lab sections
W 3:00-4:30pm
TH 5:30- 7:00pm

0-3

Instructional Staff

Completed Research Projects

21 PhD and 9 MS alumni

System	Contribution	SW release	Prototype	Funding
ADSL	equalization	Matlab	DSP/C	Freescale, TI
	2x2 testbed	LabVIEW	LabVIEW/PXI	Oil&Gas
Wimax/LTE	resource alloc.	LabVIEW	DSP/C	Freescale, TI
Underwater comm.	space-time comm. large rec. arrays	Matlab	Lake Travis testbed	UT Applied Res. Labs
Camera	image acquisition	Matlab	DSP/C	Intel, Ricoh
Display	image halftoning	Matlab	C	HP, Xerox
	video halftoning	Matlab	C	Qualcomm
Elec. design automation	fixed point conv.	Matlab	FPGA	Intel, NI
	distributed comp.	Linux/C++	Navy sonar	Navy, NI

DSP Digital Signal Processor
LTE Long-Term Evolution (cellular)
FPGA Field Programmable Gate Array
PXI PCI Extensions for Instrumentation

0-4

Current Research Projects

9 PhD students

System	Contributions	SW release	Prototype	Funding
Powerline comm.	interference reduction; testbeds	LabVIEW	Freescale, TI modems	Freescale, IBM, TI
Wi-Fi	interference reduction	Matlab	NI FPGA	Intel, NI
	time-based analog-to-digital converter		IBM 45nm TSMC 180nm	
Cellular (LTE)	cloud radio access net. baseband compression	Matlab		Huawei
Handheld camera	reducing rolling shutter artifacts	Matlab	Android	TI
EDA	reliability patterns			NI

EDA Electronic Design Automation
LTE Long-Term Evolution (cellular)
FPGA Field Programmable Gate Array
TSMC Taiwan Semicond. Manufac. Corp.

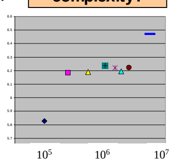
Real-Time Digital Signal Processing

- **Real-time systems** [Prof. Yale Patt, UT Austin]
  - Guarantee delivery of data by a specific time
- **Signal processing** [http://www.signalprocessingsociety.org]
  - Generation, transformation, extraction, interpretation of information
  - Algorithms with associated architectures and implementations
  - Applications related to processing information
- **Embedded systems**
  - Perform application-specific tasks
  - Work "behind the scenes" (e.g. speech compression)

Course Overview

- **Objectives**
  - Build intuition for signal processing concepts
  - Explore design tradeoffs in *signal quality vs. implementation complexity*
- **Lecture: breadth (3 hours/week)**
  - Digital signal processing (DSP) algorithms
  - Digital communication systems
  - Digital signal processor (DSP) architectures
- **Laboratory: depth (3 hours/week)**
  - Translate DSP concepts into software
  - Design/implement data transceiver
  - Test/validate implementation

Measures of signal quality? Implementation complexity?

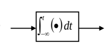


ADSL receiver design: bit rate (Mbps) vs. multiplications in equalizer training methods

[Data from Figs. 6 & 7 in B. L. Evans *et al.*, "Unification and Evaluation of Equalization Structures…", *IEEE Trans. Sig. Proc.*, 2005]

Pre-Requisites and Co-Requisites

- **Pre-Requisites**
  - *Introduction to Programming:* C programming, arrays and circular buffers, asymptotic analysis
  - *Signals & Systems:* convolution, transfer functions, frequency responses, filtering
  - *Intro. to Embedded Systems:* assembly and C languages, microprocessor organization, quantization
- **Co-Requisites**
  - *Probability:* Gaussian and uniform distributions, sum of random variables, statistical independence, random processes, correlation
  - *Engineering Communication:* technical writing



Detailed Topics

- **Digital signal processing algorithms/applications**
  Signals, convolution and sampling (*signals & systems*)
  Transfer functions & freq. responses (*signals & systems*)
  Filter design & implementation, signal-to-noise ratio
  Quantization (*embedded systems*) and data conversion

- **Digital communication algorithms/applications**
  Analog modulation/demodulation (*signals & systems*)
  Digital modulation/demodulation, pulse shaping, pseudo noise
  Signal quality: matched filtering, bit error probability

- **Digital signal processor (DSP) architectures**
  Assembly language, interfacing, pipelining (*embedded systems*)
  Harvard architecture, addressing modes, real-time prog.

Digital Signal Processors In Products

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 (WPI)


Thad Welch (Boise State)

Cameron Wright (Wyoming)


Michael Morrow (Wisconsin)

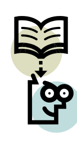
***Real-Time Digital Signal Processing from Matlab to C with the TMS320C6x DSPs*, Dec. 2011**
Matlab simulation
Mapping algorithms to C

Supplemental (Optional) Textbooks

- **J. H. McClellan, R. W. Schafer & M. A. Yoder, *DSP First: A Multimedia Approach*, 1998**
  DSP theory and algorithms at sophomore level
  Demos: http://users.ece.gatech.edu/~dspfirst/

R. Schafer's 1975 book seminal for DSP theory

- **B. P. Lathi, *Linear Systems & Signals*, or M. J. Roberts, *Signals and Systems*, or Oppenheim & Willsky, *Signals and Systems***
  Textbook for pre-requisite signals & systems course



- **Steve Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 1997**
  Available **free** online: http://www.dspguide.com

Related BS ECE Technical Cores

Signal/image processing

Real-Time Dig. Sig. Proc. Lab
Digital Signal Processing
Introduction to Data Mining
Digital Image & Video Processing

Courses with the highest workload at UT Austin?

Undergraduate students may take grad courses upon request and at their own risk ☺

Communication/networking

Real-Time Dig. Sig. Proc. Lab
Digital Communications
Wireless Communications Lab
Telecommunication Networks

Embedded Systems

Embedded & Real-Time Systems
Real-Time Dig. Sig. Proc. Lab
Digital System Design (FPGAs)
Computer Architecture
Introduction to VLSI Design

Grading

- **Calculation of numeric grades**
  21% midterm #1
  21% midterm #2
  14% homework (*drop lowest grade of eight*)
  5% pre-lab quizzes (*drop lowest grade of six*)
  39% lab reports (*drop lowest grade of seven*)
- **21% for each midterm exam**
  Focus on design tradeoffs in signal quality vs. complexity
  Based on in-lecture discussion and homework/lab assignments
  Open books, open notes, open computer (but no networking)
  Dozens of old exams (most with solutions) in course reader
  Test dates on course descriptor and lecture schedule

Average GPA has been ~3.1
MyEdu.com

No final exam

Grading

- **14% homework – eight assignments (*drop lowest*)**
  Strengthen theory and analysis
  Translate signal processing concepts into Matlab simulations
  Evaluate design tradeoffs in signal quality vs. complexity
- **5% pre-lab quizzes – for labs 2-7 (*drop lowest*)**
  10 questions on course Blackboard site taken individually
- **39% lab reports – for labs 1-7 (*drop lowest*)**
  Work ***individually*** on labs 1 and 7
  Work ***in team of two*** on labs 2-6 and receive same base grade
  Attendance/participation in lab section required and graded
- **Course ranks in graduate school recommendations**

Maximizing Your Numeric Grade

- **Attend every lecture**
  Most important information not on slides [fall 2010 student]
- **Complete every homework**
- **Submit only your own work**
  Independent solutions on all homework assignments, lab 1/7 reports and all pre-lab quizzes
  Lab team on lab 2-6 reports
  Cite sources for all other work

Spring 2011		
Lowest Grades	Lecture Absences	Zeros on homework
55.13	10	6
68.12	10	6
73.96	0	0
74.43	5	4
74.80	12	2
74.90	2	1
75.89	6	2

"In May 2006, William Swanson, CEO of Raytheon … was docked approximately US $1 million in pay by the company after it was revealed he had plagiarized 16 of the 33 rules in his popular 2004 book, *Swanson's Unwritten Rules of Management*." [Sept. 8, 2006, issue of IEEE's The Institute electronic newsletter]

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**

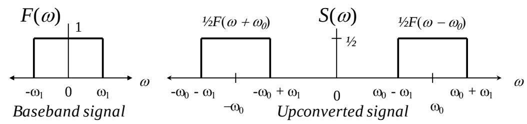
  Lowpass filter message signal (e.g. AM/FM radio)

  Digital: Add redundancy to message bit stream to aid receiver in detecting and possibly correcting bit errors

Communication System Structure

- **Carrier circuits in transmitter**

  Upconvert baseband signal into transmission band



  Then apply bandpass filter to enforce transmission band

Communication System Structure

- **Channel – wired or wireless**

  Propagating signals spread and attenuate over distance

  Boosting improves signal strength and reduces noise

- **Receiver**

  Carrier circuits downconvert bandpass signal to baseband

  Baseband processing extracts/enhances message signal

Single Carrier Transceiver Design

- **Design/implement transceiver**

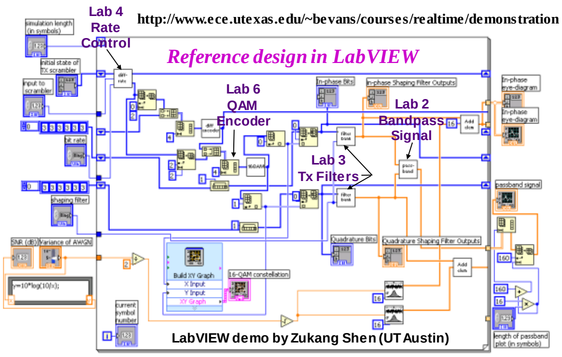
  Design different algorithms for each subsystem

  Translate algorithms into real-time software

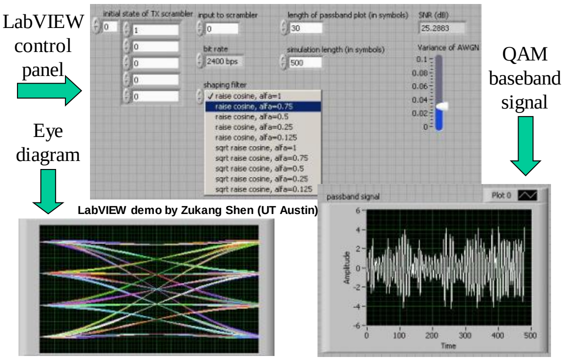
  Test implementations using signal generators & oscilloscopes

Laboratory	Transceiver Subsystems
1 introduction	block diagram of transmitter
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

Lab 1: QAM Transmitter Demo

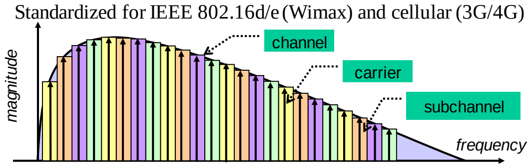


Lab 1: QAM Transmitter Demo



Got Anything Faster?

- **Multicarrier modulation divides broadband (wideband) channel into narrowband subchannels**
  - Uses Fourier series computed by fast Fourier transform (FFT)
  - Standardized for ADSL (1995) & VDSL (2003) wired modems
  - Standardized for IEEE 802.11a/g wireless LAN
  - Standardized for IEEE 802.16d/e (Wimax) and cellular (3G/4G)



Each ADSL/VDSL subchannel is 4.3 kHz wide (about width of voiceband channel) and carries a QAM signal

Conclusion

- **Objectives**
  - Build intuition for signal processing concepts
  - Translate signal processing concepts into real-time digital communications software
- **Deliverables and takeaways**
  - Tradeoffs of signal quality vs. implementation complexity
  - Design/implement voiceband transceiver in real time
  - Test/validate implementation
- **Role of technology**
  - Matlab for algorithm development
  - TI DSPs and Code Composer Studio for real-time prototyping
  - LabVIEW for test and measurement

Plug into network of 1,400+ course alumni

All software/hardware used lead in usage in their respective markets

EE 445S Real-Time Digital Signal Processing Lab *Spring 2014*

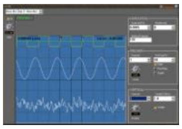
Generating Sinusoidal Signals

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/rtdsp*

Outline

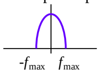
- **Bandwidth**
- **Sinusoidal amplitude modulation**
- **Sinusoidal generation**
- **Design tradeoffs**



Bandwidth

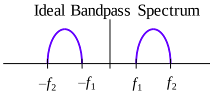
- **Non-zero extent in *positive* frequencies**

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" spectrum from 0 to 10^{15} Hz
  - Finite observation time of signal leads to infinite bandwidth
- **Alternatives to "non-zero extent"?**

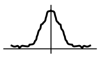
Lowpass Signal in Noise

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

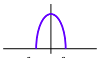
$$\min_{f_{max}} \int_0^{f_{max}} |H(f)|^2 \, df \geq 0.9 \text{ Energy}$$

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

Lowpass Spectrum in Noise



Idealized Lowpass Spectrum



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

Baseband signal: energy in frequency domain concentrated around DC

Bandpass Signal in Noise

- **How to find f_1 and f_2?**

  Apply threshold and eyeball it

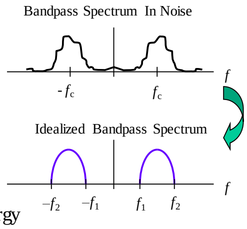
  –OR–

  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 $0 \leq W \leq 2 f_c$

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

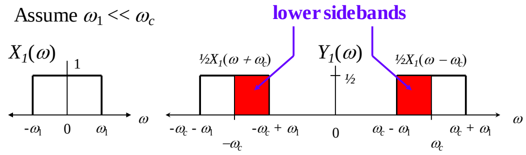


Amplitude Modulation by Cosine

- $y_1(t) = x_1(t) \cos(\omega_c t)$ $\qquad Y_1(\omega) = \frac{1}{2} X_1(\omega + \omega_c) + \frac{1}{2} X_1(\omega - \omega_c)$

  Assume $x_1(t)$ is an ideal lowpass signal with bandwidth ω_1

  Assume $\omega_1 << \omega_c$



  $Y_1(\omega)$ has (transmission) bandwidth of $2\omega_1$

  $Y_1(\omega)$ is real-valued if $X_1(\omega)$ is real-valued

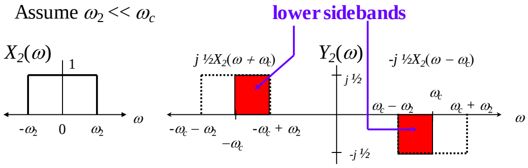
- **Demodulation: modulation then lowpass filtering**

Amplitude Modulation by Sine

- $y_2(t) = x_2(t) \sin(\omega_c t)$ $\qquad Y_2(\omega) = \frac{j}{2} X_2(\omega + \omega_c) - \frac{j}{2} X_2(\omega - \omega_c)$

  Assume $x_2(t)$ is an ideal lowpass signal with bandwidth ω_2

  Assume $\omega_2 << \omega_c$



  $Y_2(\omega)$ has (transmission) bandwidth of $2\omega_2$

  $Y_2(\omega)$ is imaginary-valued if $X_2(\omega)$ is real-valued

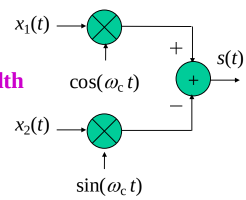
- **Demodulation: modulation then lowpass filtering**

How to Use Bandwidth Efficiently?

- **Send lowpass signals $x_1(t)$ and $x_2(t)$ with $\omega_1 = \omega_2$ over same transmission bandwidth**

  Called Quadrature Amplitude Modulation (QAM)

  Used in DSL, cable, Wi-Fi, and LTE cellular communications

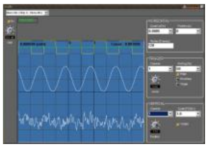


$$S(\omega) = \frac{1}{2}\left(X_1(\omega + \omega_c) + X_1(\omega - \omega_c)\right) - j\frac{1}{2}\left(X_2(\omega + \omega_c) - X_2(\omega - \omega_c)\right)$$

- **Cosine modulated signal is *orthogonal* to sine modulated signal at transmitter**

  Receiver separates $x_1(t)$ and $x_2(t)$ through demodulation

Lab 2: Sinusoidal Generation



- **Compute sinusoidal waveform**
  - Function call
  - Lookup table
  - Difference equation
- **Output waveform off chip**
  - Polling data transmit register
  - Software interrupts
  - Quantization effects in digital-to-analog (D/A) converter
- **Expected outcomes are to understand**
  - Signal quality vs. implementation complexity tradeoff
  - Interrupt mechanisms

Sinusoidal Waveforms

- **One-sided discrete-time cosine (or sine) signal with fixed-frequency ω_0 in rad/sample has form**

  $$\cos(\omega_0 n)\, u[n]$$

- **Consider one-sided continuous-time analog-amplitude cosine of frequency f_0 in Hz**

  $$\cos(2\,\pi\, f_0\, t)\, u(t)$$

  Sample at rate of f_s by substituting $t = n\, T_s = n \,/\, f_s$

  $$(1/T_s) \cos(2\,\pi\,(f_0 \,/\, f_s)\, n)\, u[n]$$

  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, $\omega_0 = 3/10\,\pi$
- **How to determine gain for D/A conversion?**

Math Library Call in C

- **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 11^{th} 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)

Efficient Polynomial Implementation

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

  $$a_{11}\, x^{11} + a_{10}\, x^{10} + a_9\, x^9 + \ldots + a_0 =$$
  $$(\ \ldots\ (((a_{11}\, x + a_{10})\, x + a_9)\, x \ \ldots\ ) + a_0$$

- **Comparison**

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

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 $\sin(\omega_0 n)\, u[n]$

  Initial conditions are all zero
- **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

Difference Equation

- **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-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 $2 \cos(\omega_0)$

Lookup Table

- **Precompute samples offline and store them in table**
- **Cosine frequency $\omega_0 = 2\, \pi\, N / L$**

  Remove all common factors between integers N and L

  Continuous-time period for $\cos(2\, \pi\, f_0\, t)$ is $T_0 = 1 / f_0$

  Discrete-time period for $\cos(2\, \pi\, (N / L)\, n)$ is L samples

  Store L samples in lookup table (N continuous-time periods)
- **Built-in lookup tables in read-only memory (ROM)**

  Samples of $\cos(\theta)$ and $\sin(\theta)$ at uniformly spaced values for θ

  Interpolate values to generate sinusoids of various frequencies

  Allows adaptation of ω_0 if desired

Design Tradeoffs

- **Signal quality vs. implementation complexity in generating $\cos(\omega_0\, n)\, u[n]$ with $\omega_0 = 2\, \pi\, N / L$**

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

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

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

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

$$dt = \frac{1}{a} du$$

limits $t \to \infty \Rightarrow u \to \infty$

$t \to -\infty \Rightarrow u \to -\infty$

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

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

$$dt = \frac{1}{a} du$$

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

$t \to -\infty \Rightarrow u \to \infty$

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

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

for $a \neq 0$

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

Discrete-Time Periodicity

Prof. B. L. Evans

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

$x[n + N_0] = x[n]$ for all n. N_0 is positive.

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

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

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

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

<u>if</u> $\frac{N}{L} N_0$ 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

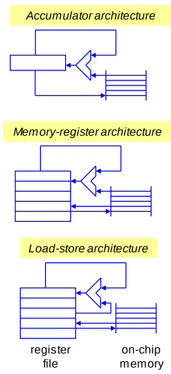
INTRODUCTION TO DIGITAL SIGNAL PROCESSORS

Prof. Brian L. Evans

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

Embedded Signal Processing Laboratory
The University of Texas at Austin

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



Outline

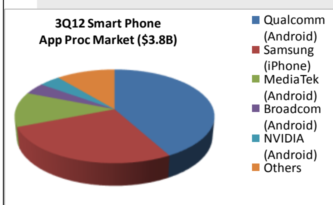
- Embedded processors and systems
- Signal processing applications
- TI TMS320C6000 digital signal processor
- Conventional digital signal processors
- Pipelining
- RISC vs. DSP processor architectures
- Conclusion

Embedded Processors and Systems

- Embedded system works
  - On application-specific tasks
  - "Behind the scenes" (little/no direct user interaction)
- Units of consumer products shipped in 2012
  - *1750M* cell phones ↑
  - *350M* PCs ↓
  - *115M* DVD/Blu-ray players
  - *100M* digital still cameras
  - *75M* DSL/VDSL modems ↓
  - *70M* cars/light trucks ↑
  - *34M* game consoles ↓
- How many embedded processors are in each?
- How much should an embedded processor cost?
  - 2011: average US prices were $73 for traditional cell phone and $191 for digital still camera
  - 2012: iPhone5 costs $749 (16GB) & $849 w/o contract

Smart Phone Application Processors

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



Source: Cellular News, 11 Jan. 2013
http://www.cellular-news.com/story/58089.php

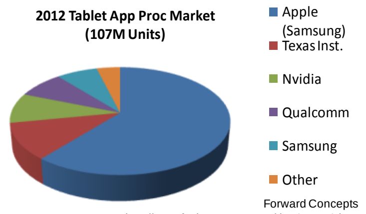
iPhone5 (10+ cores)
- Touchscreen: Broadcom (probably 2 ARM cores)
- Apps: Samsung (2 ARM + 3 GPU cores)
- Audio: Cirrus Logic (1 DSP core + 1 codec)
- Wi-Fi: Broadcom
- Baseband: Qualcomm
- Inertial sensors: STMicroelectronics

"iPhone 5 Tear Down"
http://www.ifixit.com/Teardown/iPhone-5-Teardown/10525/

Market for Application Processors

- $2.3B in tablets, $12.4B in smart phones, 2012
- $3.5B in tablets, $16.1B in smart phones, 2013 (est.)
  - 32% of revenue for all microprocessors sold in 2013 (est.)

["Tablet and Cellphone Processors Offset PC MPU Weakness," Aug 2013]



Forward Concepts
http://www.fwdconcepts.com/dsp071513.htm

Signal Processing Applications

- Embedded system cost & input/output rates
  - *Low-cost, low-throughput:* sound cards, 2G cell phones, MP3 players, car audio, guitar effects
  - *Medium-cost, medium-throughput:* printers, disk drives, 3G cell phones, ADSL modems, digital cameras, video conferencing
  - *High-cost, high-throughput:* high-end printers, audio mixing boards, wireless basestations, 3-D medical reconstruction from 2-D X-rays
- Embedded processor requirements
  - Inexpensive with small area and volume
  - Predictable input/output (I/O) rates to/from processor
  - Low power (e.g. smart phone uses 200mW average for voice and 500mW for video; battery gives 5 W-hours)

Single DSP

Multiple DSP chips or cores + accelerators

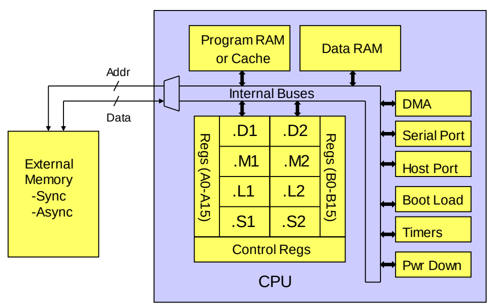
Multiple multicore DSPs

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

Modern Digital Signal Processor Example

TI TMS320C6000 Family, Simplified Architecture



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/compares
  - Shifter unit - 32-bit integer ALU and 40-bit shifter

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

Modern DSP: TMS320C6000 Instruction Set

C6000 Instruction Set by Functional Unit

.S Unit

ADD	NEG
ADDK	NOT
ADD2	OR
AND	SET
B	SHL
CLR	SHR
EXT	SSHL
MV	SUB
MVC	SUB2
MVK	XOR
MVKH	ZERO

.L Unit

ABS	NOT
ADD	OR
AND	SADD
CMPEQ	SAT
CMPGT	SSUB
CMPLT	SUB
LMBD	SUBC
MV	XOR
NEG	ZERO
NORM	

.D Unit

ADD	ST
ADDA	SUB
LD	SUBA
MV	ZERO
NEG	

.M Unit

MPY	SMPY
MPYH	SMPYH

Other

NOP	IDLE

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

Modern DSP: TMS320C6000 Instruction Set

Arithmetic: ABS, ADD, ADDA, ADDK, ADD2, MPY, MPYH, NEG, SMPY, SMPYH, SADD, SAT, SSUB, SUB, SUBA, SUBC, SUB2, ZERO

Logical: AND, CMPEQ, CMPGT, CMPLT, NOT, OR, SHL, SHR, SSHL, XOR

Bit Management: CLR, EXT, LMBD, NORM, SET

Data Management: LD, MV, MVC, MVK, MVKH, ST

Program Control: B, IDLE, NOP

C6000 Instruction Set by Category

(un)signed multiplication
saturation/packed arithmetic

C5000 vs. C6000 Addressing Modes

	TI C5000	TI C6000
Immediate Operand part of instruction	`ADD #0Fh`	`mvk .D1 15, A1` `add .L1 A1, A6, A6`
Register Operand specified in a register	`(implied)`	`add .L1 A7, A6, A7`
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`

C6700 Extensions

C6700 Floating Point Extensions by Unit

.S Unit		.L Unit	
ABSDP	CMPLTSP	ADDDP	INTSP
ABSSP	RCPDP	ADDSP	SPINT
CMPEQDP	RCPSP	DPINT	SPTRUNC
CMPEQSP	RSARDP	DPSP	SUBDP
CMPGTDP	RSQRSP	DPTRUNC	SUBSP
CMPGTSP	SPDP	INTDP	
CMPLTDP			

.D Unit		.M Unit	
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.

Selected TMS320C6700 Floating-Point DSPs

DSP	MHz	MIPS	Data (kbits)	Program (kbits)	Level 2 (kbits)	Price	Applications
C6701	150	1200	512	512	0	$ 88	C6701 EVM board
	167	1336	512	512	0	$141	
C6711	150	1200	32	32	512	n/a	C6711 DSK board
	250	2000				$ 18	
C6712	150	1200	32	32	512	$ 14	
C6713	167	1336	32	32	1000	$ 19	
	225	*1800*	*32*	*32*	*1000*	*$ 25*	*C6713 DSK board*
	300	2400	32	32	1000	$ 33	
C6722	250	2000	1000	3072	256	$ 10	Professional audio
C6726	266	2128	2000	3072	256	$ 15	Professional audio
C6727	300	2400	2000	3072	256	$ 22	C6727 EVM board
	350	2800	2000	3072	256	$ 30	Professional audio
C6748	300	2400	256	256	2048	$ 18	Pro-audio and video
	375	*3000*	*256*	*256*	*2048*	*$ 20*	*C6748 XK & EVM boards*

DSK: DSP Starter Kit. EVM: Evaluation Module.

Unit price for 100 units. Prices effective February 1, 2009.

For more information: http://www.ti.com

Selected TMS320C6000 Fixed-Point DSPs

DSP	MHz	MIPS	Data (kbits)	Program (kbits)	Level 2 (kbits)	Price	Applications
C6202	250	2000	1000	2000		$ 66	
	300	2400				$ 79	
C6203	250	2000	4000	3000		$ 84	modems banks ADSL1
	300	2400				$ 84	modems
C6204	200	1600	512	512		$ 11	
C6416	720	5760	128	128	8000	$114	ADSL2 modems
	1000	8000	128	128	8000	$227	3G basestations
C6418	500	4000	128	128	5000	$ 49	
	600	4800	128	128	5000	$ 49	
DM641	500	4000	128	128	1000	$ 28	Video conferencing
	600	4800	128	128	1000	$ 31	
DM642	500	4000	128	128	2000	$ 37	Video conferencing
	720	5760	128	128	2000	$ 57	
DM648	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

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

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

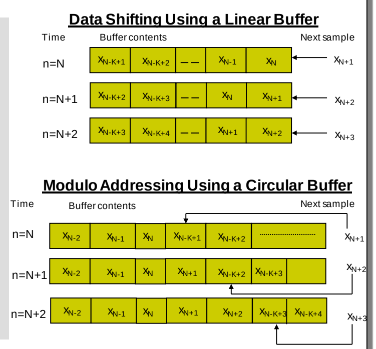
Conventional Digital Signal Processors

- 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

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

Conventional Digital Signal Processors

	Fixed-Point	Floating-Point
Cost/Unit	$2 - $79	$2 - $381
Architecture	Accumulator	load-store or memory-register
Registers	2-4 data 8 address	8 or 16 data 8 or 16 address
Data Words	16 or 24 bit integer and fixed-point	32 bit integer and fixed/floating-point
On-Chip Memory	2-64 kwords data 2-64 kwords program	8-64 kwords data 8-64 kwords program
Address Space	16-128 kw data 16-64 kw program	16 Mw – 4Gw data 16 Mw – 4 Gw program
Compilers	C, C++ compilers; poor code generation	C, C++ compilers; better code generation
Examples	TI TMS320C5000; Freescale DSP56000	TI TMS320C30; Analog Devices SHARC

2-21

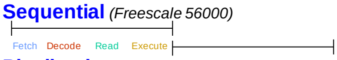
Conventional Digital Signal Processors

- 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

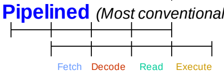
2-22

Pipelining

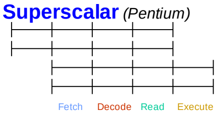
Sequential *(Freescale 56000)*



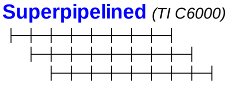
Pipelined *(Most conventional DSPs)*



Superscalar *(Pentium)*



Superpipelined *(TI C6000)*



Pipelining

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

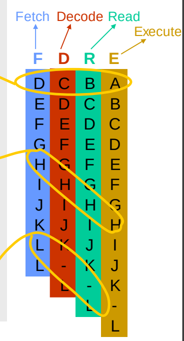
Managing Pipelines

- Compiler or programmer
- Pipeline interlocking

2-23

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

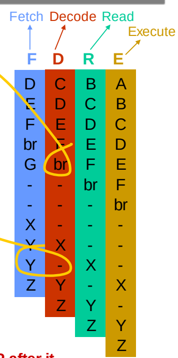


MAC means multiplication-accumulation.

2-24

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

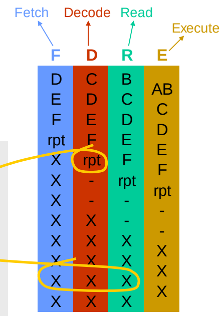
2-25

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

Pipelining: TI TMS320C6000 DSP

Pentium IV pipeline has more than 20 stages

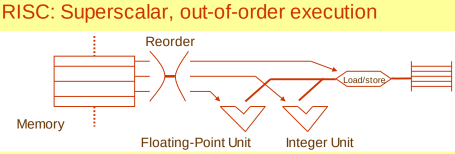
- C6000 has deep pipeline
  - 7-11 stages in C6200: fetch 4, decode 2, execute 1-5
  - 7-16 stages in C6700: fetch 4, decode 2, execute 1-10
  - Compiler and assembler must prevent pipeline hazards
- Only branch instruction: delayed unconditional
  - Processor executes next 5 instructions after branch
  - 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

Contributions by Sundararajan Sriram (TI)

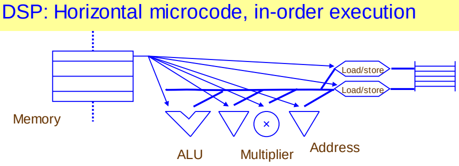
2-27

RISC vs. DSP: Instruction Encoding

- RISC: Superscalar, out-of-order execution



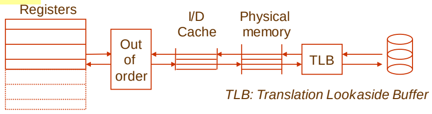
- DSP: Horizontal microcode, in-order execution



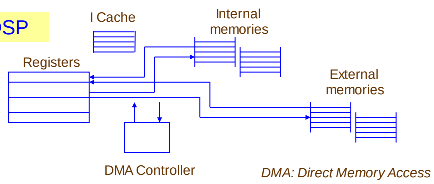
2-28

RISC vs. DSP: Memory Hierarchy

- RISC



- DSP



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)

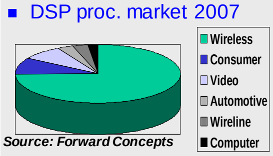
References

- Unit shipments worldwide
  - *Cars & light trucks*: http://www.plunkettresearch.com/automobiles-trucks-market-research/industry-statistics
  - *Cars & light trucks*: http://www.rwbaird.com/docs/yourreports/cruisin.pdf
  - *PCs* http://en.wikipedia.org/wiki/Market_share_of_leading_PC_vendors
  - *Mobile handsets* http://venturebeat.com/2013/02/13/gartner-samsung-apple-smartphone-sales-2012/
  - *Game consoles* http://www.statista.com/statistics/214670/global-unit-sales-of-video-game-consoles/
  - *Digital still cameras* http://www.cipa.jp/english/data/dizital.html
  - *iPhone5 teardown*: http://www.ifixit.com/Teardown/iPhone-5-Teardown/10525/
  - *DSL*:http://www.broadbandtrends.com/yahoo_site_admin/assets/docs/BBT_2012DSLMktShare_131050_TOC.44121205.pdf
- Embedded processor resources
  - Embedded Microproc. Benchmark Consortium http://www.eembc.org
  - Embedded processing comparison from 80+ processor and IP vendors: http://www.embeddedinsights.com/directory.php
  - Other: http://www.eg3.com

Optional

Digital Signal Processors

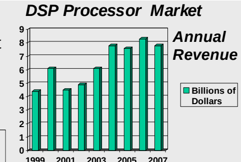
- DSP processor market
  - ~1/3 embedded DSP market
  - 2007 cholesterol lowering Pzifer Lipitor sales: $13B
- DSP proc. market 2007

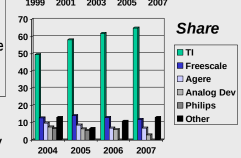


Source: Forward Concepts

- DSP proc. benchmarking
  - Berkeley Design Technology Inc. http://www.bdti.com

DSP Processor Market





Source: Forward Concepts

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 PRESPECTIVES: 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

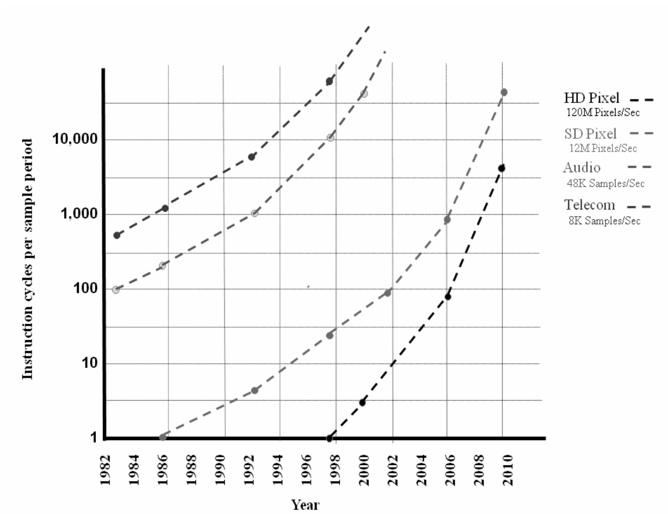


Fig. 1. Four examples of the increase of instruction cycles per sample period. It appears that the DSP becomes useful when it can perform a minimum of 100 instructions per sample period. Note that for a video system the pixel is used in place of a sample.

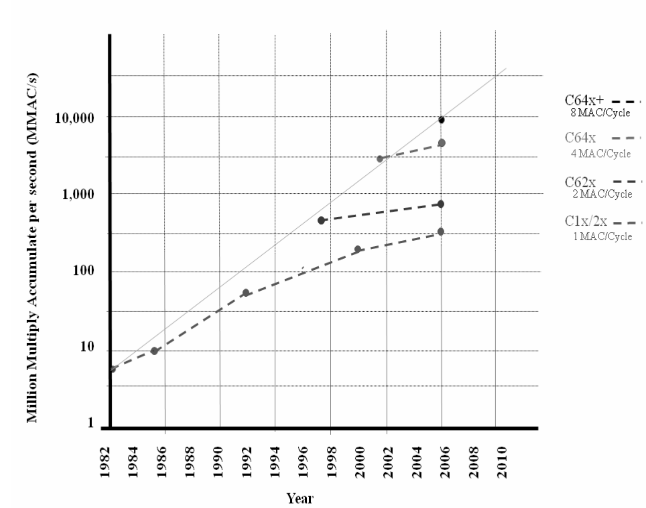


Fig. 2. Four generations of DSPs show how multi-processing has more effect on performance than clock rate. The dotted lines correspond to the increase in performance due to clock increases within an architecture. The solid line shows the increase due to both the clock increase and the parallel processing.

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 lead 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 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.

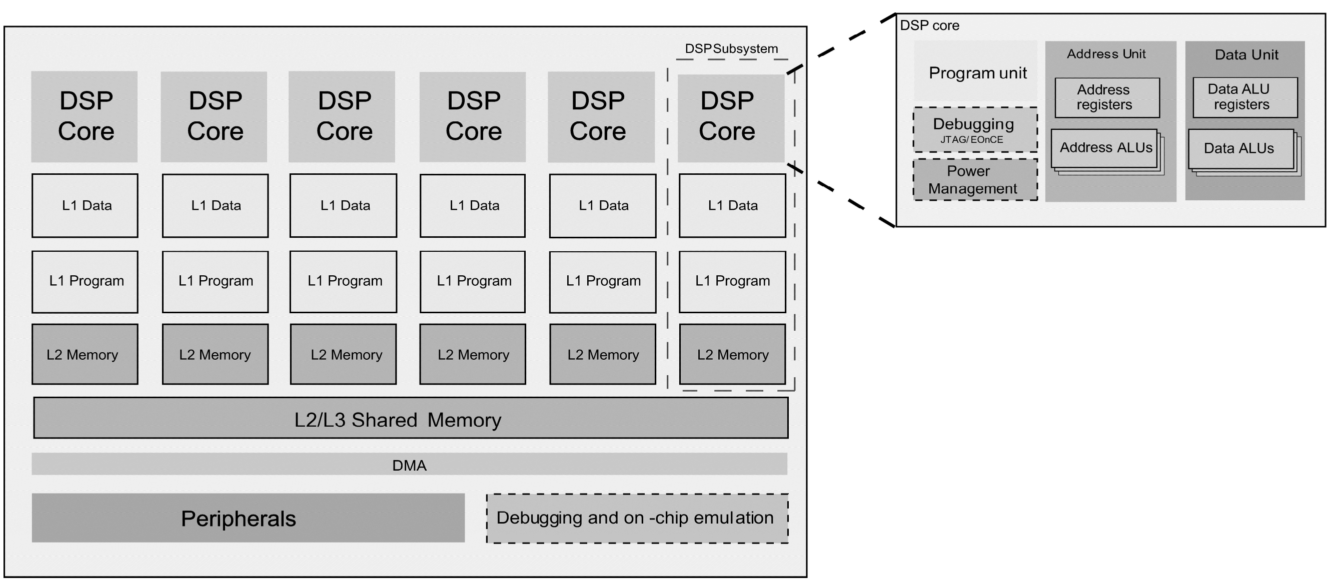


Fig.3. Typical multi-core DSP platform.

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

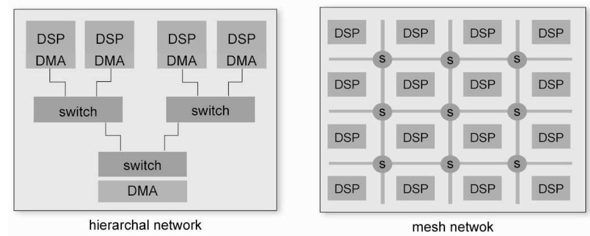


Fig.4. Interconnect types of multi-core DSP architectures.

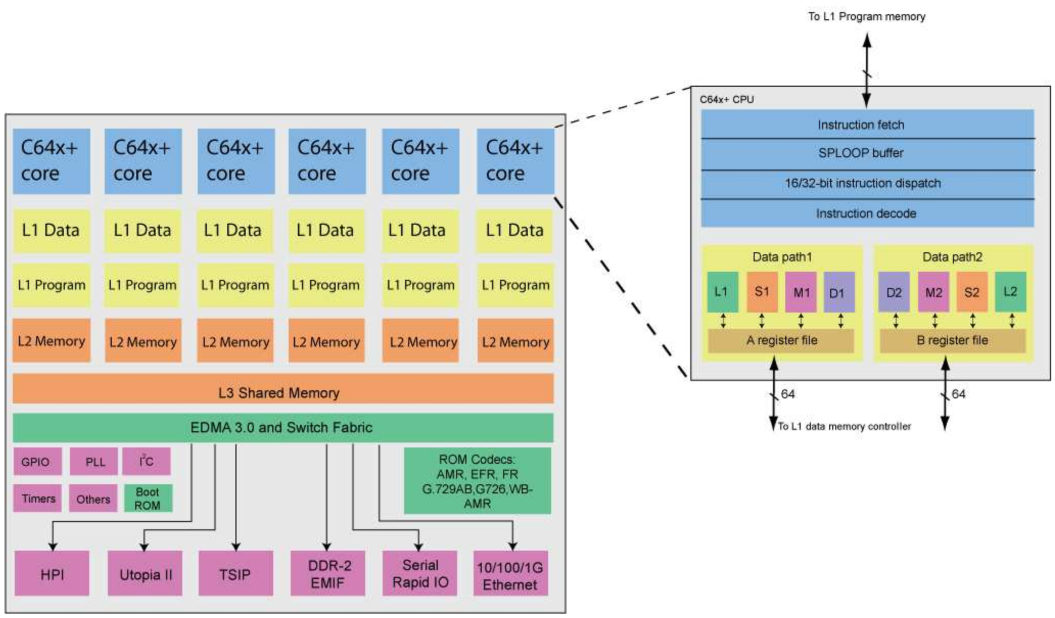


Fig.5. Texas Instruments TNETV3020 multi-core DSP processor.

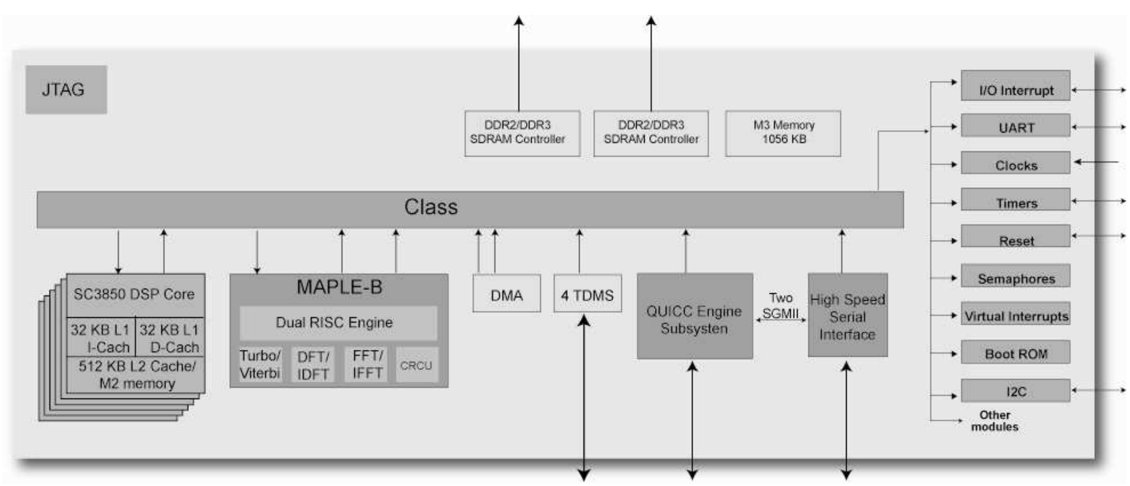


Fig.6. Freescale 8156 multi-core DSP processor.

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 hierarchal-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.

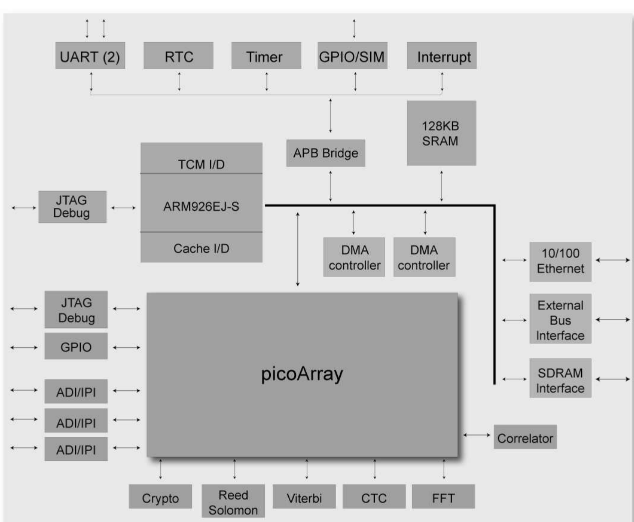


Fig.7. picoChip PC205 multi-core DSP processor.

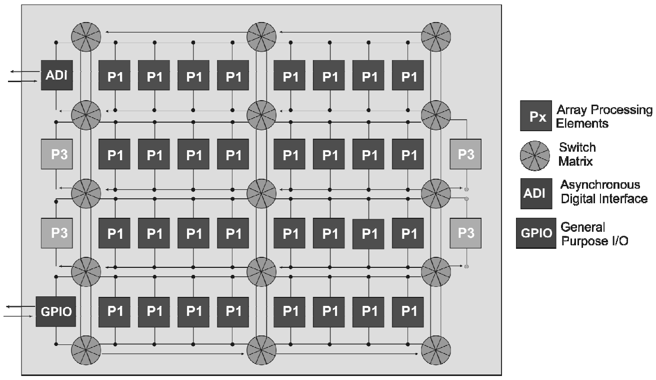


Fig. 8. picoChip picoArray.

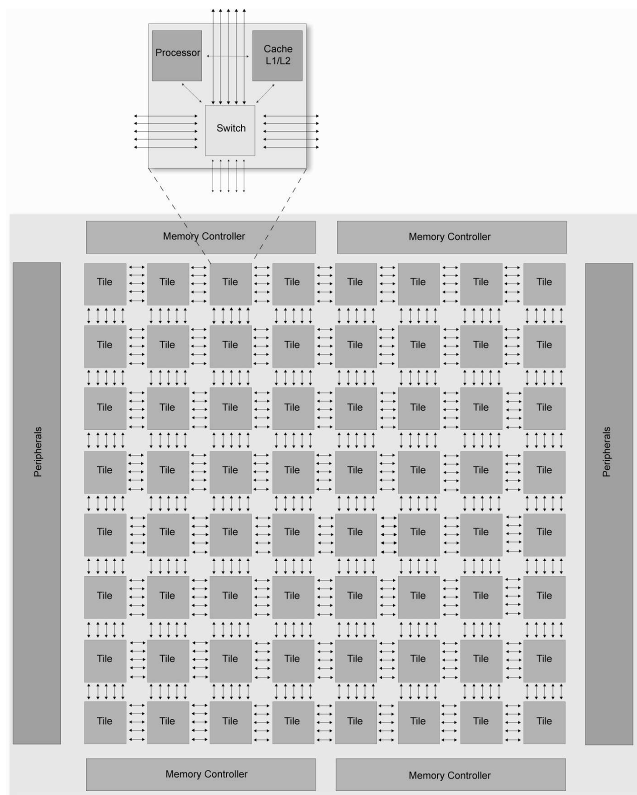


Fig. 9. Tilera TILE64 multi-core DSP processor.

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].

	Clock (MHz)	*DSP cores*	*OFDM channels*
TI TMS320C6455	1200	1	Lowest
Freescale MSC8144	1000	4	Low
Sandbridge SB3500	500	3	Medium
picoChip PC102	160	344	High
Tilera TILE64	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 Tilera 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].

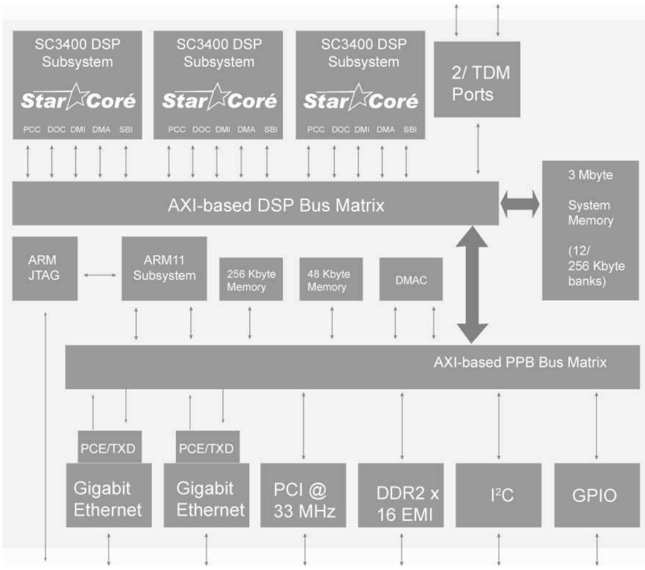


Fig.10. The Agere SP2603.

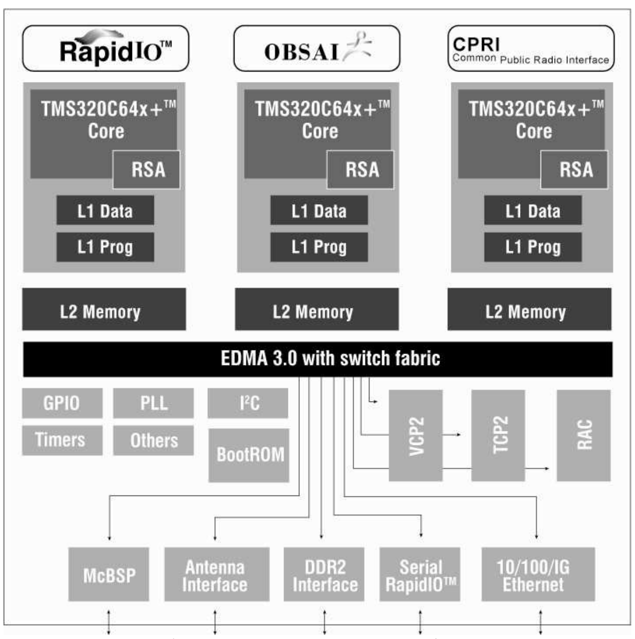


Fig. 11. Texas Instruments TCI6487.

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 Tilera architecture at a superficial glance, but also takes the mesh network principal to its logical conclusion, with very small cores ($0.17 mm^2$) 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 Tilera 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.

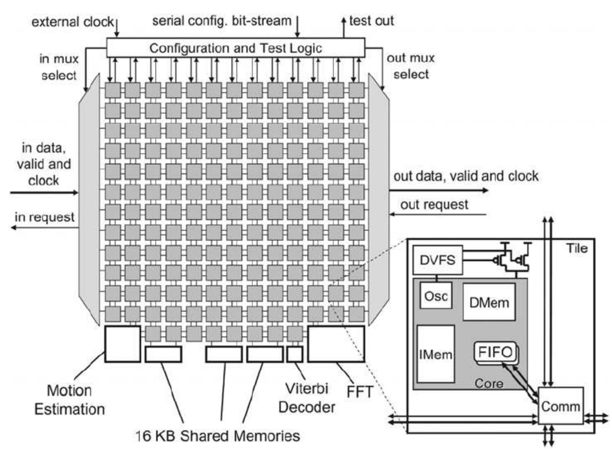


Fig.12. The AsAP processor architecture.

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 (SympoTIC)*, pp.ii-v, June 2006.

[13] J. Glossner, M. Moudgill, D. Iancu, G. Nacer, S. Jintukar, 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/ppsmp.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/

[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-kelm-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 pathss.

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, Tthe 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.

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.

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.

EE 445S Real-Time Digital Signal Processing Lab *Spring 2014*

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/rtdsp*

Outline

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

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**

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

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

Review

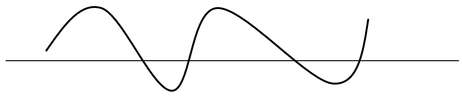
Continuous-Time vs. Discrete-Time

- **Continuous-time signals can be modeled as functions of a real argument**
  - $x(t)$ where time, t, can take any real value
  - $x(t)$ may be 0 for a given range of values of t
- **Discrete-time signals can be modeled as functions of argument that takes values from a discrete set**
  - $x[n]$ where $n \in$ {...-3,-2,-1,0,1,2,3...}
  - Integer time index, e.g. n, for discrete-time systems
- **Values for x may be real-valued or complex-valued**

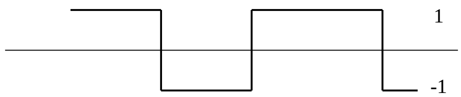
Review

Analog vs. Digital

- **Amplitude of analog signal can take any real or complex value at each time/sample**



- **Amplitude of digital signal takes values from a discrete set**



Review

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 $\int_{-\infty}^{\infty} g(t)\delta(t)\, dt = g(0)$

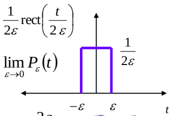
provided $g(t)$ *is* ***defined*** *at* $t = 0$

Scaling: $\int_{-\infty}^{\infty} \delta(at)\, dt = \frac{1}{|a|}$ if $a \neq 0$

- **We will leave $\delta(0)$ *undefined***

$P_\varepsilon(t) = \frac{1}{2\varepsilon}\mathrm{rect}\left(\frac{t}{2\varepsilon}\right)$

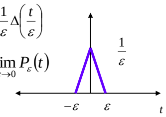
$\delta(t) = \lim_{\varepsilon \to 0} P_\varepsilon(t)$



Area $= \lim_{\varepsilon \to 0} \frac{2\varepsilon}{2\varepsilon} = 1$ **Unit Area**

$P_\varepsilon(t) = \frac{1}{\varepsilon}\Delta\left(\frac{t}{\varepsilon}\right)$

$\delta(t) = \lim_{\varepsilon \to 0} P_\varepsilon(t)$



Area $= \lim_{\varepsilon \to 0} \frac{\varepsilon}{\varepsilon} = 1$ **Unit Area**

Unit Impulse

- **We will leave $\delta(0)$ *undefined***

Some signals and systems textbooks assign $\delta(0) = \infty$

- **Plot Dirac delta as arrow at origin**

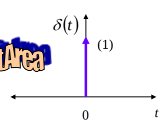
Undefined amplitude at origin

Denote area at origin as (*area*)

Height of arrow is irrelevant

Direction of arrow indicates sign of area

Unit Area



- **With $\delta(t) = 0$ for $t \neq 0$, it is tempting to think**

No!

$\phi(t)\,\delta(t) = \phi(0)\,\delta(t)$

$\phi(t)\,\delta(t-T) = \phi(T)\,\delta(t-T)$

Simplify unit impulse under integration only

Review

Unit Impulse

- **Simplifying $\delta(t)$ under integration**

$\int_{-\infty}^{\infty} \phi(t)\delta(t)dt = \phi(0)$

Assuming $\phi(t)$ is defined at t=0

- **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,

$\int_{-\infty}^{\infty} \phi(t+T)\delta(t)dt = \phi(T)$

- **Other examples**

$\int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1$

$\int_{-\infty}^{\infty} \delta(t-2)\cos\left(\frac{\pi t}{4}\right) dt = 0$

$\int_{-\infty}^{\infty} e^{-2(x-t)}\,\delta(2-t)\, dt = e^{-2(x-2)}$

- **What about at origin?**

$\int_{-\infty}^{0} \delta(t)\, dt = ?$

$\int_{-\infty}^{0^-} \delta(t)\, dt = 0$ **Before Impulse**

$\int_{-\infty}^{0^+} \delta(t)\, dt = 1$ **After Impulse**

Unit Impulse

- **Relationship between unit impulse and unit step**

$$\int_{-\infty}^{t} \delta(\tau)\, d\tau = \begin{cases} 0 & t < 0 \\ ? & t = 0 \\ 1 & t > 0 \end{cases} \iff \frac{du}{dt} = \delta(t)$$

$$= u(t)$$

- **What happens at the origin for $u(t)$?**

  $u(0^-) = 0$ and $u(0^+) = 1$, but $u(0)$ can take any value

  Common values for $u(0)$ are 0, ½, and 1

  $u(0) = ½$ is used in impulse invariance filter design:

  L. B. Jackson, "A correction to impulse invariance," *IEEE Signal Processing Letters*, vol. 7, no. 10, Oct. 2000, pp. 273-275.

3 - 9

Review

Systems

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

$$x(t) \rightarrow \boxed{T\{\bullet\}} \rightarrow y(t) \qquad x[n] \rightarrow \boxed{T\{\bullet\}} \rightarrow y[n]$$

$$y(t) = T\{\, x(t) \,\} \qquad y[n] = T\{\, x[n] \,\}$$

- **Continuous-time system examples**

  $y(t) = ½\, x(t) + ½\, x(t-1)$

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

  Squaring function can be used in sinusoidal demodulation

- **Discrete-time system examples**

  $y[n] = ½\, x[n] + ½\, x[n-1]$

  $y[n] = x^2[n]$

  Average of current input and delayed input is a simple filter

3 - 10

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**

  Quick test to identify some nonlinear systems?

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

3 - 11

Role of Initial Conditions

- **Observe a system starting at time t_0**

  Often use $t_0 = 0$ without loss of generality
- **Integrator**

$$x(t) \rightarrow \boxed{\int_{-\infty}^{t} (\bullet)\, dt} \rightarrow y(t) \qquad y(t) = \int_{-\infty}^{t} x(u)\, du = \int_{-\infty}^{t_0} x(u)\, du + \int_{t_0}^{t} x(u)\, du$$

- **Integrator observed for $t \geq t_0$**

$$x(t) \rightarrow \boxed{\int_{t_0}^{t} (\bullet)\, dt + C_0} \rightarrow y(t) \qquad C_0 = \int_{-\infty}^{t_0} x(u)\, du$$

  C_0 is due to initial conditions

  Linear system if initial conditions are zero ($C_0 = 0$)

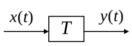
  Time-invariant system if initial conditions are zero ($C_0 = 0$)

3 - 12

Review

Continuous-Time System Properties

- **Ideal delay by *T* seconds. *Linear*?**

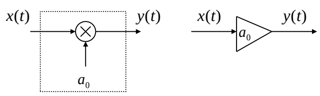


$y(t) = x(t - T)$

Role of initial conditions?

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

  Two different ways to express it in a block diagram

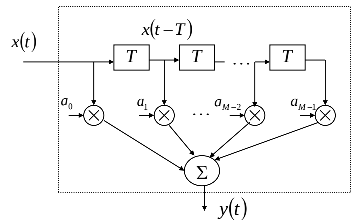


$y(t) = a_0 \, x(t)$

Linear? *Time-invariant*?

Continuous-Time System Properties

- **Tapped delay line**



Each *T* represents a delay of *T* time units

There are *M-1* delays

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

Coefficients (or taps) are a_0, a_1, $\ldots a_{M-1}$

Linear? *Time-invariant*?

Role of initial conditions?

Continuous-Time System Properties

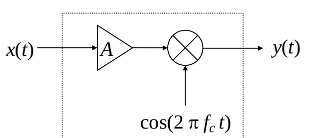
- **Amplitude Modulation (AM)**

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

  f_c is the carrier frequency (frequency of radio station)

  A is a constant

  Linear? *Time-invariant*?



- **AM modulation is AM radio if** $x(t) = 1 + k_a\, m(t)$ **where** $m(t)$ **is message (audio) to be broadcast and** $|\, k_a\, m(t)\, | < 1$ **(see lecture 19 for more info)**

Review

Generating Discrete-Time Signals

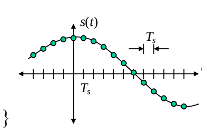
- **Many signals originate in continuous time**

  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 $n \in \{\ldots, -1, 0, 1, \ldots\}$

  How to choose sampling period T_s?

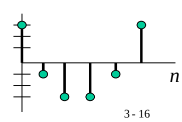


Sampled analog waveform

- **Using a formula**

  $x[n] = n^2 - 5n + 3$ on right for $0 \le n \le 5$

  How does $x[n]$ look in continuous time?



Review

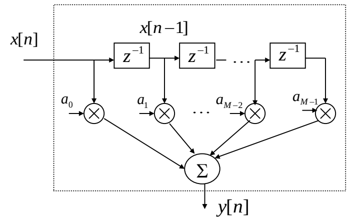
Discrete-Time System Properties

- **Let $x[n]$, $x_1[n]$ and $x_2[n]$ be inputs to a linear system**
- **Let $y[n]$, $y_1[n]$ and $y_2[n]$ 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?**

Discrete-Time System Properties

- **Tapped delay line in discrete time**

See also slide 5-4



Each z^{-1} represents a delay of 1 sample

There are *M-1* delays

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

Coefficients (or taps) are a_0, a_1, $\ldots a_{M-1}$

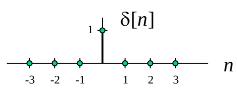
- ***Linear? Time-invariant?***

Role of initial conditions?

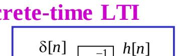
Discrete-Time System Properties

- **Let $\delta[n]$ be a discrete-time impulse function, a.k.a. Kronecker delta function:**

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$



- **Impulse response is response of discrete-time LTI system to discrete impulse function**
  - Example: delay by one sample

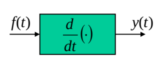


$h[n] = \delta[n-1]$

- **Finite impulse response filter**
  - Non-zero extent of impulse response is finite
  - Can be in continuous time or discrete time
  - Also called tapped delay line (slides 3-14, 3-18, 5-4)

Discrete-Time System Properties

- **Continuous time**



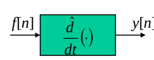
$$y(t) = \frac{d}{dt}\{f(t)\}$$

$$= \lim_{\Delta t \to 0} \frac{f(t) - f(t - \Delta t)}{\Delta t}$$

Linear?

Time-invariant?

- **Discrete time**



$$y[n] = y(nT_s) = \frac{d}{dt}\{f(t)\}\Big|_{t = nT_s}$$

$$= \lim_{T_s \to 0} \frac{f(nT_s) - f(nT_s - T_s)}{T_s}$$

$$= f[n] - f[n-1]$$

See also slide 5-18

Linear?

Time-invariant?

Conclusion

- **Continuous-time versus discrete-time:**
  discrete means quantized in time
- **Analog versus digital:**
  digital means quantized in amplitude
- **Digital signal processor**
  Discrete-time and digital system
  Well-suited for implementing LTI digital filters
- **Example of discrete-time analog system?**
- **Example of continuous-time digital system?**

EE 445S Real-Time Digital Signal Processing Lab *Spring 2014*

Sampling and Aliasing

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

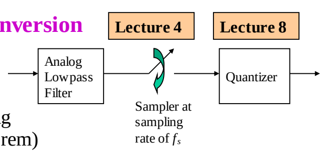
Lecture 4 *http://www.ece.utexas.edu/~bevans/courses/rtdsp*

Outline

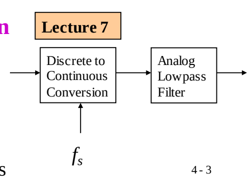
- **Data conversion**
- **Sampling**
  Time and frequency domains
  Sampling theorem
- **Aliasing**
- **Bandpass sampling**
- **Rolling shutter artifacts**
- **Conclusion**

Data Conversion

- **Analog-to-Digital Conversion**
  Lowpass filter has stopband frequency less than ½ f_s to reduce aliasing due to sampling (enforce sampling theorem)



- **Digital-to-Analog Conversion**
  Discrete-to-continuous conversion could be as simple as sample and hold
  Lowpass filter has stopband frequency less than ½ f_s reduce artificial high frequencies



Sampling: Time Domain

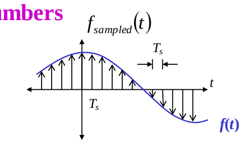
- **Many signals originate in continuous-time**
  Talking on cell phone, or playing acoustic music
- **By sampling a continuous-time signal at isolated, equally-spaced points in time, we obtain a sequence of numbers**

$f[n] = f(n\,T_s)$

$n \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

T_s is the sampling period.

$$f_{sampled}(t) = f(t) \underbrace{\sum_{n=-\infty}^{\infty} \delta(t - n\,T_s)}_{\text{impulse train}}$$



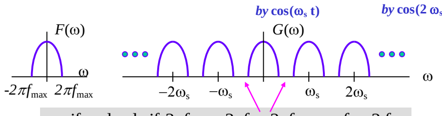
Sampled analog waveform

Sampling: Frequency Domain

- **Sampling replicates spectrum of continuous-time signal at integer multiples of sampling frequency**
- **Fourier series of impulse train where** $\omega_s = 2\,\pi\,f_s$

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

$$g(t) = f(t)\,\delta_{T_s}(t) = \frac{1}{T_s}\left(f(t) + \underbrace{2\,f(t)\cos(\omega_s\,t)}_{\text{Modulation by }\cos(\omega_s\,t)} + \underbrace{2\,f(t)\cos(2\,\omega_s\,t)}_{\text{Modulation by }\cos(2\,\omega_s\,t)} + \ldots\right)$$



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

How to recover $F(\omega)$?

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 = $f_s / 2$

What happens if $f_s = 2\,f_{max}$?

- ***Example*: Sampling audio signals**
  - Normal human hearing is from about 20 Hz to 20 kHz
  - Apply lowpass filter before sampling to pass low frequencies up to 20 kHz and reject high frequencies
  - Lowpass filter needs 10% of maximum passband frequency to roll off to zero (2 kHz rolloff in this case)

Sampling Theorem

<u>Assumption</u>

- **Continuous-time signal has absolutely no frequency content above f_{max}**
- **Sampling time is exactly the same between any two samples**
- **Sequence of numbers obtained by sampling is represented in exact precision**
- **Conversion of sequence to continuous time is ideal**

<u>In Practice</u>

Sampling and Oversampling

- **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 time-domain waveform shape
- ***DSP First*, Ch. 4, Sampling/Interpolation demo** link
  - For username/password help link

Aliasing

- **Continuous-time sinusoid**

  $x(t) = A \cos(2\pi f_0 t + \phi)$
- **Sample at** $T_s = 1/f_s$

  $x[n] = x(T_s n) = A \cos(2\pi f_0 T_s n + \phi)$
- **Keeping the sampling period same, sample**

  $y(t) = A \cos(2\pi (f_0 + l f_s) t + \phi)$

  where l **is an integer**

$$
\begin{aligned}
y[n] &= y(T_s n) \\
&= A \cos(2\pi(f_0 + l f_s)T_s n + \phi) \\
&= A \cos(2\pi f_0 T_s n + 2\pi l f_s T_s n + \phi) \\
&= A \cos(2\pi f_0 T_s n + 2\pi l n + \phi) \\
&= A \cos(2\pi f_0 T_s n + \phi) \\
&= x[n]
\end{aligned}
$$

Here, $f_s T_s = 1$

Since l is an integer, $\cos(x + 2\pi l) = \cos(x)$

- $y[n]$ **indistinguishable from** $x[n]$

Aliasing

- **Since** l **is any integer, a countable but infinite number of sinusoids give same sampled sequence**
- **Frequencies** $f_0 + l f_s$ **for** $l \neq 0$

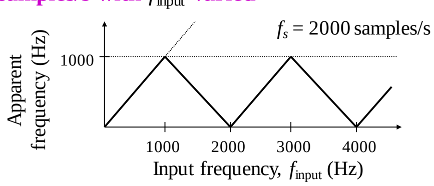
  Called aliases of frequency f_0 with respect to f_s

  All aliased frequencies appear same as f_0 due to sampling
- ***Signal Processing First*, Continuous to Discrete Sampling demo (con2dis)**

link

Aliasing

- **Sinusoid** $\sin(2\pi f_{input} t)$ **sampled at** $f_s = 2000$ **samples/s with** f_{input} **varied**



- **Mirror image effect about** $f_{input} = ½ f_s$ **gives rise to name of folding**

Bandpass Sampling

- **Reduce sampling rate**

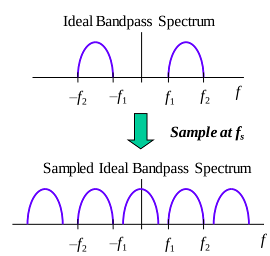
  Bandwidth: $f_2 - f_1$

  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 $f_{center} = ½(f_1 + f_2) = k f_s$
- **Practical issues**

  Sampling clock tolerance: $f_{center} = k f_s$

  Effects of noise



Lowpass filter to extract baseband

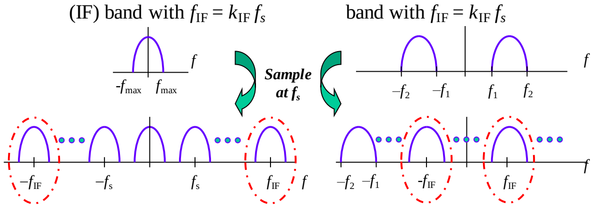
Sampling for Up/Downconversion

- **Upconversion method**

  Sampling plus bandpass filtering to extract intermediate frequency (IF) band with $f_{IF} = k_{IF} f_s$

- **Downconversion method**

  Bandpass sampling plus bandpass filtering to extract intermediate frequency (IF) band with $f_{IF} = k_{IF} f_s$



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

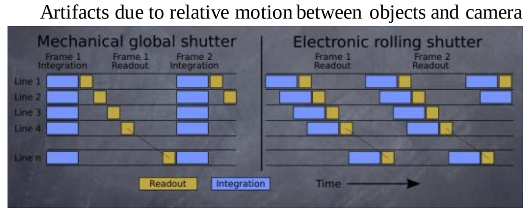


Figure from tutorial by Forssen *et al.* at 2012 *IEEE Conf. on Computer Vision & Pattern Recognition*

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

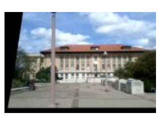
- **Fast camera motion**

  Pan camera fast left/right

  Pole wobbles and bends

  Building skewed

Warped frame



Compensated using gyroscope readings (i.e. camera rotation) and video features

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.

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**

EE445S Real-Time Digital Signal Processing Lab   Spring 2014

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/rtdsp*

Outline

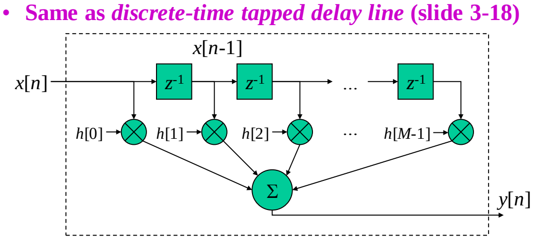
- **Many Roles for Filters**
- **Convolution**
- ***Z*-transforms**
- **Linear time-invariant systems**
  - Transfer functions
  - Frequency responses
- **Finite impulse response filters**
  - Cascading FIR filters demonstration
  - Symmetric FIR filters
  - Filter design

Many Roles for Filters

- **Noise removal**
  - Signal and noise spectrally separated
  - Example: bandpass filtering to suppress out-of-band noise
- **Analysis, synthesis, and compression**
  - Spectral analysis
  - Examples: calculating power spectra (slides 14-10 and 14-11) and polyphase filter banks for pulse shaping (lecture 13)
- **Spectral shaping**
  - Data conversion (lectures 10 and 11)
  - Channel equalization (slides 16-8 to 16-10)
  - Symbol timing recovery (slides 13-17 to 13-20 and slide 16-7)
  - Carrier frequency and phase recovery

Finite Impulse Response (FIR) Filter

- **Same as *discrete-time tapped delay line* (slide 3-18)**



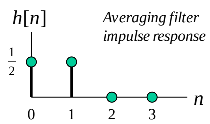
- **Impulse response** $h[n]$ **has finite extent** $n = 0, \ldots, M-1$

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

Discrete-time convolution

Discrete-time Convolution Derivation

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



Averaging filter impulse response

$y[n] = h[0]\,x[n] + h[1]\,x[n-1]$

$= (\,x[n] + x[n-1]\,)\,/\,2$

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**

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

  For each n, compute different (possibly) infinite summation
- **LTI system**

  Characterized uniquely by its impulse response

  Its output is convolution of input and impulse response

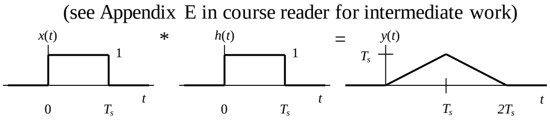
Convolution Demos

- **The Johns Hopkins University Demonstrations**

  http://www.jhu.edu/~signals

  Convolution applet to animate convolution of simple signals and hand-sketched signals

  Convolving two rectangular pulses of same width gives triangle with width of twice the width of rectangular pulses (see Appendix E in course reader for intermediate work)



What about convolving two pulses of different lengths?

Z-transform Definition

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

  Bilateral Forward *z*-transform $\qquad$ Bilateral Inverse *z*-transform

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

  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**

Three Common *Z*-transform Pairs

- ***h*[*n*] = *δ*[*n*]**

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

Region of convergence: entire *z*-plane

- ***h*[*n*] = *δ*[*n-1*]**

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

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

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

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

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

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

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

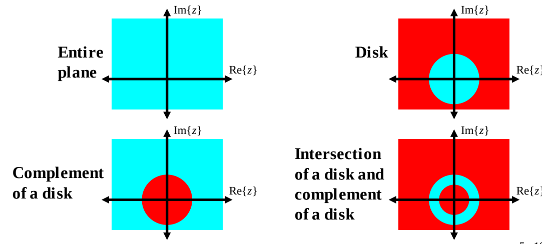
Region of convergence for summation: $|z| > |a|$

$|z| > |a|$ is the complement of a disk

Finite extent sequences | **Infinite extent sequence**

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)**



System Transfer Function

- ***Z*-transform of system's impulse response**

Impulse response uniquely represents an LTI system

- **Example: FIR filter with *M* taps (*slide 5-4*)**

$$H(z)=\sum_{n=-\infty}^{\infty}h[n]\,z^{-n}=\sum_{n=0}^{M-1}h[n]\,z^{-n}=h[0]+h[1]\,z^{-1}+\ \ldots\ +h[M-1]\,z^{-(M-1)}$$

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**

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

Example: Ideal Delay

- **Continuous Time**

Delay by *T* seconds

$x(t) \rightarrow \boxed{T} \rightarrow y(t)$

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

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**

Delay by 1 sample

$x[n] \rightarrow \boxed{z^{-1}} \rightarrow y[n]$

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

Impulse response

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

Frequency response

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

$|H(\omega)|=1$

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

Zero Initial Conditions

$\omega = \Omega T$

All pass Filter

Linear Phase

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 input signal and shift it in phase

  Example in continuous time: see handout F

  Example in discrete time. Let $x[n] = e^{j\omega n}$,

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

  $H(\omega)$ is discrete-time Fourier transform of $h[n]$
  $H(\omega)$ is also called the frequency response

Frequency Response

- **Continuous-time LTI system**

$$e^{j\Omega t} \longrightarrow \boxed{h(t)} \longrightarrow H(j\Omega)\, e^{j\Omega t}$$

$$\cos(\Omega t) \longrightarrow \boxed{h(t)} \longrightarrow |H(j\Omega)| \cos(\Omega t + \angle H(j\Omega))$$

- **Discrete-time LTI system**

$$e^{j\omega n} \longrightarrow \boxed{h[n]} \longrightarrow H(e^{j\omega})\, e^{j\omega n}$$

$$\cos(\omega n) \longrightarrow \boxed{h[n]} \longrightarrow |H(e^{j\omega})| \cos(\omega n + \angle H(e^{j\omega}))$$

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

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

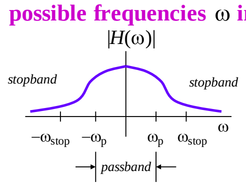
  Output $\quad H(e^{-j\omega})e^{-j\omega n} + H(e^{j\omega})e^{j\omega n} = H^*(e^{j\omega})e^{-j\omega n} + H(e^{j\omega})e^{j\omega n} =$

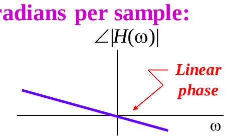
$$|H(e^{j\omega})|\, e^{-j\angle H(e^{j\omega})}\, e^{-j\omega n} + |H(e^{j\omega})|\, e^{j\angle H(e^{j\omega})}\, e^{j\omega n} =$$

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

Frequency Response

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





$$delay(\omega) = -\frac{d}{d\omega}\angle H(\omega) = k_{\text{delay}}$$

  Lowpass filter: passes low and attenuates high frequencies

  Linear phase: must be FIR filter with impulse response that is symmetric or anti-symmetric about its midpoint

- **Not all FIR filters exhibit linear phase**

Filter Design

- **Specify a desired piecewise constant magnitude response**
- **Lowpass filter example**

  $\omega \in [0, \omega_p]$, mag $\in [1-\delta_p, 1]$

  $\omega \in [\omega_s, \pi]$, mag $\in [0, \delta_s]$

  Transition band unspecified
- **Symmetric FIR filter design methods**

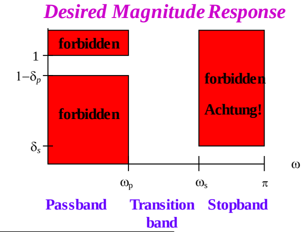
  Windowing

  Least squares

  Remez (Parks-McClellan)

Lowpass Filter Example

Desired Magnitude Response



δ_p **passband ripple**
δ_s **stopband ripple**

Red region is forbidden

Example: Two-Tap Averaging Filter

- **Input-output relationship**

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

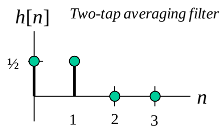
  $h[n]=\frac{1}{2}\delta[n]+\frac{1}{2}\delta[n-1]$
- **Frequency response**

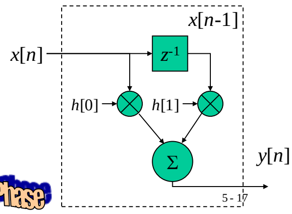
  $H(\omega)=\frac{1}{2}+\frac{1}{2}e^{-j\omega}$

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

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

Linear Phase





5 - 17

Example: First-Order Difference

- **Input-output relationship**

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

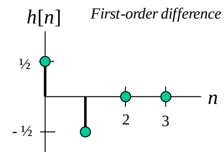
  $h[n]=\frac{1}{2}\delta[n]-\frac{1}{2}\delta[n-1]$
- **Frequency response**

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

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

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

Generalized Linear Phase



5 - 18

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

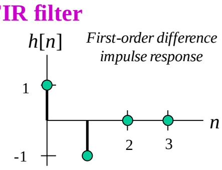
  Lowpass filter (smooth/blur input signal)

  Impulse response is {1, 1, 1, 1, 1}
- **First-order difference FIR filter**

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

  Highpass filter (sharpens input signal)

  Impulse response is {1, -1}



5 - 19

Cascading FIR Filters Demo

- ***DSP First*, Ch. 6, Freq. Response of FIR Filters**

  http://www.ece.gatech.edu/research/DSP/DSPFirstCD/visible/chapters/6firfreq/demos/blockd/index.htm

  For username/password help
- **From lowpass filter to highpass filter**

  original image $\rightarrow$ blurred image $\rightarrow$ sharpened/blurred image
- **From highpass to lowpass filter**

  original image $\rightarrow$ sharpened image $\rightarrow$ 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***

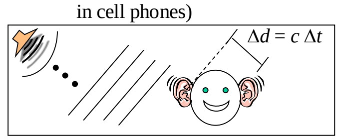
5 - 20

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
  - Difference filter subtracts two numbers to create output pixel
- **Full output precision is 16 bits per pixel**
  - Demonstration uses double-precision floating-point data and arithmetic (53 bits of mantissa + sign; 11 bits for exponent)
  - No output precision was harmed in the making of this demo ☺

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)

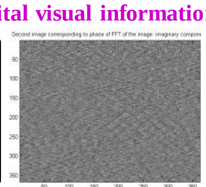


- **Linear phase crucial**
  - Audio
  - Images
  - Communication systems
- **Linear phase response**
  - Need FIR filters
  - Realizable IIR filters cannot achieve linear phase response over all frequencies

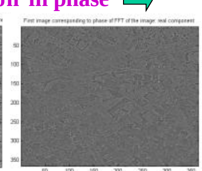
Importance of Linear Phase

- **For images, vital visual information in phase** code

Take FFT of image
Set phase to zero
Take inverse FFT



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.ece.utexas.edu/~bevans/courses/rtdsp/lectures/05_FIR_Filters/Original%20Image.tif

Finite Impulse Response Filters

- **Duration of impulse response $h[n]$ is finite, i.e. zero-valued for n outside interval $[0, M\text{-}1]$:**

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

  - Output depends on current input and previous M-1 inputs
  - Summation to compute $y[k]$ reduces to a vector dot product between M input samples in the vector

$$\{\, x[n], x[n-1], \ldots, x[n-(M-1)] \,\}$$

  and M values of the impulse response in vector

$$\{\, h[0], h[1], \ldots, h[M-1] \,\}$$

- ***What instruction set architecture features would you add to accelerate FIR filtering?***

Infinite Impulse Response Filters

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

Lecture 6

Outline

- **Many roles for filters**
- **Two IIR filter structures**
  - Biquad structure
  - Direct form implementations
- **Stability**
- **Z and Laplace transforms**
- **Cascade of biquads**
  - Analog and digital IIR filters
  - Quality factors
- **Conclusion**

Many Roles for Filters

- **Noise removal**
  - Signal and noise spectrally separated
  - Example: bandpass filtering to suppress out-of-band noise
- **Analysis, synthesis, and compression**
  - Spectral analysis
  - Examples: calculating power spectra (slides 14-10 and 14-11) and polyphase filter banks for pulse shaping (lecture 13)
- **Spectral shaping**
  - Data conversion (lectures 10 and 11)
  - Channel equalization (slides 16-8 to 16-10)
  - Symbol timing recovery (slides 13-17 to 13-20 and slide 16-7)
  - Carrier frequency and phase recovery

Digital IIR Filters

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

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

- **How to implement the IIR filter by computer?**
  - Let $x[k]$ be the input signal and $y[k]$ the output signal,

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

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

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

Different Filter Representations

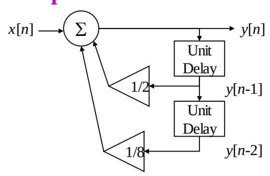
- **Difference equation**

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

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

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

- **Block diagram representation**



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

- **Transfer function**

  Assumes LTI system

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

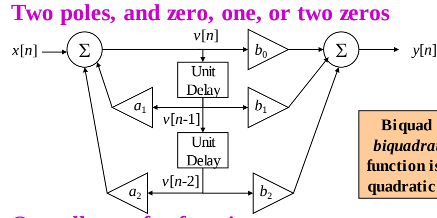
$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2} z^{-1} - \frac{1}{8} z^{-2}}$$

Poles at –0.183 and +0.683

6-5

Discrete-Time IIR Biquad

- **Two poles, and zero, one, or two zeros**



Biquad is short for *biquadratic* – transfer function is ratio of two quadratic polynomials

- **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

Discrete-Time IIR Filter Design

- **Biquad w/ zeros z_0 and z_1 and poles p_0 and p_1**

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

  Magnitude response

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

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

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

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

- **When poles and zeros are separated in angle**

  Poles near unit circle indicate filter's passband(s)

  Zeros on/near unit circle indicate stopband(s)

6-7

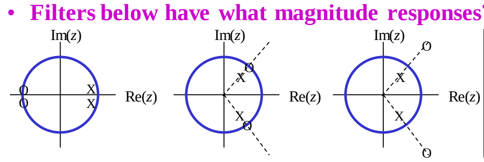
Discrete-Time IIR Biquad Examples

- **Transfer function** $H(z) = C \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)} = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$
- **When transfer function coefficients are real-valued**

  Poles (X) are conjugate symmetric or real-valued

  Zeros (O) are conjugate symmetric or real-valued

- **Filters below have what magnitude responses?**



lowpass highpass bandpass bandstop allpass notch?

Zeros are on the unit circle

Poles have radius r
Zeros have radius $1/r$

6-8

A Direct Form IIR Realization

- **IIR filters having rational transfer functions**

$$H(z)=\frac{Y(z)}{X(z)}=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+\ldots+b_Nz^{-N}}{1-a_1z^{-1}-\ldots-a_Mz^{-M}} \Rightarrow Y(z)\left(1-\sum_{m=1}^{M}a_m\,z^{-m}\right)=X(z)\sum_{k=0}^{N}b_k\,z^{-k}$$

- **Direct form realization**

$$y[n]=\sum_{m=1}^{M}a_m\,y[n-m]+\sum_{k=0}^{N}b_k\,x[n-k]$$

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

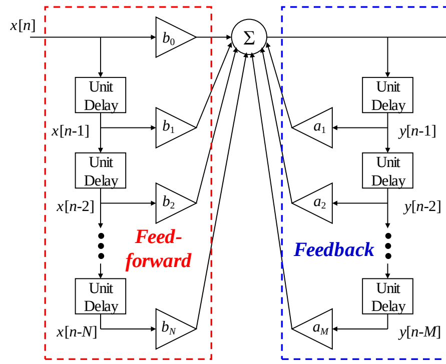
Dot product of vector of M coefficients and vector of previous M outputs (“*FIR” filtering of previous output values*)

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

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

6 - 9

Filter Structure As a Block Diagram



$$y[n]=\sum_{k=0}^{N}b_k\,x[n-k]+\sum_{m=1}^{M}a_m\,y[n-m]$$

Full Precision
Wordlength of $y[0]$ is 2 words. Wordlength of $y[n]$ increases with n for $n > 0$.

M and N may be different

Yet Another Direct Form IIR

- **Rearrange transfer function to be cascade of an all-pole IIR filter followed by an FIR filter**

$$Y(z)=\frac{X(z)B(z)}{A(z)}=V(z)B(z) \text{ where } V(z)=\frac{X(z)}{A(z)}$$

Here, $v[n]$ is the output of an all-pole filter applied to $x[n]$:

$$v[n]=x[n]+\sum_{m=1}^{M}a_m\,v[n-m]$$

$$y[n]=\sum_{k=0}^{N}b_k\,v[n-k]$$

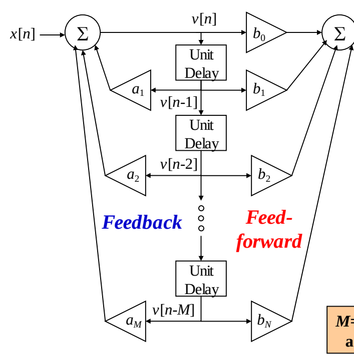
- **Implementation complexity (assuming $M \geq N$)**

Computation: $M + N + 1$ MACs

Memory: M double words for past values of $v[n]$ and $M + N + 1$ words for coefficients

6 - 11

Filter Structure As Block Diagram



$$v[n]=x[n]+\sum_{m=1}^{M}a_m\,v[n-m]$$

$$y[n]=\sum_{k=0}^{N}b_k\,v[n-k]$$

Full Precision
Wordlength of $y[0]$ is 2 words and wordlength of $v[0]$ is 1 word. Wordlength of $v[n]$ and $y[n]$ increases with n.

M=2 yields a biquad

M and N may be different

Demonstrations

- ***Signal Processing First*, PEZ Pole Zero Plotter (pezdemo)** link
- ***DSP First* demonstrations, Chapter 8** link

  IIR Filtering Tutorial (Link)

  Connection Betweeen the Z and Frequency Domains (Link)

  Time/Frequency/Z Domain Movies for IIR Filters (Link)

  For username/password help link

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

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:** $a^n\, u[n] \overset{z}{\leftrightarrow} \dfrac{1}{1 - a\, z^{-1}}$ for $|z| > |a|$ pole at z=a

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

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

Z and Laplace Transforms

- **Transform difference/differential equations into algebraic equations that are easier to solve**
- **Are complex-valued functions of a complex frequency variable**

  Laplace: $s = \sigma + j\, 2\, \pi\, f$

  Z: $z = r\, e^{\,j\,\omega}$

- **Transform kernels are complex exponentials: eigenfunctions of linear time-invariant systems**

  Laplace: $e^{-s\,t} = e^{-\sigma\, t \,-\, j\, 2\, \pi\, f\, t} \;=\; e^{-\sigma\, t} \;\; e^{-j\, 2\, \pi\, f\, t}$

  Z: $z^{-n} = (r\, e^{\,j\,\omega})^{-n} \;=\; r^{-n} \;\; e^{-j\,\omega\, n}$

  dampening factor **oscillation term**

Z and Laplace Transforms

- **No unique mapping from Z to Laplace domain or from Laplace to Z domain**

  Mapping one complex domain to another is not unique

- **One possible mapping is impulse invariance**

  Make impulse response of a discrete-time linear time-invariant (LTI) system be a sampled version of the impulse response for the continuous-time LTI system

Z

$f[n] \rightarrow \boxed{H(z)} \rightarrow y[n]$

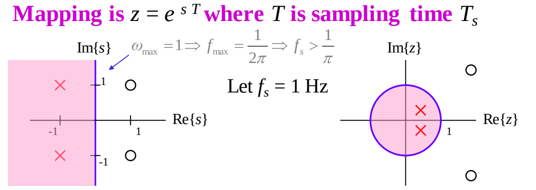
Laplace

$\tilde{f}(t) \rightarrow \boxed{H(s)} \rightarrow \tilde{y}(t)$

$$H(s) = H(z)\big|_{z=e^{sT}}$$

Impulse Invariance Mapping

- **Mapping is $z = e^{sT}$ where T is sampling time T_s**



$s = j\,2\pi\,f$

Poles: $s = -1 \pm j \Rightarrow z = 0.198 \pm j\,0.31$ (T = 1 s)

Zeros: $s = 1 \pm j \Rightarrow z = 1.469 \pm j\,2.287$ (T = 1 s)

Laplace Domain	*Z Domain*
Left-hand plane	Inside unit circle
Imaginary axis	Unit circle
Right-hand plane	Outside unit circle

lowpass, highpass bandpass, bandstop allpass or notch?

Continuous-Time IIR Biquad

- **Second-order filter section with 2 poles & 0-2 zeros**

  Transfer function is a ratio of two real-valued polynomials

  Poles and zeros occur in conjugate symmetric pairs

- ***Quality factor*: technology independent measure of sensitivity of pole locations to perturbations**

  For an analog biquad with poles at $a \pm j\,b$, where $a < 0$,

$$Q = \frac{\sqrt{a^2+b^2}}{-2a} \quad \text{where } \frac{1}{2} \le Q < \infty$$

  Real poles: $b = 0$ so $Q = ½$ (exponential decay response)

  Imaginary poles: $a = 0$ so $Q = \infty$ (oscillatory response)

Continuous-Time IIR Biquad

- **Impulse response with biquad with poles $a \pm j\,b$ with $a < 0$ but no zeroes:** $h(t) = C\,e^{a\,t}\cos(b\;t + \theta)$

  Pure sinusoid when $a = 0$ and pure decay when $b = 0$

- **Breadboard implementation**

  Consider a single pole at $-1/(R\,C)$. With 1% tolerance on breadboard R and C values, tolerance of pole location is 2%

  How many decimal digits correspond to 2% tolerance?

  How many bits correspond to 2% tolerance?

  Maximum quality factor is about 25 for implementation of analog filters using breadboard resistors and capacitors.

  Switched capacitor filters: $Q_{max} \approx 40$ (tolerance $\approx$ 0.2%)

  Integrated circuit implementations can achieve $Q_{max} \approx 80$

Discrete-Time IIR Biquad

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

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

  - Real poles: $b = 0$ and $-1 < a < 1$, so $r = |a|$ and $y = \pm 2\, a$ and $Q = ½$ (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 $Q = \frac{1}{2}\frac{1+r^2}{1-r^2} = \frac{1}{2}\frac{1+b^2}{1-b^2}$
  - 16-bit fixed-point digital signal processors with 40-bit accumulators: $Q_{max} \approx 40$

Filter design programs often use *r* as approximation of quality factor

IIR Filter Implementation

- **Same approach in discrete and continuous time**
- **Classical IIR filter designs**
  - Filter of order *n* will have *n*/2 conjugate roots if *n* is even or one real root and (*n*-1)/2 conjugate roots if *n* is odd
  - Response is very sensitive to perturbations in pole locations
- **Rule-of-thumb for implementing IIR filter**
  - Decompose IIR filter into second-order sections (biquads)
  - Cascade biquads from input to output in order of ascending *quality factors*
  - For each pair of conjugate symmetric poles in a biquad, conjugate zeroes should be chosen as those closest in Euclidean distance to the conjugate poles

Classical IIR Filter Design

- **Classical IIR filter designs differ in the shape of their magnitude responses**
  - Butterworth: monotonically decreases in passband and stopband (no ripple)
  - Chebyshev type I: monotonically decreases in passband but has ripples in the stopband
  - Chebyshev type II: has ripples in passband but monotonically decreases in the stopband
  - Elliptic: has ripples in passband and stopband
- **Classical IIR filters have poles and zeros, except**
  - Continuous-time lowpass Butterworth filters only have poles
- **Classical filters have biquads with high Q factors**

Analog IIR Filter Optimization

- **Start with an existing (e.g. classical) filter design**
- **IIR filter optimization packages from UT Austin (in Matlab) simultaneously optimize**
  - Magnitude response
  - Linear phase in passband
  - Peak overshoot in step response
  - Quality factors

Analog IIR Filter Optimization

- **Analog lowpass IIR filter design specification**

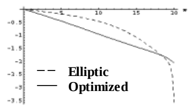
  δ_{pass}= 0.21 at ω_{pass}= 20 rad/s and δ_{stop}= 0.31 at ω_{stop}= 30 rad/s

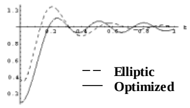
  Minimized deviation from linear phase in passband

  Minimized peak overshoot in step response

  Maximum quality factor per second-order section is 10

Linearized phase in passband





Minimized peak overshoot

elliptic

Q	poles	zeros
1.7	$-5.3533 \pm j16.9547$	$0.0 \pm j20.2479$
61.0	$-0.1636 \pm j19.9899$	$0.0 \pm j28.0184$

optimized

Q	poles	zeros
0.68	$-11.4343 \pm j10.5092$	$-3.4232 \pm j28.6856$
10.00	$-1.0926 \pm j21.8241$	$-1.2725 \pm j35.5476$

MATLAB Demos Using fdatool #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
- **Under analysis menu**

  Show magnitude response
- **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

MATLAB Demos Using fdatool #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

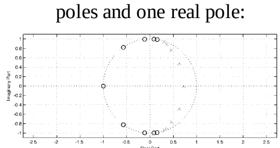
  – Poles near unit circle indicate passband

  – Two poles very close to unit circle
- **IIR filter – elliptic**

  Use second-order sections

  Increase filter order to 9

  Eight complex symmetric poles and one real pole:



Same observations on left

MATLAB Demos Using fdatool #3

- **IIR filter – elliptic**

  Use second-order sections

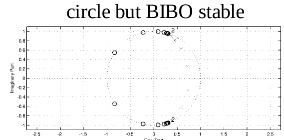
  Increase filter order to 20

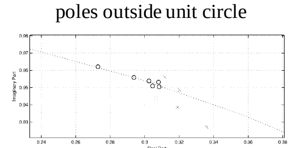
  Two poles very close to unit circle but BIBO stable

Use single section (Edit menu)

– Oscillation frequency ~9 kHz appears in passband

– BIBO unstable: two pairs of poles outside unit circle

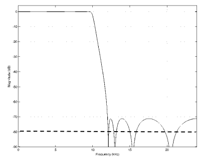


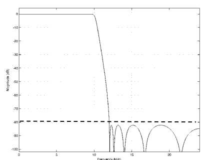


IIR filter design algorithms return poles-zeroes-gain (PZK format): Impact on response when expanding polynomials in transfer function from factored to unfactored form

MATLAB Demos Using fdatool #4

- **IIR filter - constrained least pth-norm design**
  - Use second-order sections
  - Limit pole radii ≤ 0.95
  - Increase weighting in stopband (Wstop) to 10
  - Filter order 8 does **not** meet stopband specification
  - Filter order 10 does meet stopband specification





Filter order might increase but worth it for more robust implementation

Conclusion

	FIR Filters	IIR Filters
Implementation complexity (1)	Higher	Lower (sometimes by factor of four)
Minimum order design	Parks-McClellan (Remez exchange) algorithm (2)	Elliptic design algorithm
Stable?	Always	May become unstable when implemented (3)
Linear phase	If impulse response is symmetric or anti-symmetric about midpoint	No, but phase may 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 Kaiser may be off by 10%. Search for minimum order is often needed.
(3) Algorithms can tune design to implementation target to minimize risk

Conclusion

- **Choice of IIR filter structure matters for both analysis and implementation**
- **Keep roots computed by filter design algorithms**
  - Polynomial deflation (rooting) reliable in floating-point
  - Polynomial inflation (expansion) may degrade roots
- **More than 20 IIR filter structures in use**
  - Direct forms and cascade of biquads are very common choices
- **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

Conclusion

- **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
- **When filter order is fixed, there exists no solution, one solution or an infinite number of solutions**
- **Minimum order design not always most efficient**
  - Efficiency depends on target implementation
  - Consider power-of-two coefficient design
  - Efficient designs may require search of infinite design space

Derivation of

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

condition

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 $|x[n]| \le B_1$ for all n.

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

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

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

$$\le \sum_{m=-\infty}^{\infty} |h[m]\, x[n-m]|$$

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

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

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

Special case: FIR Filter that has M coefficents.

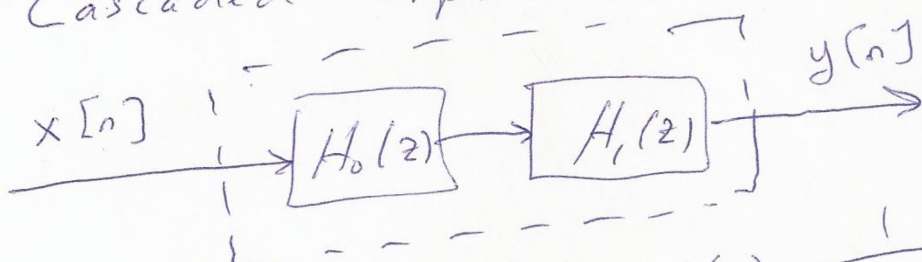
$$\sum_{m=-\infty}^{\infty} |h[m]| = \sum_{m=0}^{M-1} |h[m]| < \infty$$

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



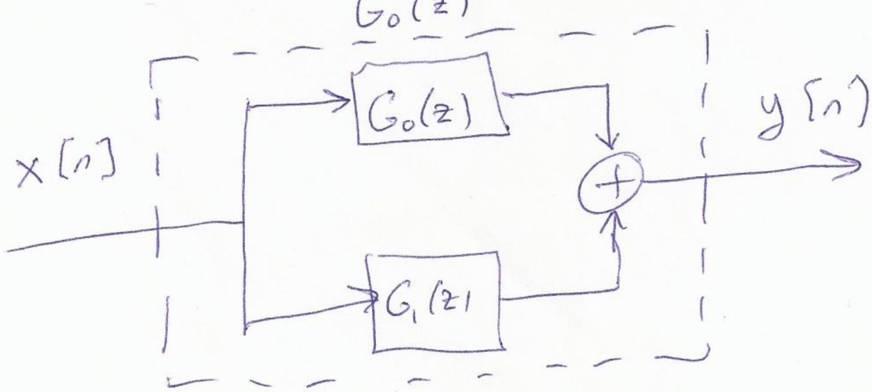
$$H_0(z) = \frac{1}{z-p_0} \qquad H_1(z) = \frac{1}{z-p_1}$$

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

Parallel Implementation

- Use partial fractions decomposition

$$H(z) = \underbrace{\frac{A}{z-p_0}}_{G_0(z)} + \underbrace{\frac{B}{z-p_1}}_{G_1(z)} \qquad A = \frac{1}{p_0 - p_1} \qquad B = \frac{1}{p_1 - p_0}$$



EE445S Real-Time Digital Signal Processing Lab   Spring 2014

Interpolation and Pulse Shaping

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

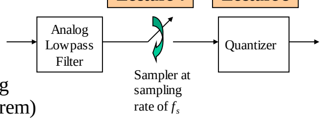
Lecture 7

Outline

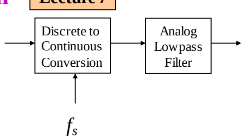
- **Discrete-to-continuous conversion**
- **Interpolation**
- **Pulse shapes**
  - Rectangular
  - Triangular
  - Sinc
  - Raised cosine
- **Sampling and interpolation demonstration**
- **Conclusion**

Data Conversion

- **Analog-to-Digital Conversion** **Lecture 4** **Lecture 8**
  - Lowpass filter has stopband frequency less than ½ f_s to reduce aliasing due to sampling (enforce sampling theorem)



- **Digital-to-Analog Conversion** **Lecture 7**
  - Discrete-to-continuous conversion could be as simple as sample and hold
  - Lowpass filter has stopband frequency less than ½ f_s to reduce artificial high frequencies



Discrete-to-Continuous Conversion

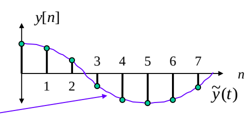
- ***Input*: sequence of samples $y[n]$**
- ***Output*: smooth continuous-time function obtained through interpolation (by "connecting the dots")**

If $f_0 < ½ f_s$, 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 ½ f_s

Discrete-to-Continuous Conversion

- **General form of interpolation is sum of weighted pulses**

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

Sequence $y[n]$ converted into continuous-time signal that is an approximation of $y(t)$

Pulse function $p(t)$ could be rectangular, triangular, parabolic, sinc, truncated sinc, raised cosine, etc.

Pulses overlap in time domain when pulse duration is greater than or equal to sampling period T_s

Pulses generally have unit amplitude and/or unit area

Above formula is related to discrete-time convolution

Interpolation From Tables

- **Using mathematical tables of numeric values of functions to compute a value of the function**

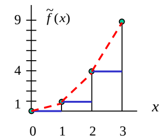
x	*f*(*x*)
0	0.0
1	1.0
2	4.0
3	9.0

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

Zero-order hold: take value to be $f(1)$ to make $f(1.5) = 1.0$ ("stairsteps")

Linear interpolation: average values of nearest two neighbors to get $f(1.5) = 2.5$

Curve fitting: fit four points in table to polynomial $a_0 + a_1 x + a_2 x^2 + a_3 x^3$ which gives $f(1.5) = x^2 = 2.25$

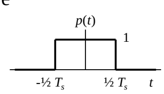


Rectangular Pulse

- **Zero-order hold**

Easy to implement in hardware or software

$$p(t) = \text{rect}\left(\frac{t}{T_s}\right) = \begin{cases} 1 & \text{if } -\frac{1}{2}T_s < t \le \frac{1}{2}T_s \\ 0 & \text{otherwise} \end{cases}$$



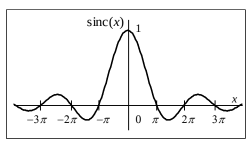
The Fourier transform is

$$P(f) = T_s \,\text{sinc}(\pi\, f\, T_s) = T_s \frac{\sin(\pi\, f\, T_s)}{\pi\, f\, T_s} \quad \text{where } \text{sinc}(x) = \frac{\sin(x)}{x}$$

In time domain, no overlap between $p(t)$ and adjacent pulses $p(t - T_s)$ and $p(t + T_s)$

In frequency domain, sinc has infinite two-sided extent; hence, the spectrum is not bandlimited

Sinc Function



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

How to compute sinc(0)? As $x \to 0$, numerator and denominator are both going to 0. How to handle it?

Even function (symmetric at origin)

Zero crossings at $x = \pm\pi, \pm 2\pi, \pm 3\pi, \ldots$

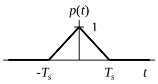
Amplitude decreases proportionally to 1/x

Triangular Pulse

- **Linear interpolation**

  It is relatively easy to implement in hardware or software, although not as easy as zero-order hold

  $$p(t) = \Delta\left(\frac{t}{T_s}\right) = \begin{cases} 1 - \frac{|t|}{T_s} & \text{if } T_s < t \le T_s \\ 0 & \text{otherwise} \end{cases}$$



  Overlap between $p(t)$ and its adjacent pulses $p(t - T_s)$ and $p(t + T_s)$ but with no others

- **Fourier transform is** $P(f) = T_s \operatorname{sinc}^2(f\, T_s)$

  How to compute this? *Hint:* Triangular pulse is equal to $1 / T_s$ times the convolution of rectangular pulse with itself

  In frequency domain, $\operatorname{sinc}^2(f\, T_s)$ has infinite two-sided extent; hence, the spectrum is not bandlimited

7 - 9

Sinc Pulse

- **Ideal bandlimited interpolation**

  $$p(t) = \operatorname{sinc}\left(\frac{\pi}{T_s} t\right) = \frac{\sin\left(\frac{\pi}{T_s} t\right)}{\frac{\pi}{T_s} t} \Leftrightarrow P(f) = \frac{1}{T_s} \operatorname{rect}\left(\frac{f}{T_s}\right) \qquad W = \frac{1}{2\, T_s}$$

  In time domain, infinite overlap between other pulses

  Fourier transform has extent $f \in [-W, W]$, where

  $P(f)$ is ideal lowpass frequency response with bandwidth W

  In frequency domain, sinc pulse is bandlimited

- **Interpolate with infinite extent pulse in time?**

  Truncate sinc pulse by multiplying it by rectangular pulse

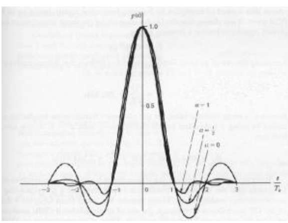
  Causes smearing in frequency domain (multiplication in time domain is convolution in frequency domain)

7 - 10

Raised Cosine Pulse: Time Domain

- **Pulse shaping used in communication systems**

  $$p(t) = \underbrace{\operatorname{sinc}\left(\frac{t}{T_s}\right)}_{\substack{\textit{ideal lowpass filter} \\ \textit{impulse response}}} \underbrace{\frac{\cos(2\pi\, \alpha\, W\, t)}{1 - 16\, \alpha^2\, W^2\, t^2}}_{\substack{\textit{Attenuation by } 1/t^2 \textit{ for} \\ \textit{large } t \textit{ to reduce tail}}}$$



  W is bandwidth of an ideal lowpass response

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

  Zero crossings at $t = \pm\, T_s, \pm\, 2\, T_s, \ldots$

- **See handout G in reader on raised cosine pulse**

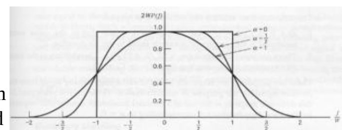
7 - 11

Raised Cosine Pulse Spectra

- **Pulse shaping used in communication systems**

  Bandwidth increased by factor of $(1 + \alpha)$: $(1 + \alpha)\, W = 2\, W - f_1$

  f_1 marks transition from passband to stopband



  $$P(f) = \begin{cases} \frac{1}{2W} & \text{if } 0 \le |f| < f_1 \\ \frac{1}{4W}\left(1 - \sin\left(\frac{\pi(|f| - W)}{2W - 2 f_1}\right)\right) & \text{if } f_1 \le |f| < 2W - f_1 \\ 0 & \text{otherwise} \end{cases} \qquad W = \frac{1}{2\, T_s} \qquad \alpha = 1 - \frac{f_1}{W}$$

  Bandwidth generally scarce in communication systems

7 - 12

Sampling and Interpolation Demo

- **DSP First, Ch. 4, Sampling and interpolation,**
  http://www.ece.gatech.edu/research/DSP/DSPFirstCD/
  - Sample sinusoid $y(t)$ to form $y[n]$
  - Reconstruct sinusoid using rectangular, triangular, or truncated sinc pulse $p(t)$

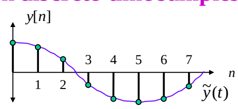
$$\tilde{y}(t) = \sum_{n=-\infty}^{\infty} y[n]\, p(t - T_s\, n)$$

- **Which pulse gives the best reconstruction?**
- **Sinc pulse is truncated to be four sampling periods long. Why is the sinc pulse truncated?**
- **What happens as the sampling rate is increased?**

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 raised cosine for practical bandlimited interpolation
- **Truncation causes smearing in frequency domain**

Quantization

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

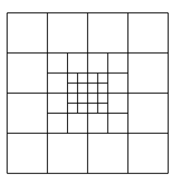
Lecture 8

Outline

- **Introduction**
- **Uniform amplitude quantization**
- **Audio**
- **Quantization error (noise) analysis**
- **Noise immunity in communication systems**
- **Conclusion**
- **Digital vs. analog audio** (optional)

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

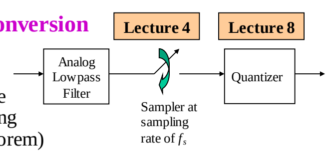


Foveated grid: point of focus in middle

- **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

Data Conversion

- **Analog-to-Digital Conversion**



Lowpass filter has stopband frequency less than ½ f_s to reduce aliasing due to sampling (enforce sampling theorem)

System properties:			
Linearity			
Time-Invariance			
Causality			
Memory			

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

Uniform Amplitude Quantization

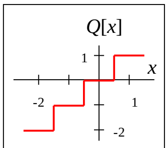
- **Round to nearest integer (midtread)**

  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



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

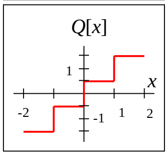
- **Rounding with offset (midrise)**

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

  Represent levels by {11, 10, 00, 01} …

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

  Used in slide 8-10



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?***

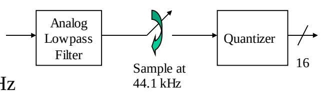
  Standard two's complement behavior

  Addition performed modulo set of integers

  Used in address calculations, array indexing

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 (and hence called an anti-aliasing filter)

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

  1.76 dB + 6.02 dB/bit * *B* = 98.08 dB

  This loose *upper bound* is derived later in slides 8-10 to 8-14

  In practice, audio CDs have dynamic range of about 95 dB

Dynamic Range

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

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \frac{\text{Signal Power}}{\text{Noise Power}}$$
$$= 10 \log_{10} \text{Signal Power} - 10 \log_{10} \text{Noise Power}$$

Why 10 $\log_{10}$?

For amplitude *A*,

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

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

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

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

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

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

  Ideal magnitude response of 0 dB over passband

  A_{stopband} = 0 dB – Noise Power in dB = -98.08 dB

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

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

- **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

8 - 9

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

$$m \longrightarrow \boxed{Q_B[\,\cdot\,]} \longrightarrow v$$

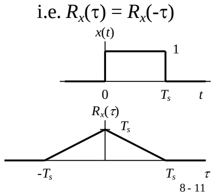
  Quantization error
  $q = Q_B[m] - m = v - m$

- **Assumptions**
  $m \in (-m_{\max}, m_{\max})$
  Uniform midrise quantizer
  Input does not overload quantizer
  Quantization error (noise) is uniformly distributed
  Number of quantization levels $L = 2^B$ is large enough so that $\frac{1}{L-1} \approx \frac{1}{L}$

8 - 10

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
  $X(f)\, X^*(f) = F\left\{ x(\tau) * x^*(-\tau) \right\}$
- **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. $R_x(\tau) = R_x(-\tau)$



8 - 11

Quantization Error (Noise) Analysis

- **Two-sided random signal** $n(t)$ $\quad P_n(f) = F\left\{ R_n(\tau) \right\}$
  Fourier transform may not exist, but power spectrum exists

$$R_n(\tau) = E\left\{ n(t)\, n^*(t+\tau) \right\} = \int_{-\infty}^{\infty} n(t)\, n^*(t+\tau)\, dt$$

$$R_n(-\tau) = E\left\{ n(t)\, n^*(t-\tau) \right\} = \int_{-\infty}^{\infty} n(t)\, n^*(t-\tau)\, dt = n(\tau) * n^*(-\tau)$$

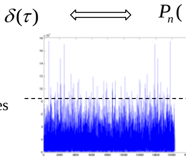
  For zero-mean Gaussian random process $n(t)$ with variance σ^2

$$R_n(\tau) = E\left\{ n(t)\, n^*(t+\tau) \right\} = 0 \;\text{ when }\; \tau \neq 0$$

$$R_n(\tau) = E\left\{ n(t)\, n^*(t+\tau) \right\} = \sigma^2\, \delta(\tau) \quad \Longleftrightarrow \quad P_n(f) = \sigma^2$$

- **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 - 12

Quantization Error (Noise) Analysis

- **Quantizer step size**

$$\Delta = \frac{2\, m_{max}}{L-1} \approx \frac{2\, m_{max}}{L}$$

- **Quantization error**

$$-\frac{\Delta}{2} \le q \le \frac{\Delta}{2}$$

  q is sample of zero-mean random process Q

  q is uniformly distributed

$$\sigma_Q^2 = E\{Q^2\} - \underbrace{\mu_Q^2}_{zero}$$

$$\sigma_Q^2 = \frac{\Delta^2}{12} = \frac{1}{3} m_{max}^2\, 2^{-2B}$$

- **Input power: $P_{average,m}$**

$$\text{SNR} = \frac{\text{Signal Power}}{\text{Noise Power}}$$

$$\text{SNR} = \frac{P_{average,m}}{\sigma_Q^2} = \left(\frac{3P_{average,m}}{m_{max}^2}\right) 2^{2B}$$

  SNR exponential in B

  Adding 1 bit increases SNR by factor of 4

- **Derivation of SNR in deciBels on next slide**

Quantization Error (Noise) Analysis

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

Loose upper bound

$$10 \log_{10} \text{SNR} = 10 \log_{10}\left(\left(\frac{3P_{average,m}}{m_{max}^2}\right) 2^{2B}\right)$$

$$= 10 \log_{10} 3 + 10 \log_{10}(P_{average,m}) - 20 \log_{10}(m_{max}) + 20\, \text{B} \log_{10}(2)$$

$$= \underbrace{0.477 + 10 \log_{10}(P_{average,m}) - 20 \log_{10}(m_{max})}_{\textit{1.76 and 1.17 are common constants used in audio}} + 6.02\, B$$

- **What is maximum number of bits of resolution for**

  Audio CD signal with SNR of 95 dB

  TI TLV320AIC23B stereo codec used on TI DSP board

  – ADC 90 dB SNR (14.6 bits) and 80 dB THD (13 bits) page 2-2

  – DAC has 100 dB SNR (16 bits) and 88 dB THD (14.3 bits) page 2-3

Total Harmonic Distortion (THD)

- **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

  SNR measure is then taken

- **In audio, sinusoidal signal is often at 1 kHz**

  "Sweet spot" for human hearing – strongest response

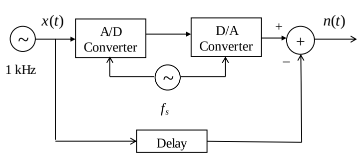
- **Example**

  "System" is ADC

  Calibrated DAC

  Signal is $x(t)$

  "Noise" is $n(t)$



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.

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

Optional

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

Optional

Digital vs. Analog Audio

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

$$A(v) = \begin{cases} V_0 + (v - V_0)^{1/3} & \text{for } v > V_0 \\ v & \text{for } -V_0 \le v \le V_0 \\ -V_0 - (V_0 - v)^{1/3} & \text{for } v < -V_0 \end{cases} \qquad D(v) = \begin{cases} V_0 & \text{for } v > V_0 \\ v & \text{for } -V_0 \le v \le V_0 \\ -V_0 & \text{for } v < -V_0 \end{cases}$$

- **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

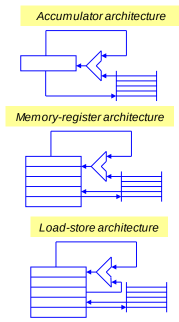
INTRODUCTION TO THE TMS320C6000 VLIW DSP

Prof. Brian L. Evans

in collaboration with
Dr. Niranjan Damera-Venkata and
Mr. Magesh Valliappan

Embedded Signal Processing Laboratory
The University of Texas at Austin
Austin, TX 78712-1084

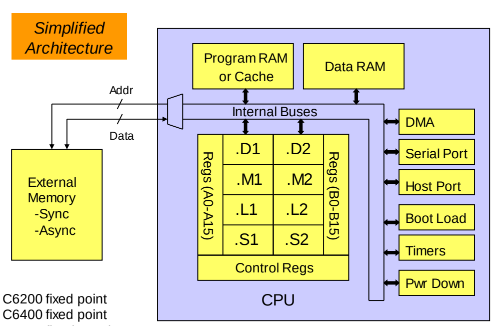
http://signal.ece.utexas.edu/



Outline

- C6000 instruction set architecture review
- Vector dot product example
- Pipelining
- Finite impulse response filtering
- Vector dot product example
- Conclusion

TI TMS320C6000 DSP Architecture (Review)



C6200 fixed point
C6400 fixed point
C6700 floating point

TI TMS320C6000 DSP Architecture (Review)

- Address 8/16/32 bit data + 64-bit data on C67x
- Load-store RISC architecture with 2 data paths
  - 16 32-bit registers per data path (A0-A15 and B0-B15)
  - 48 instructions (C6200) and 79 instructions (C6700)
- Two parallel data paths with 32-bit RISC units
  - **D**ata unit - 32-bit address calculations (modulo, linear)
  - **M**ultiplier unit - 16 bit x 16 bit with 32-bit result
  - **L**ogical unit - 40-bit (saturation) arithmetic & compares
  - **S**hifter unit - 32-bit integer ALU and 40-bit shifter
  - Conditionally executed based on registers A1-2 & B0-2
  - Can work with two 16-bit halfwords packed into 32 bits

TI TMS320C6000 DSP Architecture (Review)

- .M multiplication unit
  - 16 bit x 16 bit signed/unsigned packed/unpacked
- .L arithmetic logic unit
  - Comparisons and logic operations (and, or, and xor)
  - Saturation arithmetic and absolute value calculation
- .S shifter unit
  - Bit manipulation (set, get, shift, rotate) and branching
  - Addition and packed addition
- .D data unit
  - Load/store to memory
  - Addition and pointer arithmetic

C6000 Restrictions on Register Accesses

- Function unit access to register files
  - Data path 1 (2) units read/write A (B) registers
  - Data path 2 (1) can read one A (B) register per instruction cycle with one-cycle latency
  - Two simultaneous memory accesses cannot use registers of same register file as address pointers
  - Limit of four 32-bit reads per register per inst. cycle
- 40-bit longs stored in adjacent even/odd registers
  - Extended precision accumulation of 32-bit numbers
  - Only one 40-bit result can be written per cycle
  - 40-bit read cannot occur in same cycle as 40-bit write
  - 4:1 performance penalty using 40-bit mode

Other C6000 Disadvantages

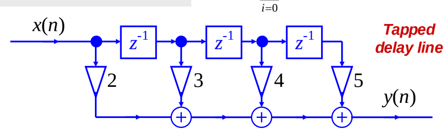
- No ALU acceleration for bit stream manipulation
  - 50% computation in MPEG-2 decoder spent on variable length decoding on C6200 in C
  - C6400 direct memory access controllers shred bit streams (for video conferencing & wireless basestations)
- Branch in pipeline disables interrupts:
  Avoid branches by using conditional execution
- No hardware protection against pipeline hazards:
  Programmer and tools must guard against it
- Must emulate many conventional DSP features
  - No hardware looping: use register/conditional branch
  - No bit-reversed addressing: use fast algorithm by Elster
  - No status register: only saturation bit given by .L units

FIR Filter

- Difference equation (vector dot product)

  $y(n) = 2\,x(n) + 3\,x(n-1) + 4\,x(n-2) + 5\,x(n-3)$

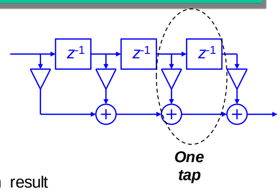
- Signal flow graph $y(n) = \sum_{i=0}^{N-1} a(i)\; x(n-i)$



- Dot product of inputs vector and coefficient vector
- Store input in circular buffer, coefficients in array

FIR Filter

- Each tap requires
  - Fetching data sample
  - Fetching coefficient
  - Fetching operand
  - Multiplying two numbers
  - Accumulating multiplication result
  - Possibly updating delay line (see below)



- Computing an FIR tap in one instruction cycle
  - Two data memory and one program memory accesses
  - Auto-increment or auto-decrement addressing modes
  - Modulo addressing to implement delay line as circular buffer

Example: Vector Dot Product (Unoptimized)

- A vector dot product is common in filtering

$$Y = \sum_{n=1}^{N} a(n)\ x(n)$$

  - Store $a(n)$ and $x(n)$ into an array of N elements
- C6000 peaks at 8 RISC instructions/cycle
  - For 300-MHz C6000, RISC instructions per sample
    - 300,000 for speech (sampling rate 8 kHz)
    - 54,421 for audio CD (sampling rate 44.1 kHz)
    - 230 for luminance NTSC digital video (sampling rate 10,368 kHz)
  - Generally requires hand coding for peak performance

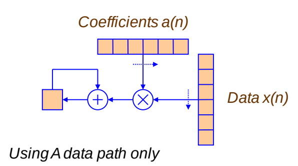
Example: Vector Dot Product (Unoptimized)

- Prologue
  - Initialize pointers: A5 for $a(n)$, A6 for $x(n)$, and A7 for Y
  - Move number of times to loop (N) into A2
  - Set accumulator (A4) to zero
- Inner loop
  - Put $a(n)$ into A0 and $x(n)$ into A1
  - Multiply $a(n)$ and $x(n)$
  - Accumulate multiplication result into A4
  - Decrement loop counter (A2)
  - Continue inner loop if counter is not zero
- Epilogue
  - Store the result into Y

Assuming coefficients & data are 16 bits wide

Reg	Meaning
A0	$a(n)$
A1	$x(n)$
A2	$N - n$
A3	$a(n)\ x(n)$
A4	Y
A5	$\&a$
A6	$\&x$
A7	$\&Y$

Example: Vector Dot Product (Unoptimized)



Reg	Meaning
A0	$a(n)$
A1	$x(n)$
A2	$N - n$
A3	$a(n)\ x(n)$
A4	Y
A5	$\&a$
A6	$\&x$
A7	$\&Y$

```
; clear A4 and initialize pointers A5, A6, and A7
        MVK  .S1  40,A2       ; A2 = 40 (loop counter)
loop    LDH  .D1  *A5++,A0    ; A0 = a(n), H = halfword
        LDH  .D1  *A6++,A1    ; A1 = x(n), H = halfword
        MPY  .M1  A0,A1,A3    ; A3 = a(n) * x(n)
        ADD  .L1  A3,A4,A4    ; Y = Y + A3
        SUB  .L1  A2,1,A2     ; decrement loop counter
[A2]    B    .S1  loop        ; if A2 != 0, then branch
        STH  .D1  A4,*A7      ; *A7 = Y
```

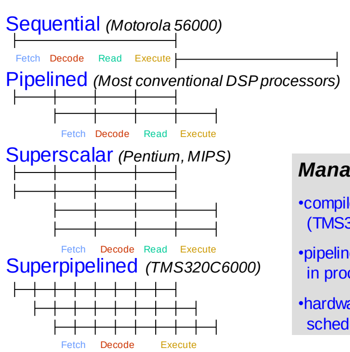
Example: Vector Dot Product (Unoptimized)

- MoVeKonstant
  - MVK .S 40,A2 ; A2 = 40
  - Lower 16 bits of A2 are loaded
- Conditional branch
  - [condition] B .S loop
  - [A2] means to execute instruction if A2 != 0 (same as C language)
  - Only A1, A2, B0, B1, and B2 can be used (not symmetric)
- Loading registers
  - LDH .D *A5, A0 ;Loads half-word into A0 from memory
- Registers may be used as pointers (*A1++)
- Implementation not efficient due to pipeline effects

Pipelining

- CPU operations
  - **F**etch instruction from (on-chip) program memory
  - **D**ecode instruction
  - **E**xecute instruction including reading data values
- Overlap operations to increase performance
  - Pipeline CPU operations to increase clock speed over a sequential implementation
  - Separate parallel functional units
  - Peripheral interfaces for I/O do not burden CPU

Pipelining



Managing Pipelines

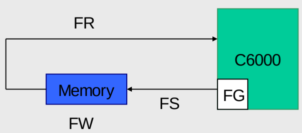
- compiler or programmer (TMS320C6000)
- pipeline interlocking in processor (TMS320C30)
- hardware instruction scheduling

TMS320C6000 Pipeline

- One instruction cycle every clock cycle
- Deep pipeline
  - 7-11 stages in C62x: fetch 4, decode 2, execute 1-5
  - 7-16 stages in C67x: fetch 4, decode 2, execute 1-10
  - If a branch is in the pipeline, interrupts are disabled
  - Avoid branches by using conditional execution
- No hardware protection against pipeline hazards
  - Compiler and assembler must prevent pipeline hazards
- Dispatches instructions in packets

Program Fetch (F)

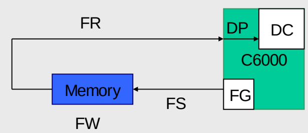
- Program fetching consists of 4 phases
  - Generate fetch address (FG)
  - Send address to memory (FS)
  - Wait for data ready (FW)
  - Read opcode (FR)
- Fetch packet consists of 8 32-bit instructions



9-17

Decode Stage (D)

- Decode stage consists of two phases
  - Dispatch instruction to functional unit (DP)
  - Instruction decoded at functional unit (DC)



9-18

Execute Stage (E)

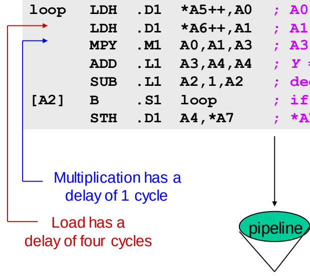
Type	*Description*	*# Instr*	*Delay*
ISC	Single cycle	38	0
IMPY	Multiply	2	1
LDx	Load	3	4
B	Branch	1	5

Execute Phase	***Description***
E1	ISC instructions completed
E2	Int. mult. instructions completed
E3	
E4	
E5	Load memory value into register
E6	Branch to destination complete

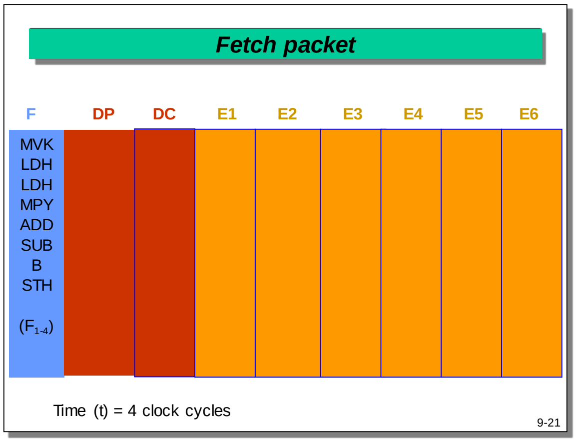
9-19

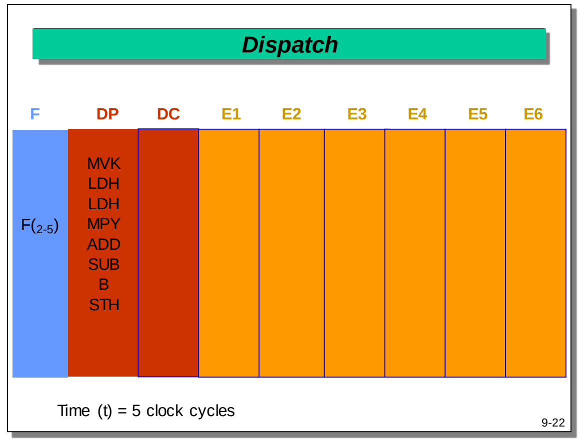
Vector Dot Product with Pipeline Effects

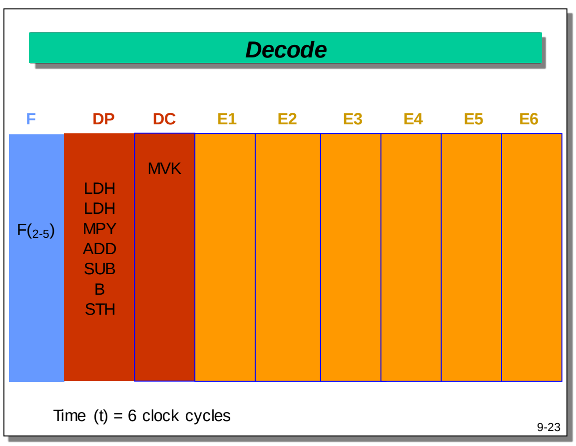
```
; clear A4 and initialize pointers A5, A6, and A7
        MVK   .S1   40,A2       ; A2 = 40 (loop counter)
loop    LDH   .D1   *A5++,A0    ; A0 = a(n), H = halfword
        LDH   .D1   *A6++,A1    ; A1 = x(n), H = halfword
        MPY   .M1   A0,A1,A3    ; A3 = a(n) * x(n)
        ADD   .L1   A3,A4,A4    ; Y = Y + A3
        SUB   .L1   A2,1,A2     ; decrement loop counter
[A2]    B     .S1   loop        ; if A2 != 0, then branch
        STH   .D1   A4,*A7      ; *A7 = Y
```

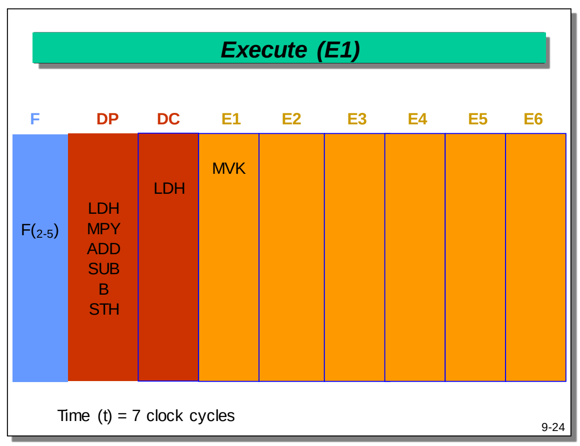


9-20

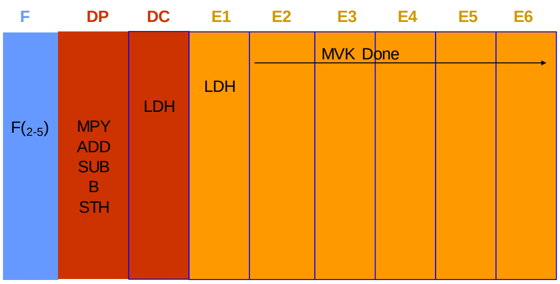

Fetch packet
F
DP
DC
E1
E2
E3
E4
E5
E6
MVK
LDH
LDH
MPY
ADD
SUB
B
STH
(F1-4)
Time (t) = 4 clock cycles
9-21


Dispatch
F
DP
DC
E1
E2
E3
E4
E5
E6
F(2-5)
MVK
LDH
LDH
MPY
ADD
SUB
B
STH
Time (t) = 5 clock cycles
9-22


Decode
F
DP
DC
E1
E2
E3
E4
E5
E6
F(2-5)
LDH
LDH
MPY
ADD
SUB
B
STH
MVK
Time (t) = 6 clock cycles
9-23


Execute (E1)
F
DP
DC
E1
E2
E3
E4
E5
E6
F(2-5)
LDH
MPY
ADD
SUB
B
STH
LDH
MVK
Time (t) = 7 clock cycles
9-24

Execute (MVK done LDH in E1)



Time (t) = 8 clock cycles

9-25

Vector Dot Product with Pipeline Effects

```
; clear A4 and initialize pointers A5, A6, and A7
        MVK  .S1  40,A2      ; A2 = 40 (loop counter)
loop    LDH  .D1  *A5++,A0   ; A0 = a(n)
        LDH  .D1  *A6++,A1   ; A1 = x(n)
        NOP  4
        MPY  .M1  A0,A1,A3   ; A3 = a(n) * x(n)
        NOP
        ADD  .L1  A3,A4,A4   ; Y = Y + A3
        SUB  .L1  A2,1,A2    ; decrement loop counter
[A2]    B    .S1  loop       ; if A2 != 0, then branch
        NOP  5
        STH  .D1  A4,*A7     ; *A7 = Y
```

Assembler will automatically insert NOP instructions

Assembler can also make sequential code parallel

9-26

Optimized Vector Dot Product on the C6000

16-bit data/ coefficients

- Split summation into two summations
- Prologue
  - Initialize pointers: A5 for $a(n)$, B6 for $x(n)$, A7 for $y(n)$
  - Move number of times to loop (N) divided by 2 into A2
- Inner loop
  - Put $a(n)$ and $a(n+1)$ in A0 and $x(n)$ and $x(n+1)$ in A1 (***packed data***)
  - Multiply $a(n)$ $x(n)$ and $a(n+1)$ $x(n+1)$
  - Accumulate even (odd) indexed terms in A4 (B4)
  - Decrement loop counter (A2)
- Store result

Reg	Meaning
A0	$a(n) \mid\mid a(n+1)$
B1	$x(n) \mid\mid x(n+1)$
A2	$(N-n)/2$
A3	$a(n)\,x(n)$
B3	$a(n+1)\,x(n+1)$
A4	$y_{even}(n)$
B4	$y_{odd}(n)$
A5	$\&a$
B6	$\&x$
A7	$\&Y$

9-27

FIR Filter Implementation on the C6000

```
        MVK  .S1  0x0001,AMR ; modulo block size 2^2
        MVKH .S1  0x4000,AMR ; modulo addr register B6
        MVK  .S2  2,A2       ; A2 = 2 (four-tap filter)
        ZERO .L1  A4         ; initialize accumulators
        ZERO .L2  B4
; initialize pointers A5, B6, and A7
fir     LDW  .D1  *A5++,A0   ; load a(n) and a(n+1)
        LDW  .D2  *B6++,B1   ; load x(n) and x(n+1)
        MPY  .M1X A0,B1,A3   ; A3 = a(n) * x(n)
        MPYH .M2X A0,B1,B3   ; B3 = a(n+1) * x(n+1)
        ADD  .L1  A3,A4,A4   ; yeven(n) += A3
        ADD  .L2  B3,B4,B4   ; yodd(n) += B3
[A2]    SUB  .S1  A2,1,A2    ; decrement loop counter
[A2]    B    .S2  fir        ; if A2 != 0, then branch
        ADD  .L1  A4,B4,A4   ; Y = Yodd + Yeven
        STH  .D1  A4,*A7     ; *A7 = Y
```

Throughput of two multiply-accumulates per instruction cycle

9-28

Conclusion

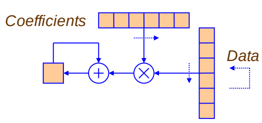
- Conventional digital signal processors
  - High performance vs. power consumption/cost/volume
  - Excel at one-dimensional processing
  - Have instructions tailored to specific applications
- TMS320C6000 VLIW DSP
  - High performance vs. cost/volume
  - Excel at multidimensional signal processing
  - Maximum of 8 RISC instructions per cycle

Conclusion

- Web resources
  - comp.dsp news group: FAQ www.bdti.com/faq/dsp_faq.html
  - embedded processors and systems: www.eg3.com
  - on-line courses and DSP boards: www.techonline.com
- References

  - R. Bhargava, R. Radhakrishnan, B. L. Evans, and L. K. John, "Evaluating MMX Technology Using DSP and Multimedia Applications," *Proc. IEEE Sym. Microarchitecture*, pp. 37-46, 1998.http://www.ece.utexas.edu/~ravib/mmxdsp/
  - B. L. Evans, "EE345S Real-Time DSP Laboratory," UT Austin. http://www.ece.utexas.edu/~bevans/courses/realtime/
  - B. L. Evans, "EE382C Embedded Software Systems," UT Austin.http://www.ece.utexas.edu/~bevans/courses/ee382c/

FIR Filter on a TMS320C5000




```
COEFFP .set 02000h       ; Program mem address
X      .set 037Fh        ; Newest data sample
LASTAP .set 037FH        ; Oldest data sample

       …
       LAR AR3, #LASTAP    ; Point to oldest sample
       RPT #127            ; Repeat next inst. 126 times
       MACD COEFFP, *-     ; Compute one tap of FIR
       APAC
       SACH Y,1            ; Store result -- note shift
```


TMS320C6200 vs. StarCore S140

Feature	*C6200*	*S140*
Functional Units	8	16
multipliers	2	4
adders	6	4
other	--	8
Instructions/cycle	8	6 + branch
RISC instructions *	8	11
conditionals	8	2
Instruction width (bits)	256	128
Total instructions	48	180
Number of registers	32	51
Register size (bits)	32	40
Accumulation precision (bits) **	32 or 40	40
Pipeline depth (cycle)	7-11	5

* Does not count equivalent RISC operations for modulo addressing
** On the C6200, there is a performance penalty for 40-bit accumulation

EE445S Real-Time Digital Signal Processing Lab Spring 2014

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

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)
  - Example uses 16 different thresholds in a 4 × 4 mask
  - Periodic artifacts appear as if screen has been overlaid

No noise shaping

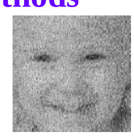
Image Halftoning

- ***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**

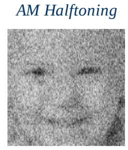
Digital Halftoning Methods

Clustered Dot Screening
AM Halftoning

Dispersed Dot Screening
FM Halftoning



Error Diffusion
FM Halftoning 1975



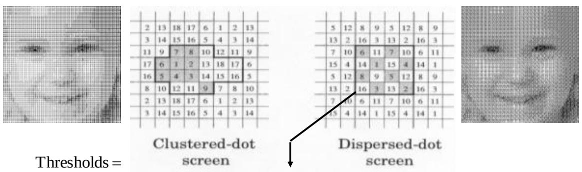
Blue-noise Mask
FM Halftoning 1993

Green-noise Halftoning
AM-FM Halftoning 1992

Direct Binary Search
FM Halftoning 1992

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

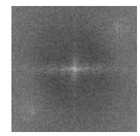


$$\text{Thresholds} = \left\{ \frac{1}{32}, \frac{3}{32}, \frac{5}{32}, \frac{7}{32}, \frac{9}{32}, \frac{11}{32}, \frac{13}{32}, \frac{15}{32}, \frac{17}{32}, \frac{19}{32}, \frac{21}{32}, \frac{23}{32}, \frac{25}{32}, \frac{27}{32}, \frac{29}{32}, \frac{31}{32} \right\} * 256$$

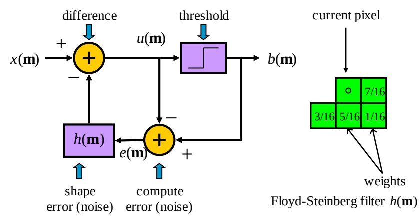
Grayscale Error Diffusion

- **Shapes quantization error (noise) into high frequencies**
- **Type of sigma-delta modulation**
- **Error filter $h(\mathbf{m})$ is lowpass**

Error Diffusion Halftone

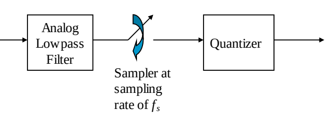


Spectrum

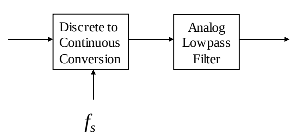


Old-Style A/D and D/A Converters

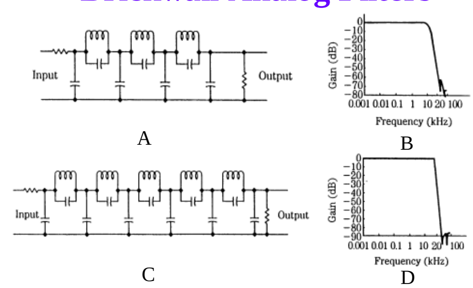
- **Used discrete components (before mid-1980s)**
- **A/D Converter**
  - Lowpass filter has stopband frequency of ½ f_s



- **D/A Converter**
  - Lowpass filter has stopband frequency of ½ f_s
  - Discrete-to-continuous conversion could be as simple as sample and hold

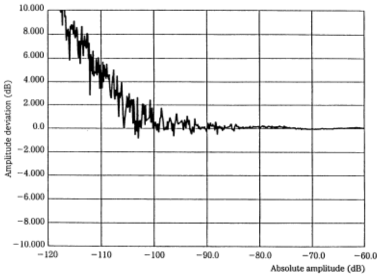


Cost of Multibit Conversion Part I: Brickwall Analog Filters



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.

Cost of Multibit Conversion Part II: Low- Level Linearity



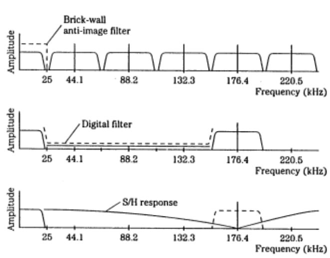
Pohlmann Fig. 4-3 An example of a low-level linearity measurement of a D/A converter showing increasing non-linearity with decreasing amplitude.

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	5th / 7th order	5th / 7th order	5th / 7th order
Dynamic range	110 dB	120 dB	120 dB

Solution 1: Oversampling



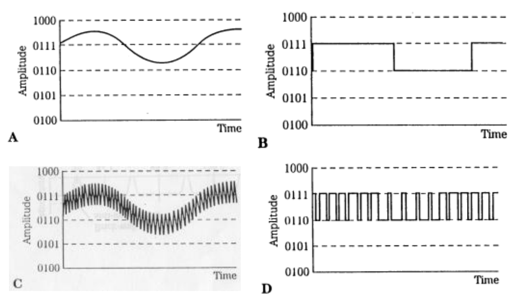
A. A brick-wall filter must sharply bandlimit the output spectra.

B. With four-times oversampling, images appear only at the oversampling frequency.

C. The output sample/hold (S/H) circuit can be used to further suppress the oversampling spectra.

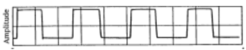
Pohlmann Fig. 4-15 Image spectra of nonoversampled and oversampled reconstruction. Four times oversampling simplifies reconstruction filter.

Solution 2: Add Dither



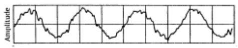
Pohlmann Fig. 2-8 Adding dither at quantizer input alleviates effects of quantization error.
A. An undithered input signal with amplitude on the order of one LSB.
B. Quantization results in a coarse coding over two levels. C. Dithered input signal.
D. Quantization yields a PWM waveform that codes information below the LSB.

Time Domain Effect of Dither

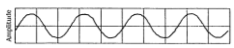


A A 1 kHz sinewave with amplitude of one-half LSB without dither produces a square wave.

B Dither of one-third LSB rms amplitude is added to the sinewave before quantization, resulting in a PWM waveform.



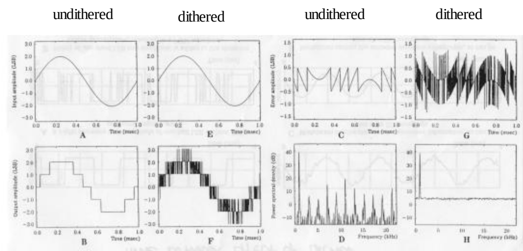
C Modulation carries the encoded sinewave information, as can be seen after 32 averagings.



D Modulation carries the encoded sinewave 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.

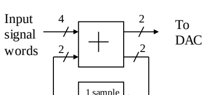
Frequency Domain Effect of Dither



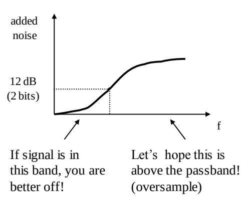
Pohlmann Fig. 2-10 Computer-simulated quantization of a low-level 1- kHz sinewave without, and with dither. A. Input signal. B. Output signal (no dither). C. Total error signal (no dither). D. Power spectrum of output signal (no dither). E. Input signal. F. Output signal (triangualr pdf dither). G. Total error signal (triangular pdf dither). H. Power spectrum of output signal (triangular pdf dither) Lipshitz, Wannamaker, and Vanderkooy

Solution 3: Noise Shaping

We have a two-bit DAC and four-bit input signal words. Both are unsigned.



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.



Assume input = 1001 constant

Time	Adder Inputs Upper	Adder Inputs 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

Periodic

Average output = 1/4(10+10+10+11)=1001

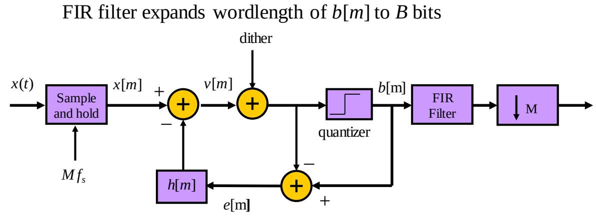
⟹ 4-bit resolution at DC!

Putting It All Together

- **A/D converter samples at f_s and quantizes to B bits**
- **Sigma delta modulator implementation**

  Internal clock runs at $M f_s$

  FIR filter expands wordlength of $b[m]$ to B bits



EE 445S Real-Time Digital Signal Processing Lab Spring 2014

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.

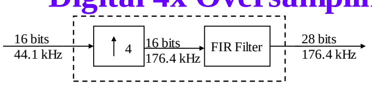
Lecture 11

Solutions

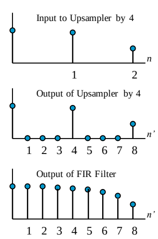
- **Oversampling eases analog filter design**
  Also creates spectrum to put noise at inaudible frequencies
- **Add dither (noise) at quantizer input**
  Breaks up harmonics (idle tones) caused by quantization
- **Shape quantization noise into high frequencies**
  Auditory system is less sensitive at 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

Digital 4x Oversampling Filter

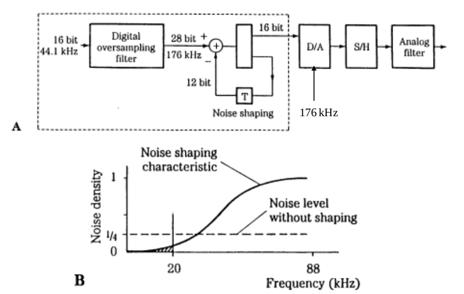


Digital 4x Oversampling Filter



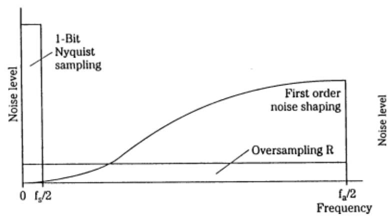
- **Upsampling by 4 (denoted by ↑4)**
  For each input sample, output the input sample followed by three zeros
  Four times the samples on output as input
  Increases sampling rate by factor of 4
- **FIR filter performs interpolation**
  Multiplying 16-bit data and 8-bit coefficient: 24-bit result
  Adding two 24-bit numbers: 25-bit result
  Adding 16 24-bit numbers: 28-bit result

Oversampling Plus Noise Shaping



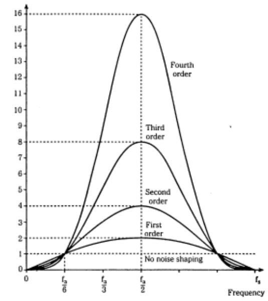
Pohlmann Fig. 4-17 Noise shaping following oversampling decreases in-band quantization error. A. Simple noise-shaping loop. B. Noise shaping suppresses noise in the audio band; boosted noise outside the audio band is filtered out.

Oversampling and Noise Shaping



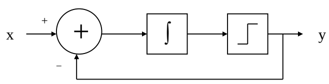
Pohlmann Fig. 16-4 With 1-bit conversion, quantization noise is quite high. In-band noise is reduced with oversampling. With noise shaping, quantization noise is shifted away from the audio band, further reducing in-band noise.

Oversampling and Noise Shaping



Pohlmann Fig. 16- 6 Higher orders of noise shaping result in more pronounced shifts in requantization noise.

First-Order Delta-Sigma Modulator

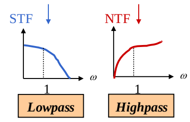


Assume quantizer adds uncorrelated white noise n *(model nonlinearity as additive noise)*

Continuous time: $\int = \frac{1}{s}$

$$Y(s) = \frac{X(s) - Y(s)}{s} + N(s)$$

$$Y(s) = \underbrace{\frac{1}{s+1}}_{\text{signal transfer function (STF)}} X(s) + \underbrace{\frac{s}{s+1}}_{\text{noise transfer function (NTF)}} N(s)$$



Discrete time: $\int = \frac{1}{1-z^{-1}}$

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

$$Y(z) = \underbrace{\frac{\frac{1}{2}}{1-\frac{1}{2}z^{-1}}}_{\text{signal transfer function (STF)}} X(z) + \underbrace{\frac{1}{2}\,\frac{1-z^{-1}}{1-\frac{1}{2}z^{-1}}}_{\text{noise transfer function (NTF)}} N(z)$$

Lowpass **Highpass**

Higher-order modulators

- Add more integrators
- Stability is a major issue

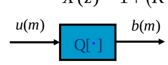
Noise-Shaped Feedback Coder

- **Type of sigma-delta modulator (see slide 9-6)**
- **Model quantizer as LTI** **[Ardalan & Paulos, 1988]**

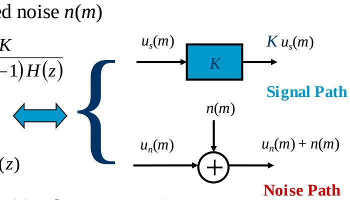
  Scales input signal by a gain by K (where $K > 1$)

  Adds uncorrelated noise $n(m)$

$$STF = \frac{B_s(z)}{X(z)} = \frac{K}{1 + (K-1)H(z)}$$

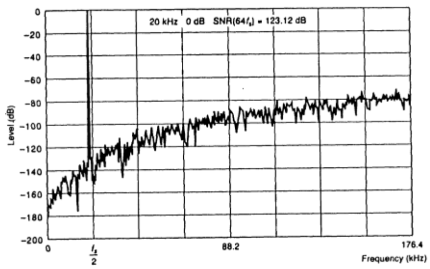


$$NTF = \frac{B_n(z)}{N(z)} = 1 - H(z)$$



NTF is highpass $\Longrightarrow$ H(**z**) is lowpass $\Longrightarrow$ STF passes low frequencies and amplifies high frequencies

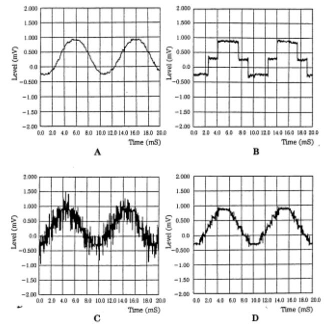
Third-order Noise Shaper Results



Pohlmann Fig. 16-13 Reproduction of a 20 kHz waveform showing the effect of third-order noise shaping. Matsushita Electric

19-Bit Resolution from a CD: Part I

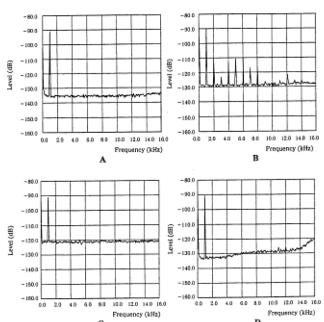
Poh1man Fig. 6-27 An example of noise shaping showing a 1 kHz sinewave with -90 dB amplitude; measurements are made with a 16 kHz lowpass filter.
A. Original 20 bit recording.
B. Truncated 16 bit signal.
C. Dithered 16 bit signal.
D. Noise shaping preserves information in lower 4 bits.



19-Bit Resolution from a CD: Part II

Pohlmann Fig. 16-28 An example of noise shaping showing the spectrum of a 1 kHz, -90 dB sinewave (from Fig. 16-27).
A. Original 20-bit recording
B. Truncated 16-bit signal
C. Dithered 16-bit signal
D. Noise shaping reduces low and medium frequency noise.

Sony's Super Bit Mapping uses psycho-acoustic noise shaping (instead of sigma-delta modulation) to convert studio masters recorded at 20-24 bits/sample into CD audio at 16 bits/sample. All Dire Straits albums are available in this format.



Open Issues in Audio CD Converters

- **Oversampling systems used in 44.1 kHz converters**
  Digital anti-imaging filters (anti-aliasing filters in the case of A/D converters) can be improved (from paper by J. Dunn)
- ***Ripple*: Near-sinusoidal ripple of passband can be interpreted as due to sum of original signal and smaller pre- and post-echoes of original signal**
  Ripple magnitude and no. of cycles in passband correspond to echoes up to 0.8 ms either side of direct signal and between -120 and -50 dB in amplitude relative to direct
  Post-echo masked by signal, but pre-echo is not masked
  Solution is to reduce passband ripple. Human hearing is no better than 0.1 dB at its most sensitive, but associated pre-echo from 0.1 dB passband ripple is audible.

Open Issues in Audio CD Converters

- ***Stopband rejection* (A/D Converter)**
  - Anti-aliasing filters are often half-band type with only 6 dB attenuation at 1/2 of sampling rate.
  - Do not adequately reject frequencies that will alias.
  - Ideal filter rolls off at 20 kHz and attenuates below the noise floor by 22.05 kHz, but many converter designs do not achieve this
- ***Stopband rejection* (D/A Converter)**
  - Same as for A/D converters
  - Additional problem: intermodulation products in passband. Signal from the D/A converter fed to a (power) amplifier which may have nonlinearity, especially at high frequencies where the open loop gain is falling.

Audio-Only DVDs

- **Sampling rate of 96 kHz with resolution of 24 bits**
  - Dynamic range of 6.02 B + 1.17 = 145.17 dB
  - Marketing ploy to get people to buy more disks
- **Cannot provide better performance than CD**
  - Hearing limited to 20 kHz: sampling rates > 40 kHz wasted
  - Dynamic range in typical living room is 70 dB SPL
    - **Noise floor 40 dB Sound Pressure Level (SPL)**
    - **Most loudspeakers will not produce even 110 dB SPL**
  - Dynamic range in a quiet room less than 80 dB SPL
  - No audio A/D or D/A converter has true 24-bit performance
- **Why not release a tiny DVD with the same capacity as a CD, with CD format audio on it?**

Super Audio CD (SACD) Format

- **One-bit digital audio bitstream**
  - Being promoted by Sony and Philips (CD patents expired)
  - SACD player uses a green laser (rather than CD's infrared)
  - Dual-layer format for play on an ordinary CD player
- **Direct Stream Digital (DSD) bitstream**
  - Produced by 1-bit 5th-order sigma-delta converter operating at 2.8224 MHz (oversampling ratio of 64 vs. CD sampling)
  - Problems with 1-bit converters: distortion, noise modulation, and high out-of-band noise power.
- **Problems with 1-bit stream (S. Lipshitz, AES 2000)**
  - Cannot properly add dither without overloading quantizer
  - Suffers from distortion, noise modulation, and idle tones

Conclusion on Audio Formats

- **Audio CD format**
  - Fine as a delivery format
  - Converters have some room for improvement
- **Audio DVD format**
  - Not justified from audio perspective
  - Appears to be a marketing ploy
- **Super Audio CD format**
  - Good specifications on paper
  - Not needed: conventional audio CD is more than adequate
  - 1-bit quantization cannot be made to work correctly
  - Another marketing ploy (17-year patents expiring)

EE445S Real-Time Digital Signal Processing Lab Spring 2014

Channel Impairments

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

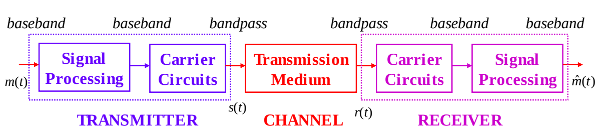
Lecture 12 http://courses.utexas.edu/

Outline

- Analog communication systems
- Channel impairments
- Hybrid communication systems
- Analog pulse amplitude modulation

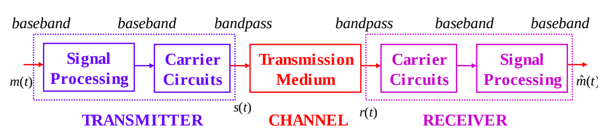
Communication System Structure

- **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



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**

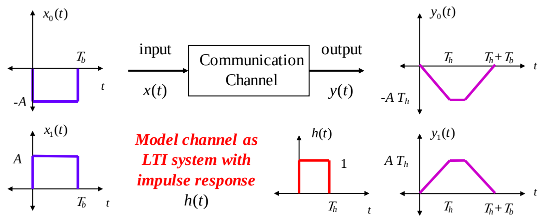


Wireline Channel Impairments

- **Linear time-invariant effects**

  Attenuation: dependent on channel frequency response

  Spreading: finite extent of each transmitted pulse increases



Bit of '0' or '1'

Assume that $T_h < T_b$

Wireline Channel Impairments

- **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.

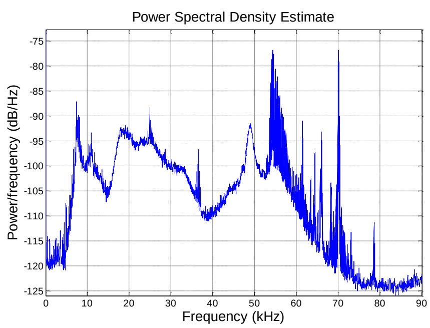
- **Nonlinear effects**

  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)

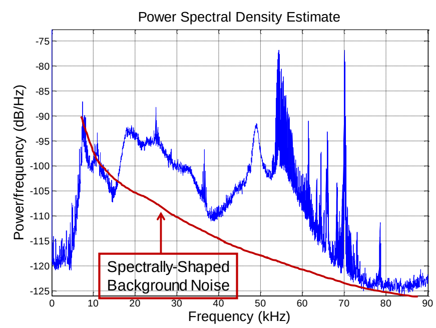
  Additive interference: arises from other systems operating in transmission band (e.g. microwave oven in 2.4 GHz band)

Home Power Line Noise/Interference



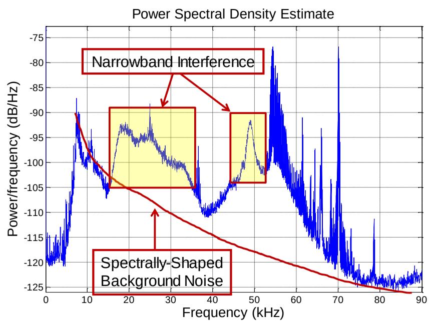
Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011

Home Power Line Noise/Interference



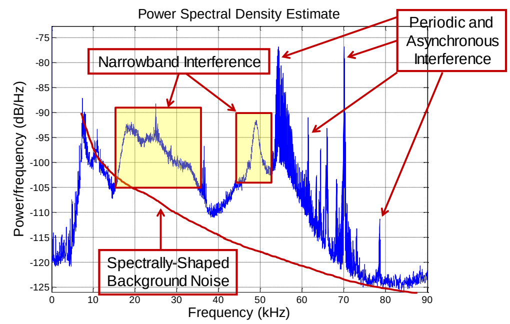
Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011

Home Power Line Noise/Interference



Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011

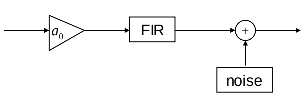
Home Power Line Noise/Interference



Measurement taken on a wall power plug in an apartment in Austin, Texas, on March 20, 2011

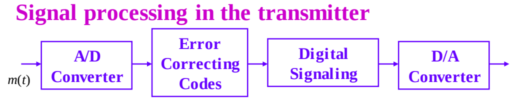
Wireless Channel Impairments

- **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
- **Simplified channel model for fading, LTI effects and additive noise**

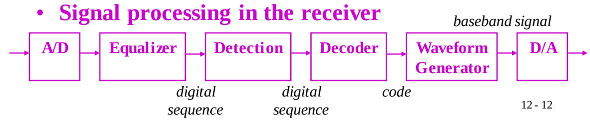


Hybrid Communication Systems

- **Mixed analog and digital signal processing in the transmitter and receiver**
  - Example: message signal is digital but broadcast over an analog channel (compressed speech in digital cell phones)
- **Signal processing in the transmitter**



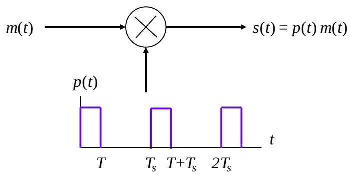
- **Signal processing in the receiver**



Optional

Pulse Amplitude Modulation (PAM)

- **Amplitude of periodic pulse train is varied with a sampled message signal $m(t)$**
  - **Digital PAM**: coded pulses of the sampled and quantized message signal are transmitted (lectures 13 and 14)
  - **Analog PAM**: periodic pulse train with period T_s is the carrier (below)



Optional

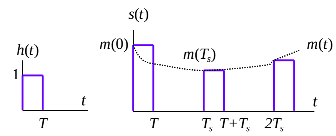
Analog PAM

- **Pulse amplitude varied with amplitude of sampled message**
  - Sample message every T_s
  - Hold sample for T seconds ($T < T_s$)
  - Bandwidth $\propto 1/T$
- **Transmitted signal**

$$s(t) = \sum_{n=-\infty}^{\infty} \underbrace{m(T_s\, n)}_{sample}\ \underbrace{h(t - T_s n)}_{hold}$$

  - $h(t)$ is a rectangular pulse of duration T units

$$h(t) = \begin{cases} 1 & \text{for } 0 < t < T \\ 1/2 & \text{for } t = 0, t = T \\ 0 & \text{otherwise} \end{cases}$$



As $T \to 0$, $\frac{1}{T} h(t) \to \delta(t)$

Optional

Analog PAM

- **Transmitted signal**

$$\begin{aligned} s(t) &= \sum_{n=-\infty}^{\infty} m(T_s\, n)\, h(t - T_s\, n) \\ &= \sum_{n=-\infty}^{\infty} m(T_s\, n) \left(\delta(t - T_s\, n) * h(t)\right) \\ &= \underbrace{\left[\sum_{n=-\infty}^{\infty} m(T_s\, n)\, \delta(t - T_s\, n)\right]}_{m_{sampled}(t)} * h(t) \end{aligned}$$

- **Fourier transform**

$$\begin{aligned} S(f) &= M_{\text{sampled}}(f)\, H(f) \\ &= f_s \sum_{k=-\infty}^{\infty} M(f - f_s\, k)\, H(f) \end{aligned}$$

$$\begin{aligned} H(f) &= T \operatorname{sinc}(\pi\, f\, T)\, e^{-j\, 2\pi\, f\, T/2} \\ &= T \operatorname{sinc}(\pi\, f\, T)\, e^{-j\pi\, f\, T} \end{aligned}$$

- **Equalization of sample and hold distortion added in transmitter**
  - $H(f)$ causes amplitude distortion and delay of $T/2$
  - Equalize amplitude distortion by post-filtering with magnitude response

$$\frac{1}{H(f)} = \frac{1}{T \operatorname{sinc}(\pi\, f\, T)} = \frac{\pi\, f}{\sin(\pi\, f\, T)}$$

  - Negligible distortion (less than 0.5%) if $\frac{T}{T_s} \leq 0.1$

Optional

Analog PAM

- **Requires transmitted pulses to**
  - *Not* be significantly corrupted in amplitude
  - Experience roughly uniform delay
- **Useful in time-division multiplexing**
  - public switched telephone network T1 (E1) line time-division multiplexes 24 (32) voice channels
  - Bit rate of 1.544 (2.048) Mbps for duty cycle < 10%
- **Other analog pulse modulation methods**
  - Pulse-duration modulation (PDM), a.k.a. pulse width modulation (PWM)
  - Pulse-position modulation (PPM): used in some optical pulse modulation systems.

Digital Pulse Amplitude Modulation (PAM)

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

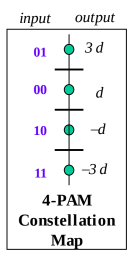
Lecture 13

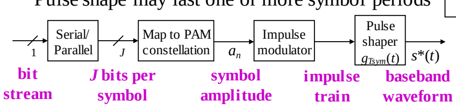
Outline

- **Introduction**
- **Pulse shaping**
- **Pulse shaping filter bank**
- **Design tradeoffs**
- **Symbol recovery**

Introduction

- **Convert bit stream into pulse stream**
  - Group stream of bits into symbols of J bits
  - Represent symbol of bits by unique amplitude
  - Scale pulse shape by amplitude
- **M-level PAM or simply M-PAM ($M = 2^J$)**
  - Symbol period is T_{sym} and bit rate is $J\, f_{sym}$
  - Impulse train has impulses separated by T_{sym}
  - Pulse shape may last one or more symbol periods





Pulse Shaping

- **Without pulse shaping**
  - One impulse per symbol period
  - Infinite bandwidth used (not practical)

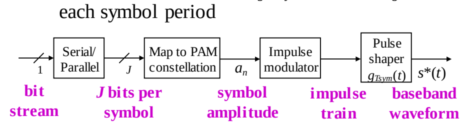
$$s^*(t) = \sum_{k=-\infty}^{\infty} a_k\, \delta(t - k\, T_{sym})$$

k is a symbol index

- **Limit bandwidth by pulse shaping (FIR filtering)**
  - Convolution of discrete-time signal a_k and continuous-time pulse shape

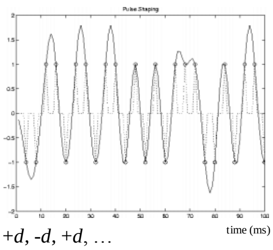
$$s^*(t) = \sum_{k=-\infty}^{\infty} a_k\, g_{T_{sym}}\left(t - k\, T_{sym}\right)$$

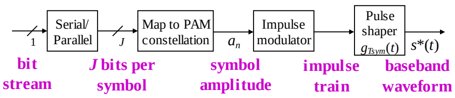
  - For a pulse shape lasting $N_g\, T_{sym}$ seconds, N_g pulses overlap in each symbol period



2-PAM Transmission

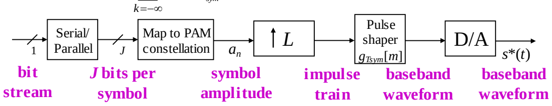
- **2-PAM example (*right*)**
  Raised cosine pulse with peak value of 1
  What are d and T_{sym}?
  How does maximum amplitude relate to d?
- **Highest frequency ½ f_{sym}**
  Alternating symbol amplitudes $+d$, $-d$, $+d$, …



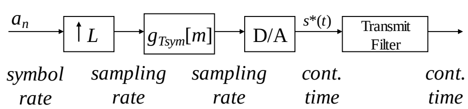


PAM Transmission

- **Transmitted signal**
  $$s^*(t) = \sum_{k=-\infty}^{\infty} a_k \, g_{T_{sym}}\left(t - k\,T_{sym}\right)$$
- ***Sample* at sampling time T_s : let $t = (n\,L + m)\,T_s$**
  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 nth symbol (i.e., $m = 0, 1, \ldots, L\text{-}1$)
  $$s^*[Ln+m] = \sum_{k=-\infty}^{\infty} a_k \, g_{T_{sym}}[L\,(n-k)+m]$$

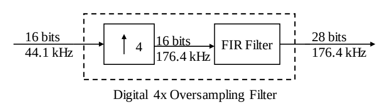


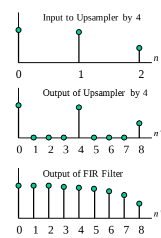
Pulse Shaping Block Diagram



- **Upsampling by L denoted as $\uparrow L$**
  Outputs input sample followed by L-1 zeros
  Upsampling by L converts symbol rate to sampling rate
- **Pulse shaping (FIR) filter $g_{Tsym}[m]$**
  Fills in zero values generated by upsampler
  Multiplies by zero most of time (L-1 out of every L times)

Digital Interpolation Example

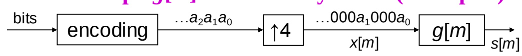




- **Upsampling by 4 (denoted by $\uparrow$ 4)**
  Output input sample followed by 3 zeros
  Four times the samples on output as input
  Increases sampling rate by factor of 4
- **FIR filter performs interpolation**
  Lowpass filter with stopband frequency $\omega_{stopband} \le \pi / 4$
  For f_{sampling} = 176.4 kHz, $\omega = \pi / 4$ corresponds to 22.05 kHz

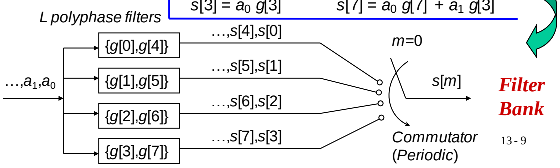
Pulse Shaping Filter Bank Example

- ***L* = 4 samples per symbol**
- **Pulse shape *g*[*m*] lasts for 2 symbols (8 samples)**



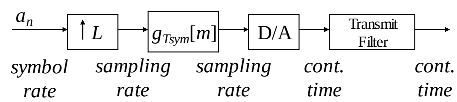
$s[m] = x[m] * g[m]$

$s[0] = a_0\, g[0]$ $\quad s[4] = a_0\, g[4] + a_1\, g[0]$
$s[1] = a_0\, g[1]$ $\quad s[5] = a_0\, g[5] + a_1\, g[1]$
$s[2] = a_0\, g[2]$ $\quad s[6] = a_0\, g[6] + a_1\, g[2]$
$s[3] = a_0\, g[3]$ $\quad s[7] = a_0\, g[7] + a_1\, g[3]$



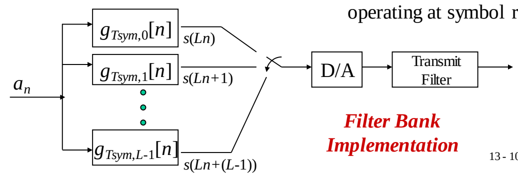
13 - 9

Pulse Shaping Filter Bank



- **Simplify by avoiding multiplication by zero**

  Split long pulse shaping filter into L short polyphase filters operating at symbol rate



13 - 10

Pulse Shaping Filter Bank Example

- **Pulse length 24 samples and *L* = 4 samples/symbol**

$$s^*[L\,n+m] = \sum_{k=n-2}^{n+3} a_k\, g_{T_{sym}}[L\,(n-k)+m]$$

Six pulses contribute to each output sample

- **Derivation: let *t* = (*n* + *m/L*) *T*$_{sym}$**

$$s^*\!\left(nT_{sym} + \frac{m}{L}T_{sym}\right) = \sum_{k=n-2}^{n+3} a_k\, g_{T_{sym}}\!\left(nT_{sym} + \frac{m}{L}T_{sym} - kT_{sym}\right) \qquad m = 0, 1, \ldots, L-1$$

- **Define *m*th polyphase filter**

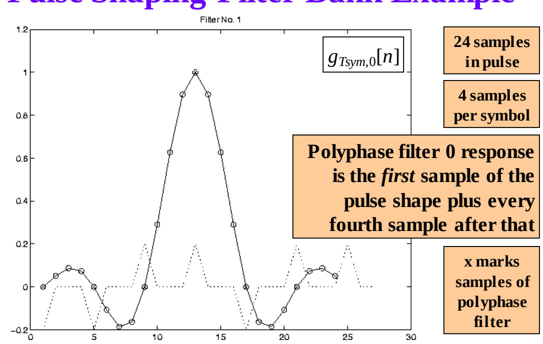
$$g_{T_{sym},m}[n] = g_{T_{sym}}\!\left(nT_{sym} + \frac{m}{L}T_{sym}\right) \qquad m = 0, 1, \ldots, L-1$$

- **Four six-tap polyphase filters (next slide)**

$$s^*\!\left(nT_{sym} + \frac{m}{L}T_{sym}\right) = \sum_{k=n-2}^{n+3} a_k\, g_{T_{sym},m}[n-k]$$

13 - 11

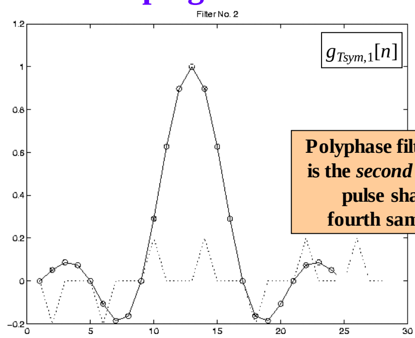
Pulse Shaping Filter Bank Example



Polyphase filter 0 has only one non-zero sample.

13 - 12

Pulse Shaping Filter Bank Example



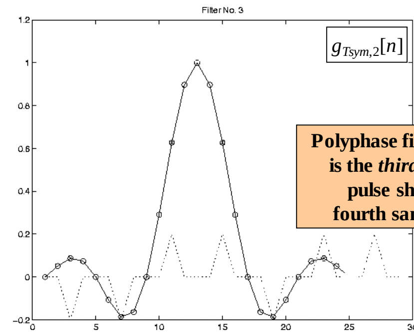
24 samples in pulse

4 samples per symbol

Polyphase filter 1 response is the *second* sample of the pulse shape plus every fourth sample after that

x marks samples of polyphase filter

Pulse Shaping Filter Bank Example



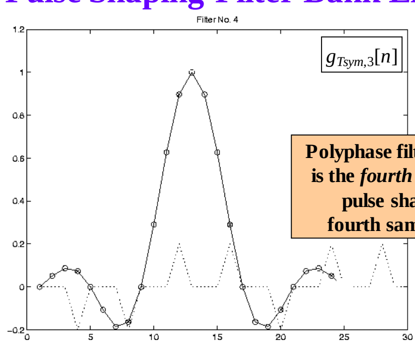
24 samples in pulse

4 samples per symbol

Polyphase filter 2 response is the *third* sample of the pulse shape plus every fourth sample after that

x marks samples of polyphase filter

Pulse Shaping Filter Bank Example



24 samples in pulse

4 samples per symbol

Polyphase filter 3 response is the *fourth* sample of the pulse shape plus every fourth sample after that

x marks samples of polyphase filter

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 13-7)	$(L\ N_g)(L\ f_{sym})$			
Filter bank structure (slide 13-10)	$L\ N_g\ f_{sym}$			

f_{sym} symbol rate
L samples/symbol
N_g duration of pulse shape in symbol periods

Optional

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

Optional

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

$$p(t) = q^2(t) = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} a_k \, a_m \, g_1(t - kT_{sym}) \, g_1(t - mT_{sym})$$

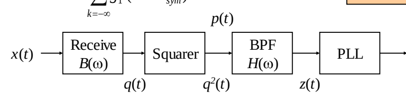
$g_1(t)$ is deterministic

$$E\{p(t)\} = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} E\{a_k \, a_m\} \, g_1(t - kT_{sym}) \, g_1(t - mT_{sym})$$

$E\{a_k \, a_m\} = a^2 \, \delta[k-m]$

$$= a^2 \sum_{k=-\infty}^{\infty} g_1^2(t - kT_{sym})$$

Periodic with period T_{sym}



Optional

Symbol Clock Recovery

- Fourier series representation of E{ $p(t)$ }

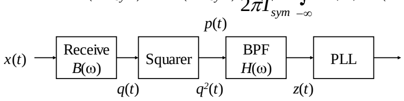
$$E\{p(t)\} = \sum_{k=-\infty}^{\infty} p_k \, e^{j k \omega_{sym} t} \quad \text{where} \quad p_k = \frac{1}{T_{sym}} \int_0^{T_{sym}} E\{p(t)\} e^{-j k \omega_{sym} t} dt$$

- 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{ $z(t)$ }

$$z_k = p_k H(k\omega_{sym}) = H(k\omega_{sym}) \frac{a^2}{2\pi T_{sym}} \int_{-\infty}^{\infty} G_1(\omega) G_1(k\omega_{sym} - \omega) d\omega$$



Optional

Symbol Clock Recovery

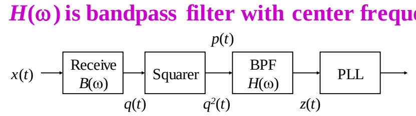
- With $G_1(\omega) = X(\omega) \, B(\omega)$
  - Choose $B(\omega)$ to pass $\pm ½\omega_{sym}$ → $p_k = 0$ except k = -1, 0, 1

$$Z_k = p_k H(k\omega_{sym}) = H(k\omega_{sym}) \frac{a^2}{2\pi \, T_{sym}} \int_{-\infty}^{\infty} G_1(\omega) G_1(k\omega_{sym} - \omega) d\omega$$

  - Choose $H(\omega)$ to pass $\pm\omega_{sym}$ → $Z_k = 0$ 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 $\omega_{passband}$ = ½ ω_{sym}
- $H(\omega)$ is bandpass filter with center frequency ω_{sym}



EE445S Real-Time Digital Signal Processing Lab   Spring 2014

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

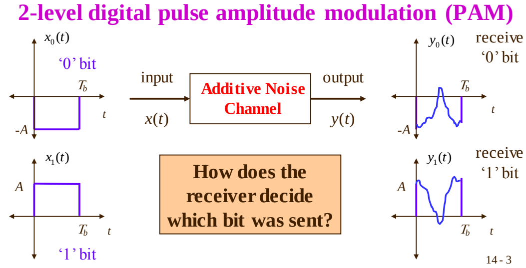
Lecture 14

Outline

- **Transmitting one bit at a time** (Part I)
- **Matched filtering**
- **PAM system** (Part II)
- **Intersymbol interference**
- **Communication performance**
  - Bit error probability for binary signals
  - Symbol error probability for M-ary (multilevel) signals
- **Eye diagram**

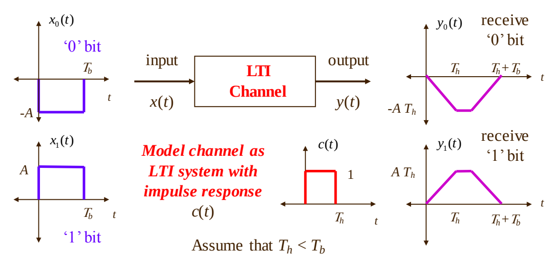
Transmitting One Bit

- **Transmission on communication channels is analog**
- **One way to transmit digital information is called 2-level digital pulse amplitude modulation (PAM)**



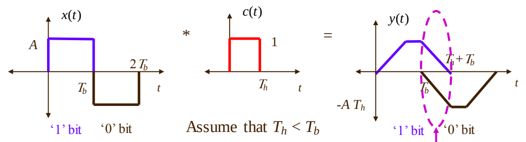
Transmitting One Bit

- **Two-level digital pulse amplitude modulation over channel that has memory but does not add noise**



Transmitting Two Bits (Interference)

- **Transmitting two bits (pulses) back-to-back will cause overlap (interference) at the receiver**

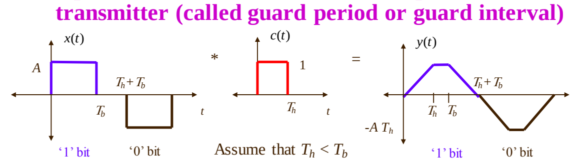


- **Sample $y(t)$ at T_b, $2\ T_b$, …, and threshold with threshold of zero**
- **How do we prevent intersymbol interference (ISI) at the receiver?**

Intersymbol interference

Preventing ISI at Receiver

- **Option #1: wait T_h seconds between pulses in transmitter (called guard period or guard interval)**



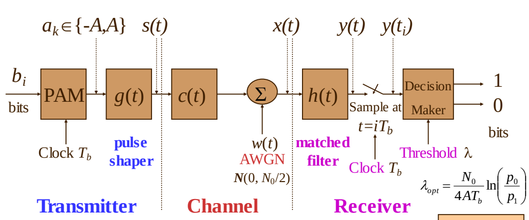
Disadvantages?

- **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

Digital 2-level PAM System



p_0 is the probability bit '0' sent

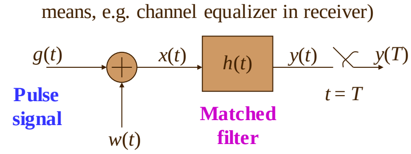
- **Transmitted signal** $s(t) = \sum_k a_k\ g(t - k\ T_b)$
- **Requires synchronization of clocks between transmitter and receiver**

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)



T is the symbol period

Additive white Gaussian noise (AWGN) with zero mean and variance N_0 /2

$$y(t) = g(t) * h(t) + w(t) * h(t) = g_0(t) + n(t)$$

Matched Filter Derivation

- **Design of matched filter**

  Maximize signal power i.e. power of $g_0(t) = g(t) * h(t)$ at $t = T$

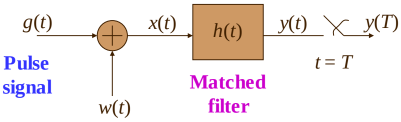
  Minimize noise i.e. power of $n(t) = w(t) * h(t)$

- **Combine design criteria**

  max η, where η is peak pulse SNR

  $$\eta = \frac{|g_0(T)|^2}{E\{n^2(t)\}} = \frac{\text{instantaneous power}}{\text{average power}}$$

T is the symbol period



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

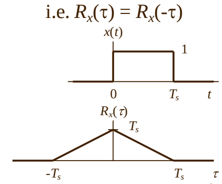
  $$X(f)\, X^*(f) = F\{x(\tau) * x^*(-\tau)\}$$

- **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. $R_x(\tau) = R_x(-\tau)$



Power Spectra

- **Two-sided random signal $n(t)$** $\qquad P_n(f) = F\{R_n(\tau)\}$

  Fourier transform may not exist, but power spectrum exists

  $$R_n(\tau) = E\{n(t)\, n^*(t+\tau)\} = \int_{-\infty}^{\infty} n(t)\, n^*(t+\tau)\, dt$$

  $$R_n(-\tau) = E\{n(t)\, n^*(t-\tau)\} = \int_{-\infty}^{\infty} n(t)\, n^*(t-\tau)\, dt = n(\tau) * n^*(-\tau)$$

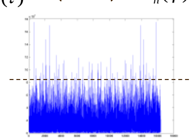
  For zero-mean Gaussian random process $n(t)$ with variance σ^2

  $$R_n(\tau) = E\{n(t)\, n^*(t+\tau)\} = 0 \text{ when } \tau \neq 0$$

  $$R_n(\tau) = E\{n(t)\, n^*(t+\tau)\} = \sigma^2\, \delta(\tau) \iff P_n(f) = \sigma^2$$

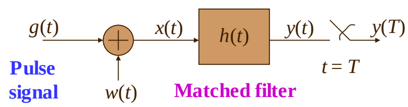
- **Estimate noise power spectrum in Matlab**

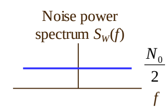
```
N = 16384;   % finite no. of samples
gaussianNoise = randn(N,1);
plot( abs(fft(gaussianNoise)) .^ 2 );
```



approximate noise floor

Matched Filter Derivation





- **Noise** $\quad n(t) = w(t) * h(t) \qquad S_N(f) = \underbrace{S_W(f)}_{\text{AWGN}}\, \underbrace{S_H(f)}_{\text{Matched filter}} = \frac{N_0}{2} |H(f)|^2$

  $$E\{n^2(t)\} = \int_{-\infty}^{\infty} S_N(f)\, df = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df$$

- **Signal** $\quad g_0(t) = g(t) * h(t) \qquad G_0(f) = H(f) G(f)$

  $$g_0(t) = \int_{-\infty}^{\infty} H(f)\, G(f)\, e^{j 2 \pi f t}\, df$$

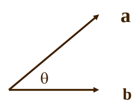
  $$|g_0(T)|^2 = \left| \int_{-\infty}^{\infty} H(f)\, G(f)\, e^{j 2 \pi f T}\, df \right|^2$$

T is the symbol period

Matched Filter Derivation

- **Find $h(t)$ that maximizes pulse peak SNR η**

$$\eta = \frac{\left| \int_{-\infty}^{\infty} H(f)\, G(f)\, e^{j 2\pi f T}\, df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df}$$



- **Schwartz's inequality**

For vectors: $|\mathbf{a}^T \mathbf{b}^*| \le \|\mathbf{a}\|\ \|\mathbf{b}\| \Leftrightarrow \cos\theta = \dfrac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\|\ \|\mathbf{b}\|}$

For functions: $\left| \int_{-\infty}^{\infty} \phi_1(x)\, \phi_2^*(x)\, dx \right|^2 \le \int_{-\infty}^{\infty} |\phi_1(x)|^2 dx \int_{-\infty}^{\infty} |\phi_2(x)|^2 dx$

upper bound reached iff $\phi_1(x) = k\, \phi_2(x)\ \ \forall k \in R$

Matched Filter Derivation

Let $\phi_1(f) = H(f)$ and $\phi_2(f) = G^*(f)\, e^{-j 2\pi f T}$

$$\left| \int_{-\infty}^{\infty} H(f)\, G(f)\, e^{j 2\pi f T}\, df \right|^2 \le \int_{-\infty}^{\infty} |H(f)|^2\, df \int_{-\infty}^{\infty} |G(f)|^2\, df$$

$$\eta = \frac{\left| \int_{-\infty}^{\infty} H(f)\, G(f)\, e^{j 2\pi f T}\, df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df} \le \frac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2\, df$$

T is the symbol period

$\eta_{\max} = \dfrac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2\, df$, which occurs when

$H_{opt}(f) = k\, G^*(f)\, e^{-j 2\pi f T}\ \ \forall k$ by Schwartz 's inequality

Hence, $h_{opt}(t) = k\, g^*(T - t)$

Matched Filter

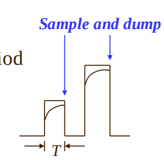
- **Impulse response is $h_{\text{opt}}(t) = k\, g^*(T - t)$**
  - Symbol period T, transmitter pulse shape $g(t)$ and gain k
  - Scaled, conjugated, time-reversed, and shifted version of $g(t)$
  - Duration and shape determined by pulse shape $g(t)$
- **Maximizes peak pulse SNR**

$$\eta_{\max} = \frac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2\, df = \frac{2}{N_0} \int_{-\infty}^{\infty} |g(t)|^2\, dt = \frac{2 E_b}{N_0} = \text{SNR}$$

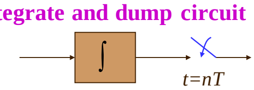
  - Does not depend on pulse shape $g(t)$
  - Proportional to signal energy (energy per bit) E_b
  - Inversely proportional to power spectral density of noise

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

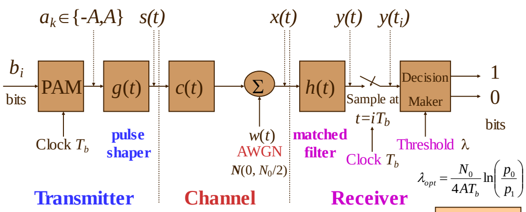


- **Integrate and dump circuit**



$h(t)$ = ____

Digital 2-level PAM System



- **Transmitted signal** $s(t) = \sum_k a_k \, g(t - k\,T_b)$
- **Requires synchronization of clocks between transmitter and receiver**

p_0 is the probability bit '0' sent

Digital 2-level PAM System

- **Why is $g(t)$ a pulse and not an impulse?**

  Otherwise, $s(t)$ would require infinite bandwidth

  $$s(t) = \sum_k a_k \delta(t - k\,T_b)$$

  We limit its bandwidth by using a pulse shaping filter
- **Neglecting noise, would like $y(t) = g(t) * c(t) * h(t)$ to be a pulse, i.e. $y(t) = \mu\, p(t)$, to eliminate ISI**

$$y(t) = \mu \sum_k a_k\, p(t - kT_b) + n(t) \;\; \text{where} \;\; n(t) = w(t) * h(t)$$

$$\Rightarrow y(t_i) = \underbrace{\mu\, a_i \;\, p(t_i - iT_b)}_{\substack{\text{actual value} \\ \text{(note that } t_i = i\,T_b)}} + \underbrace{\mu \sum_{k, k \neq i} a_k \;\, p\big((i-k)T_b\big)}_{\substack{\text{intersymbol} \\ \text{interference (ISI)}}} + \underbrace{n(t_i)}_{\text{noise}}$$

$p(t)$ is centered at origin

Eliminating ISI in PAM

- **One choice for $P(f)$ is a rectangular pulse**

  W is the bandwidth of the system

  Inverse Fourier transform of a rectangular pulse is is a sinc function

  $p(t) = \text{sinc}(2\,\pi\,W\,t)$

$$P(f) = \begin{cases} \dfrac{1}{2\,W} & , -W < f < W \\ 0 & , |\,f\,| > W \end{cases}$$

$$P(f) = \frac{1}{2\,W}\,\text{rect}\left(\frac{f}{2\,W}\right)$$

- **This is called the Ideal Nyquist Channel**
- **It is not realizable because pulse shape is not causal and is infinite in duration**

Eliminating ISI in PAM

- **Another choice for $P(f)$ is a raised cosine spectrum**

$$P(f) = \begin{cases} \dfrac{1}{2\,W} & 0 \le |\,f\,| < f_1 \\ \dfrac{1}{4W}\left(1 - \sin\left(\dfrac{\pi(|\,f\,| - W)}{2W - 2\,f_1}\right)\right) & f_1 \le |\,f\,| < 2W - f_1 \\ 0 & 2W - f_1 \le |\,f\,| \le 2W \end{cases}$$

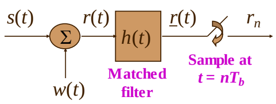
- **Roll-off factor gives bandwidth in excess of bandwidth W for ideal Nyquist channel** $\alpha = 1 - \dfrac{f_1}{W}$
- **Raised cosine pulse has zero ISI when sampled correctly**

$$p(t) = \underbrace{\text{sinc}\left(\frac{t}{T_s}\right)}_{\substack{\text{ideal Nyquist channel} \\ \text{impulse response}}} \underbrace{\frac{\cos(2\pi\,\alpha\,W\,t)}{1 - 16\,\alpha^2\,W^2\,t^2}}_{\substack{\text{dampening adjusted by} \\ \text{rolloff factor } \alpha}}$$

- **Let $g(t)$ and $h(t)$ be square root raised cosine pulses**

Bit Error Probability for 2-PAM

- T_b **is bit period (bit rate is** $f_b = 1/T_b$**)**



$$s(t) = \sum_k a_k \, g(t - k\,T_b)$$

$$r(t) = s(t) + w(t)$$

$$\underline{r}(t) = h(t) * r(t)$$

$w(t)$ is AWGN 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 ½** f_b

14 - 21

Bit Error Probability for 2-PAM

- **Noise power at matched filter output**

Filtered noise **$T = T_{sym}$**

$$v(nT) = \int_{-\infty}^{\infty} h(\tau) w(nT - \tau) d\tau$$

Noise power

$$E\{v^2(nT)\} = E\left\{\left[\int_{-\infty}^{\infty} h(\tau) w(nT - \tau) d\tau\right]^2\right\}$$

$$= E\{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_T(\tau_1) w(nT - \tau_1) g_T(\tau_2) w(nT - \tau_2) d\tau_1 d\tau_2\}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_T(\tau_1) g_T(\tau_2) \underbrace{E\{w(nT - \tau_1) w(nT - \tau_2)\}}_{\sigma^2\,\delta(\tau_1 - \tau_2)} d\tau_1 d\tau_2$$

$$= \sigma^2 \int_{-\infty}^{\infty} h^2(\tau) d\tau = \sigma^2 \frac{1}{2\pi} \int_{-\omega_{sym}/2}^{\omega_{sym}/2} H^2(\omega) d\omega = \frac{\sigma^2}{T}$$

14 - 22

Bit Error Probability for 2-PAM

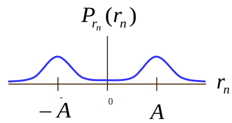
- **Symbol amplitudes of** *+A* **and** *-A*
- **Rectangular pulse shape with amplitude 1**
- **Bit duration (**T_b**) of 1 second**
- **Matched filtering with gain of one (*see slide 14-15*)**

  Integrate received signal over *n*th bit period and sample

$$r_n = \int_n^{n+1} r(t)\, dt$$

$$= \pm A + \int_n^{n+1} w(t)\, dt$$

$$= \pm A + v_n$$



Probability density function (PDF)

14 - 23

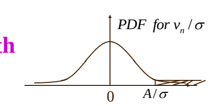
Bit Error Probability for 2-PAM

- **Probability of error given that transmitted pulse has amplitude** *–A*

$T_b = 1$

$$P(\text{error} \mid s(nT_b) = -A) = P(-A + v_n > 0) = P(v_n > A) = P\left(\frac{v_n}{\sigma} > \frac{A}{\sigma}\right)$$

- **Random variable** $\frac{v_n}{\sigma}$ **is Gaussian with zero mean and variance of one**



PDF for ***N*(0, 1)**

$$P(\text{error} \mid s(nT) = -A) = P\left(\frac{v_n}{\sigma} > \frac{A}{\sigma}\right) = \int_{\frac{A}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}}\, dv = Q\left(\frac{A}{\sigma}\right)$$

Q function on next slide

14 - 24

Q Function

- **Q function**

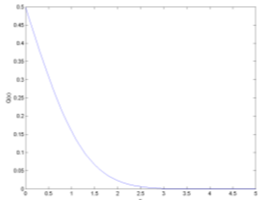
$$Q(x)=\frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-y^2/2}dy$$

- **Complementary error function erfc**

$$erfc(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}dt$$

- **Relationship**

$$Q(x)=\frac{1}{2}erfc\left(\frac{x}{\sqrt{2}}\right)$$



Erfc[x] in Mathematica

erfc(x) in Matlab

14 - 25

Bit Error Probability for 2-PAM

$T_b = 1$

- **Probability of error given that transmitted pulse has amplitude A**

$$P(\text{error} \mid s(nT_b)=A)=Q(A/\sigma)$$

- **Assume that 0 and 1 are equally likely bits**

$$P(\text{error})=P(A)P(\text{error} \mid s(nT_b)=A)+P(-A)P(\text{error} \mid s(nT_b)=-A)$$
$$=\frac{1}{2}Q\left(\frac{A}{\sigma}\right)+\frac{1}{2}Q\left(\frac{A}{\sigma}\right)=Q\left(\frac{A}{\sigma}\right)=Q\left(\sqrt{\rho}\right)$$

where, $\rho = \text{SNR} = \frac{A^2}{\sigma^2}$

$$Q(\sqrt{\rho}) \le \frac{1}{\sqrt{2\pi}}\frac{e^{-\frac{\rho}{2}}}{\sqrt{\rho}}$$

for large positive ρ

- **Probability of error exponentially decreases with SNR (see *slide 8-16*)**

14 - 26

PAM Symbol Error Probability

- **Set symbol time (T_{sym}) to 1 second**
- **Average transmitted signal power**

$$P_{Signal}=E\{a_n^2\}\times\frac{1}{2\pi}\int_{-\infty}^{\infty}|G_T(\omega)|^2\,d\omega=E\{a_n^2\}$$

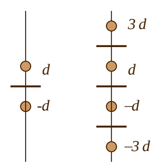
$G_T(\omega)$ square root raised cosine spectrum

- ***M*-level PAM symbol amplitudes**

$$l_i=d(2i-1),\quad i=-\frac{M}{2}+1,\ldots,0,\ldots,\frac{M}{2}$$

- **With each symbol equally likely**

$$P_{Signal}=\frac{1}{M}\sum_{i=-\frac{M}{2}+1}^{\frac{M}{2}} l_i^2=\frac{2}{M}\sum_{i=1}^{\frac{M}{2}}\left(d(2i-1)\right)^2=(M^2-1)\frac{d^2}{3}$$



Constellation points with receiver decision boundaries

14 - 27

PAM Symbol Error Probability

- **Noise power and SNR**

$$P_{Noise}=\frac{1}{2\pi}\int_{-\omega_{sym}/2}^{\omega_{sym}/2}\underbrace{\frac{N_0}{2}}_{\text{two-sided power spectral density of AWGN}}d\omega=\frac{N_0}{2}$$

$$\text{SNR}=\frac{P_{Signal}}{P_{Noise}}=\frac{2(M^2-1)}{3}\times\frac{d^2}{N_0}$$

- **Assume ideal channel, i.e. one without ISI**

$$r_n=a_n+\underbrace{v_n}_{\text{channel noise after matched filtering and sampling}}$$

- **Consider M-2 inner levels in constellation**

Error only if $|v_n|>d$

where $\sigma^2=N_0/2$

Probability of error is

$$P(|v_n|>d)=2\,Q\left(\frac{d}{\sigma}\right)$$

- **Consider two outer levels in constellation**

$$P(v_n>d)=Q\left(\frac{d}{\sigma}\right)$$

14 - 28

PAM Symbol Error Probability

- Assuming that each symbol is equally likely, symbol error probability for M-level PAM

$$P_e = \underbrace{\frac{M-2}{M}\left(2\,Q\left(\frac{d}{\sigma}\right)\right)}_{\textit{M-2 interior points}} + \underbrace{\frac{2}{M}Q\left(\frac{d}{\sigma}\right)}_{\textit{2 exterior points}} = \frac{2(M-1)}{M}Q\left(\frac{d}{\sigma}\right)$$

- Symbol error probability in terms of SNR

$$P_e = 2\frac{M-1}{M}Q\left(\left(\frac{3}{M^2-1}\text{SNR}\right)^{\frac{1}{2}}\right) \quad \text{since SNR} = \frac{P_{Signal}}{P_{Noise}} = \frac{d^2}{3\sigma^2}\left(M^2-1\right)$$

Visualizing ISI

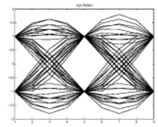
- Eye diagram is empirical measure of signal quality

$$x(n) = \sum_{k=-\infty}^{\infty} a_k\, g(n\,T_{sym} - k\,T_{sym}) = g(0)\left(a_n + \sum_{\substack{k=-\infty \\ k \neq n}}^{+\infty} a_k \frac{g(nT_{sym} - kT_{sym})}{g(0)}\right)$$

- Intersymbol interference (ISI):

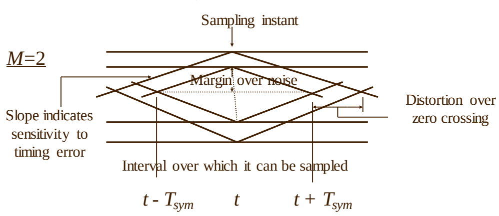
$$D \le (M-1)\,d \sum_{k=-\infty,k\neq n}^{\infty} \left|\frac{g(nT_{sym} - kT_{sym})}{g(0)}\right| = (M-1)\,d \sum_{k=-\infty,k\neq n}^{\infty} \left|\frac{g(kT_{sym})}{g(0)}\right|$$

Raised cosine filter has zero ISI when correctly sampled



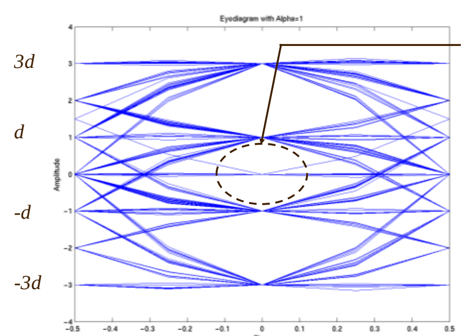
Eye Diagram for 2-PAM

- Useful for PAM transmitter and receiver analysis and troubleshooting



- The more open the eye, the better the reception

Eye Diagram for 4-PAM



Due to startup transients.

Fix is to discard first few symbols equal to number of symbol periods in pulse shape.

Quadrature Amplitude Modulation (QAM) Transmitter

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

Lecture 15

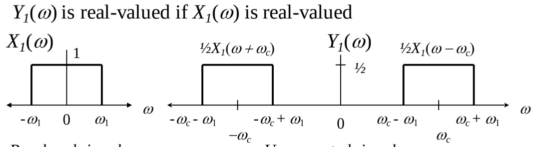
Introduction

- **Digital Pulse Amplitude Modulation (PAM)**
  - Modulates digital information 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 frequency)
- **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

Review

Amplitude Modulation by Cosine

- $y_1(t) = x_1(t)\cos(\omega_c t)$ $\quad Y_1(\omega) = \frac{1}{2}X_1(\omega+\omega_c) + \frac{1}{2}X_1(\omega-\omega_c)$
  - Assume $x_1(t)$ is an ideal lowpass signal with bandwidth ω_1
  - Assume $\omega_1 << \omega_c$
  - $Y_1(\omega)$ is real-valued if $X_1(\omega)$ is real-valued



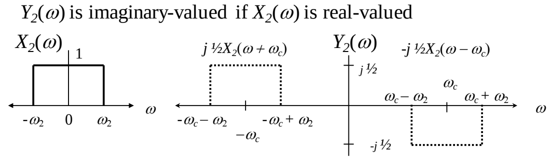
Baseband signal *Upconverted signal*

- **Demodulation: modulation then lowpass filtering**

Review

Amplitude Modulation by Sine

- $y_2(t) = x_2(t)\sin(\omega_c t)$ $\quad Y_2(\omega) = \frac{j}{2}X_2(\omega+\omega_c) - \frac{j}{2}X_2(\omega-\omega_c)$
  - Assume $x_2(t)$ is an ideal lowpass signal with bandwidth ω_2
  - Assume $\omega_2 << \omega_c$
  - $Y_2(\omega)$ is imaginary-valued if $X_2(\omega)$ is real-valued

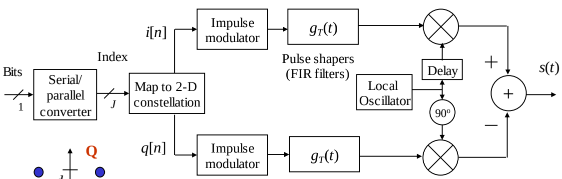


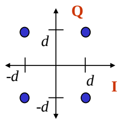
Baseband signal *Upconverted signal*

- **Demodulation: modulation then lowpass filtering**

Baseband Digital QAM Transmitter

- **Continuous-time filtering and upconversion**





4-level QAM Constellation

Delay matches delay through 90° phase shifter

Delay required but often omitted in diagrams

Phase Shift by 90 Degrees

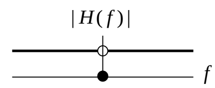
- **90° phase shift performed by Hilbert transformer**

cosine => sine $\quad \cos(2\pi\, f_0\, t) \Rightarrow \frac{1}{2}\delta(f + f_0) + \frac{1}{2}\delta(f - f_0)$

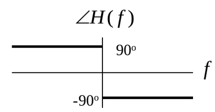
sine => – cosine $\quad \sin(2\pi\, f_0\, t) \Rightarrow \frac{j}{2}\delta(f + f_0) - \frac{j}{2}\delta(f - f_0)$

- **Frequency response** $\boxed{H(f) = -j\,\mathrm{sgn}(f)}$

Magnitude Response



Phase Response



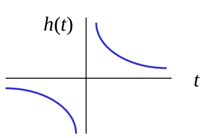
All-pass except at origin

Hilbert Transformer

- **Continuous-time ideal Hilbert transformer**

$H(f) = -j\,\mathrm{sgn}(f)$

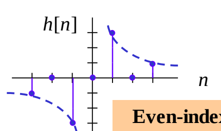
$$h(t) = \begin{cases} 1/(\pi\, t) & \text{if } t \neq 0 \\ 0 & \text{if } t = 0 \end{cases}$$



- **Discrete-time ideal Hilbert transformer**

$H(\omega) = -j\,\mathrm{sgn}(\omega)$

$$h[n] = \begin{cases} \dfrac{2}{\pi}\dfrac{\sin^2(\pi n/2)}{n} & \text{if } n \neq 0 \\ 0 & \text{if } n = 0 \end{cases}$$



Even-indexed samples are zero

Discrete-Time Hilbert Transformer

- **Approximate by odd-length linear phase FIR filter**

Truncate response to 2 L + 1 samples: L samples left of origin, L samples right of origin, and origin

Shift truncated impulse response by L samples to right to make it causal

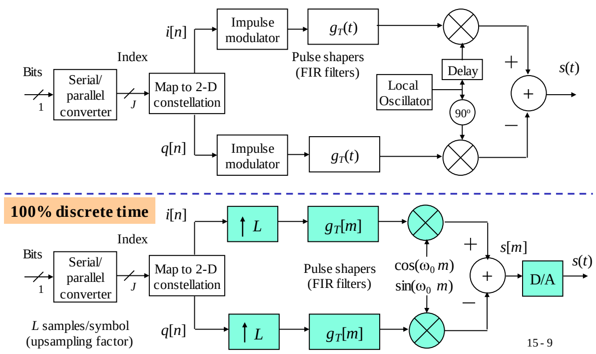
L is odd because every other sample of impulse response is 0

- **Linear phase FIR filter of length *N* has same phase response as an ideal delay of length (*N*-1)/2**

(N-1)/2 is an integer when N is odd (here N = 2 L + 1)

- **Matched delay block on slide 15-5 would be an ideal delay of *L* samples**

Baseband Digital QAM Transmitter



Performance Analysis of PAM

- **If we sample matched filter output at correct time instances, nT_{sym}, without any ISI, received signal**

$x(nT_{sym}) = s(nT_{sym}) + v(nT_{sym})$ **$v(nT) \sim N(0;\ \sigma^2/T_{sym})$**

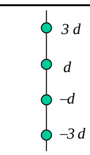
where transmitted signal is

$s(nT_{sym}) = a_n = (2i-1)d$ for $i = -M/2+1, \ldots, M/2$

$v(t)$ 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

Performance Analysis of PAM

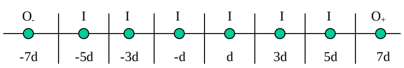
- **Decision error for inner points** $P_I(e) = P(|v(nT_{sym})| > d) = 2\,Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$
- **Decision error for outer points** $P_{O_-}(e) = P(v(nT_{sym}) > d) = Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$

$P_{O_+}(e) = P(v(nT_{sym}) < -d) = P(v(nT_{sym}) > d) = Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$

- **Symbol error probability**

$$P(e) = \frac{M-2}{M} P_I(e) + \frac{1}{M} P_{O_+}(e) + \frac{1}{M} P_{O_-}(e) = \frac{2(M-1)}{M} Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$

8-level PAM Constellation



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**

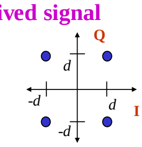
$s(nT_{sym}) = a_n + j\, b_n = (2i-1)d + j\,(2k-1)d$

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

Performance Analysis of 16-QAM

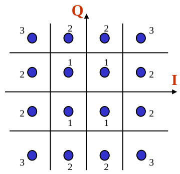
- **Type 1 correct detection**

$$P_1(c) = P\left(|v_I(nT_{sym})| < d \;\&\; |v_Q(nT_{sym})| < d\right)$$

$$= P\left(|v_I(nT_{sym})| < d\right) P\left(|v_Q(nT_{sym})| < d\right)$$

$$= \Big(1 - \underbrace{P\left(|v_I(nT_{sym})| > d\right)}_{2Q(\frac{d}{\sigma}\sqrt{T})}\Big)\Big(1 - \underbrace{P\left(|v_Q(nT_{sym})| > d\right)}_{2Q(\frac{d}{\sigma}\sqrt{T})}\Big)$$

$$= \left(1 - 2Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right)^2$$



16-QAM

1 - interior decision region
2 - edge region but not corner
3 - corner region

Performance Analysis of 16-QAM

- **Type 2 correct detection**

$$P_2(c) = P(v_I(nT_{sym}) < d \;\&\; |v_Q(nT_{sym})| < d)$$

$$= P(v_I(nT_{sym}) < d) P(|v_Q(nT_{sym})| < d)$$

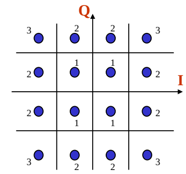
$$= \left(1 - Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right)\left(1 - 2Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right)$$

- **Type 3 correct detection**

$$P_3(c) = P(v_I(nT_{sym}) < d \;\&\; v_Q(nT_{sym}) > -d)$$

$$= P(v_I(nT_{sym}) < d) P(v_Q(nT_{sym}) > -d)$$

$$= \left(1 - Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right)^2$$



16-QAM

1 - interior decision region
2 - edge region but not corner
3 - corner region

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 QAM constellations?**

15 - 15

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 9 d^2 + d^2 + d^2 + 9 d^2 = 20 d^2

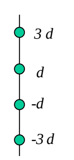
  Average power per symbol 5 d^2

- **4-QAM constellation points**

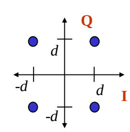
  Points are in set { $-d - jd$, $-d + jd$, $d + jd$, $d - jd$ }

  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

Quadrature Amplitude Modulation (QAM) Receiver

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

Lecture 16

Outline

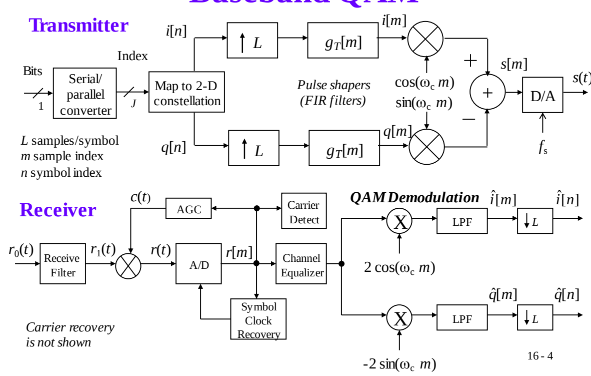
- **Introduction**
- **Automatic gain control**
- **Carrier detection**
- **Symbol clock recovery**
- **Channel equalization**
- **QAM demodulation**

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**

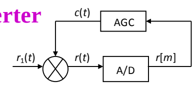
Fading	Automatic gain control (AGC)
Additive noise	Matched filters
Linear distortion	Channel equalizer
Carrier mismatch	Carrier recovery
Symbol timing mismatch	Symbol clock recovery

Baseband QAM



Automatic Gain Control

- **Scales input voltage to A/D converter**
  Increase gain for low signal level
  Decrease gain for high signal level



- **Consider A/D converter with 8-bit signed output**
  When $c(t)$ is zero, A/D output is 0
  When $c(t)$ is infinity, A/D output is -128 or 127
  Let f_{-128}, f_0 and f_{127} represent how frequently outputs -128, 0 and 127 occur over a window of previous samples
  Each frequency value is between 0 and 1, inclusive
  Update: $c(t) = (1 + 2\, f_0 - f_{-128} - f_{127})\, c(t - \tau)$
  Initial values: $f_{-128} = f_0 = f_{127} = 1 / 256$. Zero also works.

16 - 5

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 - 6

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)

Pole locations?

- **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 $|\omega_{sym}\, \tau| << 1$

See Reader handout M

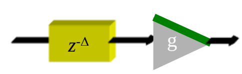
  Smooth timing error estimate to compute phase advancement
  $p[n] = \beta\; p[n-1] + \alpha\; v[n]$

Lowpass IIR filter

16 - 7

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**



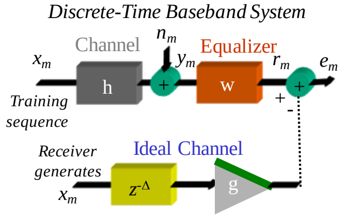
  Cascade of delay Δ and gain g
  Impulse response: impulse delayed by Δ with amplitude g
  Frequency response: allpass and linear phase (no distortion)
  Undo effects by discarding Δ samples and scaling by $1/g$

16 - 8

Channel Equalizer

- **IIR equalizer**
  - Ignore noise n_m
  - Set error e_m to zero
  - $H(z)\,W(z) = g\,z^{-\Delta}$
  - $W(z) = g\,z^{-\Delta} / H(z)$
  - Issues?

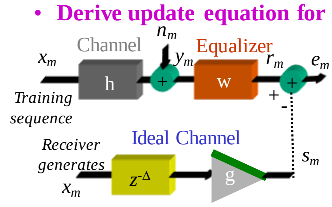
Discrete-Time Baseband System



- **FIR equalizer**
  - Adapt equalizer coefficients when transmitter sends training sequence to reduce measure of error, e.g. square of e_m

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**



$e[m] = r[m] - s[m]$

$s[m] = g\,x[m-\Delta]$

$r[m] = y[m] + w_1\,y[m-1]$

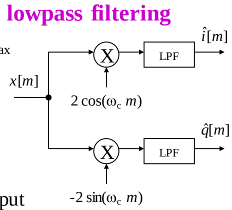
$w_1[m+1] = w_1[m] - \mu \left.\dfrac{\partial J_{LMS}[m]}{\partial w_1}\right|_{w_1 = w_1[m]}$

Using least mean squares (LMS) $\quad J_{LMS}[m] = \dfrac{1}{2} e^2[m]$

Step size $0 < \mu < 1$ $\quad w_1[m+1] = w_1[m] - \mu\, e[m]\, y[m-1]$

Baseband QAM Demodulation

- **Recovers baseband in-phase/quadrature signals**
- **Assumes perfect AGC, equalizer, symbol recovery**
- **QAM modulation followed by lowpass filtering**
  - Receiver $f_{max} = 2\,f_c + B$ and $f_s > 2\,f_{max}$
- **Lowpass filter has other roles**
  - Matched filter
  - Anti-aliasing filter



- **Matched filters**
  - Maximize SNR at downsampler output
  - Hence minimize symbol error at downsampler output

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

$$\hat{i}[m] = 2x[m]\cos(\omega_c m) = 2i[m]\cos^2(\omega_c t) - 2q[m]\sin(\omega_c m)\cos(\omega_c m)$$

$$= \underbrace{i[m]}_{\text{baseband}} + \underbrace{i[m]\cos(2\omega_c m) - q[m]\sin(2\omega_c m)}_{\text{high frequency component centered at } 2\,\omega_c}$$

$$\hat{q}[m] = -2x[m]\sin(\omega_c m) = -2i[m]\cos(\omega_c m)\sin(\omega_c m) + 2q[m]\sin^2(\omega_c m)$$

$$= \underbrace{q[m]}_{\text{baseband}} - \underbrace{i[m]\sin(2\omega_c m) - q[m]\cos(2\omega_c m)}_{\text{high frequency component centered at } 2\,\omega_c}$$

$\cos^2\theta = \frac{1}{2}(1+\cos 2\theta) \qquad 2\cos\theta\sin\theta = \sin 2\theta \qquad \sin^2\theta = \frac{1}{2}(1-\cos 2\theta)$

EE445S Real-Time Digital Signal Processing Lab Spring 2014

Fast Fourier Transform

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

Lecture 17

Discrete-Time Fourier Transform

- **Forward transform of discrete-time signal $x[n]$**

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\,\omega\, n}$$

Assumes that $x[n]$ is two-sided and infinite in duration

Produces $X(\omega)$ that is periodic in ω (in units of rad/sample) with period 2π due to exponential term

- **Inverse discrete-time Fourier transform** $x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$

- **Basic transform pairs**

$$x[n] = \delta[n] \Leftrightarrow X(\omega) = 1$$

$$x[n] = 1 \Leftrightarrow X(\omega) = \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi k)$$

Discrete Fourier Transform (DFT)

- **Discrete Fourier transform (DFT) of a discrete-time signal $x[n]$ with finite extent $n \in [0, N\text{-}1]$**

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk} = X(\omega)\Big|_{\omega=\frac{2\pi}{N}k} \quad \text{for } k = 0, 1, \ldots, N\text{-}1$$

$X[k]$ periodic with period N due to exponential

Also assumes $x[n]$ periodic with period N

Two-Point DFT

$X[0] = x[0] + x[1]$

$X[1] = x[0] - x[1]$

- **Inverse discrete Fourier transform** $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}$

- **Twiddle factor** $W_N = e^{j\frac{2\pi}{N}} \Rightarrow x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{nk}$

Discrete Fourier Transform (con't)

- **Forward transform**

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{-nk}$$

for $k = 0, 1, \ldots, N$-1

Exponent of W_N has period N

- **Memory usage**

$x[n]$: N complex words of RAM

$X[k]$: N complex words of RAM

W_N: N complex words of ROM

- **Halve memory usage**

Allow output array $X[k]$ to write over input array $x[n]$

Exploit twiddle factors symmetry

- **Computation**

N^2 complex multiplications

$N(N-1)$ complex additions

N^2 integer multiplications

N^2 modulo indexes into lookup table of twiddle factors

- **Inverse transform**

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{nk}$$

for $n = 0, 1, \ldots, N$-1

Memory usage?

Computational complexity?

Fast Fourier Transform Algorithms

- **Communication system application: multicarrier modulation using harmonically related carriers**
  - Discrete multitone modulation in ADSL & VDSL modems
  - OFDM in IEEE 802.11a/g Wi-Fi and cellular LTE
- **Efficient divide-and-conquer algorithm**
  - Compute discrete Fourier transform of length $N = 2^{\nu}$
  - ½ $N \log_2 N$ complex multiplications and additions
  - How many real complex multiplications and additions?
- **Derivation: Assume N is even and power of two**

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} = \sum_{n=even}^{N-1} x[n] W_N^{nk} + \sum_{n=odd}^{N-1} x[n] W_N^{nk}$$

Fast Fourier Transform (cont'd)

- **Substitute $n = 2r$ for n even and $n = 2r+1$ for odd**

$$X[k] = \sum_{r=0}^{N/2-1} x[2r] W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k}$$
$$= \sum_{r=0}^{N/2-1} x[2r] \left(W_N^2\right)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] \left(W_N^2\right)^{rk}$$

- **Using the property** $W_N^{2l} = e^{-j2\pi \frac{2l}{N}} = e^{-j\frac{2\pi l}{N/2}} = W_{N/2}^{l}$

$$X[k] = \sum_{r=0}^{N/2-1} x[2r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] W_{N/2}^{rk} = G[k] + W_N^k\, H[k]$$

  - One FFT length N => two FFTs length $N/2$
  - Repeat process until two-point FFTs remain
  - Computational complexity of two-point FFT?

Two-Point FFT

$X[0] = x[0] + x[1]$

$X[1] = x[0] - x[1]$

Linear Convolution by FFT

- **Linear convolution**
  - $x[n]$ has length N_x and $h[n]$ has length N_h
  - $y[n]$ has length N_x+N_h-1

$y[n] = \sum_m h[m]\, x[n-m]$

- **Linear convolution requires $N_x N_h$ *real-valued* multiplications and $2N_x + 2N_h - 1$ words of memory**
- **Linear convolution by FFT of length $N = N_x+N_h - 1$**
  - Zero pad $x[n]$ and $h[n]$ to make each N samples long
  - Compute forward DFTs of length N to obtain $X[k]$ and $H[k]$
  - $Y[k] = H[k]\, X[k]$ for $k = 0\ldots N-1$: may overwrite $X[k]$ with $Y[k]$
  - Take inverse DFT of length N of $Y[k]$ to obtain $y[n]$
- **If $h[n]$ is fixed, then precompute and store $H[k]$**

Linear Convolution by FFT

- **Implementation complexity using N-length FFTs**
  - 3 $N \log_2 N$ complex multiplications and additions
  - 2 N complex words of memory if $Y[k]$ overwrites $X[k]$
- **FFT approach requires fewer computations if**
  - $12(N_x + N_h - 1)\log_2(N_x + N_h - 1) < N_x N_h$
- **Disadvantages of FFT approach**
  - Uses twice the memory: $2(N_x + N_h - 1)$ complex words vs. $2N_x + 2N_h - 1$ words
  - Often requires floating-point arithmetic
  - Adds delay of N_x samples to buffer $x[n]$ whereas linear convolution is computed sample-by-sample
  - Creates discontinuities at boundaries of blocks of input data: use overlapping blocks and windowing

FFT under fixed-point arithmetic?

THE UNIVERSITY OF TEXAS AT AUSTIN

UT ECE ESPL Wireless Networking and Communications Group

Review for Midterm #2

Prof. Brian L. Evans

EE 445S Real-Time Digital Signal Processing Laboratory

9 January 2014

Outline

2

- Introduction
- Signal processing building blocks
  - Filters
  - Data conversion
  - Rate changers
- Communication systems

Design tradeoffs in signal quality vs. implementation complexity

Introduction

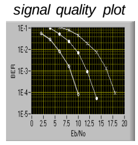
3

- Signal processing algorithms
  - Multirate processing: e.g. interpolation
  - Local feedback: e.g. IIR filters
  - Iteration: e.g. phase locked loops
- Signal representations
  - Bits, symbols
  - Real-valued symbol amplitudes
  - Complex-valued symbol amplitudes (I-Q)
  - Vectors/matrices of scalar data types
- Algorithm implementation
  - Dominated by multiplication/addition
  - High-throughput input/output

Often iterative

Do not need recursion

Communication signal quality plot

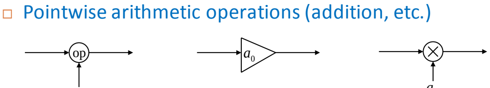


Bit error rate vs. Signal-to-noise ratio (Eb/No)

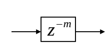
Finite Impulse Response Filters

4

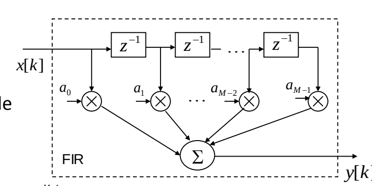
- Pointwise arithmetic operations (addition, etc.)



- Delay by m samples



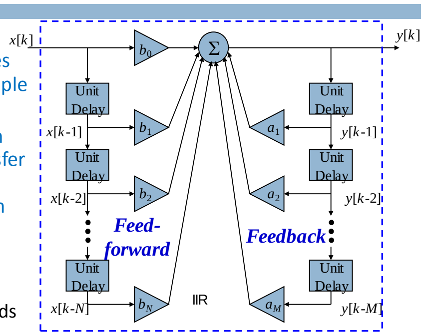
- Finite impulse response filter
  - Always stable
  - Each input sample produces one output sample
  - DSP processor architecture



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

Infinite Impulse Response Filters

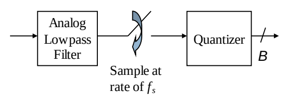
- Each input sample produces one output sample
- Pole locations perturbed when expanding transfer function into unfactored form
- 20+ filter structures
  - Direct form
  - Cascade biquads
  - Lattice



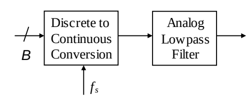
$$y[k] = \sum_{n=0}^{N} b_n\, x[k-n] + \sum_{m=1}^{M} a_m\, y[k-m]$$

Data Conversion

- Analog-to-Digital



- Digital-to-Analog



- Quantize to *B* bits
  - Quantization error = noise
  - $\text{SNR}_{\text{dB}} \approx C_0 + 6.02\, B$
  - Dynamic range $\approx$ SNR

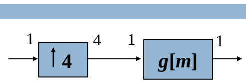
$$\text{SNR}_{dB} = 10 \log_{10} \frac{\text{Signal Power}}{\text{Noise Power}}$$

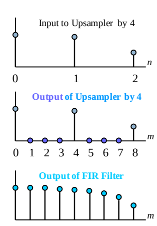
$$\text{SNR}_{dB} = 10 \log_{10} P^{signal} - 10 \log_{10} P^{noise}$$

$$\text{SNR}_{dB} = P^{signal}_{dB} - P^{noise}_{dB}$$

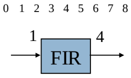
- A/D and D/A lowpass filter
  - $f_{\text{stop}} < ½\, f_s$ $\quad f_{\text{pass}} \le 0.9\, f_{\text{stop}}$
  - $A_{\text{stop}} = \text{SNR}_{\text{dB}}$ $\quad A_{\text{pass}} = \Delta_{\text{dB}}$
  - $\Delta_{\text{dB}} = 20 \log_{10} (2 m_{\text{max}} / (2^B - 1))$
  - Δ is quantization step size
  - m_{max} is max quantizer voltage

Increasing Sampling Rate

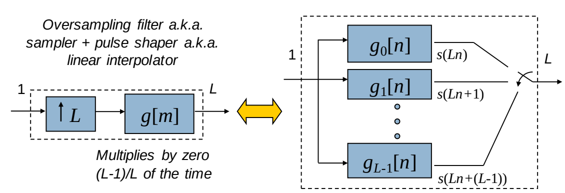




- Upsampling by *L* denoted as ↑*L*
  - Outputs input sample followed by *L*-1 zeros
  - Increases sampling rate by factor of *L*
- Finite impulse response (FIR) filter $g[m]$
  - Fills in zero values generated by upsampler
  - Multiplies by zero most of time (*L*-1 out of every *L* times)
- Sometimes combined into rate changing FIR block



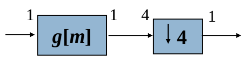
Polyphase Filter Bank Form

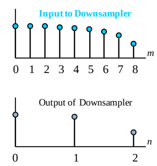


- Filter bank (right) avoids multiplication by zero
  - Split filter $g[m]$ into *L* shorter polyphase filters operating at the lower sampling rate (no loss in output precision)
  - Saves factor of *L* in multiplications and previous inputs stored **and** increases parallelism by factor of *L*

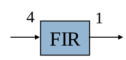
Decreasing Sampling Rate

9



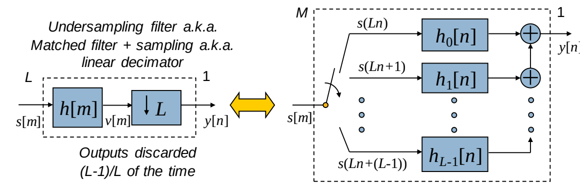


- Finite impulse response (FIR) filter $g[m]$
  - Typically a lowpass filter
  - Enforces sampling theorem
- Downsampling by L denoted as $\downarrow L$
  - Inputs L samples
  - Outputs first sample and discards L-1 samples
  - Decreases sampling rate by factor of L
- Sometimes combined into rate changing FIR block



Polyphase Filter Bank Form

10



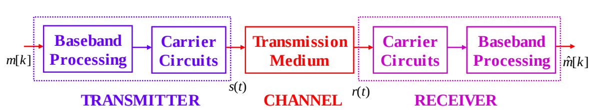
$y[1] = v[L] = h[0]\, s[L] + h[1]\, s[L\text{-}1] + \ldots + h[L\text{-}1]\, s[1] + h[L]\, s[0]$

- 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

Communication Systems

11

- Message signal $m[k]$ 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 carrier circuits include digital-to-analog converter, analog/RF upconverter, and transmit filter



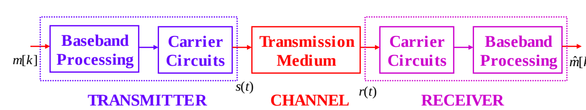
Communication Systems

12

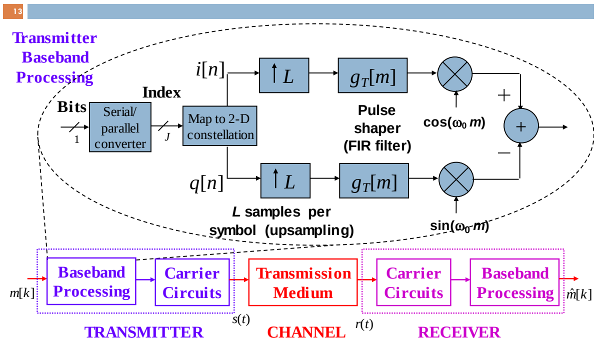
- Propagating signals experience attenuation & spreading w/ distance

Model the environment

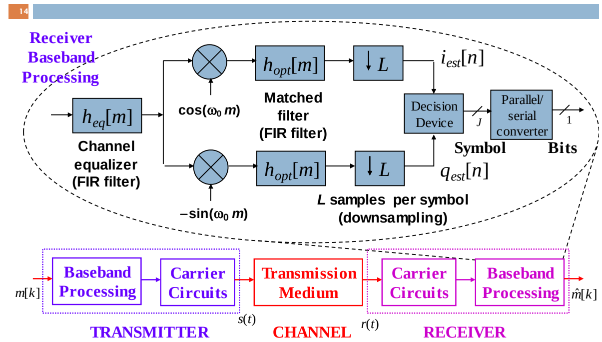
- Receiver carrier circuits include receive filter, carrier recovery, analog/RF downconverter, automatic gain control and analog-to-digital converter
- Receiver baseband processing extracts/enhances baseband signal



Quadrature Amplitude Modulation

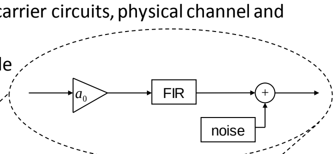


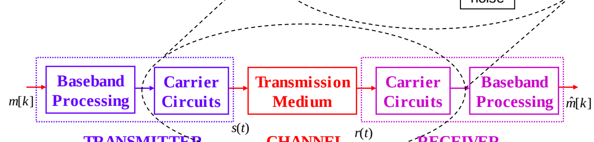
Quad. Amplitude Demodulation



Modeling of Points In-Between

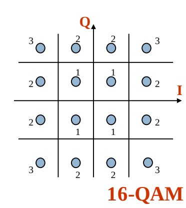
- Baseband discrete-time channel model
  - Combines transmitter carrier circuits, physical channel and receiver carrier circuits
  - One model uses cascade of gain, FIR filter, and additive noise





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 $\sqrt{\text{SNR}}$

EE445S Real-Time Digital Signal Processing Lab (Spring 2014)

Lecture:	MWF 11:00am–12:00pm in ETC 5.148
Instructor:	Prof. Brian L. Evans, ENS 433B, 512-232-1457, bevans@ece.utexas.edu
Office Hours:	MW 12:00–12:30pm and TH 12:30–2:30pm
Lab Sections:	M 6:30–9:30pm (Sinno), T 6:30–9:30pm (Sinno),
(ENS 252B)	W 6:30–9:30pm (Jia), F 1:00–4:00pm (Jia)
TA Office Hours:	Ms. Zeina Sinno, W 3:00–4:30pm and TH 5:30–7:00pm, zeina@utexas.edu
(ENS 137)	Mr. Chao Jia, TH 3:30–5:30pm and F 9:30–10:30am, kurtjc@gmail.com
Course Web Page:	http://users.ece.utexas.edu/~bevans/courses/rtdsp

This course covers basic discrete-time signal processing concepts and gives hands-on experience in translating these concepts into real-time digital communications software. The goal is to understand design tradeoffs in signal quality vs. implementation complexity.

Prerequisites

EE 312 and 319K with a grade of at least C- in each; BME 343 or EE 313 with a grade of at least C-; credit with a grade of at least C- or registration for BME 333T or EE 333T; and credit with a grade of at least C- or registration for BME 335 or EE 351K.

Topical Outline

System-level design tradeoffs in signal quality vs. implementation complexity; prototyping of baseband transceivers in real-time embedded software; addressing nodes, parallel instructions, pipelining, and interfacing in digital signal processors; sampling, filtering, quantization, and data conversion; modulation, pulse shaping, pseudo-noise sequences, carrier recovery, and equalization; and desktop simulation of digital communication systems.

Required Texts

1. C. R. Johnson Jr., W. A. Sethares and A. G. Klein, *Software Receiver Design*, Cambridge University Press, Oct. 2011, ISBN 978-0521189446. Paperback. Matlab code.
2. T. B. Welch, C. H. G. Wright and M. G. Morrow, *Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs*, CRC Press, 2nd ed., Dec. 2011, ISBN 978-1439883037.
3. B. L. Evans, *EE 445S Real-Time DSP Lab Course Reader*. Available on course Web page and on-demand from the HKN Office (ENS 129).

Supplemental Texts

4. B. P. Lathi, *Linear Systems and Signals*, 2nd ed., Oxford, ISBN 0-19-515833-4, 2005.
5. M. J. Roberts, *Signals and Systems*, McGraw-Hill, ISBN 978-0072930443, June 2003.
6. A. O. Oppenhiem and R. W. Schafer, *Signals and Systems*, 2nd ed., Prentice Hall, 1999.
7. J. H. McClellan, R. W. Schafer, and M. A. Yoder, *DSP First: A Multimedia Approach*, Prentice-Hall, ISBN 0-13-243171-8, 1998. On-line Multimedia CD ROM.

Grading

14% Homework, 21% Midterm #1, 21% Midterm #2, 5% Pre-lab quizzes, 39% Laboratory. Midterms will be held during lecture, with midterm #1 on Friday, Mar. 7th, and midterm #2 on Friday, May 2nd. Attendance/participation in laboratory is mandatory and graded. Lecture helps connect together all of the pieces of the class— laboratory, reading, and homework assignments. Lecture attendance is helpful in landing internships and permanent positions, and allows you to get the most for your tuition dollar. Plus and minus grades will be assigned for the final letter grades. There is no final exam. 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. *Discussion of homework questions is encouraged. Please submit your own independent homework solutions. Late assignments will not be accepted.*

University Honor Code
"The core values of The University of Texas at Austin are learning, discovery, freedom, leadership, individual opportunity, and responsibility. Each member of the University is expected to uphold these values through integrity, honesty, fairness, and respect toward peers and community." http://www.utexas.edu/about-ut/mission-core-purpose-honor-code

Religious Holidays
By UT Austin policy, you must notify the instructor of any pending absence at least fourteen (14) days prior to the date of observance of a religious holy day, or on the first class day if the observance takes place during the first fourteen days of the semester. If you must miss class, lab section, exam, or assignment to observe a religious holiday, you will have an opportunity to complete the missed work within a reasonable amount of time after the absence.

College of Engineering Drop/Add Policy
The Dean must approve adding or dropping courses after the fourth class day of the semester.

Students with Disabilities
UT provides upon request appropriate academic accommodations for qualified students with disabilities. Please contact Office of Dean of Students at 512-471-6259 or ssd@uts.cc.utexas.edu.

Lecture Topics
Introduction
Sinusoidal Generation
Introduction to Digital Signal Processors
Signals and Systems
Sampling and Aliasing
Finite Impulse Response Filters
Infinite Impulse Response Filters
Interpolation and Pulse Shaping
Quantization
Data Conversion
Channel Impairments
Digital PAM
Matched Filtering
Quadrature Amplitude Modulation (QAM) Transmitter
QAM Receiver

EE445S 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. The TAs are Mr. Chao Jia and Ms. Zeina Sinno. Both conduct research in reducing rolling shutter artifact in smart phone cameras. Both have been TAs for this course before. A grader will grade homework assignments for the lecture component of the class.

2 Supplemental Information

Wireless Networking & Communications Seminars generally meet Fridays in ENS 637.

You can search for a topic in Google scholar to find papers and patent applications on the topic. Web address is http://scholar.google.com.

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 and 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 f̃irst 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
Ŝignal Processing Tutorial: Nyquist Sampling Theorem and Anti-Aliasing (Part 1)
*Advantages: visuals
*Disadvantages: ... a bit slow

http://www.youtube.com/watch?v=Fy9dJgGCWZI
Ŝampling 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
Ŝimple 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
F̂IR 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
D̂igital Filter Design
*Advantages: quite Advantages information - especially on design TRADEOFFs
*Disadvantages: sound quality, better off just reading slides while he lectures

The Learning Resource Center

The ECE Learning Resource Center (LRC) for instructional computing is located in the second-floor ENS lab rooms as well as ENS 507. The ECE LRC rooms are open Mondays–Fridays from 8:00 AM to 10:00 PM, and on Saturday and Sunday from 11:00 AM to 10:00 PM. 24-hour access is available Mondays–Thursdays in ENS 507 with a valid UT Austin ID card. To activate your ECE LRC accounts, present your UT identification card to an ECE LRC proctor. The LRCs are described at http://www.ece.utexas.edu/it/labs.

1 Available Hardware

The ECE LRC has about 200 workstations, including Unix workstations and Windows machines. Several Linux workstations are available for remote connection: browser, daisy, luigi, mario, peach, thwomp and yoshi. The following Sun Unix workstation is available for remote connection: sunfire1. All are in the domain ece.utexas.edu. 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 LRC machines unless otherwise noted. On the Unix machines, they are installed in the /usr/local/bin directory.

• **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 ω). 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 discontinous 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**:

```
A = [1 2 3; 4 5 6];
B = [7 8; 9 10; 11 12];
C = inv((A*B)');
```

Matlab Tutorials and Availability

Here are excellent Matlab tutorials:

1. UT Austin: http://ssc.utexas.edu/training/software-tutorials#matlab
2. Mathworks: http://www.mathworks.com/academia/student_center/tutorials/

The following Matlab tutorial book is a useful reference:

Duane C. Hanselman and Bruce Littlefield, *Mastering MATLAB*, ISBN 9780136013303, Prentice Hall, 2011.

Matlab is available in the ECE Learning Resource Centers and through remote login. A student version of Matlab may be purchased at the bookstore for roughly $100.

Although the first few computer homeworks will help step you through Matlab, it is strongly suggested that you take the short courses that the Division of Statistics and Scientific Computing will be offering. The schedule of those courses is available online at

http://ssc.utexas.edu/training/software-short-courses

Technical support is provided through free consulting services from the Division of Statistics and Scientific Computation. Simple queries can be e-mailed to stats@ssc.utexas.edu. For more complicated inquiries, please go in person to their offices located in GDC 7.504. 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

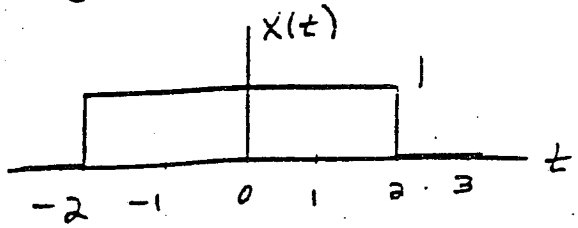
```
module load matlab
matlab
```

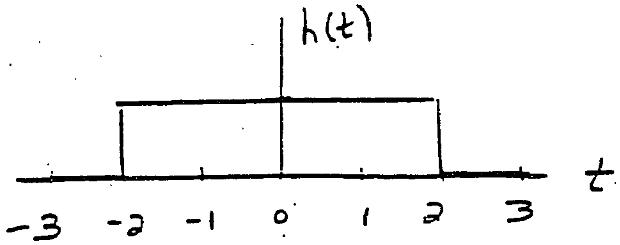
When Matlab begins running, it will automatically execute the commands in your Matlab initialization file, if you have one. On Unix systems, the initialization file must be ~/matlab/startup.m where ~ means your home directory.

Prof. Evans

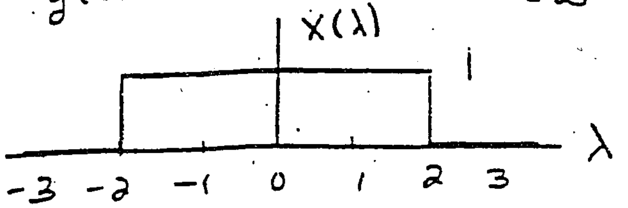
Convolution of Two Rectangular Pulses

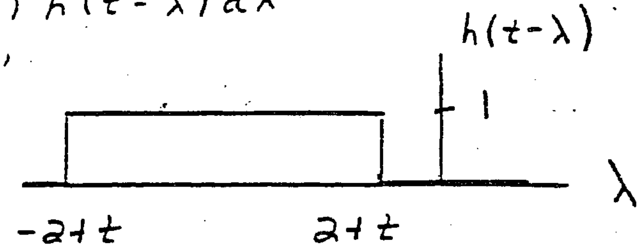
1. Continuous-Time Convolution

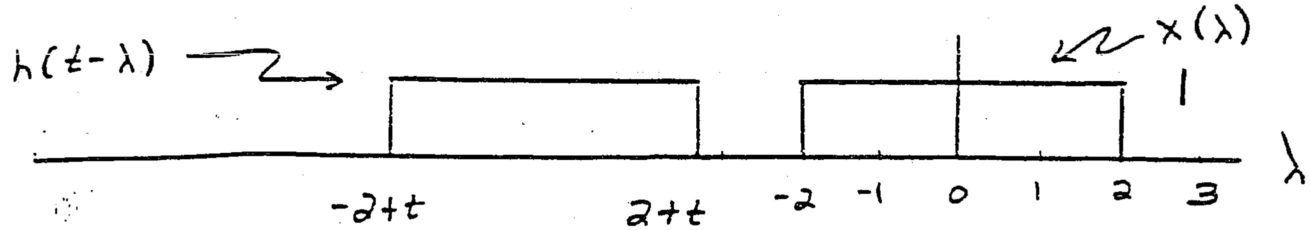




$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda)\, h(t-\lambda)\, d\lambda$$



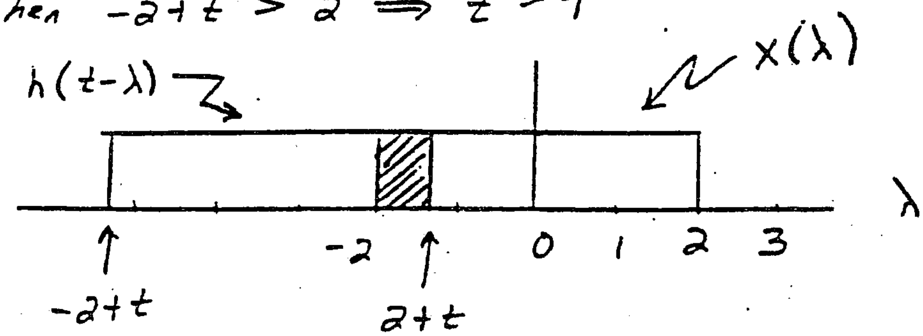




There is no overlap when $2+t < -2 \Rightarrow t < -4$

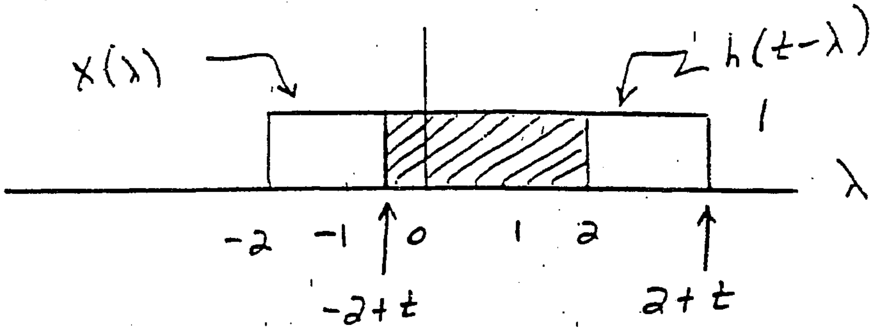
There is no overlap when $-2+t > 2 \Rightarrow t > 4$

For $-4 \le t \le 0$, there is overlap from $\lambda = -2$ to $\lambda = 2+t$



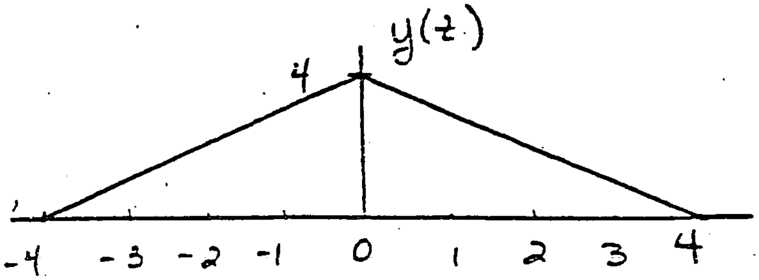
$$y(t) = \int_{-\infty}^{\infty} x(\lambda)\, h(t-\lambda)\, d\lambda = \int_{-2}^{2+t} 1 \cdot d\lambda = \Big[\lambda\Big]_{-2}^{2+t} = t+4$$

For $0 < t \leq 4$,
there is overlap
from $\lambda = -2 + t$
to $\lambda = 2$



$$y(t) = \int_{-\infty}^{\infty} x(\lambda) h(t-\lambda) d\lambda = \int_{-2+t}^{2} 1 \cdot d\lambda = [\lambda]_{-2+t}^{2} = 4 - t$$

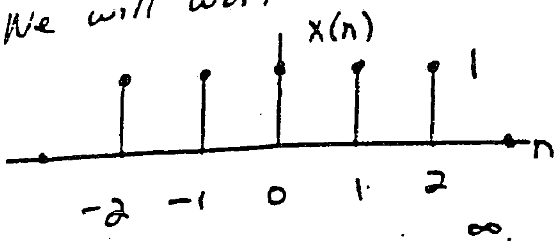
$$y(t) = \begin{cases} 0 & \text{for } t < -4 \\ 4+t & \text{for } -4 \leq t \leq 0 \\ 4-t & \text{for } 0 < t \leq 4 \\ 0 & \text{for } t > 4 \end{cases}$$

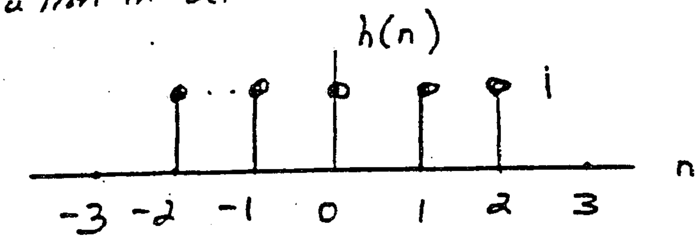


To check a convolution result, check the value of $y(t)$ at the endpoints of each interval — they should agree.

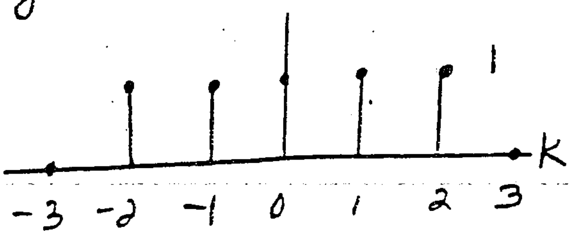
2. Discrete-Time Convolution

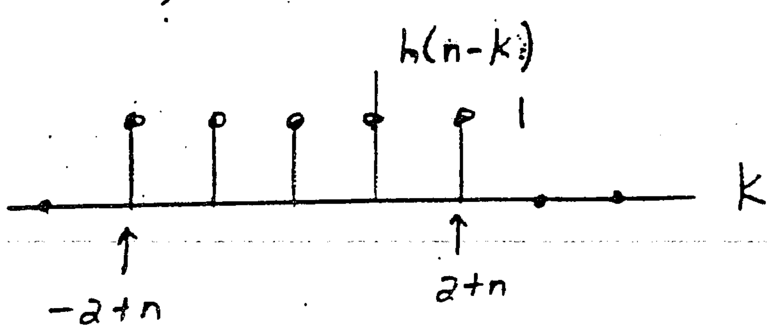
We will work a similar convolution in discrete time.

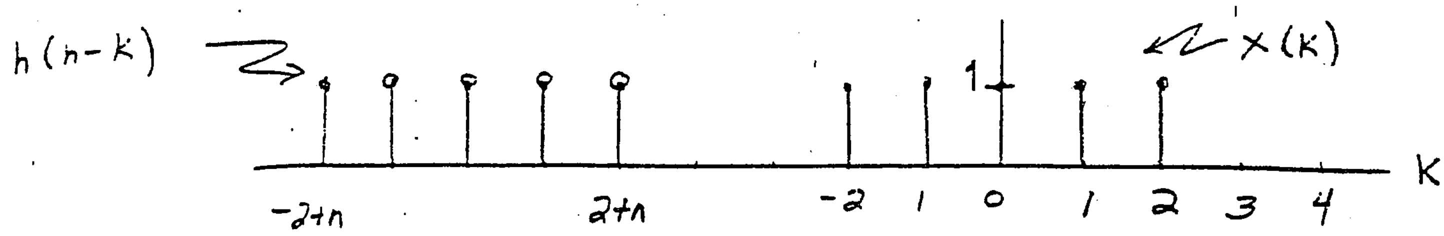




$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k),$$ so k has the same role as λ



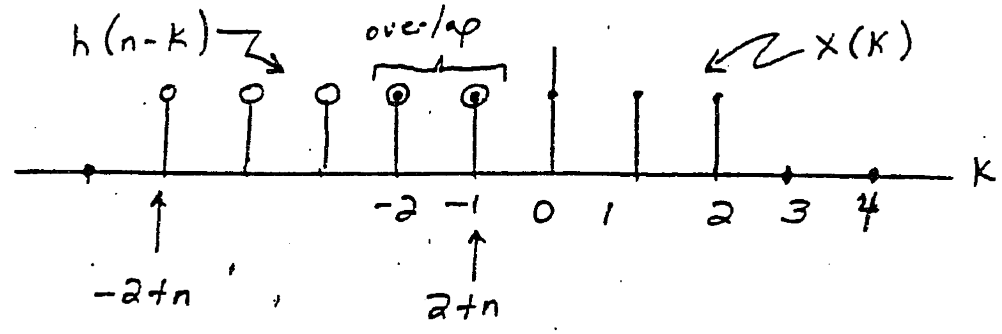




There is no overlap when $2+n < -2 \Rightarrow n < -4$

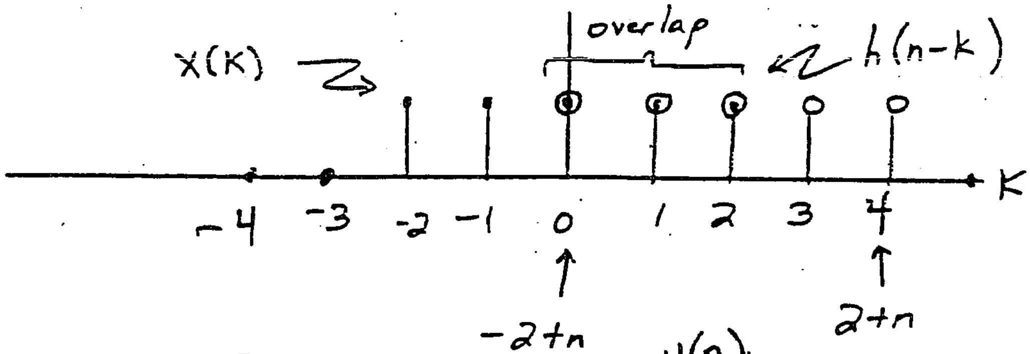
There is no overlap when $-2+n > 2 \Rightarrow n > 4$

For $-4 \le n \le 0$, there is overlap from $k=-2$ to $k = 2+n$



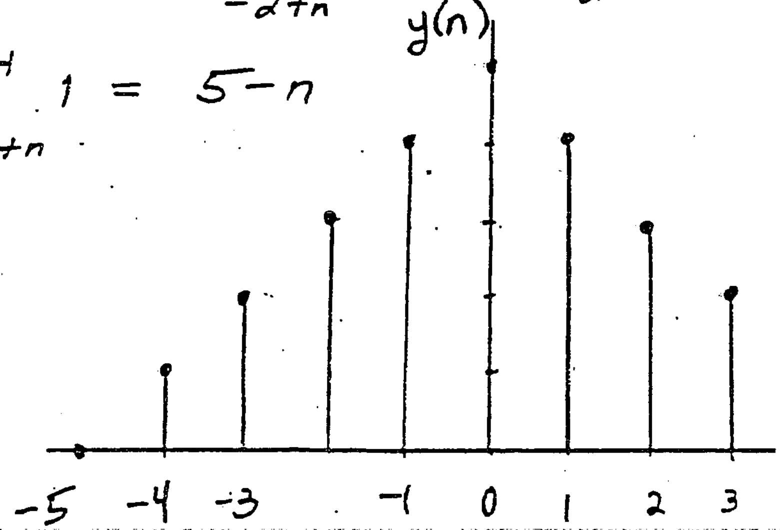
$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k) = \sum_{k=-2}^{2+n} 1 = n+5$$

For $1 \le n \le 4$, there is overlap from $k=-2+n$ to $k=2$



$$y(n) = \sum_{k=}^{\infty} x(k)\,h(n-k) = \sum_{k=-2+n}^{2} 1 = 5-n$$

$$y(n) = \begin{cases} 0 & \text{for } n < -4 \\ 5+n & \text{for } -4 \le n \le 0 \\ 5-n & \text{for } 1 \le n \le 4 \\ 0 & \text{for } n > 4 \end{cases}$$



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, $g(e_f(.), t) = g(e_f(.), 0)e_f(t) = c\, e_f(t)$.

Example: Analog RC Lowpass Filter

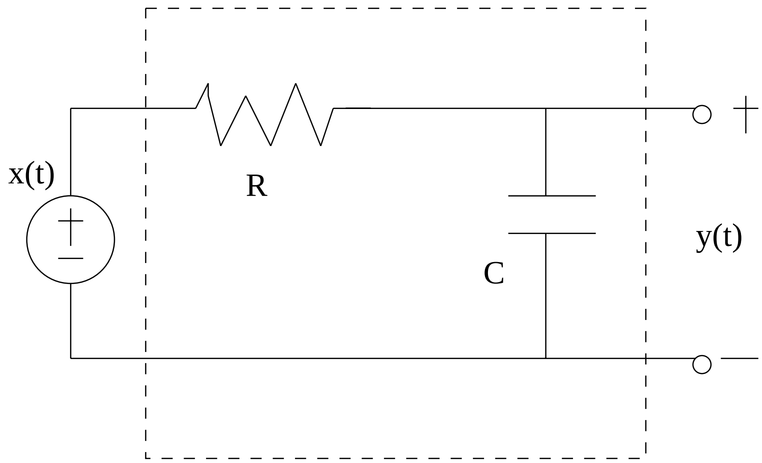


Figure 1: A First-Order Analog Lowpass Filter

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, $x(t) = e_f(t) = e^{j2\pi ft}$,

$$\begin{aligned}
y(t) &= \int_{-\infty}^{\infty} x(t-\lambda)h(\lambda)\, d\lambda \\
&= \int_{-\infty}^{\infty} e^{j2\pi f(t-\lambda)} \frac{1}{RC} e^{-\frac{1}{RC}\lambda} u(\lambda)\, d\lambda \\
&= e^{j2\pi ft} \left[\frac{1}{RC} \int_{-\infty}^{\infty} e^{-j2\pi f\lambda} e^{-\frac{1}{RC}\lambda}\, d\lambda \right] \\
&= \left[\frac{\frac{1}{RC}}{j2\pi f + \frac{1}{RC}} \right] e^{j2\pi ft} \\
&= g(e_f(.), 0)\, e_f(t)
\end{aligned}$$

So, $g(e_f(.), 0) = H(f)$, which is the transfer function of the system.

Placeholder – please ignore.

EE445S Real-Time Digital Signal Processing Laboratory

Raised Cosine Spectrum

Section 7.5, pp. 431–434, Simon Haykin, *Communication Systems*, 4th ed.

We may overcome the practical difficulties encounted 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

$$P(f) + P(f - 2W) + P(f + 2W) = \frac{1}{W}, -W \leq f \leq W \tag{1}$$

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:

$$P(f) = \begin{cases} \dfrac{1}{2W} & \text{for } 0 \leq |f| < f_1 \\ \dfrac{1}{4W}\left(1 - \sin\dfrac{\pi(|f| - W)}{2W - 2f_1}\right) & \text{for } f_1 \leq |f| < 2W - f_1 \\ 0 & \text{for } |f| \geq 2W - f_1 \end{cases} \tag{2}$$

The frequency parameter f_1 and bandwidth W are related by

$$\alpha = 1 - \frac{f_1}{W} \tag{3}$$

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)

$$p(t) = \text{sinc}(2Wt)\left(\frac{\cos 2\pi\alpha Wt}{1 - 16\alpha^2W^2t^2}\right) \tag{4}$$

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.

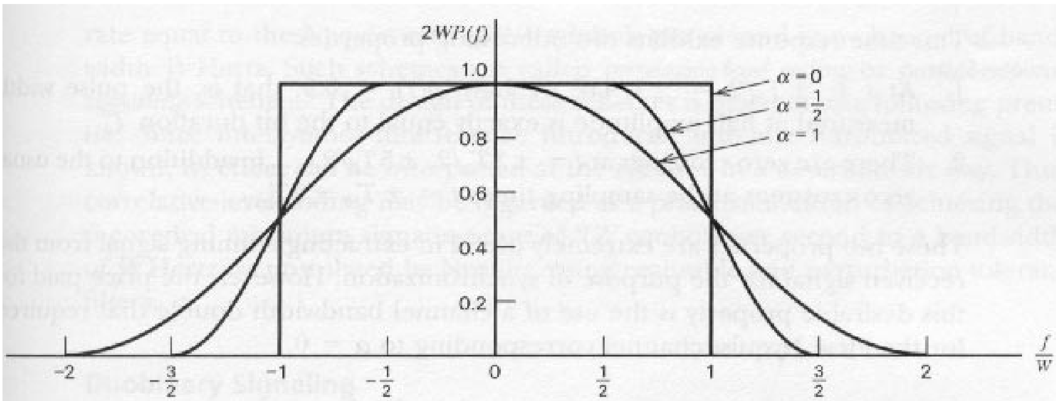


Figure 1: Frequency response for the raised cosine function.

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

$$P(f) = \begin{cases} \dfrac{1}{4W}\left(1 + \cos\dfrac{\pi f}{2W}\right) & \text{for } 0 < |f| < 2W \\ 0 & \text{if } |f| \geq 2W \end{cases}$$

Correspondingly, the time response $p(t)$ simplifies to

$$p(t) = \frac{\text{sinc}(4Wt)}{1 - 16W^2t^2} \tag{5}$$

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 $t = \pm 3T_b/2$, $\pm 5T_b/2$,... in addition to the usual zero crossings at the sampling times $t = \pm T_b, \pm 2T_b, \ldots$

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 $\alpha = 0$.

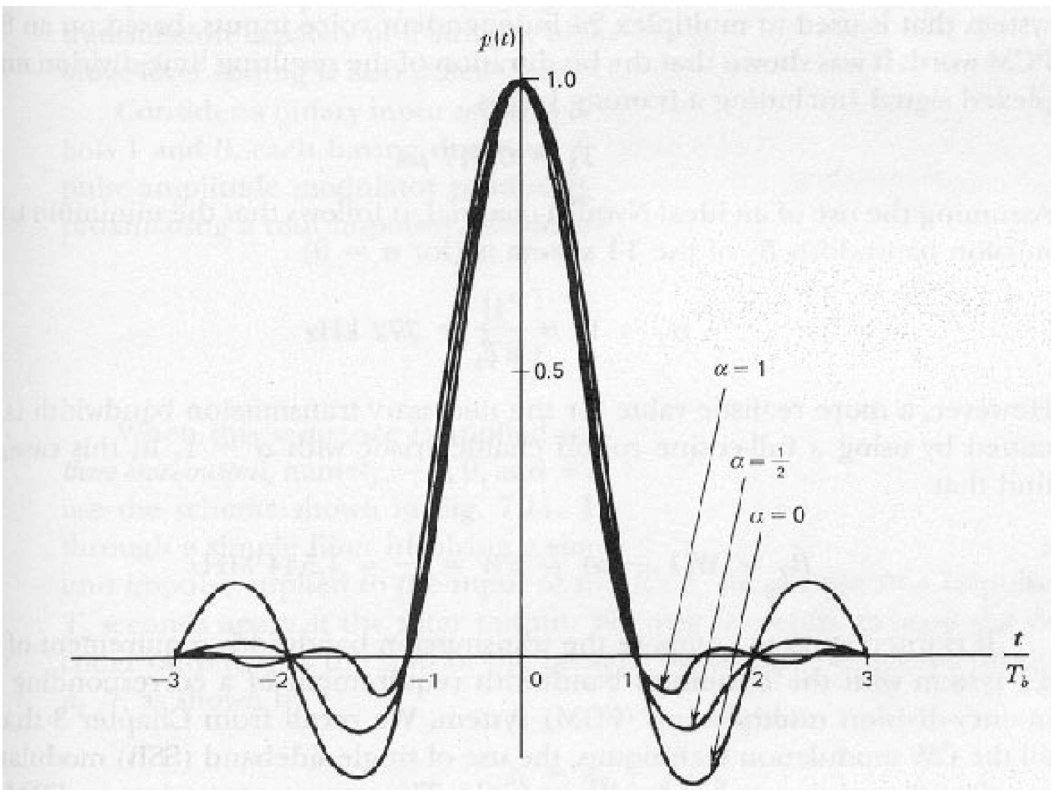
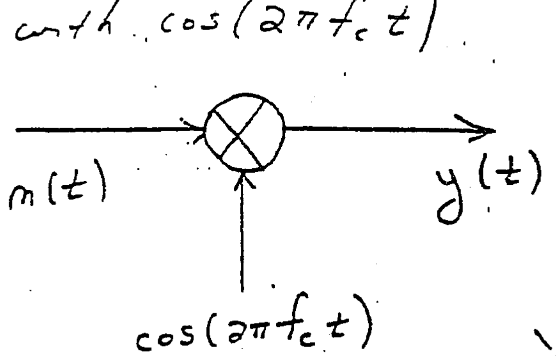


Figure 2: Time response for the raised cosine function.

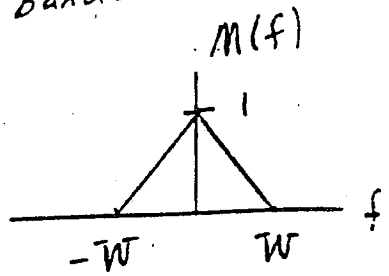
Placeholder – please ignore.

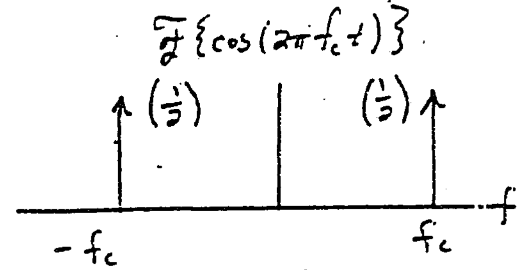
Amplitude Modulation with $\cos(2\pi f_c t)$

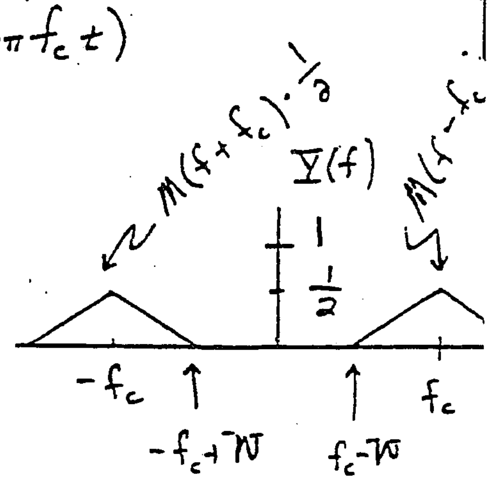
$y(t) = m(t)\cos(2\pi f_c t)$



$m(t)$ is lowpass with bandwidth W $(f_c \gg W)$







I choose an arbitrary spectrum for $M(f)$.

$$Y(f) = M(f) * \mathcal{F}\{\cos(2\pi f_c t)\}$$

$$Y(f) = M(f) * \tfrac{1}{2}\left(\delta(f+f_c) + \delta(f-f_c)\right)$$

Example uses the normalized Fourier transform

$$Y(f) = \frac{1}{2}\int_{-\infty}^{\infty}\left(\underbrace{\delta(\lambda+f_c)} + \underbrace{\delta(\lambda-f_c)}\right) M(f-\lambda)\,d\lambda$$

Deltas are non-zero @ $\lambda = -f_c$ and $\lambda = f_c$

$$Y(f) = \frac{1}{2}\left(M(f+f_c) + M(f-f_c)\right)$$

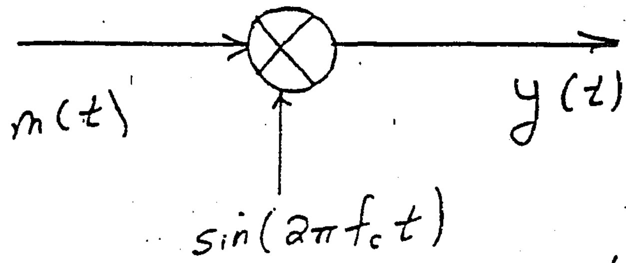
It is easier to work with amplitude modulation to draw pictures in the frequency domain when analyzing the resulting spectrum for amplitude modulation.

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 $f_c >> W$.

Example uses the normalized Fourier transform

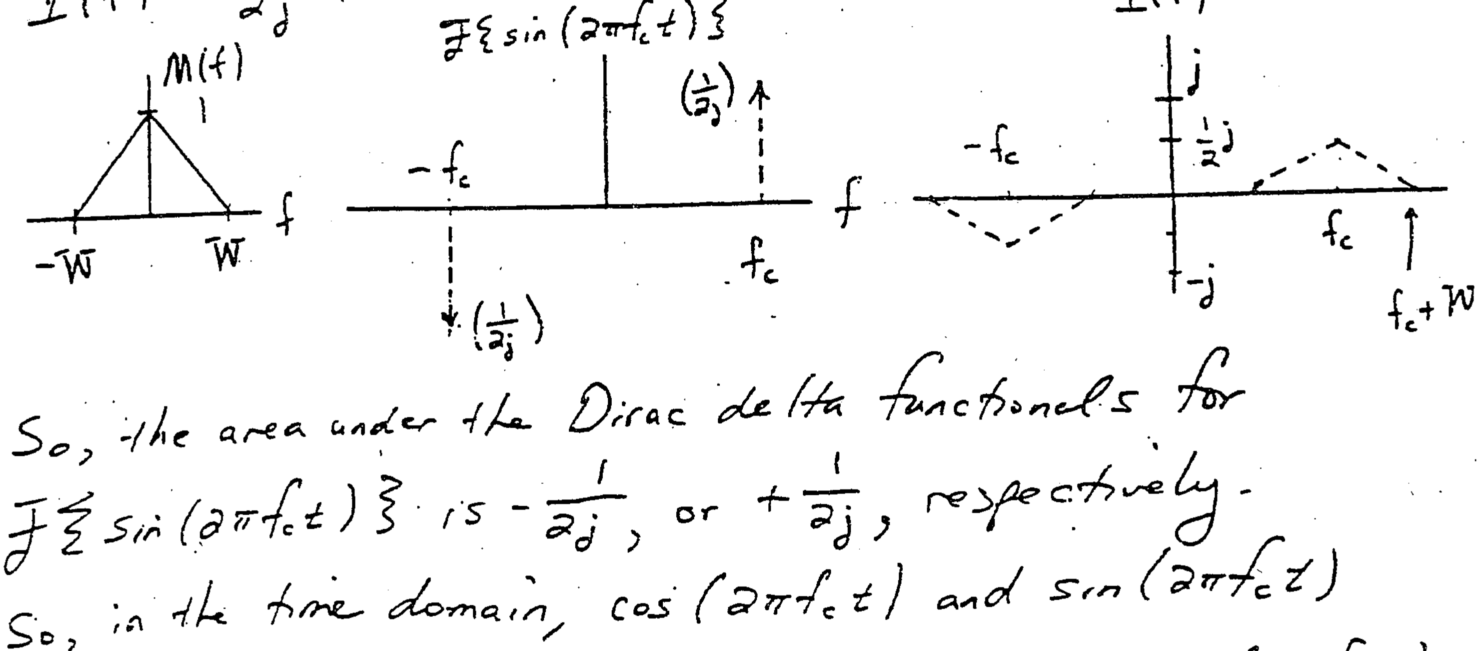
$$Y(f) = M(f) * \mathcal{F}\{\sin(2\pi f_c t)\}$$

$$Y(f) = M(f) * \frac{1}{2j}\left(-\delta(f+f_c) + \delta(f-f_c)\right)$$

$$Y(f) = \frac{1}{2j}\int_{-\infty}^{\infty}\left(-\delta(\underbrace{\lambda+f_c}) + \delta(\underbrace{\lambda-f_c})\right) M(f-\lambda)\, d\lambda$$

Deltas are non-zero at $\lambda = -f_c$ and $\lambda = f_c$

$$Y(f) = \frac{1}{2j}\left(-M(f+f_c) + M(f-f_c)\right)$$



So, the area under the Dirac delta functionals for $\mathcal{F}\{\sin(2\pi f_c t)\}$ 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 $2W$.

Analog Sinusoidal Modulation

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 vestigal sideband modulation. Vestigal 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 alterative representation for the modulated signal $x(t)$ is

$$x(t) = \Re e\{\tilde{x}(t)\, e^{j 2\pi f_c t}\}$$

Halftone Comparison

Simple Threshold

Clustered-dot dither

Error diffusion

Original

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 20, 2005

Course: EE 345S Evans

Name: Evans, Brian
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.**

Problem	*Point Value*	*Your score*	*Topic*
1	20		Digital Filter Analysis
2	20		Digital Filter Design
3	20		Sampling and Interpolation
4	20		Phase Response
5	20		Oscillator Implementation
Total	100		

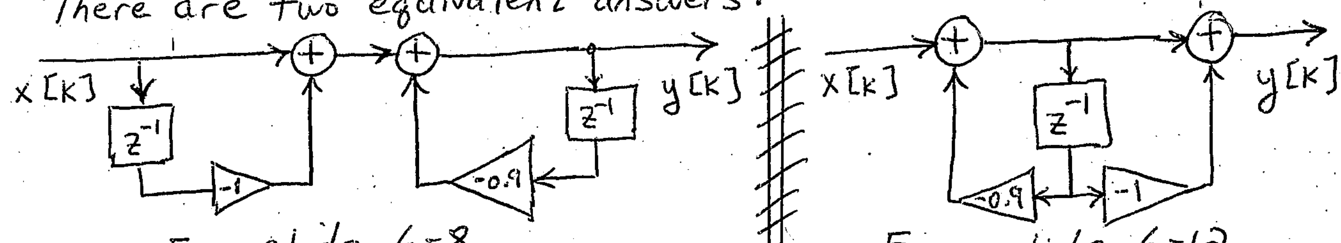
Problem 1.1 *Digital Filter Analysis*. 20 points. IIR

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.

$Y(z) = -0.9\, z^{-1} Y(z) + X(z) - z^{-1} X(z)$

$Y(z) + 0.9\, Y(z) = X(z) - z^{-1} X(z)$

$\dfrac{Y(z)}{X(z)} = \dfrac{1 - z^{-1}}{1 + 0.9\, z^{-1}} = H(z)$ for $|z| > 0.9$

(See slides 5-21 and 5-22 for the region of convergence.)

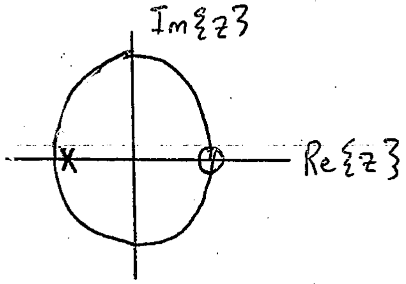
(d) Find the equation for the frequency response of the filter. 4 points.

Since the region of convergence includes the unit circle,

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{1 - e^{-j\omega}}{1 + 0.9\, e^{-j\omega}}$$

(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)

$\boxed{\text{HIGHPASS FILTER}}$

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.

$f_s = 2.2$ 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.

$f_c = 550$ 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.

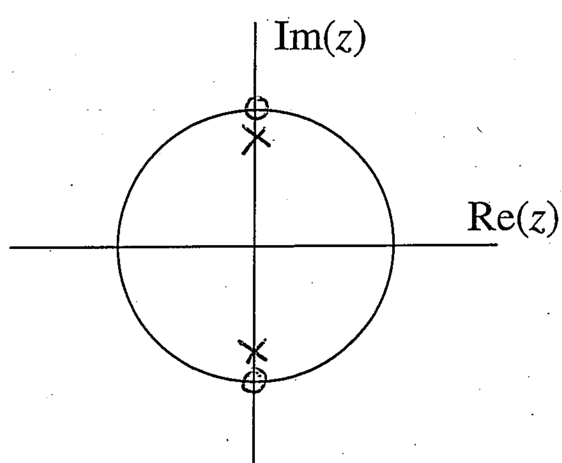
A biquad has 2 poles and 0-2 zeros (see slide 6-4).
Design a notch filter to notch out (reject) the AM radio station frequency. Pole-zero diagram shown on slide 6-6 (middle plot).

Zero locations: $z_0 = e^{+j\omega_c}$, $z_1 = e^{-j\omega_c}$

Pole locations: $p_0 = 0.9\, e^{+j\omega_c}$, $p_1 = 0.9\, e^{-j\omega_c}$

$$\omega_c = 2\pi \frac{f_c}{f_s} = \frac{\pi}{2}$$

(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 feedforward 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).

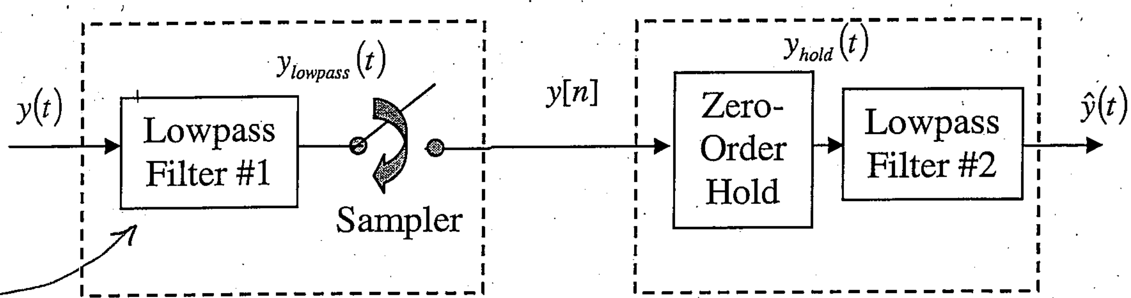
Representing a filter in terms of poles-zeros-gain (PZK) is common.

We can arbitrarily set the DC response to 1: $H(e^{j0}) = C \dfrac{(1 - z_0)(1 - z_1)}{(1 - p_0)(1 - p_1)} = 1$

This problem explores the many unrealistic assumptions in the Shanon sampling theorem (slides 4-6 and 4-7).

Problem 1.3. *Sampling and Interpolation.* 20 points.

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.



Magnitude specification: $A_{pass,1}$, $f_{pass,1}$, $A_{stop,1}$, $f_{stop,1}$

Continuous-Time to Discrete-Time Conversion

Discrete-Time to Continuous-Time Conversion

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 Shanon sampling theorem: $f_s > 2 f_{stop,1}$ (slide 4-6)
$f_{stop,1}$ 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.

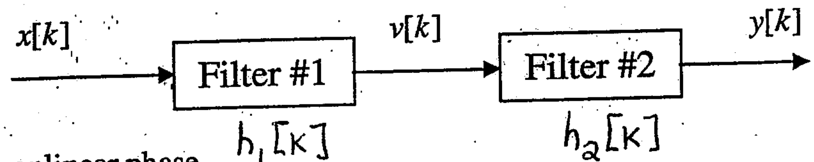
A timing error in the oscillator corresponds to a phase shift in frequency response. Since the phases of the signals in this application are not important, we can tolerate the usual 0.1% accurracy in oscillators.

(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.

<u>Counter-example #1 : FIR filters</u> (from a student solution)

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.

<u>Counter-example #2 : IIR - FIR cascade</u>

Let filter #1 be an IIR biquad (all-pole) and filter #2 be a three-tap FIR filter.

$$H_1(z) = C\,\frac{1}{(1-p_0 z^{-1})(1-p_1 z^{-1})}$$

Let $H_2(z) = (1-p_0 z^{-1})(1-p_1 z^{-1})$

$= 1-(p_0+p_1)z^{-1} + p_0 p_1 z^{-2}$

where $p_0 = re^{j\theta}$ and $p_1 = re^{-j\theta}$, $r \in (0,1)$. $= 1-(2\cos\theta)z^{-1} + r^2 z^{-2}$

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

$$y[k] = x[k] + y[k-1] - y[k-2] - \tfrac{1}{2}\, x[k-1]$$

with initial conditions $y[-1] = 0$, $y[-2] = 0$, and $x[-1] = 0$.

In IEEE floating point arithmetic, $\frac{1}{2}$ is exactly represented using one bit of mantissa.

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.

$x[k] = \delta[k] \Rightarrow x[0] = 1 \Rightarrow y[0] = x[0] + y[-1] - y[-2] + \frac{1}{2}x[-1] = x[0] = 1$

$x[1] = 0 \quad y[1] = x[1] + y[0] - y[-1] - \frac{1}{2}x[0] = y[0] - \frac{1}{2}x[0] = \frac{1}{2}$

$x[2] = 0 \quad y[2] = x[2] + y[1] - y[0] - \frac{1}{2}x[1] = y[1] - y[0] = -\frac{1}{2}$

$\boxed{\text{NO LOSS OF PRECISION}}$

(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 Lth 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 ½ 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 || notation (the || 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.**

Problem	*Point Value*	*Your score*	*Topic*
1	20		Digital Filter Analysis
2	20		IIR Filter
3	20		Sampling and Reconstruction
4	20		Linear Systems
5	20		Assembly Language
Total	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:

$$y[k] = -0.7\, y[k-1] + x[k] - x[k-1]$$

(a) Draw the block diagram for this filter. 4 points.

(b) What are the initial conditions and what values should they be assigned? 4 points.

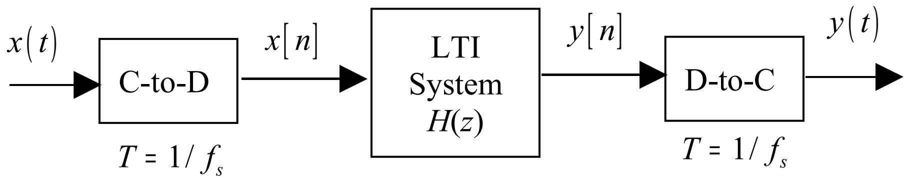
(c) Find the equation for the transfer function in the z-domain including the region of convergence. 4 points.

(d) Find the equation for the frequency response of the filter. 4 points.

(e) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.

Problem 1.2 *IIR filtering*. 20 points.

For the system shown below



The input signal $x(t)$ to the Continuous-to-Discrete converter is

$$x(t) = 4 + \cos(500\pi t) - 3\cos([(2000\pi/3)t]$$

The transfer function for the linear, time-invariant (LTI) system is $H(z)$

$$H(z) = \frac{\left(1 - z^{-1}\right)\left(1 - e^{j\pi/2}z^{-1}\right)\left(1 - e^{-j\pi/2}z^{-1}\right)}{\left(1 - 0.9e^{j2\pi/3}z^{-1}\right)\left(1 - 0.9e^{-j2\pi/3}z^{-1}\right)}$$

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.

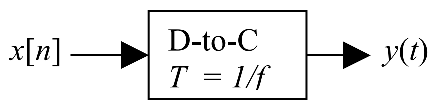
Suppose that a discrete-time signal x[n] is given by the formula

$$x[n] = 10\cos(0.2\pi n - \pi/7)$$

and that it was obtained by sampling a continuous-time signal at a sampling rate of $f_s = 2000$ samples/sec.

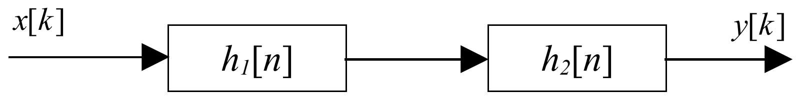
a) Determine two different continuous-time signals $x_1(t)$ and $x_2(t)$ whose samples are equal to $x[n]$; i.e. find $x_1(t)$ and $x_2(t)$ such that $x[n] = x_1(nT_s) = x_2(nT_s)$

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?

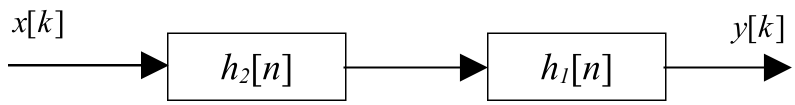


Problem 1.4. *Linear Systems*. 20 points.

Two stable discrete-time linear time-invariant (LTI) filters are in cascade as shown below.



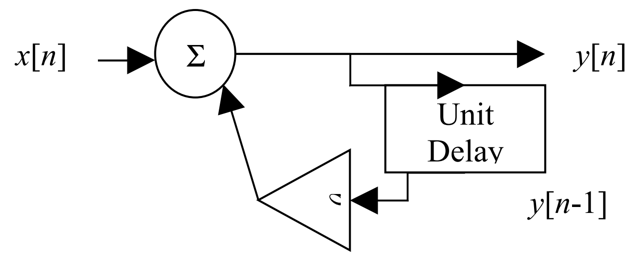
a) Show that the end-to-end system from x[k] to y[k] is equivalent to the following system where the order of the systems have been replaced, or give a counter-example. 10 points



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.**

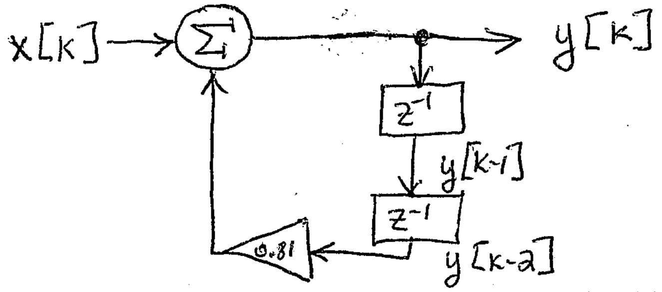
Problem	*Point Value*	*Your score*	*Topic*
1	20		Digital Filter Analysis
2	20		Digital Filter Design
3	20		Sampling
4	24		Time-Domain Response
5	16		Potpourri
Total	100		

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:

$$y[k] = 0.81\, y[k\text{-}2] + x[k]$$

(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.

$$y[0] = 0.81\, y[-2] + x[0]$$
$$y[1] = 0.81\, y[-1] + x[1]$$

$\Rightarrow$ Initial conditions are $y[-1]$ and $y[-2]$. Their values should be zero to guarantee LTI properties of the filter.

(c) Find the equation for the transfer function in the z-domain including the region of convergence. 4 points.

Take the z-transform of the difference equation with $y[-1]=0$ and $y[-2]=0$ and isolate the ratio $Y(z)/X(z)$:

$$Y(z) = 0.81\, z^{-2}\, Y(z) + X(z)$$
$$Y(z)\left(1 - 0.81 z^{-2}\right) = X(z) \Rightarrow \frac{Y(z)}{X(z)} = \frac{1}{1 - 0.81 z^{-2}} = H(z)$$

Poles are located at
$$1 - 0.81 z^{-2} = 0$$
$$(1 + 0.9 z^{-1})(1 - 0.9 z^{-1}) = 0$$
$$z = -0.9 \text{ and } z = 0.9$$

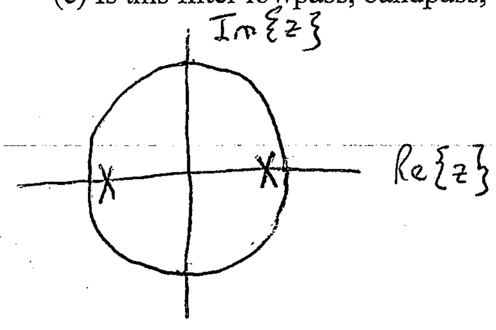
ROC is $|z| > 0.9$

(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:

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{1}{1 - 0.81 e^{-j2\omega}}$$

(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 passbands of the filter.
Passband at $\omega = 0$.
Another passband at $\omega = \pm\pi$.
Bandstop filter.

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.

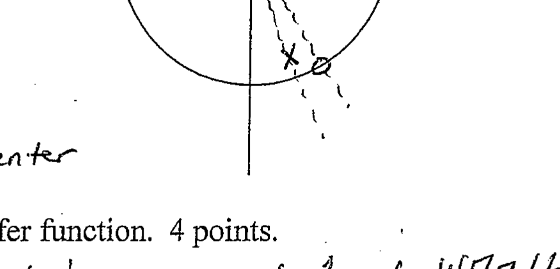
A biquad has 2 poles and 0–2 zeros.
We want to pass 1477 Hz and reject 1336 Hz.
This is a bandpass filter with a center frequency of 1477 Hz.

(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 $p_0 = 0.95 e^{j\omega_{1477}}$; $p_1 = 0.95 e^{-j\omega_{1477}}$

$\omega_{1336} = 2\pi \frac{1336\text{ Hz}}{8000\text{ Hz}}$; 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

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

$\omega_{center} = 1477\text{ Hz} \Rightarrow z_{center} = e^{j\omega_{center}}$

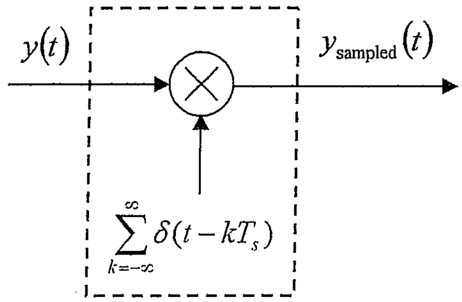
(d) Compute the scaling constant (gain) for the filter's transfer function. 4 points.

Let $H(z_{center}) = 1$, i.e. set the filter's response to 1 at 1477 Hz,

$$H(z_{center}) = C \frac{(1 - z_0 z_{center}^{-1})(1 - z_1 z_{center}^{-1})}{(1 - p_0 z_{center}^{-1})(1 - p_1 z_{center}^{-1})} = 1$$ 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:



Futher, we know that the Fourier transform of the impulse train is another impulse train:

$$\sum_{k=-\infty}^{\infty} \delta(t-kT_s) \Leftrightarrow f_s \sum_{n=-\infty}^{\infty} \delta(f-nf_s)$$

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: $u(t) \sum_{k=-\infty}^{\infty} \delta(t-kT_s) = \sum_{k=0}^{\infty} \delta(t-kT_s)$

$$\mathcal{F}\left\{\sum_{k=0}^{\infty} \delta(t-kT_s)\right\} = \sum_{k=0}^{\infty} \mathcal{F}\{\delta(t-kT_s)\} = \sum_{k=0}^{\infty} e^{-j2\pi k T_s f}$$

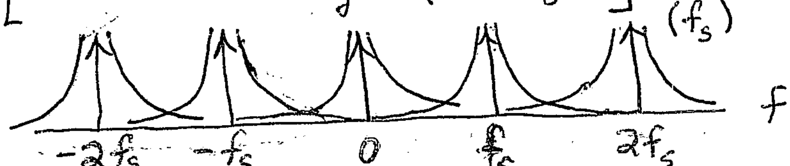
Solution #2: Multiplication in the time domain means convolution in the frequency domain:

$$\mathcal{F}\left\{u(t) \sum_{k=-\infty}^{\infty} \delta(t-kT_s)\right\} = \mathcal{F}\{u(t)\} * \mathcal{F}\left\{\sum_{k=-\infty}^{\infty} \delta(t-kT_s)\right\}$$

$$= \left[\frac{1}{2}\delta(f) + \frac{1}{j2\pi f}\right] * \left[f_s \sum_{m=-\infty}^{\infty} \delta(f-mf_s)\right]$$

(b) Sketch the magnitude of the Fourier transform of a causal impulse train. 5 points.

$$= f_s \sum_{m=-\infty}^{\infty} \left[\delta(f-mf_s) + \frac{1}{j2\pi(f-mf_s)}\right]$$

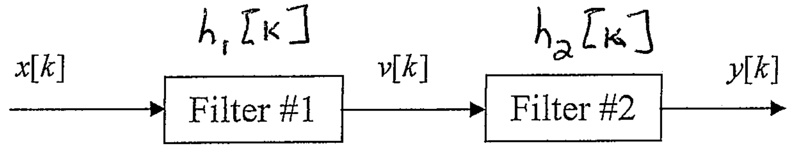


(c) Describe the difference between the Fourier transforms of the (two-sided) impulse train and the causal (one-sided) impulse train. 5 points.

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

Let $h_1[k] = a\, h_1[k-1] + x[k] \Rightarrow H_1(z) = \dfrac{1}{1 - a z^{-1}}$

and $h_2[k] = v[k] - a\, v[k-1] \Rightarrow H_2(z) = 1 - a z^{-1}$

$$H(z) = H_1(z)\, H_2(z) = \left[\frac{1}{1 - a z^{-1}}\right]\left[1 - a z^{-1}\right] = 1$$

$h[k] = \delta[k]$ 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 matters, and why the IIR filter structure matters.

(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.

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.**

Problem	*Point Value*	*Your score*	*Topic*
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
Total	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

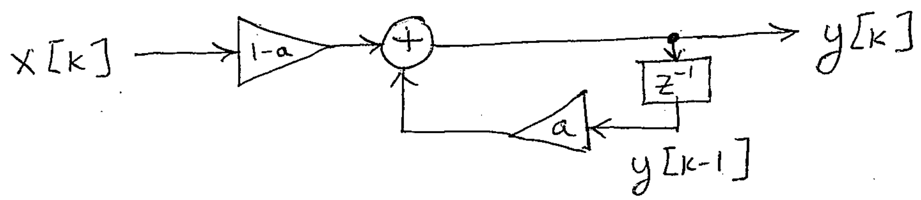
$$y[k] = a\, y[k-1] + (1-a)\, x[k]$$

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.

$$Y(z) = a z^{-1} Y(z) + (1-a) X(z)$$

$$(1-az^{-1}) Y(z) = (1-a) X(z)$$

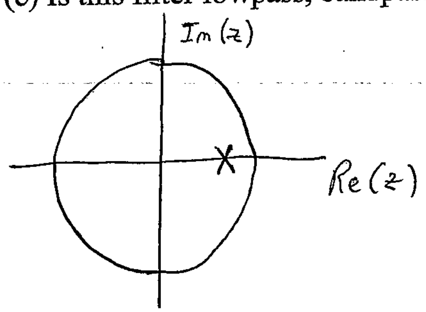
$$\frac{Y(z)}{X(z)} = \frac{1-a}{1-az^{-1}} \quad \text{for } |z| > a \text{ due to causality.}$$

(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,

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{1-a}{1-ae^{-j\omega}}$$

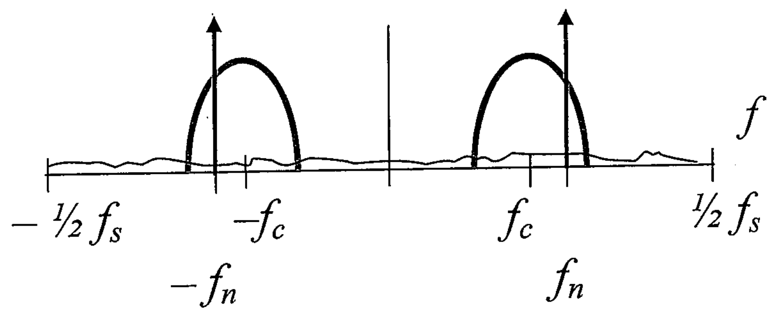
(e) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? 4 points.



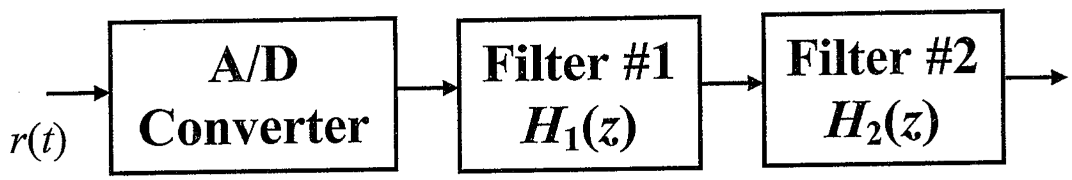
Filter has one pole at $z=a$, $0<a<1$. $\omega_{pole}=0$.
Filter has no zeros. Pole determines passband.
Filter is lowpass when $a \approx 1$ and essentially allpass when $a \approx 0$.

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 $f_s >> f_c$, 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.

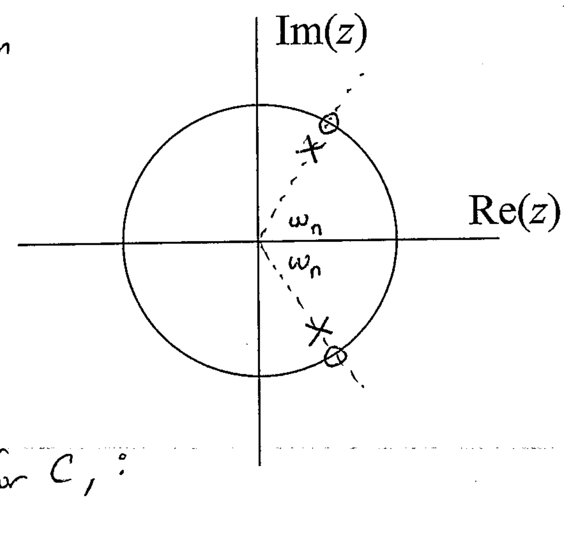
Notch. We seek to eliminate only one frequency (f_n) and pass all other frequencies as much as possible.

(b) Give formulas for the pole-zero locations of the biquad. 8 points.

$$\omega_n = 2\pi \frac{f_n}{f_s} \; ; \; p_0 = 0.9\, e^{-j\omega_n} \; ; \; p_1 = 0.9\, e^{+j\omega_n}$$

$$z_0 = e^{-j\omega_n} \; ; \; z_1 = e^{+j\omega_n}$$

(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.

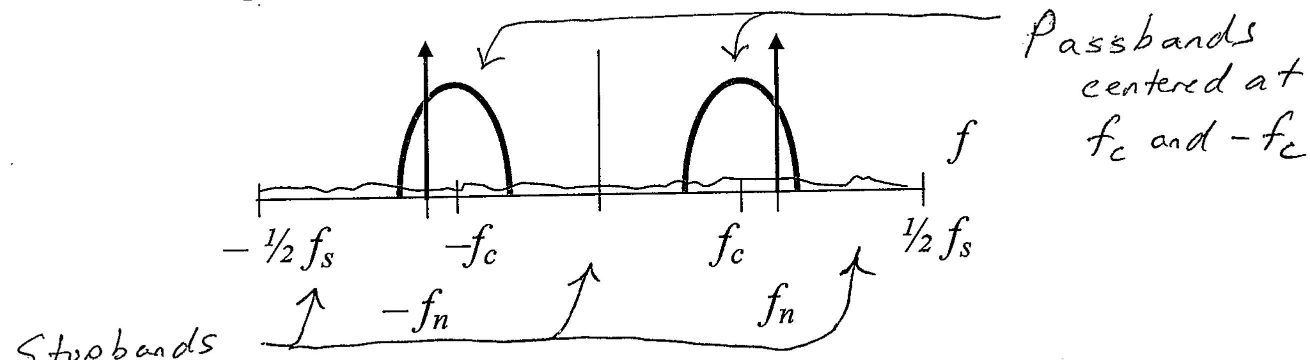
$$H_1(z) = C_1 \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}$$

Set the DC gain to be one and solve for C_1:

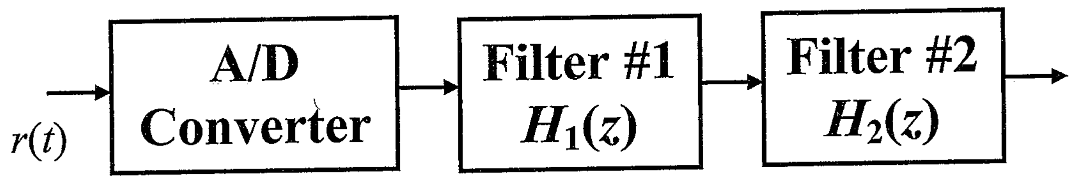
$$H(1) = C_1 \frac{(1 - z_0)(1 - z_1)}{(1 - p_0)(1 - p_1)} = 1$$

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 $f_s >> f_c$, 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.

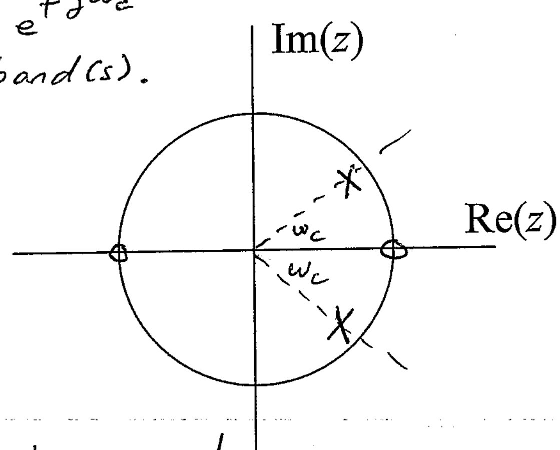
Poles near the unit circle indicate passband(s).

$\omega_c = 2\pi \frac{f_c}{f_s}$; $p_0 = 0.9\, e^{-j\omega_c}$; $p_1 = 0.9\, e^{+j\omega_c}$

Zeros on or near the unit circle indicate stopband(s).

$z_0 = 1$; $z_1 = -1$

(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: $H_2(z_c) = C_2 \frac{(1 - z_0 z_c)(1 - z_1 z_c)}{(1 - p_0 z_c)(1 - p_1 z_c)} = 1$ 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 the z-transform of both sides of the equation:

$$Y(z) = b\, X(z) + 2b\, z^{-1} X(z) + b\, z^{-2} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = b + 2b z^{-1} + b z^{-2} \quad \text{for } z \neq 0$$

2. Compute the frequency response. Since the unit circle is in the region of convergence of the transfer function,

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = b + 2be^{-j\omega} + be^{-2j\omega}$$

3. Isolate the magnitude and phase responses.

$$H_{freq}(\omega) = e^{-j\omega}\left(be^{j\omega} + 2b + be^{-j\omega}\right)$$

$$= \underbrace{e^{-j\omega}}_{\substack{\text{phase response} \\ \text{with phase } -\omega}} \underbrace{(2b\cos\omega + 2b)}_{\substack{\text{magnitude response} \\ \text{because } 2b\cos\omega + 2b \geq 0}}$$

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) to match human hearing; filtering of EKG signals

Bandstop — remove AM radio station in ADSL transmission band

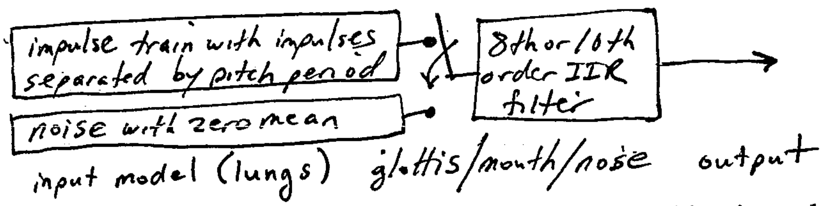
Highpass — enhance lines and texture in images (from in-class demo)

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)

(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 ≈ 100 Hz.
Female voice: pitch ≈ 200 Hz.
So, keep the same IIR filter model for the speech and change the pitch period of the input.

(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.

	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 near-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 ~~bout~~ 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.**

Problem	*Point Value*	*Your score*	*Topic*
1	25		Digital Filter Analysis
2	30		Upconversion
3	25		Digital Filter Design
4	20		Potpourri
Total	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

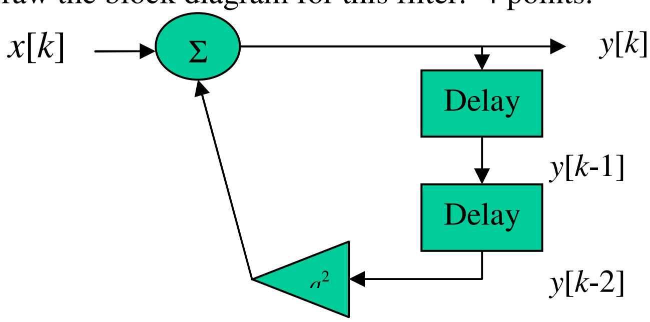
$$y[k] = a^2\, y[k-2] + (1-a)\, x[k]$$

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 $y[k\text{-}2]$. 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.

$\boldsymbol{y[0] = a^2\, y[-2] + (1-a)\, x[0]}$
$\boldsymbol{y[1] = a^2\, y[-1] + (1-a)\, x[1]}$
$\boldsymbol{y[2] = a^2\, y[0] \;\; + (1-a)\, x[2]}$

Hence, the initial conditions are y[-1] and y[-2], i.e., the initial values of the memory locations for $y[k-1]$ 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:

$\boldsymbol{Y(z) = a^2 z^{-2}\, Y(z) + (1-a)\, X(z)}$
$\boldsymbol{Y(z) - a^2 z^{-2}\, Y(z) = (1-a)\, X(z)}$
$\boldsymbol{(1 - a^2 z^{-2})\, Y(z) = (1-a)\, X(z)}$

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

Hence, poles are located at $z = a$ 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:

$$H_{freq}(\omega) = H(z)\big|_{z=e^{j\omega}} = \frac{1-a}{1-a^2 e^{-j2\omega}}$$

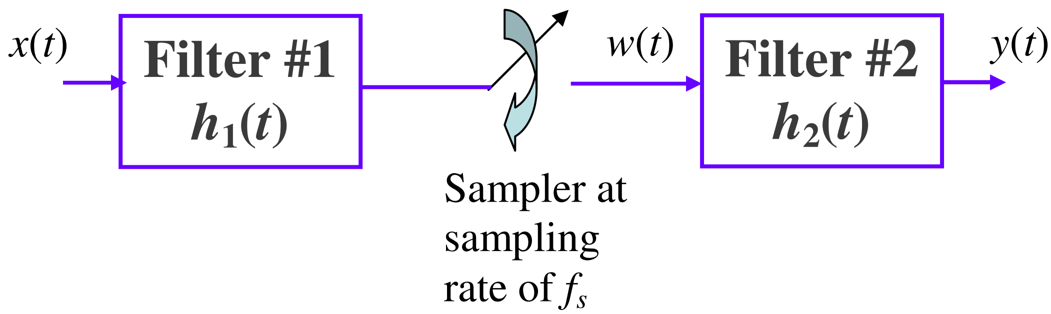
(f) Is this filter lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? What value of the parameter a would you use? 5 points.

Poles are at angles 0 rad/sample (low frequency) and π rad/sample (high frequency).
When a ≈ 0, the filter is close to allpass.
When a ≈ 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.



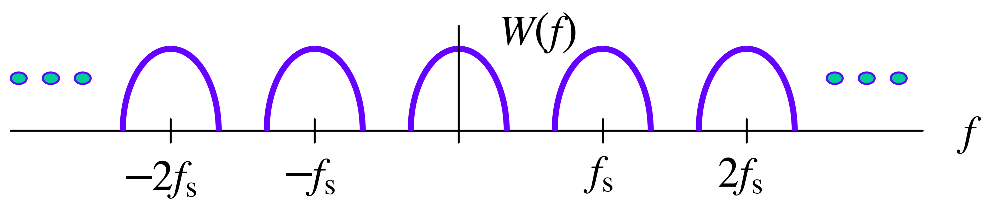
The output $y(t)$ should contain an AM radio signal at carrier frequency 590 kHz and an AM radio signal at carrier frequency 1300 kHz. The input $x(t) = 1 + k_a\, m(t)$, where $m(t)$ is the speech/audio signal to be broadcast. Since $m(t)$ could come from an audio CD, the bandwidth of $m(t)$ could be 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_{pass} = -1 dB and A_{stop} = -90 dB. The 90 dB of comes from the dynamic range of the audio CD. Also, f_{stop} < 5 kHz. We'll choose f_{pass} = 4.3 kHz and f_{stop} = 4.8 kHz. The transition region is roughly 10% of f_{pass}.**

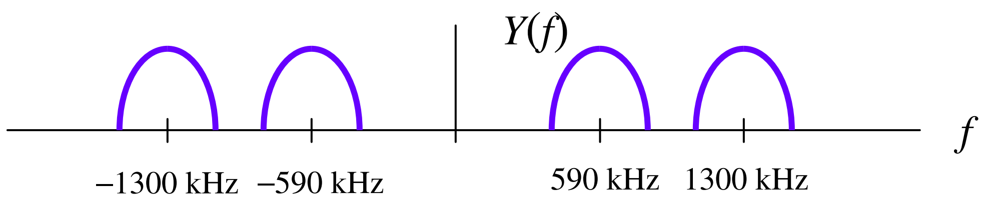
2) Give the sampling rate f_s of the sampler. **We want to produce replicas of filter #1 output centered at 590 kHz and 1300 kHz. Also $f_s > 2\, f_{max}$ and f_{max} = 4.8 kHz. So, f_s = 10 kHz.**

3) Draw the spectrum of $w(t)$. **Each lobe below is $2\, f_{max}$ wide.**

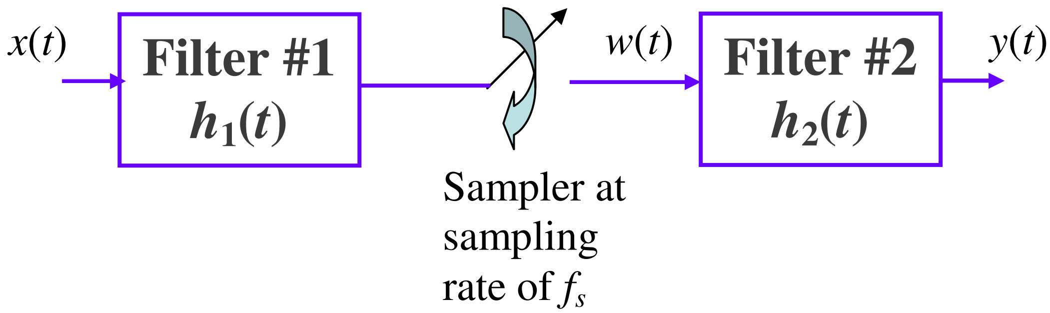


4) Give the filter specifications to design filter #2. **Filter #2 passbands are 585.7–594.3 kHz and 1295.7–1304.3 kHz, and stopbands are 0–585 kHz, 595–1295 kHz, and greater than 1305 kHz. These bands have counterparts in negative frequencies.**

5) Draw the spectrum of $y(t)$. **Each lobe below is $2\, f_{max}$ 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.**

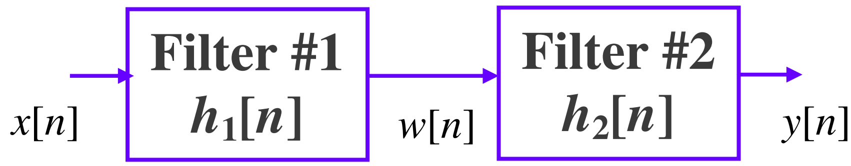
3) What filter design method would you use to design filter #1? Why?
I would use the Parks-McClellan (Remez exchange) algorithm to design the shortest linear phase FIR filter.

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.**

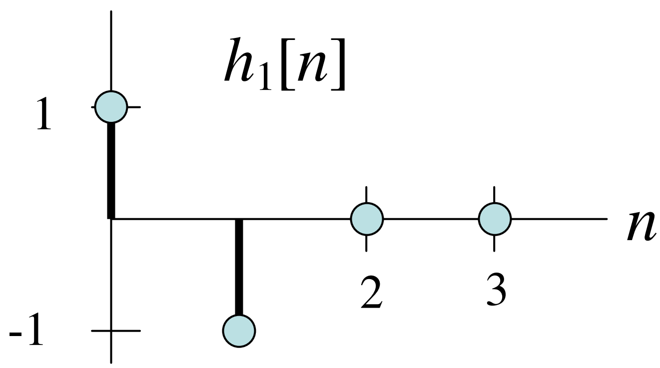
5) What filter design method would you use to design filter #2? Why? **Through homework assignments, we have designed multiband FIR filters using the Parks-McClellan (Remez exchange) algorithm. The Kaiser window method is for lowpass FIR filters. The FIR Least Squares method could be used. I would use 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 ½ 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 ½ 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 ½ 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.

$\boldsymbol{h_2[n] = \delta[n] + \delta[n-1]}$

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, round($-x$) = $-$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.**

Problem	*Point Value*	*Your score*	*Topic*
1	25		Digital Filter Analysis
2	25		Sinusoidal Generation
3	30		Digital Filter Design
4	20		Potpourri
Total	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 $1 \le a \le 2$.

(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:

$$c[n] = \cos\left(\frac{\pi}{180} n\right) \quad \text{for } n = 0, 1, ..., 359.$$

(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.

$$s[n] = \sin\left(\frac{\pi}{180} n\right) \quad \text{for } n = 0, 1, ..., 359?$$

(c) How would you most efficiently use the above ROM table $c[n]$ to compute $d[n]$ given below. 5 points.

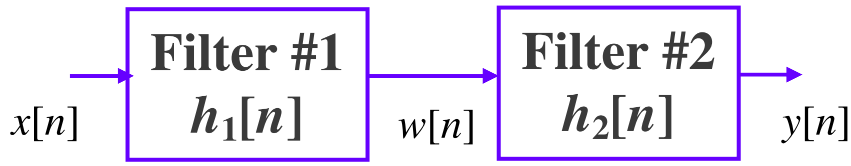
$$d[n] = \cos\left(\frac{\pi}{90} n\right) \quad \text{for } n = 0, 1, ..., 179$$

(d) How would you most efficiently use the above ROM table $c[n]$ to compute $x[n]$ given below. 10 points.

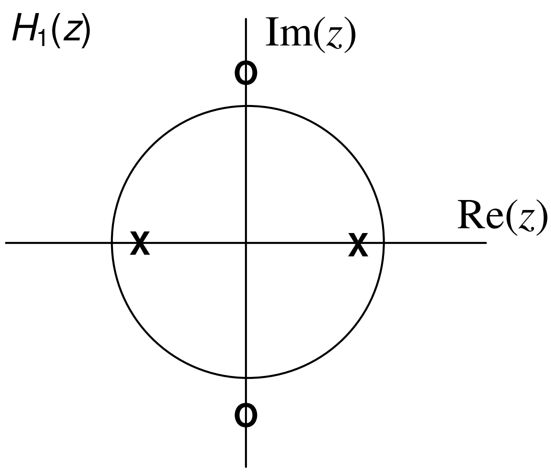
$$d[n] = \cos\left(\frac{\pi}{360} n\right) \quad \text{for } n = 0, 1, ..., 719$$

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:

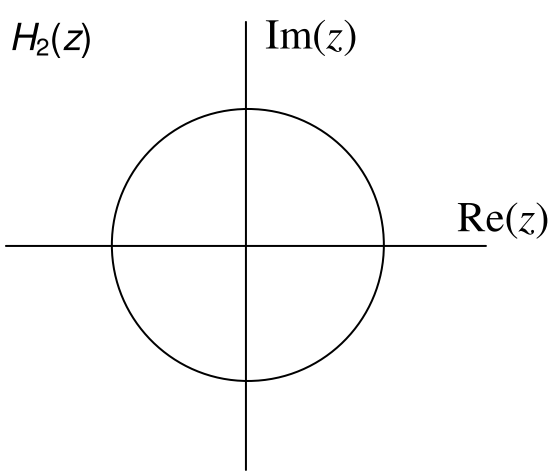


(a) Poles (X) and zeros (O) for filter #1 are shown below. Assume that the poles have radii of 0.9, and the zeros have radii of 1.2. Is filter #1 a lowpass, highpass, bandpass, bandstop, allpass or notch filter? Why? 10 points.



(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.



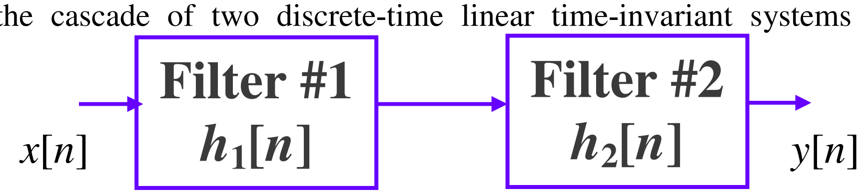
(d) Give the transfer function $H_2(z)$. 5 points.

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.**

Problem	*Point Value*	*Your score*	*Topic*
1	25		Digital Filter Analysis
2	25		Digital Filter Design
3	30		Software Defined Receiver
4	20		Potpourri
Total	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

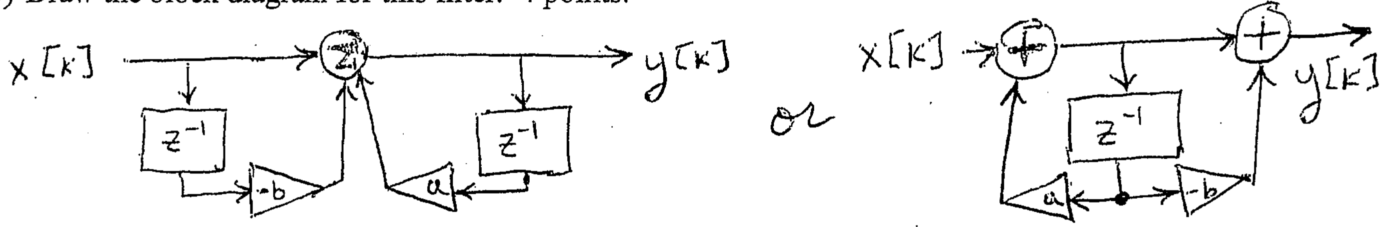
$$y[k] = a\, y[k-1] + x[k] - b\, x[k-1]$$

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 guarantee LTI properties.

(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:

$$Y(z) = a z^{-1} Y(z) + X(z) - b z^{-1} X(z)$$

$$(1 - a z^{-1}) Y(z) = (1 - b z^{-1}) X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - b z^{-1}}{1 - a z^{-1}}$$

Pole at $z = a$:

$$1 - a z^{-1} = 0$$

$$a z^{-1} = 1$$

$$z = a$$

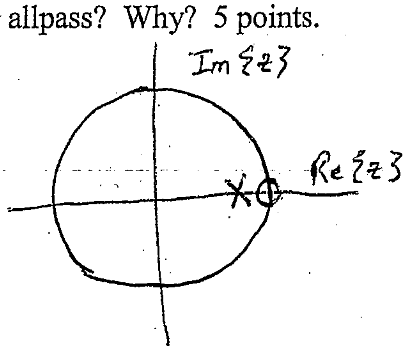
For causal system, $|z| > |a|$

(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.

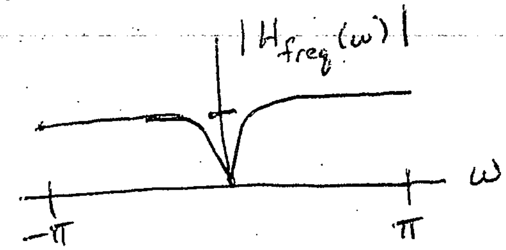


Pole at $z = 0.9$, i.e. at $\omega = 0$

Zero at $z = 1$, i.e. at $\omega = 0$

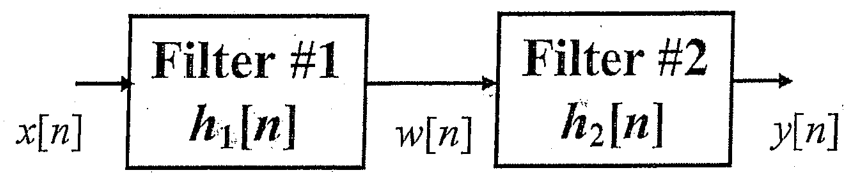
Notch filter.

DC removed.



Problem 1.2 *Digital Filter Design.* 25 points.

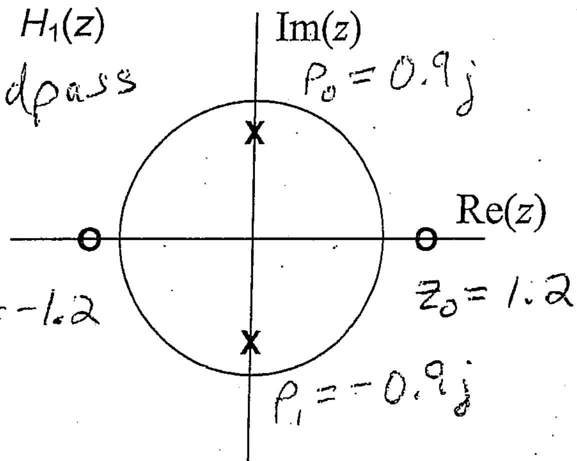
Consider the following cascade of two causal discrete-time linear time invariant (LTI) filters with impulse responses $h_1[n]$ and $h_2[n]$, respectively:



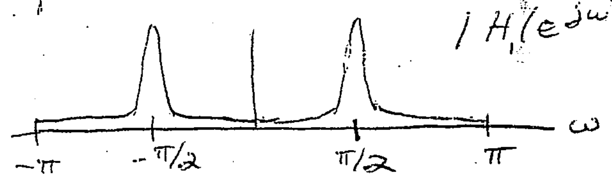
(a) Poles (X) and zeros (O) for filter #1 are shown below. Assume that the poles have radii of 0.9, and the zeros have radii of 1.2. Is filter #1 a lowpass, highpass, bandpass, bandstop, allpass or notch filter? Why? 10 points.

$H_1(z)$

Bandpass



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.

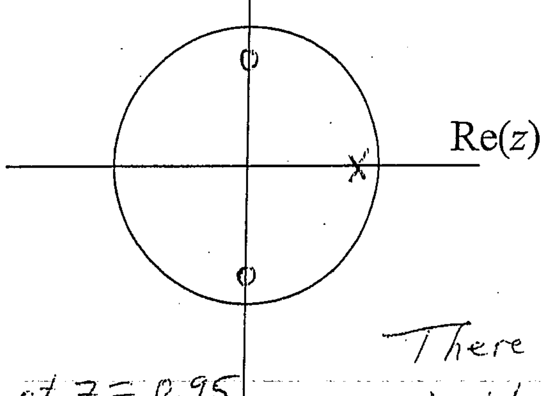
$$H_1(z) = C_1 \frac{(z-z_0)(z-z_1)}{(z-p_0)(z-p_1)}$$

Normalize at DC, i.e. $\omega = 0$ or $z = e^{j\omega} = 1$ for $\omega = 0$,

$H_1(z)\Big|_{z=1} = 1$ and solve for C_1

(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.

$H_2(z)$

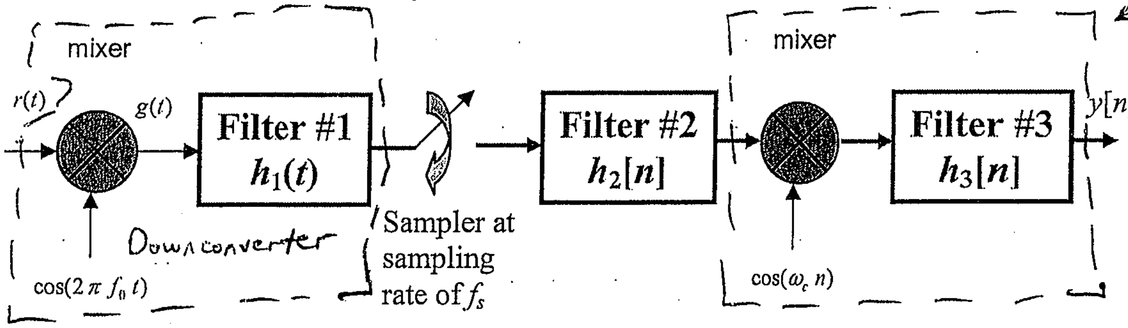


Pole at $z = 0.95$.
Upper zero at $z = 0.9j$.
Lower zero at $z = -0.9j$.

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.

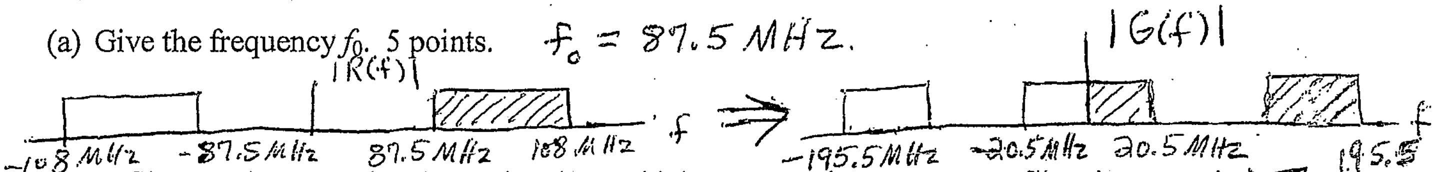
The digital FM radio below has two stages of downconversion, one in continuous time and one in discrete time. Only the discrete-time stage is dependent on the FM station selected by the user, and hence, it can be controlled by software.



The radio receives a bandpass waveform $r(t)$ containing all of the FM radios 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. $f_0 = 87.5$ 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.
$f_{pass} = 20.5$ MHz, $f_{stop} = 22.75$ MHz. $A_{pass} = 1$ dB. $A_{stop} = 40$ dB

(c) Give the constraints on the sampling rate f_s. 5 points.

$f_s > 2 f_{stop}$

Let $\omega_{band} = 2\pi \frac{200 \text{ kHz}}{f_s}$ 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 ω_{band} in its passband. Allow rolloff of $0.1\,\omega_{band}$.

(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. $\omega_{pass} = 0.9\,\omega_{stop}$. $\omega_{stop} = \frac{1}{2}\omega_{band}$.
$A_{pass} = 1$ dB. $A_{stop} = 40$ dB.

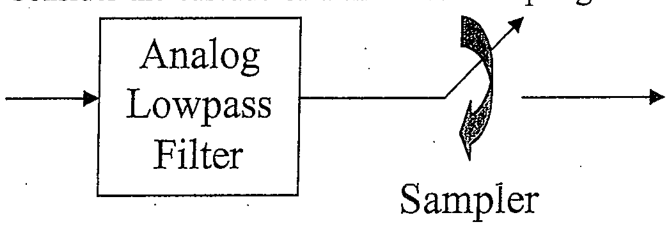
Problem 1.4. *Potpourri.* 20 points.

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 chosen according $f_s > 2 f_{max}$ 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 $\omega_0 = \pi$.
$$x[n] = \cos(\pi n) \longrightarrow \boxed{z^{-2}} \longrightarrow y[n] = \cos(\pi n)$$
$y[n] = \cos(\pi(n-2)) = \cos(\pi n - 2\pi)$ 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, $x[n] = \cos(\omega_0 n) \Rightarrow y[n] = |H_{freq}(\omega_0)| \cos(\omega_0 n + \measuredangle H_{freq}(\omega_0))$.
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.**

Problem	*Point Value*	*Your score*	*Topic*
1	25		Digital Filter Analysis
2	30		Digital Filter Implementation
3	25		Simultaneous Broadcast
4	20		Potpourri
Total	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

$$y[n] = a^2\, y[n\text{-}2] + x[n] + x[n\text{-}2]$$

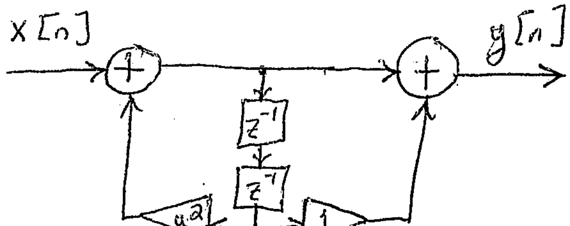
where $|a| < 1$.

(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,

$y[0] = a^2 y[-2] + x[0] + x[-2]$
$y[1] = a^2 y[-1] + x[1] + x[-1]$ $\Rightarrow$ initial conditions are $y[-1]$, $y[-2]$, $x[-1]$ and $x[-2]$

The initial conditions should be set to zero so that the filter satisfies linearity and time-invariant properties.

(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,

$$Y(z) = a^2 z^{-2} Y(z) + X(z) + z^{-2} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + z^{-2}}{1 - a^2 z^{-2}} \quad \text{for } |z| > |a|$$

(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:

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{1 + e^{-2j\omega}}{1 - a^2 e^{-2j\omega}}$$

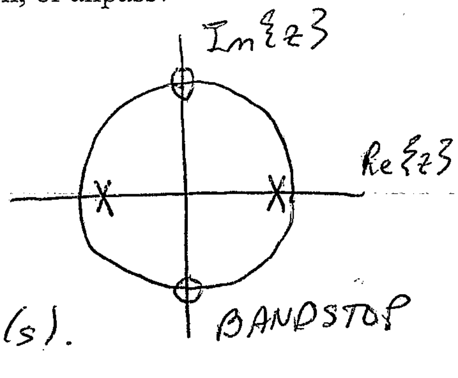
(f) If $a = 0.9$, would this filter be lowpass, bandpass, bandstop, highpass, notch, or allpass? Why? *5 points.*

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).



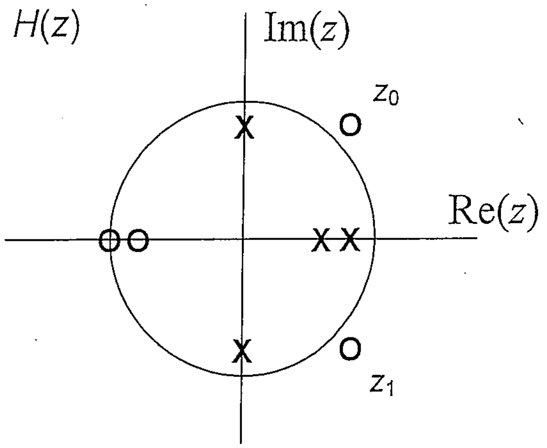
BANDSTOP

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 $h[n]$.

$$x[n] \longrightarrow \boxed{\begin{array}{c}\textbf{Filter \#1}\\ h_1[n]\end{array}} \xrightarrow{w[n]} \boxed{\begin{array}{c}\textbf{Filter \#2}\\ h_2[n]\end{array}} \longrightarrow y[n]$$

Poles (X) and zeros (O) for the cascade of the filters are shown below.

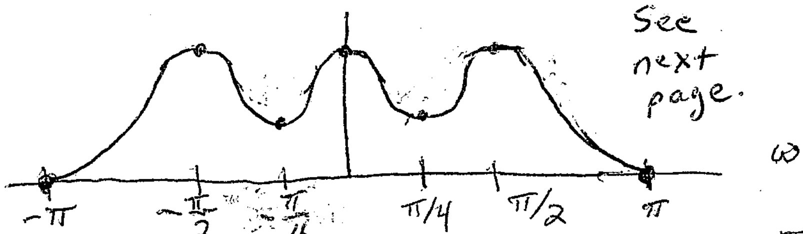


Zero locations
$z_0 = 1.05\, e^{j\pi/4}$
$z_1 = 1.05\, e^{-j\pi/4}$
$z_2 = -1$
$z_3 = -0.9$

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 $z = 1$:

$$H(1) = C\frac{(1-z_0)(1-z_1)(1-z_2)(1-z_3)}{(1-p_0)(1-p_1)(1-p_2)(1-p_3)} = 1$$

(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 $\omega = -\frac{\pi}{2}$

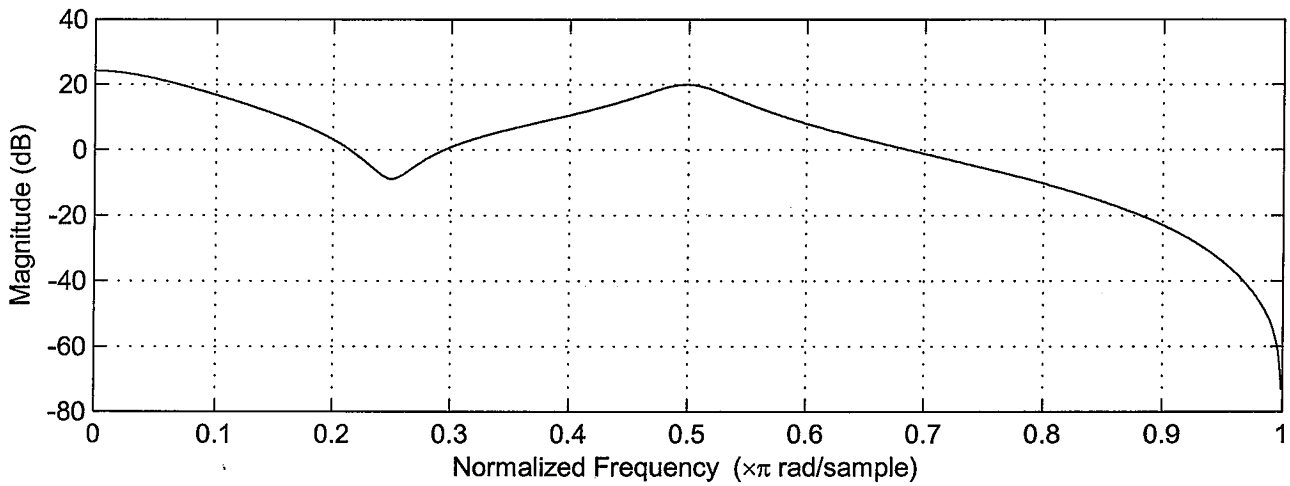
(d) Is filter #1 a lowpass, highpass, bandpass, bandstop, allpass or notch filter? Why? 5 points.

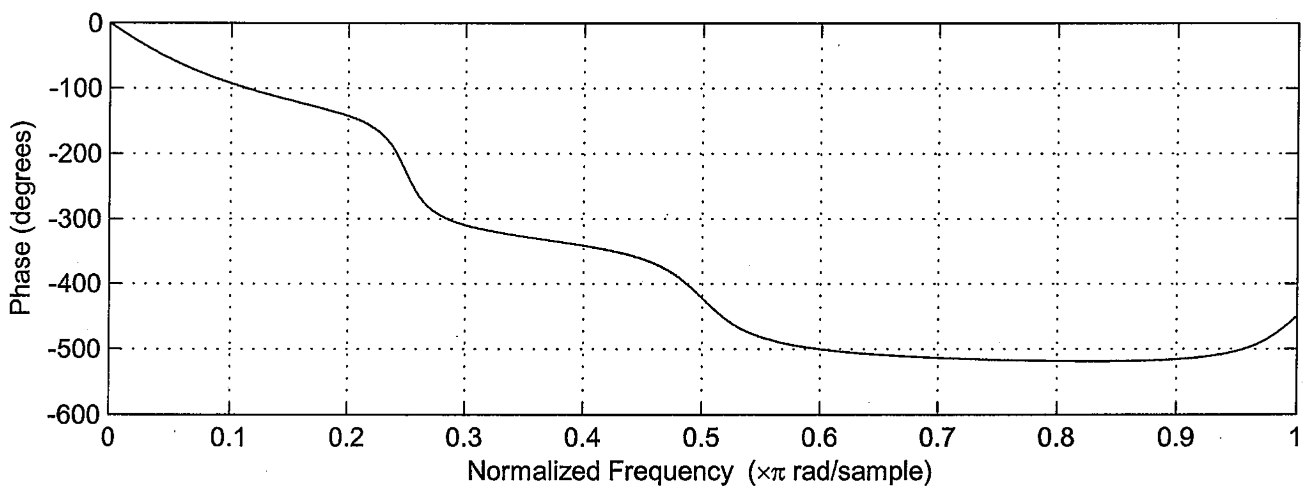
Lowpass. Passband is centered at $\omega = 0$. Stopbands at $\omega = \frac{\pi}{4}$ and $\omega = -\frac{\pi}{4}$.

(e) Which poles and zeros would you choose for filter #2? Why? 5 points.

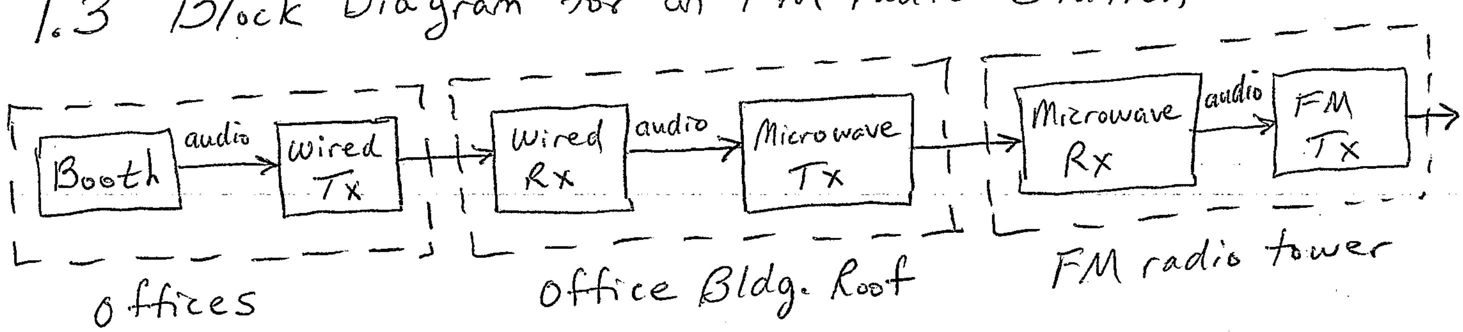
We'll choose the remaining poles p_0 and p_3 and the remaining zeros z_2 and z_3.

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..

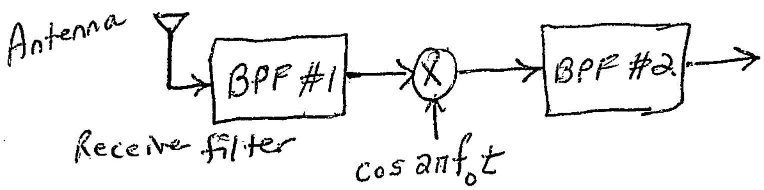
Please see previous page.

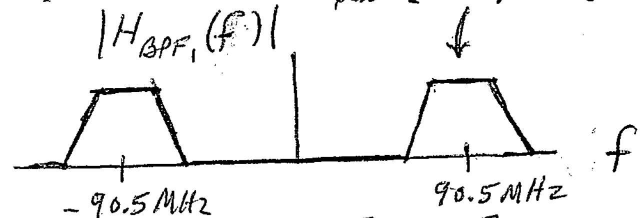
(a) Draw a block diagram using mixers and filters to directly convert an FM radio transmission from a station freqency of 90.5 MHz to a station frequency on 90.1 MHz. Describe the way you would choose the parameters for each block; e.g., for filters, give the design specifications and the filter design method you would use. In your answer, the FM transmission should **not** be downconverted to baseband. *10 points.*

$A_{pass} = 1\ dB$

$A_{stop} = 40\ dB$

$f_{pass} \in [90.41, 90.59]\ MHz$





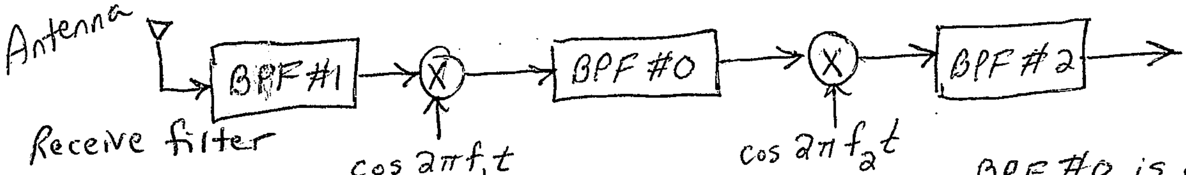
$f_0 = 90.5\ MHz - 90.1\ MHz = 0.4\ MHz$

$f_{stop_1} \in [0, 90.4]\ MHz$

BPF #2 is same shape as BPF #1 with $f_c = 90.1\ MHz$.

$f_{stop_2} \in [90.6, \infty)\ MHz$

(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.*



BPF #1 and BPF #2 are same as in (a).

$f_1 = 90.5\ MHz - 10\ MHz = 80.5\ MHz$

$f_2 = 90.1\ MHz - 10\ MHz = 80.1\ MHz$

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 $H_{freq}(\omega)$ 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 Fourier transform of $u[n]$ is

$$\frac{1}{1-e^{-j\omega}} + \sum_{m=-\infty}^{\infty} \delta(\omega - 2\pi m)$$

This does not match $\frac{1}{1-e^{-j\omega}}$

(c) Discrete-time finite impulse response (FIR) filters are always bounded-input bounded-output stable. *5 points.*

True. Assume that the FIR filter is linear and time-invariant (init. cond. are zero). LTI system is BIBO stable if

$$\sum_{n=0}^{\infty} |h[n]| < \infty$$

$$\underbrace{\sum_{n=0}^{N-1} |h[n]| < \infty}$$

Finite sum of finite values ⇒ must be finite

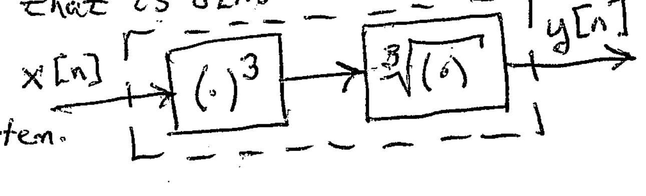
(d) If $\delta[n]$ were input to a discrete-time linear time-invariant system and the output were also $\delta[n]$, then system could only be the identity system, i.e. the output is always equal to the input. *5 points.*

This 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 system. 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.**

Problem	*Point Value*	*Your score*	*Topic*
1	25		Digital Filter Analysis
2	27		Sinusoidal Generation
3	30		ECG Filter Design
4	18		Potpourri
Total	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

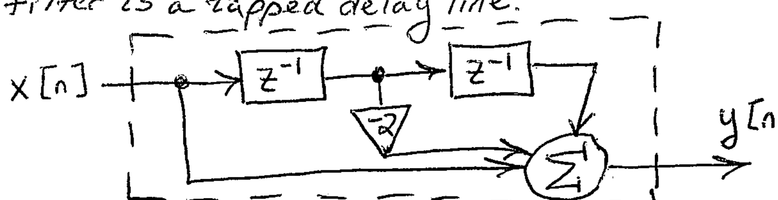
$$y[n] = x[n] - 2\,x[n-1] + x[n-2]$$

(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 non-zero extent of 3 samples.

(b) Draw the block diagram for this filter. *4 points.*

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.*

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

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

$$\frac{Y(z)}{X(z)} = 1 - 2z^{-1} + z^{-2} \quad \text{for } z \neq 0$$

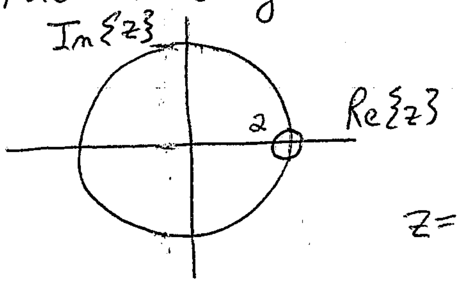
(e) Find the equation for the frequency response of the filter. *5 points.*

Since the region of convergence includes the unit circle,

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = 1 - 2e^{-j\omega} + e^{-j2\omega}$$

(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 $u(nT_s) = \frac{1}{T_s} u(n) = \frac{1}{T_s} u[n]$. This identity is used in the solution for part (a).

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

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

(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 $T_s \cos(2\pi f_0 t)\, u(t)$.

Sample at uniformly spaced samples: $t = n T_s = \frac{n}{f_s}$:

$$x[n] = \cos\left(2\pi \frac{f_0}{f_s} n\right) u[n] \Rightarrow \omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{N}{L} \Rightarrow f_0 = \frac{N}{L} f_s$$

(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 seven values for f_s.

(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 $n=0$.

For example, $x[L] = \cos\left(2\pi \frac{N}{\cancel{L}} \cancel{L}\right) = \cos(0) = x[0]$

(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 Mth sample in the lookup table for $x[n]$ by using

$$y[k] = x[M k] \quad \text{for } k = 0, 1, \ldots, L_y - 1 \text{ where } L_y = L / M$$ a.k.a. downsampling by M

How many different values of M could be used without introducing aliasing? *9 points.*

$$y[k] = x[Mk] = \cos\left(2\pi \frac{N}{L}(Mk)\right) u[Mk] = \cos\left(2\pi \frac{N}{\left(\frac{L}{M}\right)} k\right) u[Mk]$$

$$= \cos\left(2\pi \frac{N}{L_y} k\right) u[Mk]$$

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:

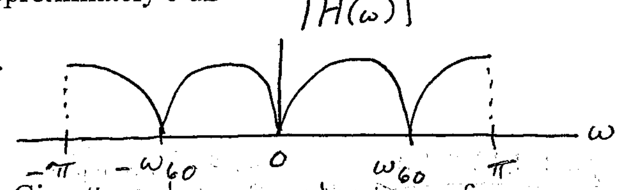
Recall from problem 1.2 that $\omega_0 = 2\pi \frac{f_0}{f_s}$

- 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

$\omega_{60} = 2\pi \frac{60}{200} = \frac{3}{5}\pi$

The IIR filter is to have an equal number of poles and zeros.

The IIR filter is to have a minimum number of poles.



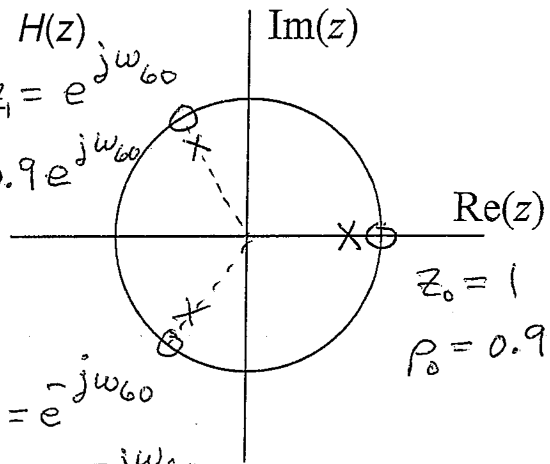
(a) Design the filter by manually placing poles and zeros. Give the pole and zero locations of the filter. *20 points.*

Use a cascade of two notch filters. First notch filter is at 0 Hz. Second notch filter is at 60 Hz.

$H(z)$

$z_1 = e^{j\omega_{60}}$

$P_1 = 0.9 e^{j\omega_{60}}$



$z_0 = 1$

$P_0 = 0.9$

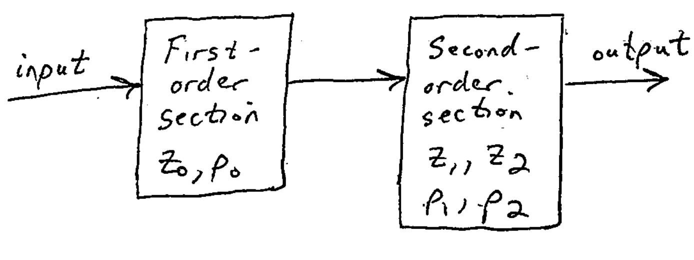
$z_2 = e^{-j\omega_{60}}$

$P_2 = 0.9 e^{-j\omega_{60}}$

(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). Use either alternate method.

(a) Consider generating a sinusoidal waveform on the TMS320C6713 digital signal processor by storing one period of the sinusoid in a lookup table, executing a difference equation, or 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 flp	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 handcoded 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 the coefficient array and past/present inputs fit into on-chip memory. C6713 on the DSK has 1 Kword of L1 data memory. Longest possible filter is 512 coefficients.

(c) Consider implementing a finite impulse response (~~IIR~~ 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.

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.**

Problem	*Point Value*	*Your score*	*Topic*
1	28		Digital Filter Analysis
2	30		Filter Design Tradeoffs
3	24		Downconversion
4	18		Potpourri
Total	100		

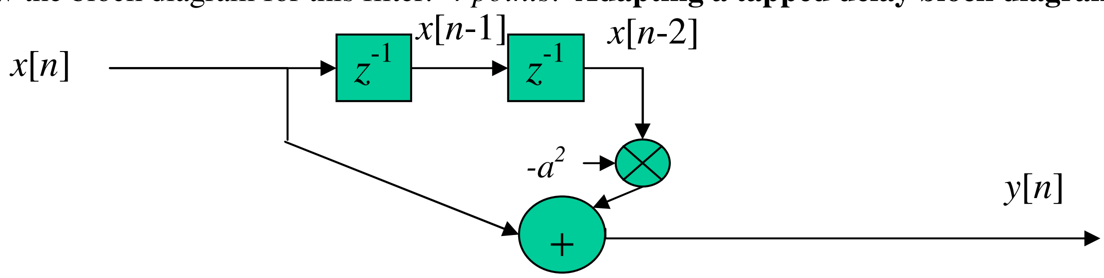
Problem 1.1 *Digital Filter Analysis*. 28 points.

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*.

$y[0] = x[0] - a^2\, x[-2]$
$y[1] = x[1] - a^2\, x[-1]$
$y[2] = x[2] - a^2\, x[0]$

Hence, the initial conditions are x[-1] and x[-2], i.e., the initial values of the memory locations for $x[n-1]$ 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**

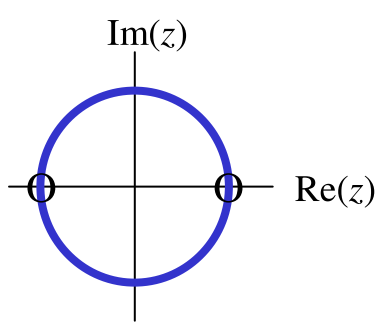
$Y(z) = X(z) - a^2 z^{-2} X(z)$, which gives $Y(z) = (1 - a^2 z^{-2})\, X(z)$: $H(z) = \dfrac{Y(z)}{X(z)} = 1 - a^2 z^{-2}$

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:**

$$H_{freq}(\omega) = H(z)\big|_{z=e^{j\omega}} = 1 - a^2 e^{-j2\omega}$$

(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**

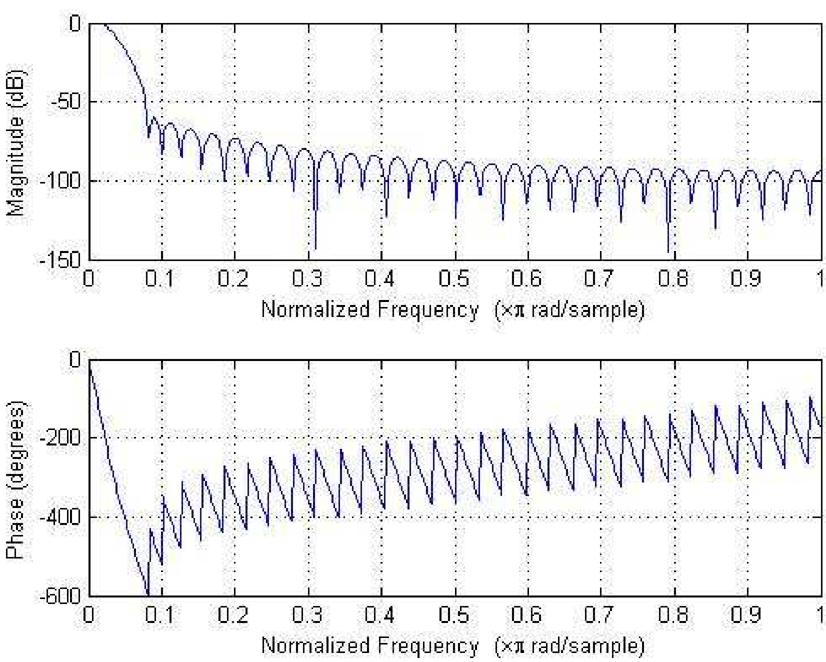
BIBO stability: YES or NO. Why? **Yes, the poles are inside the unit circle. The quality factor is quite low; hence, the poles are unlikely to become BIBO unstable when implemented.**

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 Hz) / (1000 Hz). Pole locations are at 0.95 exp($j\,\omega_0$) and 0.95 exp($-j\,\omega_0$).**

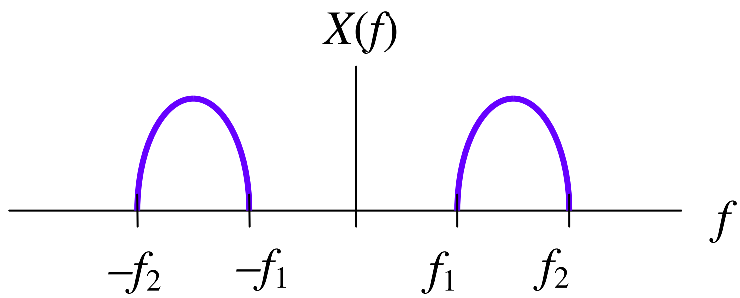
Matlab code to design the filter for part (c), which was not required for the test:

```
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 $f_c = ½ (f_1 + f_2)$ where $f_c > 2\,B$.

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

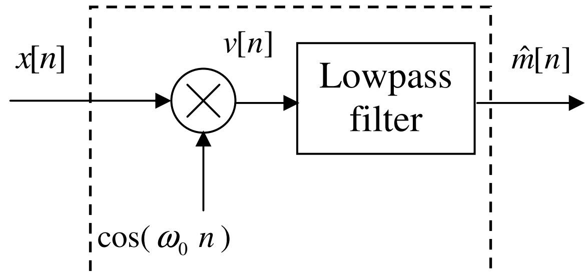
ω_{stop} be the stopband frequency of a discrete-time filter in rad/sample

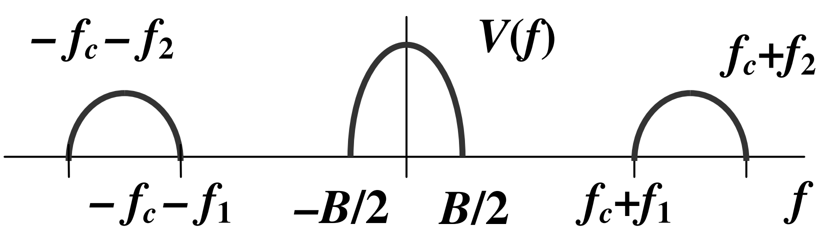
(a) Downconversion method #1. *12 points*,

Uses sinusoidal amplitude demodulation.

Give formulas for ω_0, ω_{pass}, ω_{stop} and f_s.

Analyze in continuous-time first.





(b)

$$\omega_0 = 2\pi \frac{f_c}{f_s} \qquad f_s > 2(2f_c + B/2)$$

$$\omega_{\text{pass}} = 2\pi \frac{B/2}{f_s} \qquad \omega_{\text{stop}} = 2\pi \frac{f_c + f_1}{f_s}$$

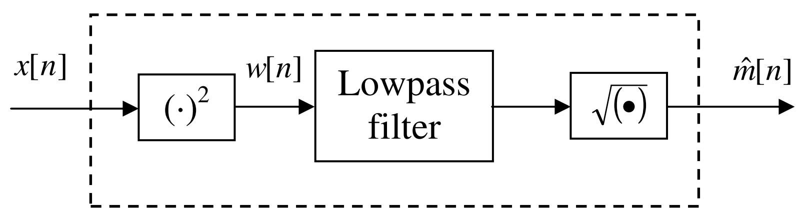
(c) Downconversion method #2. *12 points.*

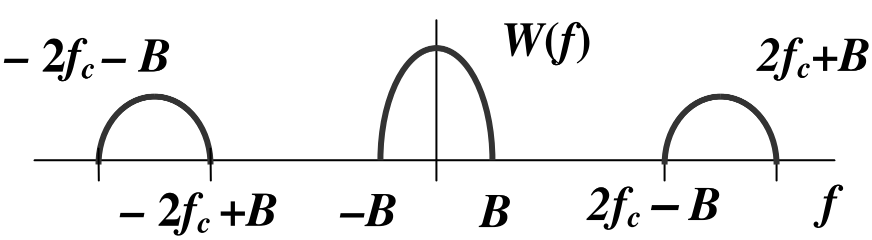
Uses a squaring device. Assume that output values of the lowpass filter are non-negative.

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 $x[n] + p_1\ y_1[n\text{-}1]$ 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.

Problem	*Point Value*	*Your score*	*Topic*
1	28		Digital Filter Analysis
2	30		Sinusoidal Generation
3	24		Upconversion
4	18		Potpourri
Total	100		

Problem 1.1 *Digital Filter Analysis*. 28 points.

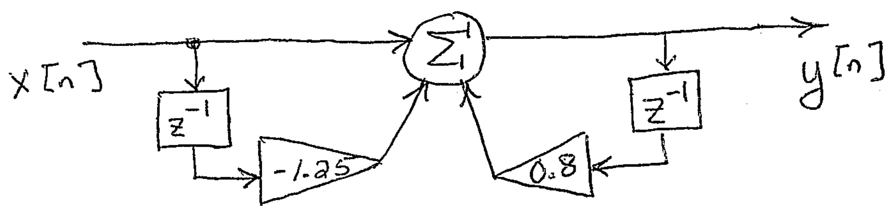
A causal discrete-time linear time-invariant filter with input $x[n]$ and output $y[n]$ is governed by the following difference equation:

$$y[n] - 0.8\, y[n\text{-}1] = x[n] - 1.25\, x[n\text{-}1]$$

(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.*

$$Y(z) - 0.8 z^{-1} Y(z) = X(z) - 1.25 z^{-1} X(z)$$

$$Y(z)(1 - 0.8 z^{-1}) = X(z)(1 - 1.25 z^{-1})$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - 1.25 z^{-1}}{1 - 0.8 z^{-1}}$$

zero at $z = 1.25$
pole at $z = 0.8$

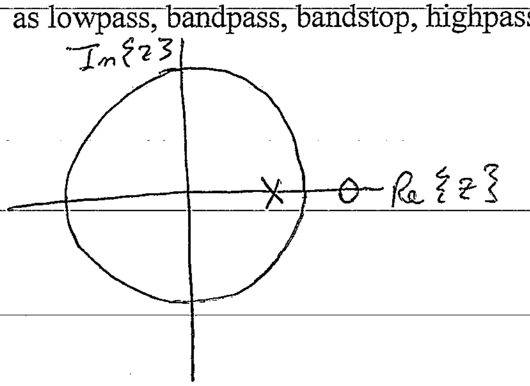
ROC $|z| > 0.8$ for causality.

(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.

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{1 - 1.25 e^{-j\omega}}{1 - 0.8 e^{-j\omega}}$$

(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 $\cos(2\pi f_0 t) u(t)$.

f_{max} is slightly greater than f_0.

(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.*

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. There are 10 terms. Assume remaining 19 MACs can be pipelined in C math library call.

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 × 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 × 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 ×~~ 2 bytes/word) 10 bytes	2	~~$2 \cdot 1 + 2 =$~~ 4 cycles

← From slide 2-16 →

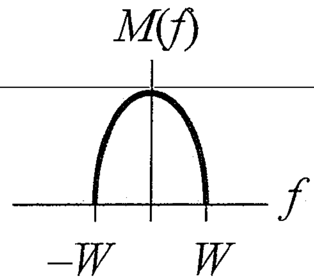
In 16-bit format, one could compute two multiplies in parallel, because there are two multiply units.

(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.

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 $f_c > 3$ W

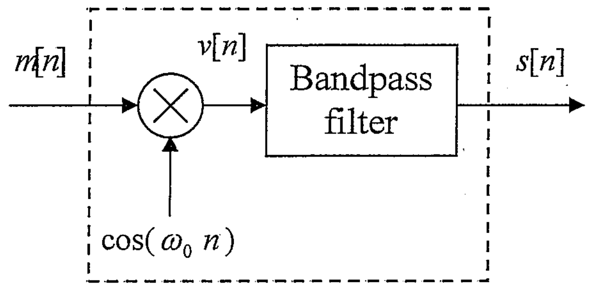
f_s : sampling rate in Hz for sampling of $m(t)$ to give $m[n]$ and $s(t)$ to give $s[n]$

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:

$\omega_0 = 2\pi \dfrac{f_c}{f_s}$

$B = 2W$

$\omega_{stop1} = 0.9\, \omega_{pass_1}$

$\omega_{pass1} = 2\pi \dfrac{f_c - W}{f_s}$

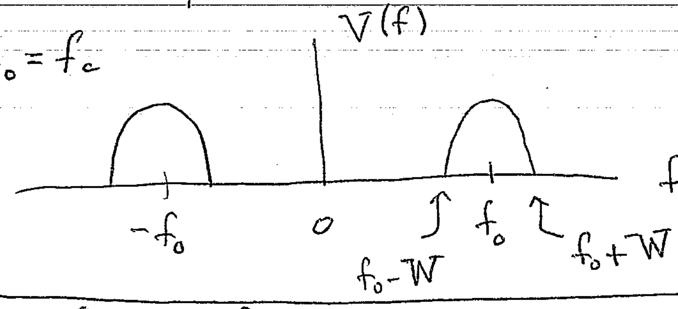
$\omega_{pass2} = 2\pi \dfrac{f_c + W}{f_s}$

$\omega_{stop2} = 1.1\, \omega_{pass_2}$

$f_s > 2\,(1.1\,(f_c + W))$

Work the problem in continuous-time first:

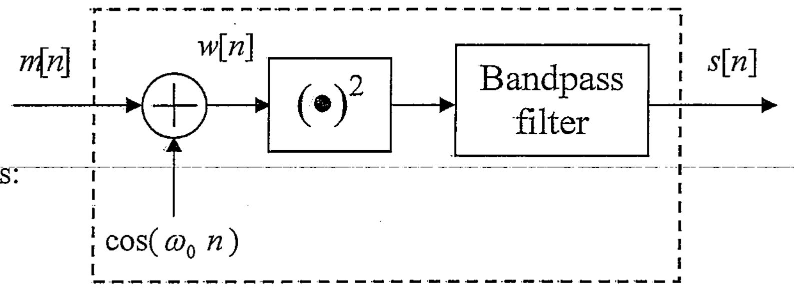
$f_0 = f_c$



Software Receiver Design Exercise 5.10

(b) Upconversion method #2. *10 points.*

Uses a squaring device and the bandpass filter from part (a).



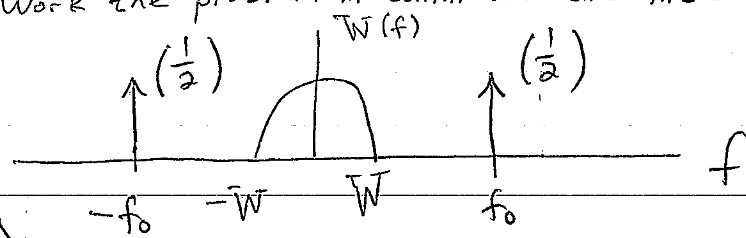
Give formulas for the following parameters:

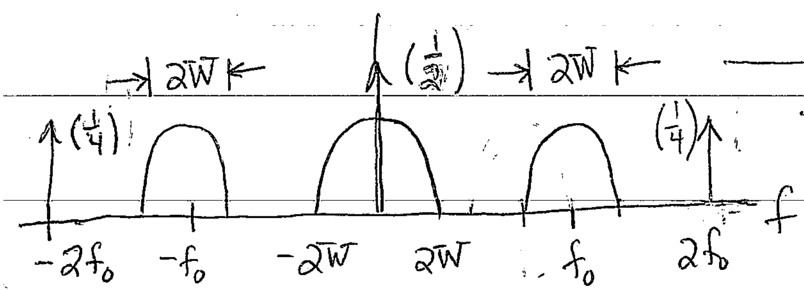
$\omega_0 = 2\pi \dfrac{f_c}{f_s}$

$B = 2W$

$f_s > 2\,(2 f_c)$

Work the problem in continuous-time first:





convolution with itself

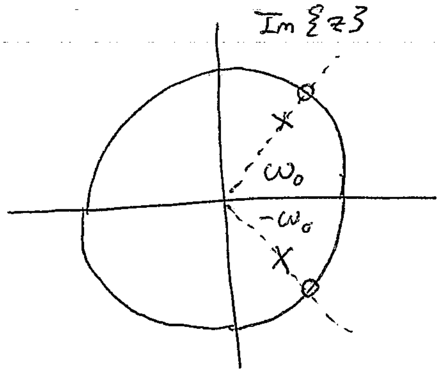
$(f_0 = f_c)$

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: October 15, 2010 Course: EE 445S 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(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**.**

Problem	*Point Value*	*Your score*	*Topic*
1	28		Digital Filter Analysis
2	30		Sinusoidal Generation
3	24		Upconversion
4	18		Potpourri
Total	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*.

(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*.

(d) Find the equation for the transfer function of the filter 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) 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*.

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 must f_s take to prevent aliasing? *6 points*.

(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*.

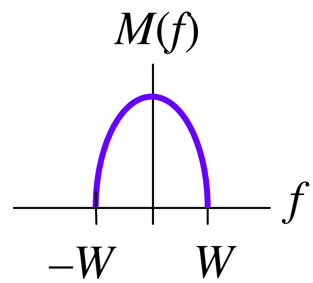
Data Size	Operation	Throughput	Delay Slots
16-bit short	addition	1 cycle	0 cycles
16-bit by 16-bit short	multiplication	1 cycle	1 cycle
32-bit floating-point	addition	1 cycle	3 cycles
32-bit x 32-bit floating-point	multiplication	1 cycle	3 cycles
64-bit floating-point	addition	2 cycles	6 cycles
64-bit x 64-bit floating-point	multiplication	4 cycles	9 cycles

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			
Difference equation using 32-bit floating-point data/arithmetic			
Difference equation using arithmetic on 16-bit (short) data and coefficients			

(c) Which method would you advocate using? Why? *6 points*.

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 $f_c > 3$ W

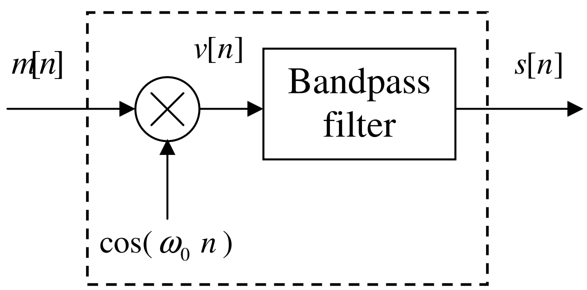
f_s : sampling rate in Hz for sampling of $m(t)$ to give $m[n]$ and $s(t)$ to give $s[n]$

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:

$\omega_0 =$

$B =$

$\omega_{stop1} =$

$\omega_{pass1} =$

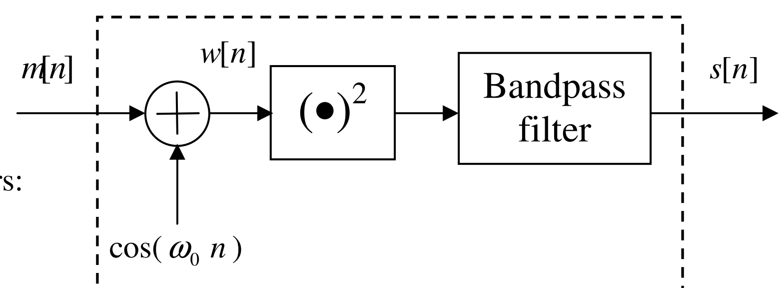
$\omega_{pass2} =$

$\omega_{stop2} =$

$f_s >$

(b) Upconversion method #2. *10 points*.

Uses a squaring device and the bandpass filter from part (a).



Give formulas for the following parameters:

$\omega_0 =$

$B =$

$f_s >$

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*.

(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*.

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 11, 2011 Course: EE 445S Evans

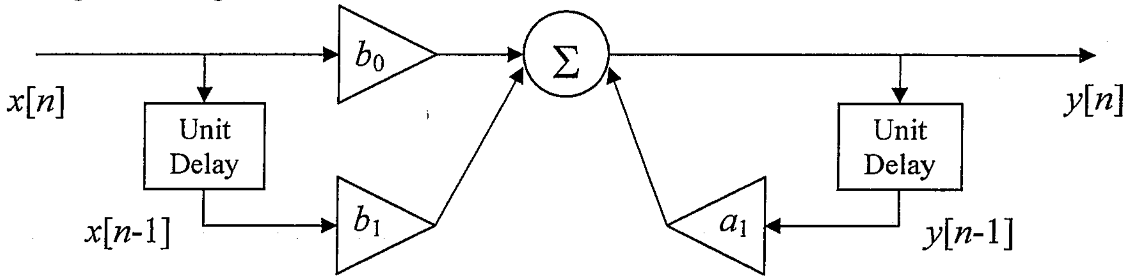
Name: Set, Solution
Last, First

- 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.

Problem	*Point Value*	*Your score*	*Topic*
1	28		Discrete-Time Filter Analysis
2	18		Discrete-Time IIR Filtering
3	30		Analysis and Synthesis
4	24		Downconversion
Total	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.*

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

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

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \; ; \quad \text{RoC is } |z| > |a_1|$$

(e) Find the equation for the frequency response of the filter. *5 points.*

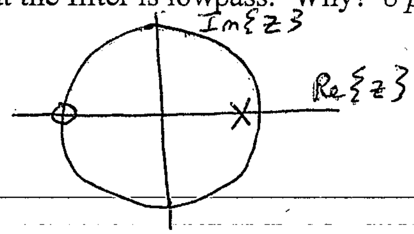
Since $|a_1| < 1$ because this is a filter, the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{b_0 + b_1 e^{-j\omega}}{1 - a_1 e^{-j\omega}}$$

(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.

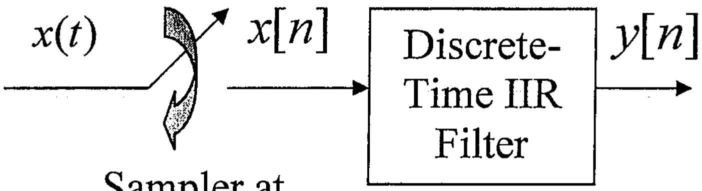
Assume that odd harmonics (180 Hz, 300 Hz, etc.) are also present.

Assume that even harmonics (120 Hz, 240 Hz, etc.) are not present.

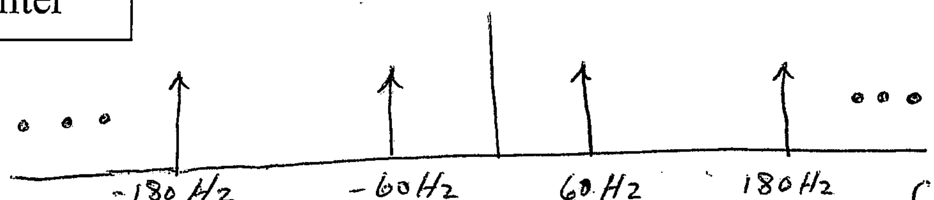
Design the following signal processing system to remove 60 Hz and its odd harmonics.

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.



Sampler at sampling rate of f_s



(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:

- $f_s > 2 f_{max}$ $\rightarrow$ $f_s \sim 120$ 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 $180 - f_s = 60$

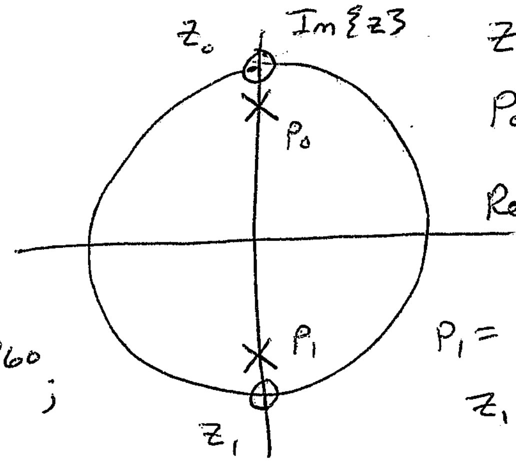
  This gives $f_s = 240$ Hz or $f_s = 120$ Hz

  We rule out $f_s = 120$ 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.*

$$\omega_0 = 2\pi \frac{f_0}{f_s}$$

$$\omega_{60} = 2\pi \frac{60\text{ Hz}}{240\text{ Hz}} = \frac{\pi}{2}$$



$z_0 = j$

$p_0 = 0.9j$

$p_1 = -0.9j$

$z_1 = -j$

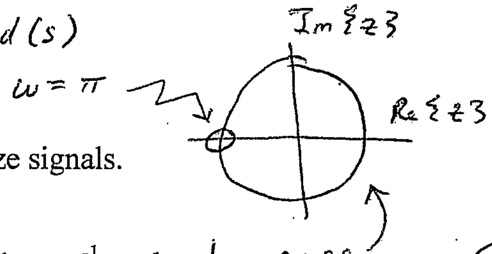
$p_0 = 0.9\, e^{j\omega_{60}}$; $p_1 = 0.9\, e^{-j\omega_{60}}$;

$z_0 = e^{j\omega_{60}}$; $z_1 = e^{-j\omega_{60}}$

$$H(z) = C \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})};$$

Set $H(1) = 1$ and solve for C

Zeros on or near unit circle indicate stopband(s)



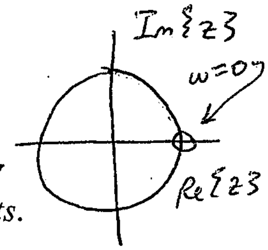
Problem 1.3. *Analysis and Synthesis*. 30 points.

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 lowpass

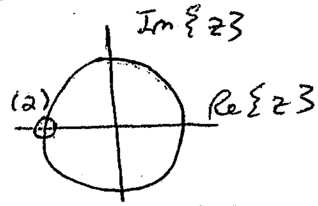
a first-order difference filter with a transfer function of $H_2(z) = 1 - z^{-1}$. highpass



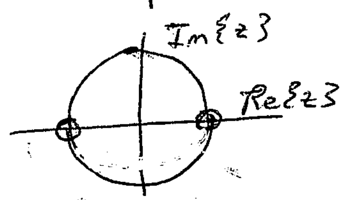
(a) What is the frequency selectivity of the following combination of these filters: *12 points*.

i. $H_1(z) + H_2(z) = (1 + z^{-1}) + (1 - z^{-1}) = 2$ allpass

ii. $H_1(z)\,H_1(z) = (1 + z^{-1})(1 + z^{-1}) = 1 + 2z^{-1} + z^{-2}$ lowpass



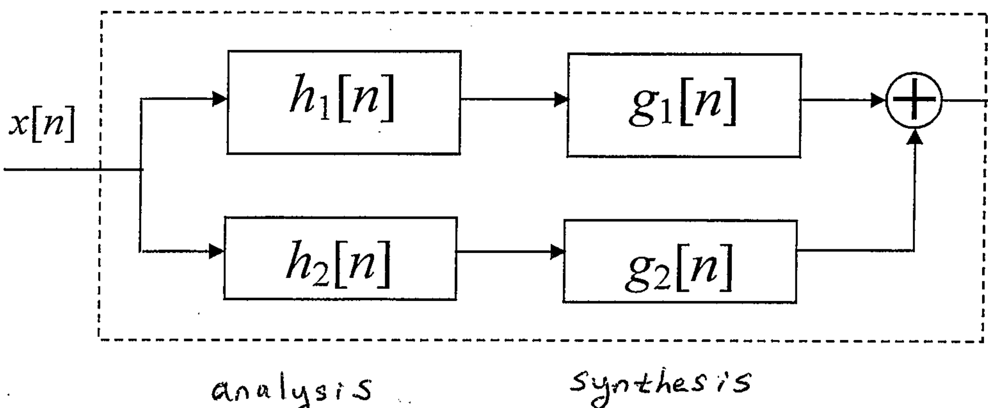
iii. $H_1(z)\,H_2(z) = (1 + z^{-1})(1 - z^{-1}) = 1 - z^{-2}$ bandpass



iv. $H_2(z)\,H_2(z) = (1 - z^{-1})(1 - z^{-1}) = 1 - 2z^{-1} + z^{-2}$ highpass

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*.



$\boxed{\text{Solution \#2}}$

$G_1(z) = G_2(z) = G(z)$ and $G(z)$ is FIR and all-pass. Works because $H_1(z) + H_2(z)$ is all-pass.

$$Y(z) = \left(H_1(z)\,G_1(z) + H_2(z)\,G_2(z)\right) X(z)$$

$$H(z) = H_1(z)\,G_1(z) + H_2(z)\,G_2(z)$$

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.

$\boxed{\text{Solution \#1}}$

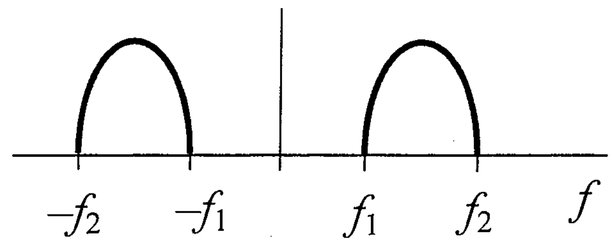
$G_1(z) = -1 - z^{-1} = -H_1(z)$

$G_2(z) = 1 - z^{-1} = H_2(z)$

$\Rightarrow H(z) = -4z^{-1}$

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

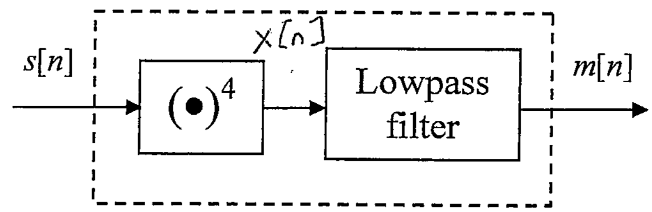
ω_{stop} : stopband frequency of discrete-time filter in rad/sample

$B = 2W$

$f_c > 9W$

(a) Downconversion method #1. *12 points*,

Uses a fourth-order static nonlinearity.



Give formulas for the following parameters:

$\omega_{pass} = 2\pi \frac{2B}{f_s}$

$\omega_{stop} = 2\pi \frac{2f_c - 2B}{f_s}$

$f_s > 2(4f_c + 2B)$

$x(t) = s^4(t)$

$X(f) = (S(f) * S(f)) * (S(f) * S(f))$

See below.

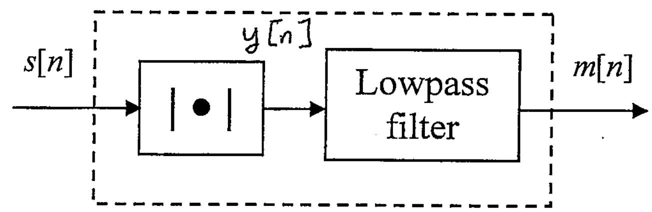
(b) Downconversion method #2. *12 points*.

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/r to carrier.

where a_0 and a_2 are real constants.



Give formulas for the following parameters:

$\omega_{pass} = 2\pi \frac{B}{f_s}$

$\omega_{stop} = 2\pi \frac{2f_c - B}{f_s}$

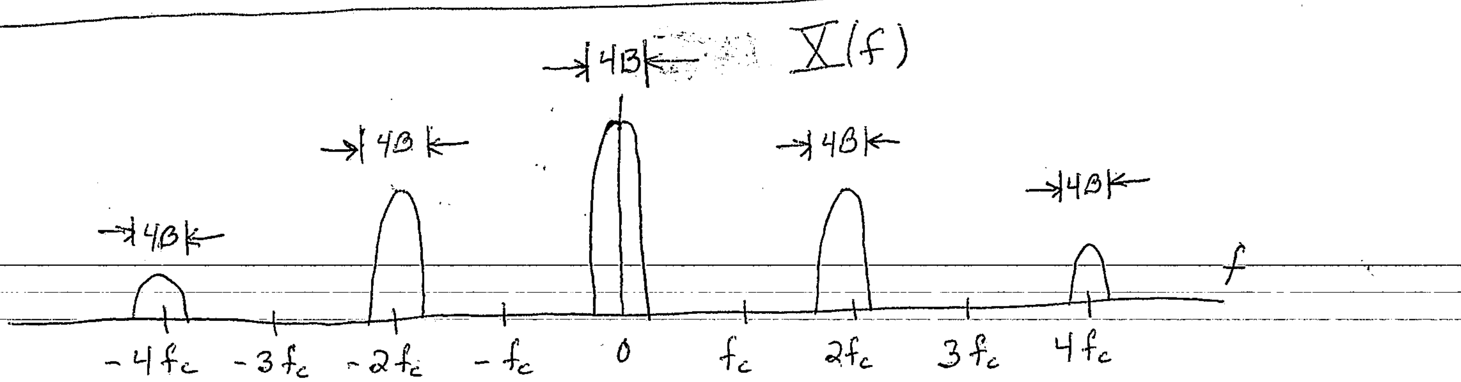
$f_s > 2(2f_c + B)$

$y(t) = |s(t)|$

$y(t) = |m(t) \cos(2\pi f_c t)|$

$y(t) = |m(t)| \cdot |\cos(2\pi f_c t)|$

$y(t) \approx |m(t)| \cdot (a_0 + a_2 \cos(4\pi f_c t))$



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.

Problem	*Point Value*	*Your score*	*Topic*
1	25		Discrete-Time Filter Analysis
2	27		Need for Speed
3	27		I'd Like to Buy a Vowel
4	21		Potpourri
Total	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:

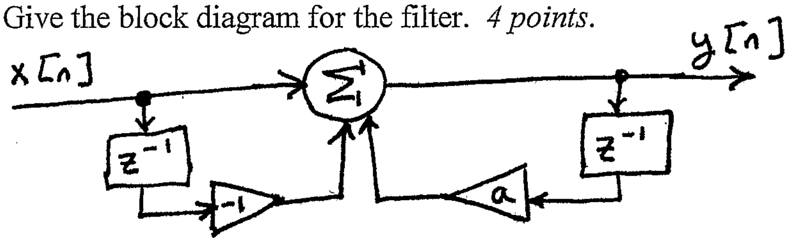
$$y[n] - a\, y[n-1] = x[n] - x[n-1]$$

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, time-invariance and causality.

(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:

$$Y(z) - a z^{-1} Y(z) = X(z) - z^{-1} X(z) \Rightarrow \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{1 - a z^{-1}} = H(z)$$

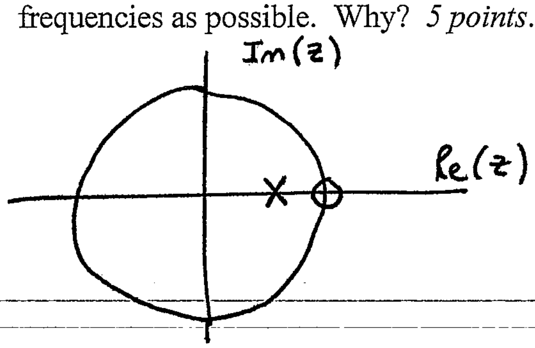
Region of convergence is $|z| > |a|$ for causality.

(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:

$$H_{freq}(\omega) = H(z)\big|_{z = e^{j\omega}} = \frac{1 - e^{-j\omega}}{1 - a e^{-j\omega}}$$

(f) Give a value for a so that the filter removes zero frequency and passes as many other frequencies as possible. Why? *5 points.*



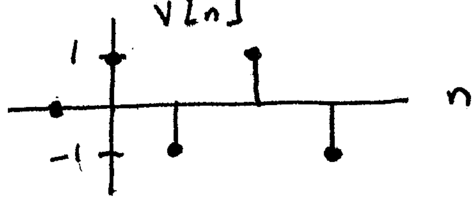
Regardless of the value of a where $0 < |a| < 0.9$, the filter has a zero at $z = 1$, i.e. $\omega = 0$. Placing the pole at $z = 0.899$ would give a notch filter at DC.

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



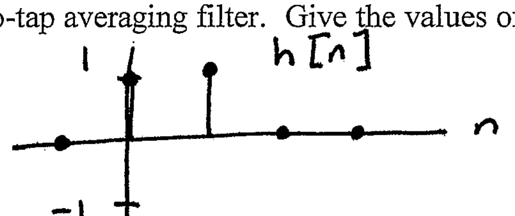
(b) Let $h[n]$ be the impulse response of a lowpass filter, and $g[n]$ is an impulse response formed by $v[n]\, h[n]$.

i. Let $h[n]$ be the impulse response of a two-tap averaging filter. Give the values of $g[n]$. *3 points.*

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

$g[n] = v[n]\, h[n]$

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

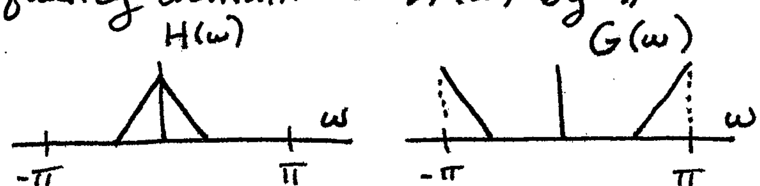


ii. If $g[n]$ from part i were an impulse response of a linear time-invariant filter, what would its frequency selectivity be? Lowpass, highpass, bandpass, bandstop, allpass, or notch? *3 points.*

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.*

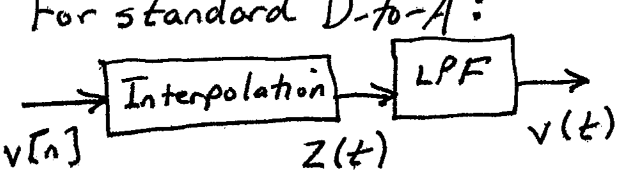
For ideal D-to-A conversion:

$\omega_0 = 2\pi \frac{f_0}{f_s} = \pi \Rightarrow f_0 = \frac{1}{2} f_s$

$v(t) = T_s \cos(\pi f_s t)\, u(t)$

Sampling $v(t)$ at $t = n T_s$ gives $v[n]$.

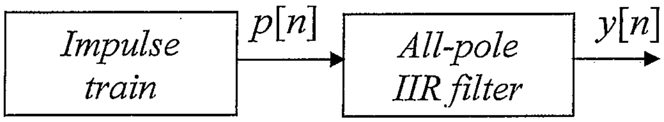
For standard D-to-A:



Depends on interpolation method and lowpass filter.

Problem 1.3. *I'd Like to Buy a Vowel.* 27 points.

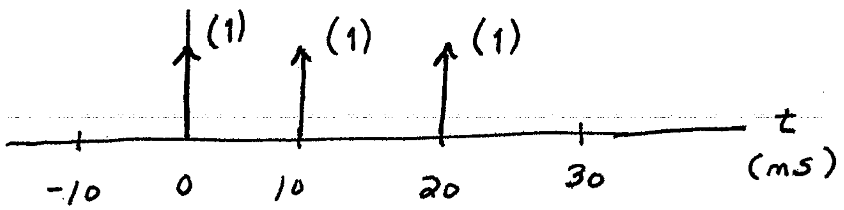
The following discrete-time block diagram synthesizes a short segment of speech $y[n]$ for a vowel sound:



Assume the following:

- segment of speech lasts from 0 ms to 25 ms, inclusive. $T_{seg} = 25\,ms$
- impulses in the impulse train are separated by the speaker's pitch period
- pitch period is 1/(100 Hz) or 10 ms $\quad T_p = 10\,ms$
- ~~samping~~ sampling rate f_s is 8000 Hz $\quad T_s = \frac{1}{8000\,Hz} = 0.125\,ms$

(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.*



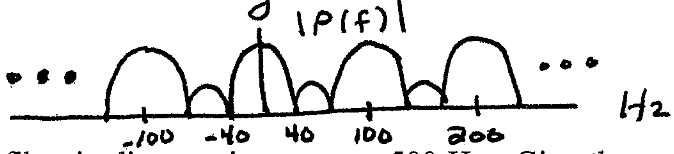
rectangular pulse of value 1 from $0 \le t < T_{seg}$

(b) Sketch the continuous-time Fourier transform for $p(t)$. *12 points.*

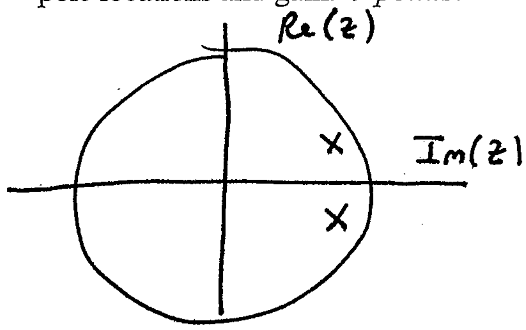
$$p(t) = \delta(t) + \delta(t - 10ms) + \delta(t - 20ms) = \delta_{T_p}(t)\, \mathrm{rect}\left(\frac{t - \frac{1}{2}T_{seg}}{T_{seg}}\right)$$

$$P(f) = \mathcal{F}\{\delta_{T_p}(t)\} * \mathcal{F}\left\{\mathrm{rect}\left(\frac{t - \frac{1}{2}T_{seg}}{T_{seg}}\right)\right\}$$

Impulse train with impulses separated by T_p



(a) Design the first biquad for the ~~all-pass~~ all-pole IIR filter in discrete-time to pass 500 Hz. Give the pole locations and gain. *9 points.*



$$\omega_{500} = 2\pi \frac{500\,Hz}{8000\,Hz} = \frac{\pi}{8}$$

Pole locations at

$$p_0 = 0.9\, e^{j\omega_{500}}$$

$$p_1 = 0.9\, e^{-j\omega_{500}}$$

$$H(z) = C \frac{1}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})};$$ Let $H(1) = 1$ 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

$$H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = 1 - e^{-j\omega}$$

$\boxed{\text{Notes}}$

$-1 = e^{-j\pi}$

$j = e^{j\frac{\pi}{2}}$

ii. Show that the phase response is linear. (*Note: The phase response may not necessarily go through the origin.*)

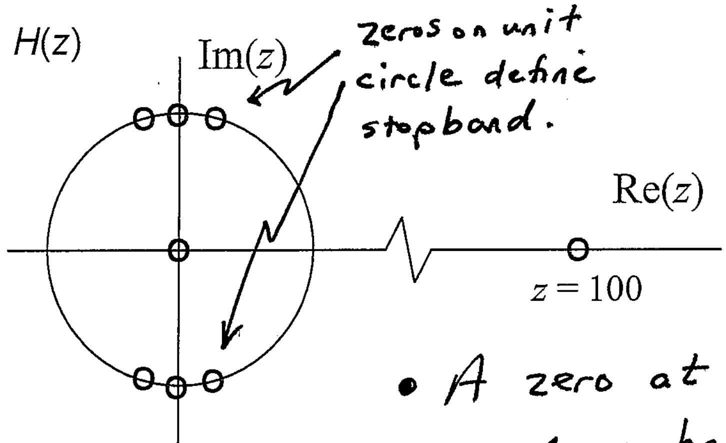
$$H_{freq}(\omega) = e^{-j\frac{\omega}{2}}\left(e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right) = e^{-j\frac{\omega}{2}}\left(2j\sin\frac{\omega}{2}\right)$$

$$\measuredangle H_{freq}(\omega) = \begin{cases} \frac{\pi}{2} - \frac{\omega}{2} & \text{for } \omega > 0 \\ -\frac{\pi}{2} - \frac{\omega}{2} & \text{for } \omega < 0 \end{cases}$$

iii. Compute the group delay

$$D(\omega) = -\frac{d}{d\omega}\measuredangle H_{freq}(\omega) = \frac{1}{2}$$

(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.*



zeros on unit circle define stopband.

- 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. <u>The minimum order is 6.</u>

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 9, 2012 Course: EE 445S Evans

Name: One, Thing
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.

Problem	*Point Value*	*Your score*	*Topic*
1	25		Discrete-Time Filter Analysis
2	24		Music Therapy
3	27		Downconversion
4	24		Discrete-Time FIR Filter Design
Total	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:

Note: A variation of this question appeared on the fall 2003 midterm #1.

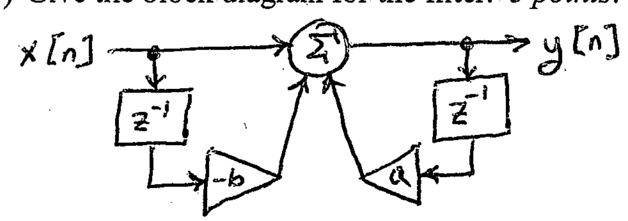
$$y[n] = a\, y[n-1] + x[n] - b\, x[n-1]$$

where $0 < |a| < 0.9$ and $|b| > 0.9$.

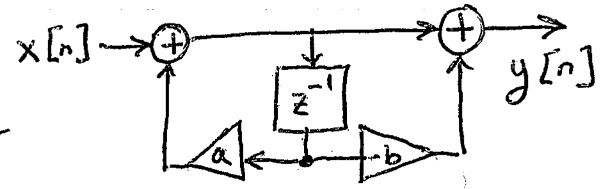
(a) Is this a finite impulse response filter or an infinite impulse response filter? Why? *2 points.*

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.*



or



(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 $|a| < 0.9$,

$$H_{freq}(\omega) = H(z)\Big|_{z = e^{j\omega}} = \frac{1 - b e^{-j\omega}}{1 - a e^{-j\omega}}$$

(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.

$a = 0.8$
$b = 1.25$

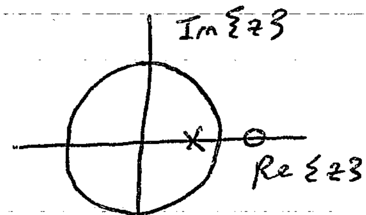
2. All-pass. *3 points.* See Appendix O in reader.

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 $\omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{5512.5\ Hz}{44100\ Hz} = \frac{\pi}{4}$.

Pole locations:

$p_0 = 0.9\, e^{j\omega_0}$ and $p_1 = 0.9\, e^{-j\omega_0}$

Zero locations:

$z_0 = e^{j\omega_0}$ and $z_1 = e^{-j\omega_0}$

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

Gain:

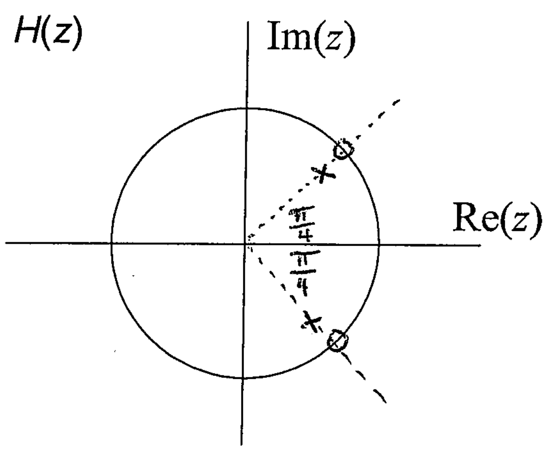
Simplest answer is $C = 1$.

Alternate answer is to set the DC gain to 1:

$H(1) = 1$

and solve for C.

(b) Draw the pole-zero diagram. *6 points.*

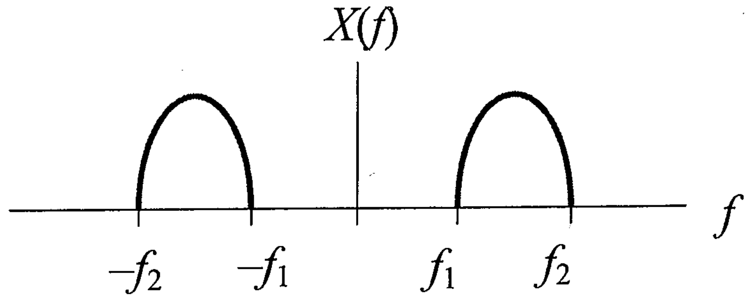


///

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.

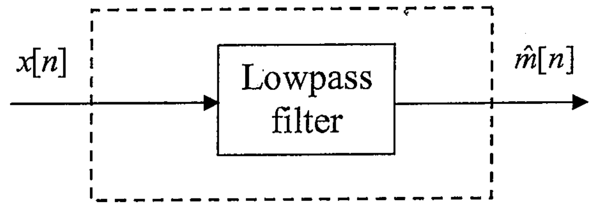


Let B be the bandpass bandwidth in Hz of $x(t)$ given by $B = f_2 - f_1$
f_c be the carrier frequency in Hz given by $f_c = ½\,(f_1 + f_2)$ where $f_c > 2\,B$.
f_s be the sampling rate in Hz for sampling $x(t)$ to produce $x[n]$
ω_{pass} be the passband frequency of a discrete-time filter in rad/sample
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.

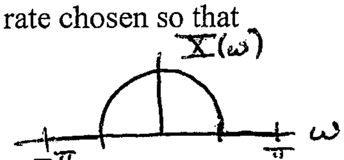
(a) Downconversion method #1. *12 points*,



With the sampling rate chosen so that

$f_s = 2\,B$

$f_c = 4\,f_s = 8B$

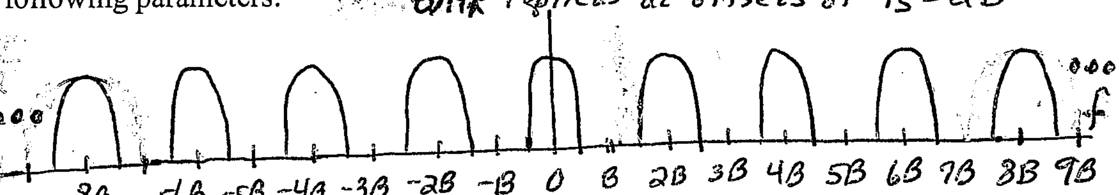


sketch the Fourier transform of $x[n]$ and give formulas for the following parameters:

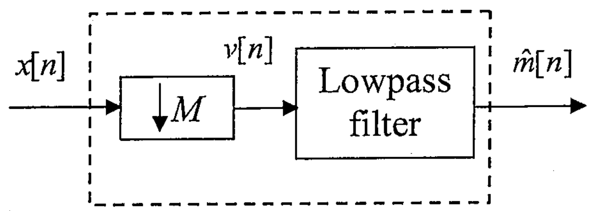
Sampling $x(t)$ by f_s gives spectrum with replicas at offsets of $f_s = 2B$

$\frac{\pi}{2} = \omega_{pass} = 2\pi \frac{B/2}{2B}$

$\pi = \omega_{stop} = 2\pi \frac{B}{2B}$



(b) Downconversion method #2. *15 points*.



Downsampling factor M is an integer.

With

$f_s > 2\,f_2$

Note: $f_c > 2B$

sketch the Fourier transform of $x[n]$ and give formulas for the following parameters:

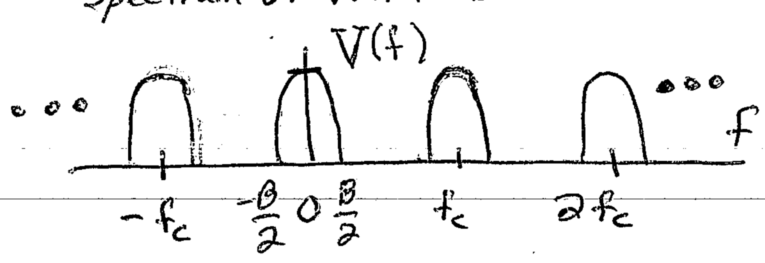
$f_s = M\,f_c$

$\omega_{pass} = 2\pi \frac{B/2}{f_s/M} = \pi \frac{B}{f_c}$

$\omega_{stop} = 2\pi \frac{B}{f_s/M} = 2\pi \frac{B}{f_c}$

Note: With $f_s = M\,f_c$, $f_s > 2\,f_2$ only for $M \geq 3$.

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:



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: $E_{total} = \sum_{n=0}^{\infty} |h[n]|^2$

$$E_{total} = \sum_{n=0}^{\infty} |0.9^n|^2 = \sum_{n=0}^{\infty} ((0.9)^n)^2 = \sum_{n=0}^{\infty} ((0.9)^2)^n$$

$$= \sum_{n=0}^{\infty} 0.81^n = \frac{1}{1-0.81} = \frac{1}{0.19} = 5.26316$$

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.

$$\sum_{n=0}^{L-1} 0.81^n \geq 0.9\, E_{total}$$

$\boxed{L = 11}$

$$\frac{1-0.81^L}{1-0.81} \geq \frac{0.9}{1-0.81} \Rightarrow 0.1 \geq 0.81^L \qquad \boxed{L = \frac{\ln 0.1}{\ln 0.81}}$$

$$\sum_{n=0}^{N} a^n = \frac{1-a^{N+1}}{1-a}$$

$$\sum_{n=0}^{L-1} 0.81^n = \frac{1-0.81^L}{1-0.81}$$

(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.

Problem	*Point Value*	*Your score*	*Topic*
1	25		Discrete-Time Filter Analysis
2	24		Discrete-Time Filter Implementation
3	27		System Identification
4	24		Upconversion
Total	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:

$$H(z) = \frac{b}{1 - a z^{-1}} = \frac{Y(z)}{X(z)}$$

for $|z| > |a|$. Here, a and b are real-valued, and b is not zero.

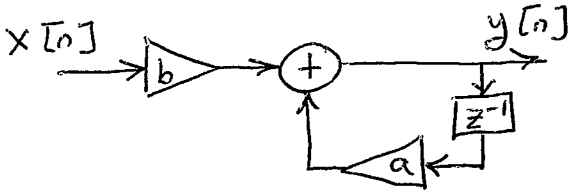
(a) From the transfer function, derive the difference equation relating input $x[n]$ and output $y[n]$. *6 points.*

$$\frac{Y(z)}{X(z)} = \frac{b}{1 - a z^{-1}} \Rightarrow \begin{aligned} (1 - a z^{-1})\, Y(z) &= b\, X(z) \\ Y(z) - a z^{-1} Y(z) &= b\, X(z) \end{aligned}$$

apply inverse z-transform: $y[n] - a\, y[n-1] = b\, x[n]$

$$\underline{y[n] = a\, y[n-1] + b\, x[n]}$$

(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 $n = 0$: $y[0] = a\, y[-1] + b\, x[0]$

Initial condition is $y[-1]$. It must be set to zero to satisfy LTI system properties.

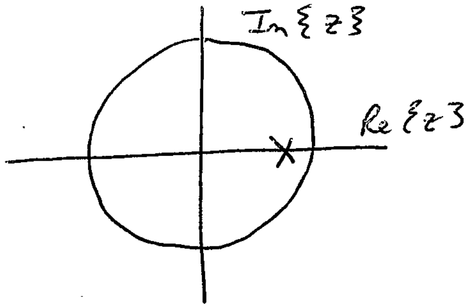
(e) Find the equation for the frequency response of the filter. Justify your approach. *6 points.*

Since the LTI system is stable, the region of convergence includes the unit circle:

$$H_{freq}(\omega) = H(z)\big|_{z = e^{j\omega}} = \frac{b}{1 - a e^{-j\omega}}$$

(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 ($\omega = 0$):

$$H_{freq}(0) = \frac{b}{1 - a} = 1$$

$$b = 1 - a \Rightarrow b = 0.1$$

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. Store complex numbers as

Amplitudes of the input and output signals are real-valued. 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.*

$a_1 = p_0 + p_1 \qquad a_2 = -p_0 p_1$

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 multiplication

iii. How many real-valued multiplication and addition operations does it take for the 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

$$H_1(z) = \frac{1-z_0 z^{-1}}{1-p_0 z^{-1}} \text{ and } H_2(z) = \frac{1-z_1 z^{-1}}{1-p_1 z^{-1}}$$

i. How many real-valued multiplications and real-valued additions are in one complex-valued multiplication? *3 points.*

$(a+jb)(c+jd) = \underbrace{(ac-bd)}_{\text{real}} + j\underbrace{(bc+ad)}_{\text{imaginary}}$ 4 real multiplies, 2 real adds

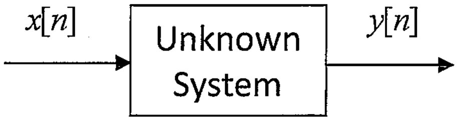
ii. How many real-valued additions are in one complex-valued addition? *3 points.*

$(a+jb)+(c+jd) = (a+c) + j(b+d)$ 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.*

$X(\omega) = 1 + e^{-j\omega}$, and at $\omega = \pi$, $X(\pi) = 0$ and $Y(\pi) = 0$.

We cannot observe the frequency response of $H(\omega)$ at $\omega = \pi$.

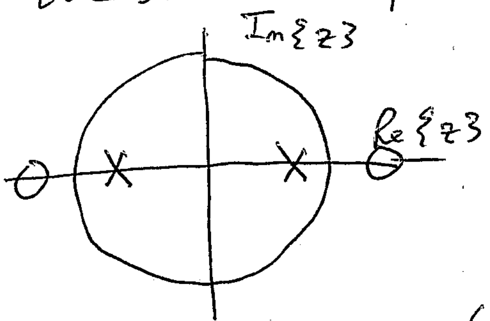
(c) Determine a real, non-zero value of a_1 in $x[n] = \delta[n] + a_1\,\delta[n-1]$ that will allow $H(\omega)$ to be measured at all frequencies. *6 points.*

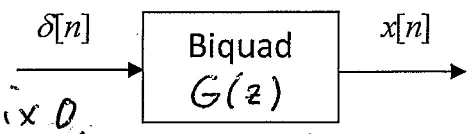
$X(\omega) = 1 + a_1 e^{-j\omega}$. When $a_1 = -1$, $X(0) = 0$, and we cannot observe the DC response of $H(\omega)$ at the output.

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 O,

$z_0 = \frac{1}{0.8} = 1.25$

$z_1 = \frac{1}{-0.8} = -1.25.$

Set DC gain to one.

$G(z)\big|_{z=1} = 1$

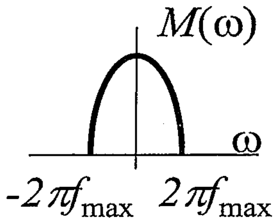
and solve for C.

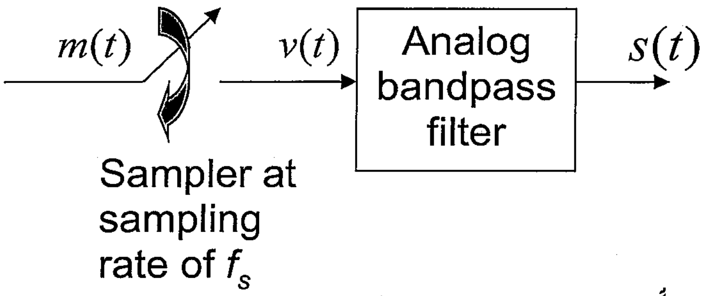
$$G(z) = C\,\frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})}{(1 - p_0 z^{-1})(1 - p_1 z^{-1})}.$$

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.



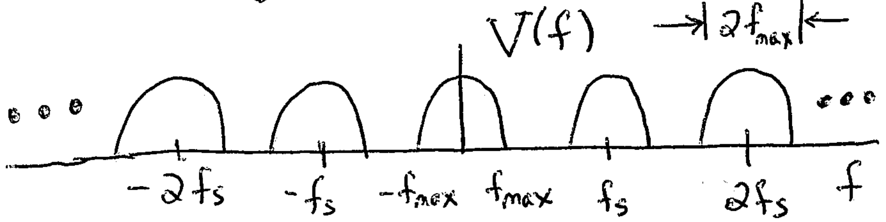


This system is entirely in continuous time.

Assume that $f_s > 2 f_{max}$.

(a) Draw the spectrum for $v(t)$. *6 points.*

Sampling replicates the input spectrum at offsets equal to multiples of f_s, because sampling in time can be modelled as multiplication by an impulse train.



(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.*

$f_{stop_1} = 0.9 f_{pass_1}$ or $f_{stop_1} = f_{pass_1} - (2 f_{max}) 0.1$

$f_{pass_1} = f_c - f_{max}$

$f_{pass_2} = f_c + f_{max}$

10% roll off

$f_{stop_2} = 1.1 f_{pass_2}$ or $f_{stop_2} = f_{pass_2} + (2 f_{max}) 0.1$

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.

Problem	*Point Value*	*Your score*	*Topic*
1	28		Filter Analysis
2	24		Filter Implementation
3	24		Filter Design
4	24		Potpourri
Total	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:

$$H(z) = 1 - z^{-3}$$

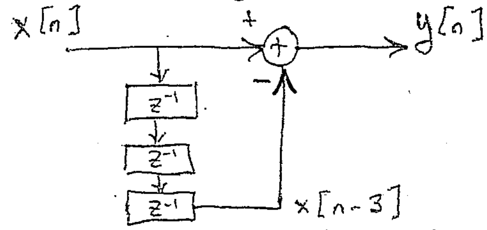
for $|z| \neq 0$.

(a) From the transfer function, derive the difference equation relating input $x[n]$ and output $y[n]$. *6 points.*

$$H(z) = \frac{Y(z)}{X(z)} = 1 - z^{-3}$$

$$Y(z) = (1 - z^{-3})X(z) = X(z) - z^{-3}X(z) \overset{Z^{-1}}{\Rightarrow} y[n] = x[n] - x[n-3]$$

(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: $H_{freq}(\omega) = H(z)\Big|_{z=e^{j\omega}} = 1 - e^{-3j\omega}$

Alternate Justification: LTI system is bounded-input bounded-output stable.

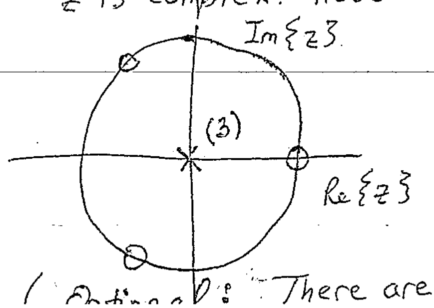
(e) What is the group delay through the filter? *3 points.* $N=4$ coefficients.

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 $\omega = \frac{2}{3}\pi$.

(Optional: There are three poles at origin.)

$H_{freq}(\omega) = e^{-j\frac{3}{2}\omega}\left(e^{j\frac{3}{2}\omega} - e^{-j\frac{3}{2}\omega}\right)$

$= \underbrace{2\sin\left(\tfrac{3}{2}\omega\right)}_{\text{amplitude}} \underbrace{j \cdot e^{-j\frac{3}{2}\omega}}_{\text{phase}}$

Note: $j = e^{j\frac{\pi}{2}}$

$gd(\omega) = -\frac{d}{d\omega} \measuredangle H_{freq}(\omega)$

$= \frac{3}{2} = \frac{N-1}{2}$

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.

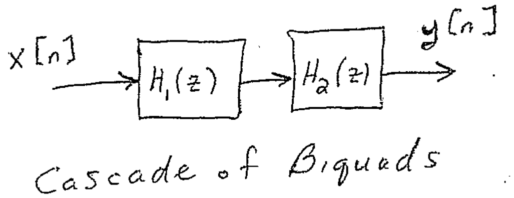
Cascade of biquads. We factor H(z) into a product of two second-order sections (biquads)

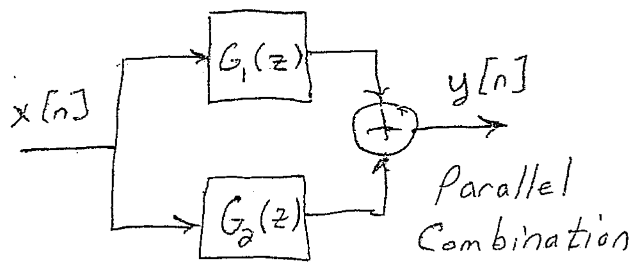
$$H(z) = H_1(z)\, H_2(z)$$

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.*





(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.*

	Cascade	Parallel
words	10 coefficients	10 coefficients
	12 current & previous inputs/outputs	12 current & previous inputs/outputs
		1 y[n]
	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.*

	Cascade	Parallel
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.

To communicate data from one core to the other through shared memory takes 10 cycles (1 store + 1 load inst.)

i. Compare the memory usage for the two structures. *3 points.*

Same as (b)i. because of shared on-chip memory

ii. Compare the execution cycles for the two structures. *6 points.*

Cascade	Parallel
Implement cascade of biquads on a single core due to the high 10-cycle inter-core communication cost. 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.

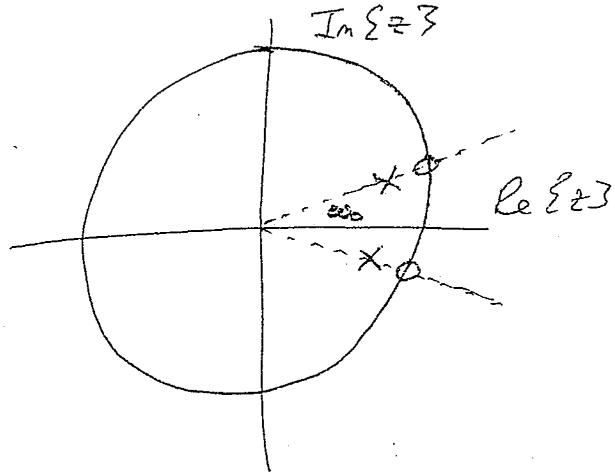
(a) Design a discrete-time second-order infinite impulse response (IIR) filter for a G.hnem transceiver to remove the 60 kHz WWVB interferer. Give poles, zeros and gain. *12 points.*

Notch filter at $\omega_0 = 2\pi \frac{60 \text{ kHz}}{800 \text{ kHz}} = 2\pi \frac{3}{40}$ 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:

$H(z)\big|_{z=1} = 1$ Solve for C_0.

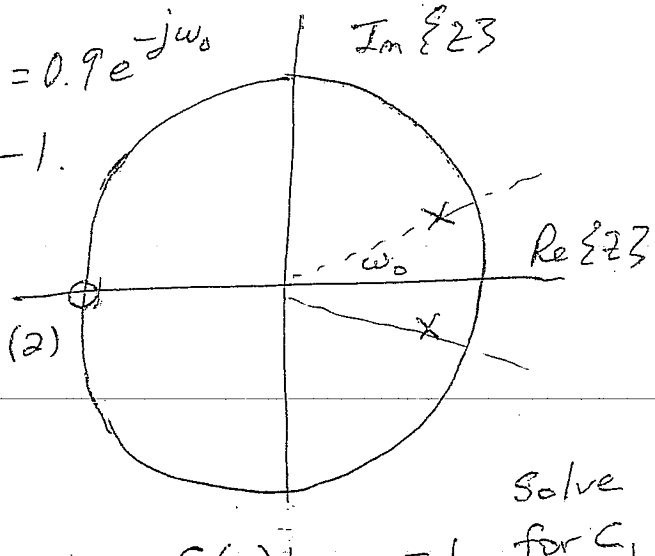
(b) Design a discrete-time second-order infinite impulse response (IIR) filter for a G.hnem transceiver to extract the 60 kHz WWVB signal for use in generating timestamps for power load profiles at the consumer's site. Give poles, zeros and gain. *12 points.*

$\omega_0 = 2\pi \frac{60 \text{ kHz}}{800 \text{ kHz}} = 2\pi \frac{3}{40}$ 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:

$G(z)\big|_{z=1} = 1$ Solve for C_1.

(Alternately, 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 x 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.

Alternate answer: Use an IIR filter design method and keep the first 10,000 samples of the impulse response.

(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 $\boxed{\text{unsigned 16-bit integers.}}$ 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.)

(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 downcounting

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

The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: October 18, 2013 Course: EE 445S Evans

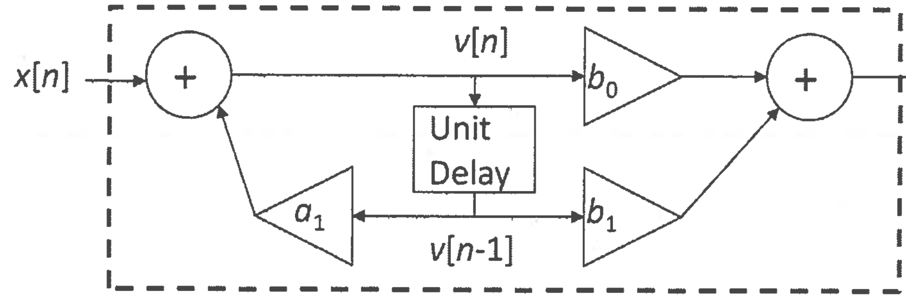
Name: Bollywood, Sally
Last, First

- 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.

Problem	*Point Value*	*Your score*	*Topic*
1	~~27~~ 28		Discrete-Time Filter Analysis
2	24		Discrete-Time Filter Design
3	24		System Identification
4	24		Modulation and Demodulation
Total	100		

Problem 1.1 *Discrete-Time Filter Analysis.* ~~27~~ 28 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.*

$$H(z) = \frac{Y(z)}{X(z)} = \frac{V(z)}{X(z)} \cdot \frac{Y(z)}{V(z)} = \frac{1}{1 - a_1 z^{-1}} \cdot (b_0 + b_1 z^{-1})$$

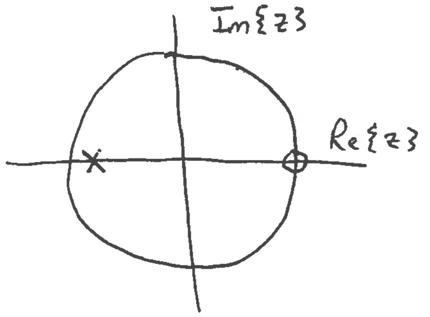
$$= \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \text{ for } |z| > |a_1|$$

(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.

$$H_{freq}(\omega) = H(z)\Big|_{z = e^{j\omega}} = \frac{b_0 + b_1 e^{-j\omega}}{1 - a_1 e^{-j\omega}}$$

(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 filte

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. Zeros are conjugate symmetric.

You will be asked to design and implement a notch filter:

> f_0 is the frequency in Hz to be eliminated, and
>
> f_s is the sampling rate in Hz where $f_s > 2 f_0$

Assume that the gain of the biquad is 1. $\Rightarrow \quad C = 1$

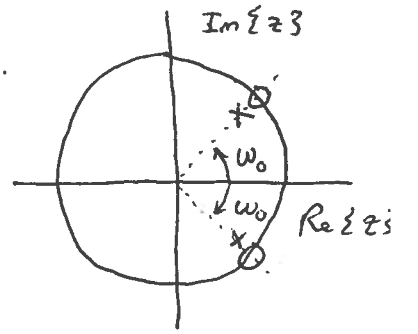
(a) Give a formula for the discrete-time frequency ω_0 in rad/sample to be eliminated. *3 points.*

$$\omega_0 = 2\pi \frac{f_0}{f_s}$$

(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

$b_0 = 1$

$b_1 = -(z_0 + z_1) = -(e^{j\omega_0} + e^{-j\omega_0}) = -2\cos\omega_0$

$b_2 = z_0 z_1 = e^{j\omega_0} e^{-j\omega_0} = 1$

Feed back 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:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{\frac{1}{1-z^{-1}}} = (1+z^{-1})(1-z^{-1}) = 1 - z^{-2} \Rightarrow h[n] = \delta[n] - \delta[n-2]$$

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.*

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

n

0 $\quad 1 = y[0] = h[0]\,x[0] \Rightarrow 1 = h[0] \Rightarrow h[0] = 1$

1 $\quad 1 = y[1] = h[0]\,x[1] + h[1]\,x[0]$

$\quad 1 = h[0] + h[1] \Rightarrow h[1] = 0$

2 $\quad y[2] = h[0]\,x[2] + h[1]\,x[1] + h[2]\,x[0]$

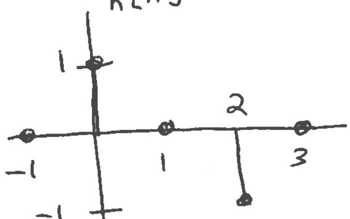
$\quad 0 = h[0] + h[1] + h[2] \Rightarrow h[2] = -h[0] \Rightarrow h[2] = -1$

3 $\quad y[3] = h[0]\,x[3] + h[1]\,x[2] + h[2]\,x[1] + h[3]\,x[0]$

$\quad 0 = h[0] + h[1] + h[2] + h[3] \Rightarrow h[3] = 0$

Check: $h[n] * u[n] \stackrel{?}{=} \delta[n] + \delta[n-1]$ $\boxed{\text{YES}}$

$h[n]$



(b) Compute the group delay through the filter as a function of frequency. *6 points.*

$$H_{freq}(\omega) = 1 - e^{-j2\omega}$$

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

$$= \underbrace{2\sin(\omega)}_{\text{amplitude term}}\; \underbrace{j\, e^{-j\omega}}_{\text{phase term}}$$

Note: $j = e^{+j\frac{\pi}{2}}$

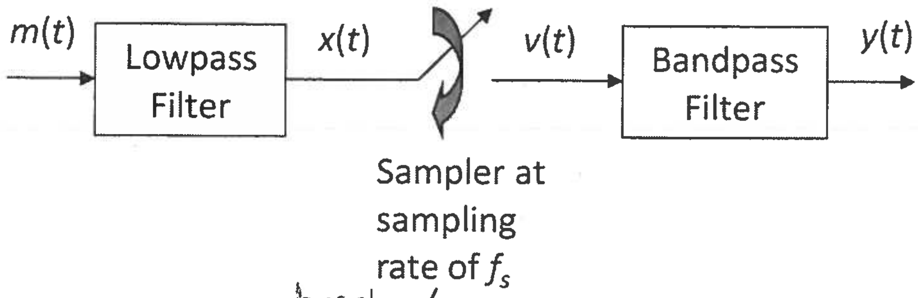
Group delay is 1 sample.

$$\text{Delay}(\omega) = -\frac{d}{d\omega} \measuredangle H_{freq}(\omega) = 1$$

Except for two points of discontinuity, $\measuredangle H_{freq}(\omega) = -\omega + \frac{\pi}{2}$

Problem 1.4. Modulation and Demodulation. *24 points.*

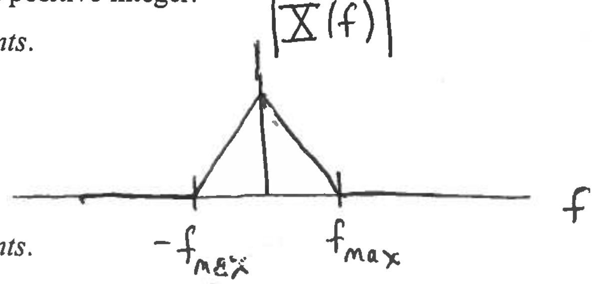
A mixer can be used to realize sinusoidal amplitude modulation $y(t) = x(t) \cos(2 \pi f_c t)$ for baseband signal $x(t)$:



Assume that $x(t)$ is a ideal ~~bandpass~~ baseband signal whose magnitude spectrum is zero for ~~$f > f_{max}$~~ $|f| > f_{max}$.

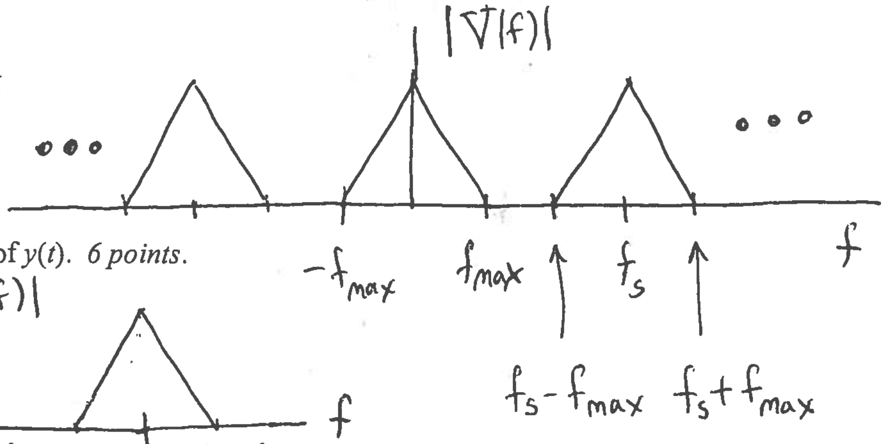
Assume that $f_s > 2 f_{max}$ and $f_c = m f_s$ where m is a positive integer.

(a) Draw the magnitude spectrum of $x(t)$. *6 points.*

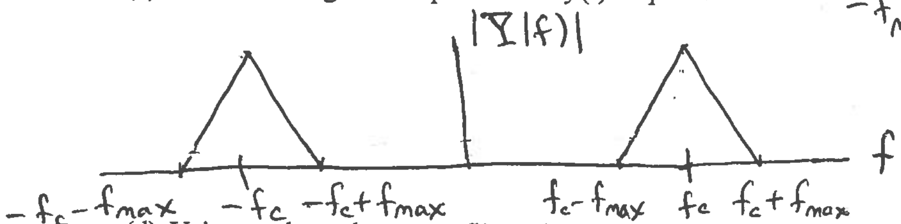


(b) Draw the magnitude spectrum of $v(t)$. *6 points.*

Spectrum of $X(f)$ is replicated at offsets in frequency equal to multiples of f_s

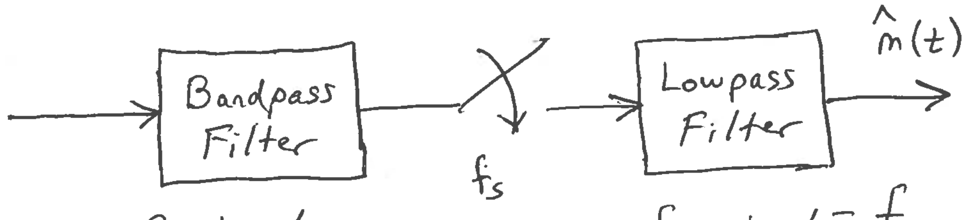


(c) Draw the magnitude spectrum of $y(t)$. *6 points.*



(d) Using only a lowpass filter, bandpass filter, and a sampler, give a block diagram for demodulation. *6 points.*

$f_c = m f_s$



Passband

$f_c - f_{max} < f < f_c + f_{max}$

$f_{passband} = f_{max}$

$f_{stopband} = f_s - f_{max}$

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.

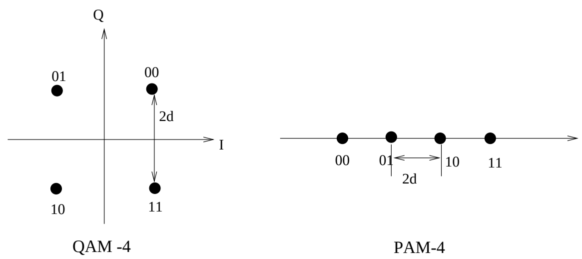


Figure 1: PAM-4 and QAM-4 (QPSK) constellation

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 $\{h_0, \ldots, h_{23}\}$ 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 $h_0, \ldots, h_{23}$. 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.

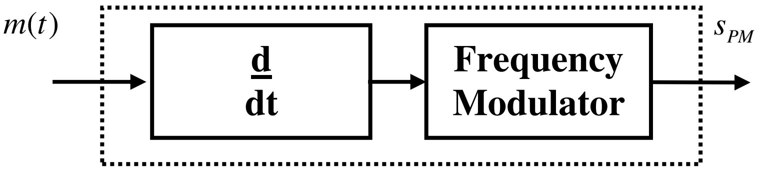
Phase modulation at a carrier frequency f_c of a causal message signal $m(t)$ is defined as

$$s_{PM}(t) = A_c \cos\left(2\pi f_c t + 2\pi k_p m(t)\right)$$

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 2.2 *Equalizer Design*. 15 points.

You are given a discretized communication channel defined by the following sampled impulse response with a representing a real number:

$$h[n] = \delta[n] - 2\,a\,\delta[n\text{-}1] + a^2\,\delta[n\text{-}2]$$

(a) For this channel, what would you propose to do at the transmitter to prevent intersymbol interference? 5 points.

(b) Find the transfer function of the discretized channel. 5 points.

(c) For this channel, design a stable linear time-invariant equalizer for the receiver so that the impulse response of the cascade of the discretized channel and equalizer yields a delayed impulse. Please state any assumptions on the value of a. 5 points.

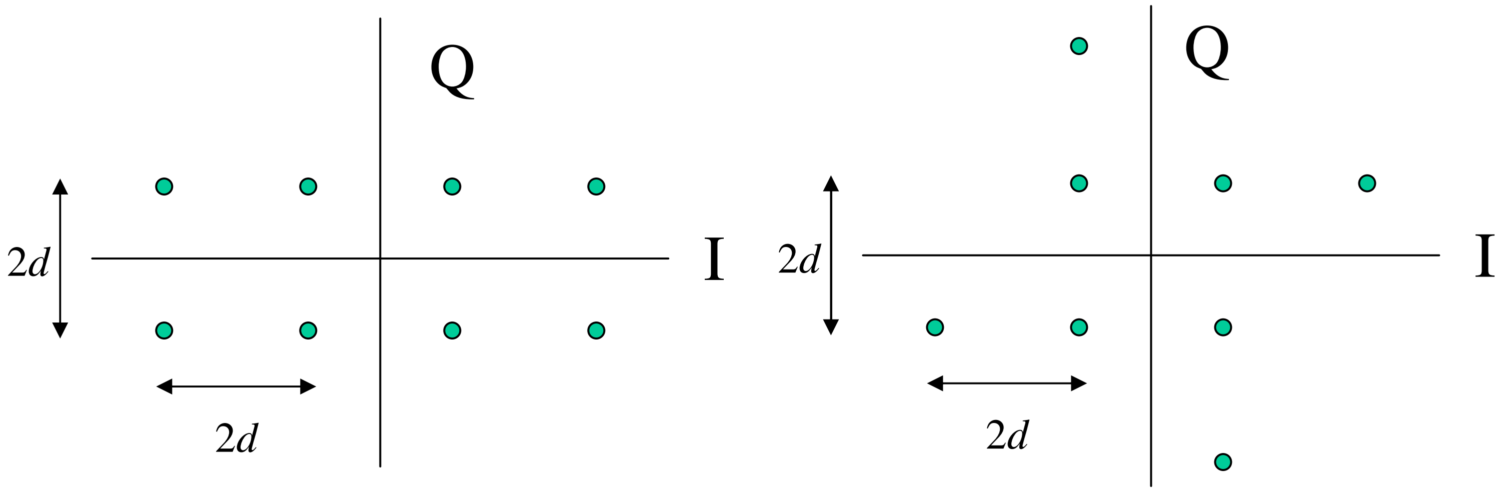
Problem 2.3 *8-QAM*. 30 points.

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.

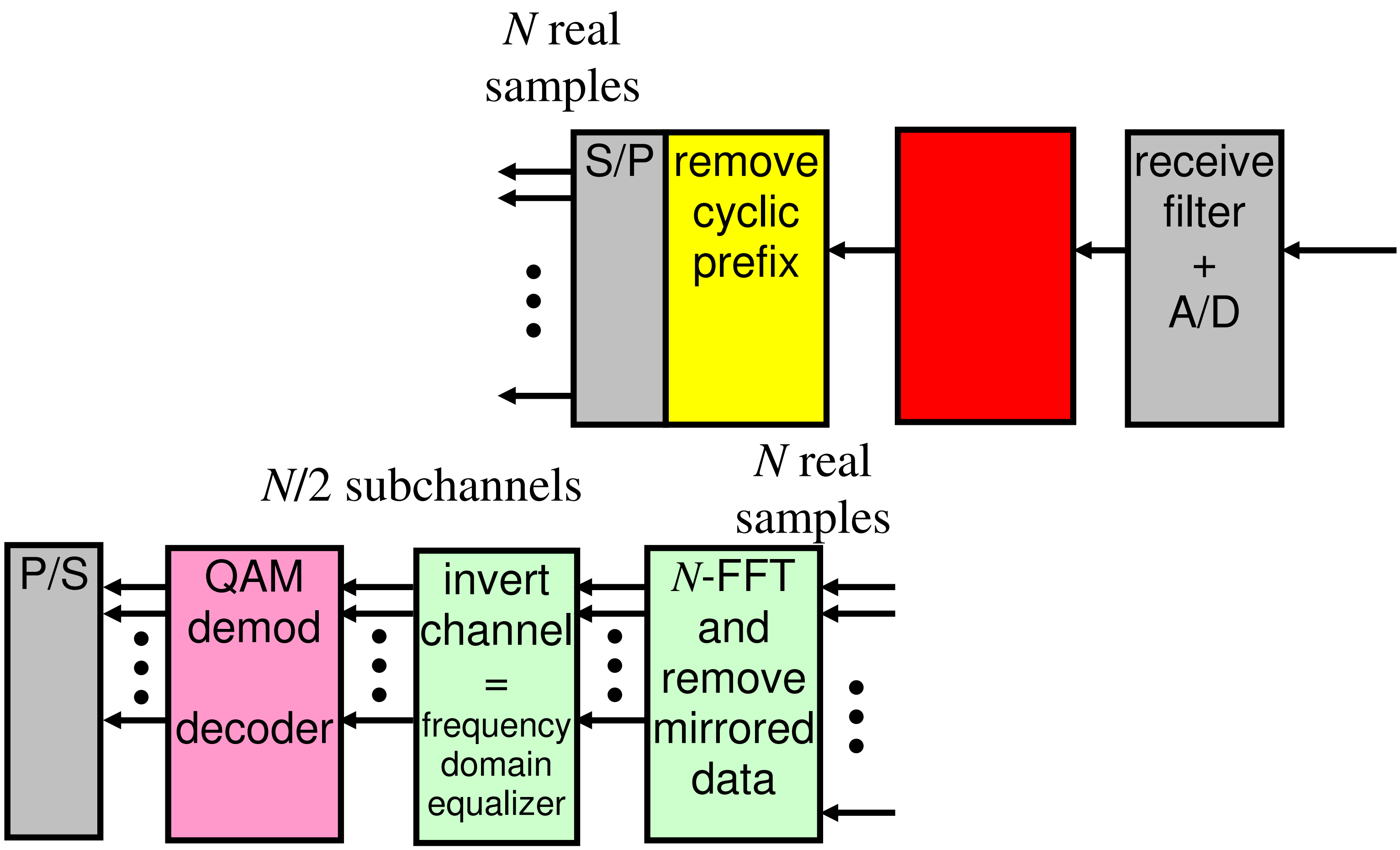
(c) Draw the optimal bit assignments for each symbol you would use for each of the constellations above on the constellations directly. 5 points.

(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

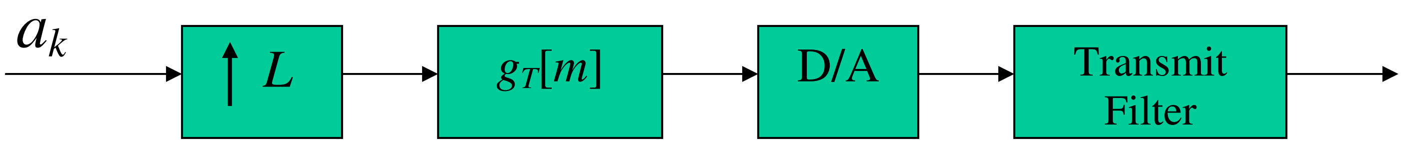
- The exam is scheduled to last 90 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected to a network.
- Please turn off all cell phones, 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.

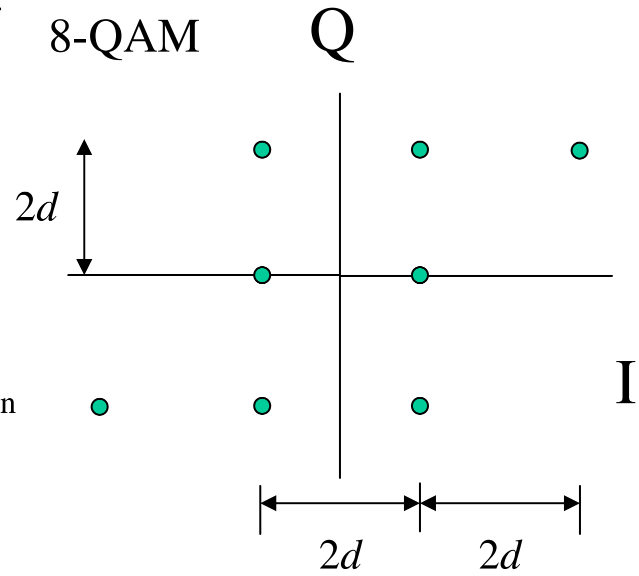
(d) Does the channel equalizer you designed in part (c) have a lowpass, highpass, bandpass, bandstop, allpass, or notch response? 2 points.

(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 69^{th} 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

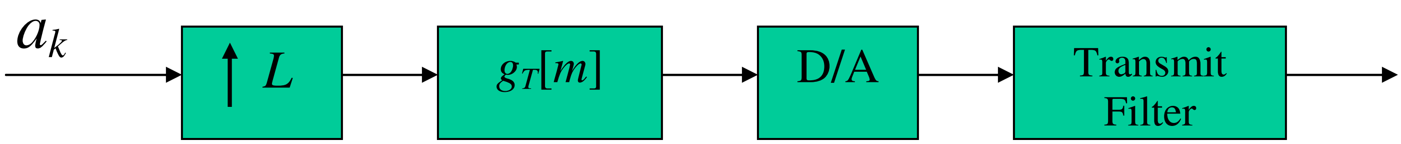
- The exam is scheduled to last 60 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected 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.

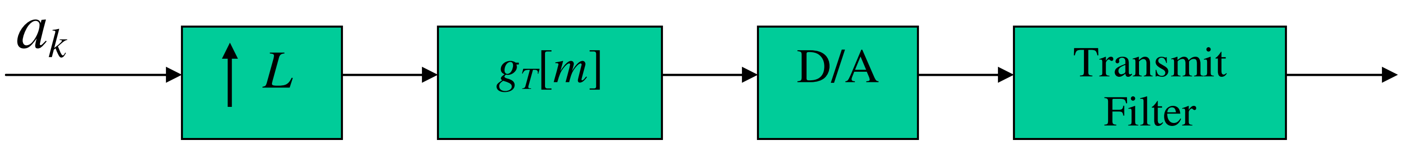
(d) Give formulas using the appropriate system parameters for the implementation complexity measures in the table below that would be required to compute the two leftmost blocks in the above block diagram (i.e., the blocks before the D/A converter). 18 points.

	Multiplication-accumulation operations per second	Memory Usage in Words	Memory Reads and Writes in words/second
As shown above			
Using a filter bank			

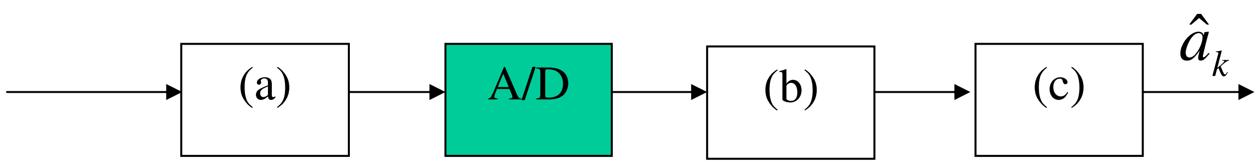
Problem 2.2 *Digital PAM Reception*. 24 points

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.

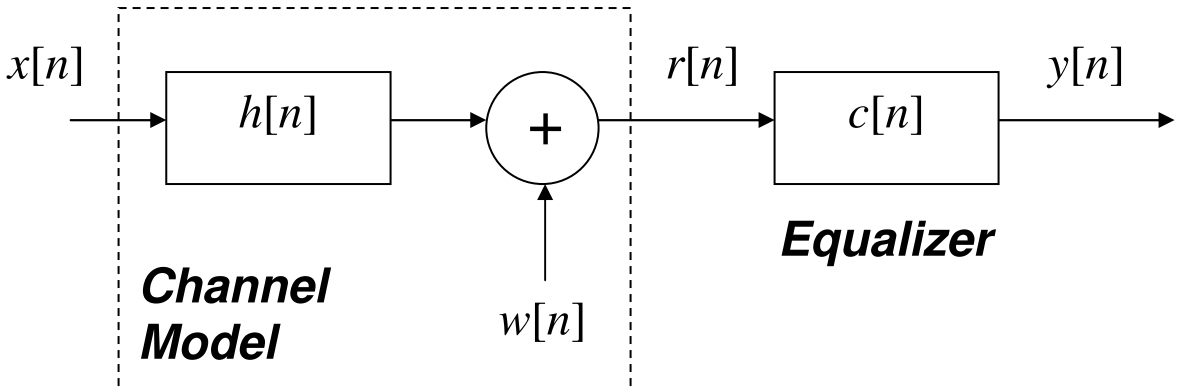
(a)

(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 includes 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

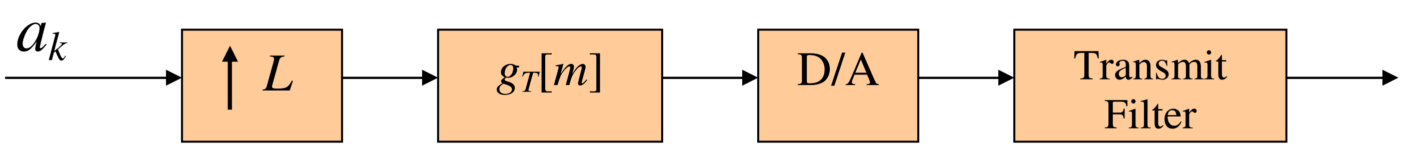
- The exam is scheduled to last 60 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any stand-alone computer system, i.e. one that is not connected 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.

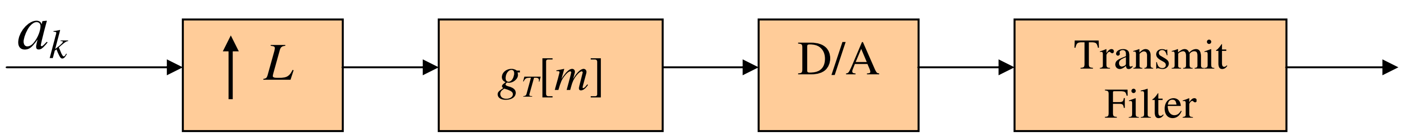
	Multiplication-accumulation operations per second	Memory usage in words	Memory reads and writes in words/second
As shown above			
Using a filter bank			

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.

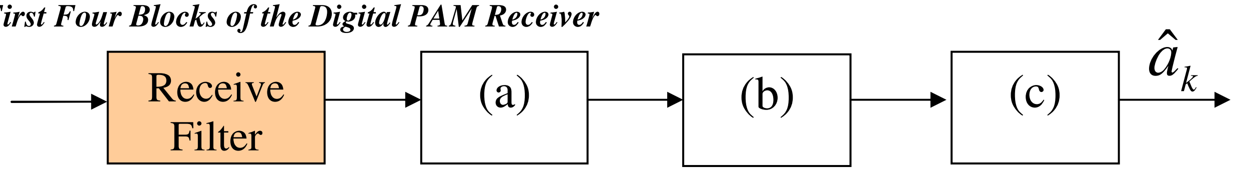
Last Four Blocks of the Digital PAM Transmitter



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 and variance σ^2 in both the in-phase (I) and quadrature (Q) components

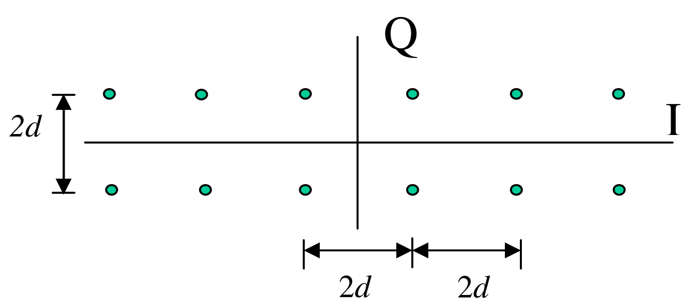
(iii) Perfect carrier frequency/phase recovery

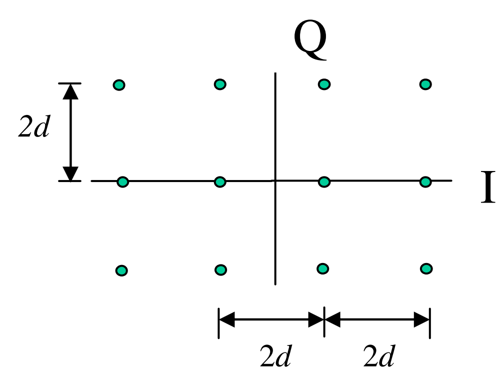
(iv) Perfect symbol timing recovery

(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.

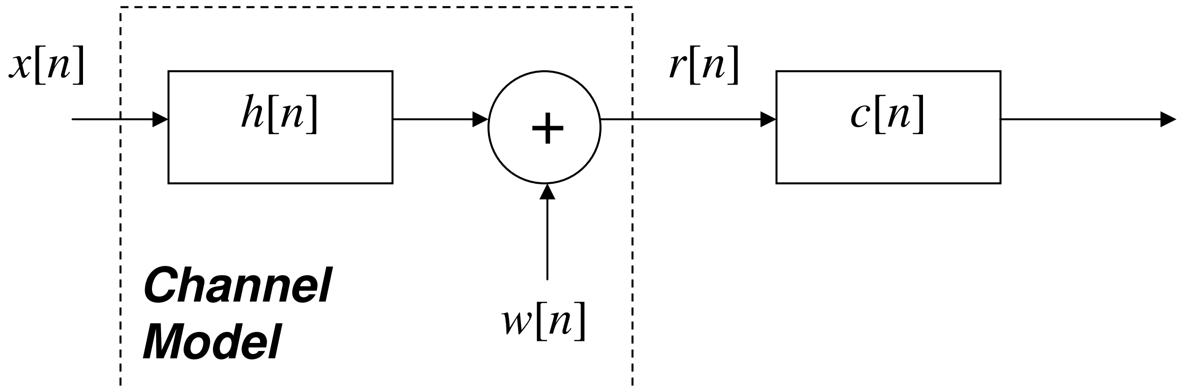
(b) Draw your **decision regions** on the 12-QAM constellations shown above. 6 points.

(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

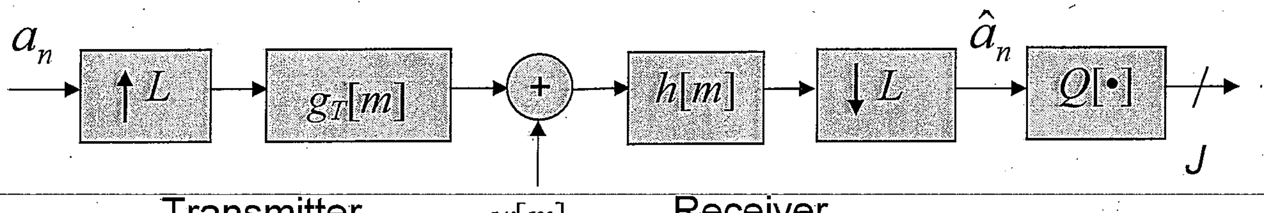
Course: EE 445S

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 $w[m]$.

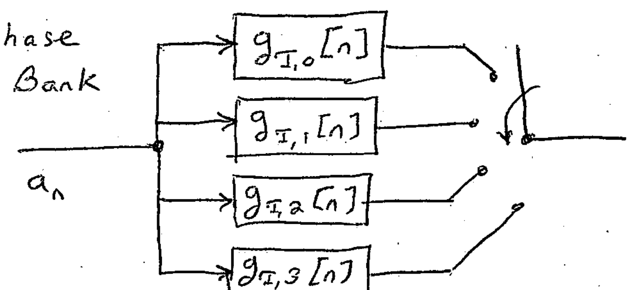
Assume that the receiver is synchronized to the transmitter.

(a) What is the bit rate in bits per second? *4 points.*

$J f_{sym}$

(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.*

Polyphase Filter Bank



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.

$h_{opt}[m] = k\, g_T^*[L-m]$

Optimizes peak pulse SNR at the output of the downsampler by L.

(d) Describe a fast algorithm for the quantizer $Q[\bullet]$ that does not involve any multiplications or additions. What is the computational complexity? *6 points.*

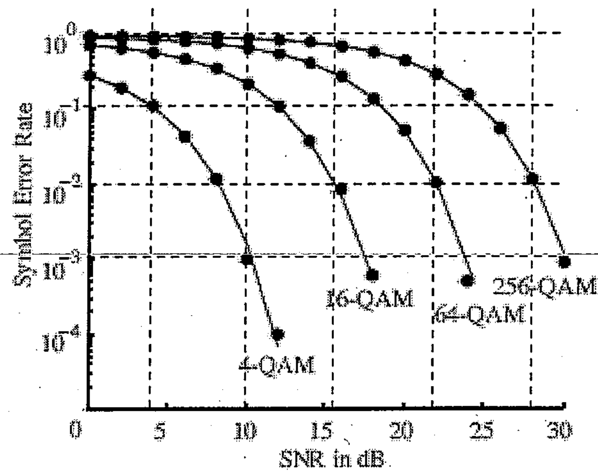
2-PAM
$+d$
$-d$
Compare $\hat{a}_n > 0$

4-PAM
1. Compare $\hat{a}_n > 0$
2. Then compare 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

1. Between 16-QAM and 4-QAM (addition of two bits/symbol). *3 points.*

   16 dB − 8 dB = 8 dB

2. Between 64-QAM to 16-QAM (addition of two bits/symbol). *3 points.*

   22 dB − 16 dB = 6 dB

(d) A formula to convert the number of bits J in a symbol to SNR in dB is $C_0 + C_1 J$.

1. 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.

   $C_0 = 4$ dB $\qquad C_1 = 3$ dB/bit

2. What is C_1 for PAM? Why the difference? *6 points*

   $C_1 = 6$ dB/bit from Quantization Lecture.

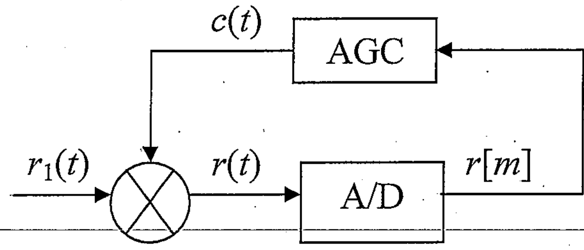
   Since PAM isn't as efficient with the spectrum as QAM, more power is needed for PAM to send the same number of bits with same probability of symbol error. QAM can transit two PAM signals in same transmission bandwidth.

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

$$c(t) = \left(1 + 2f_0 - f_{127} - f_{-128}\right) c(t-\tau_1) \quad \text{where } \tau_1 = T_s$$

Solution #2

$$c(t) = \frac{2f_0}{f_{127} + f_{-128}}\, c(t-\tau_2) \quad \text{where } \tau_2 = T_{sym}$$

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 basestations. 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

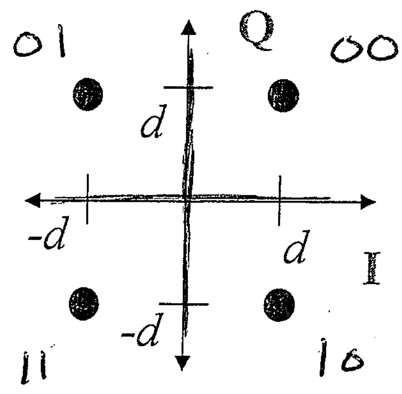
Name: Set, Solution
Last, First

- 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. $T_{sym} = 1$
Assume that the energy in the pulse shape is 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.*

Total power: $2d^2 + 2d^2 + 2d^2 + 2d^2 = 8d^2$

Average power: $\frac{8d^2}{4} = 2d^2$

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): $\hat{a}_n + j\hat{b}_n$

If ($\hat{a}_n > 0$) $s_1 = 0$ else $s_1 = 1$

If ($\hat{b}_n > 0$) $s_0 = 1$ else $s_0 = 0$

Two comparisons using divide-and-conquer strategy.

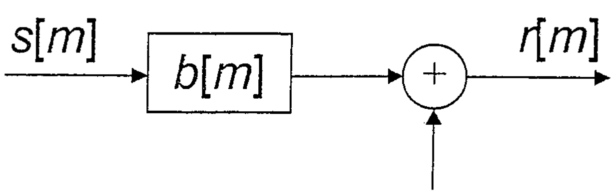
(e) Based on the decision regions in (c), give a formula for the probability of symbol error. *6 points.*

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).

$$P_3(c) = \left(1 - Q\left(\frac{d}{\sigma}\right)\right)^2$$

$$P(e) = 1 - P_3(c) = 1 - \left(1 - Q\left(\frac{d}{\sigma}\right)\right)^2 = 2Q\left(\frac{d}{\sigma}\right) - Q^2\left(\frac{d}{\sigma}\right)$$

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 $w[m]$.

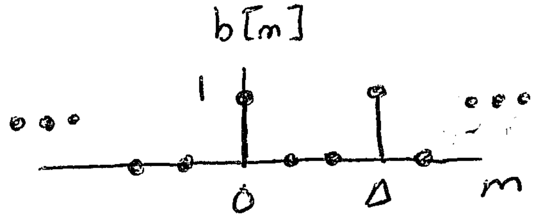
~~$b[m] = 1 + \delta[m-\Delta]$~~ where $\Delta > 1$ $\quad b[m] = \delta[m] + \delta[m - \Delta]$

$w[m]$ 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. The PN sequence would have values of +1 or −1. Longer sequences will better ~~results in~~ (c). PN length can be less than, equal to, or greater than Δ.

(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 $m = \Delta$.

(d) Give a formula for the impulse response or transfer function of a channel equalizer in the receiver to compensate for the frequency selectivity of the channel. You may ignore the noise. *9 points.*

$b[m] = \delta[m] + \delta[m-\Delta]$

$B(z) = 1 + z^{-\Delta}$ ← Δ zeros on the unit circle

Equalizer filter $G(z)$

We can't use $G(z) = \dfrac{1}{B(z)}$ because

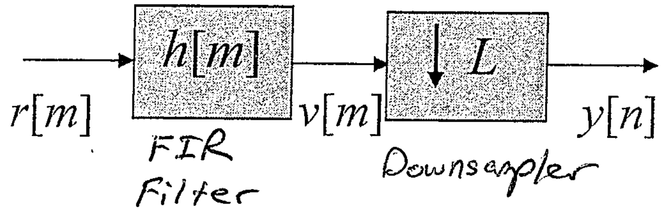
$G(z)$ would have Δ poles on unit circle.

Let $G(z) = \dfrac{1}{1 + 0.95 z^{-\Delta}}$ ← Δ poles with radius $(0.95)^{\Delta}$. $\boxed{\text{IIR Comb Filter}}$

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 $y[n] = v[L\,n]$

$h_{causal}[m] = k\, g^*[L N_g - m]$

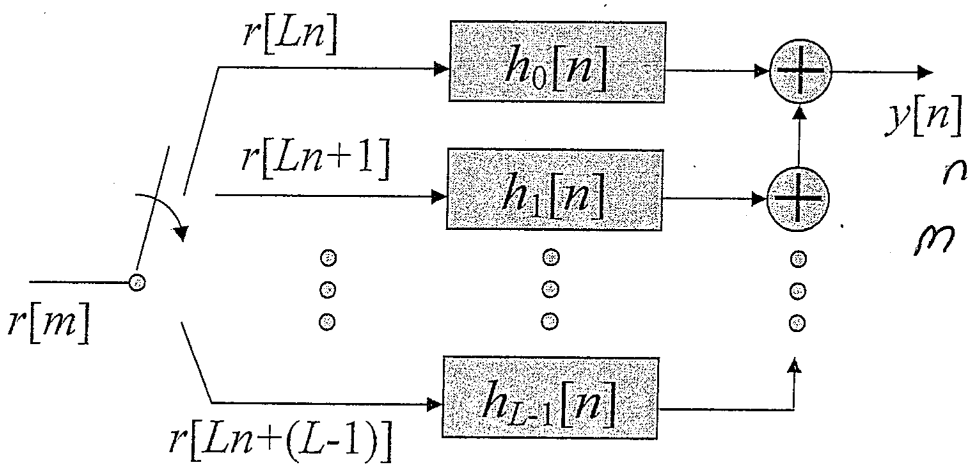
(a) Give a formula for the causal impulse response $h[m]$ that maximizes a measure of signal-to-noise ratio at $y[n]$. *6 points.*

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 $L N_g$ coefficients, $L N_g$ multiplication-accumulation (MAC) operations are needed for each output sample. MACs/s $(L N_g)(\underbrace{L f_{sym}}_{\text{sampling rate}})$

The above cascade can be efficiently implemented as a polyphase filter bank as follows:



n is symbol index.

m is sample index.

(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.*

$\boxed{h_0[n] = h[Ln] \text{ for } n = 0, 1, \ldots, N_g - 1}$

From midterm #2 review slide 10:

$y[1] = v[L] = h[0]r[L] + h[1]r[L-1] + \cdots + h[L-1]r[1] + h[L]r[0]$

(d) How many multiplication-accumulation operations per second are needed to implement the above polyphase filter bank? *9 points.*

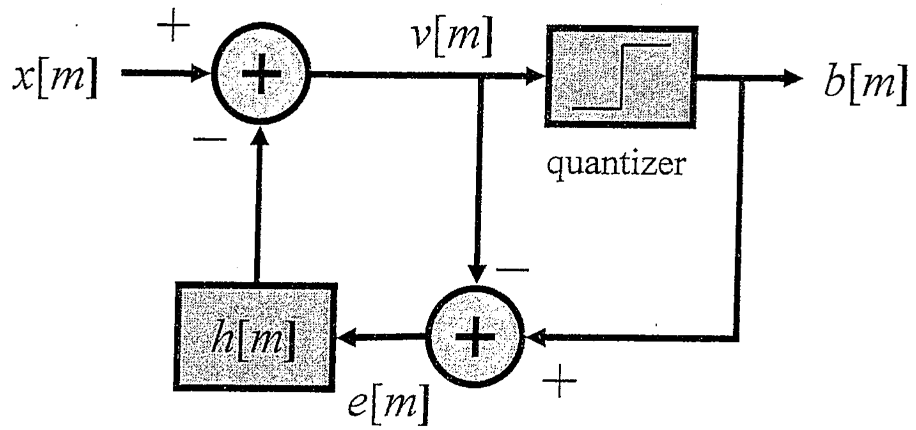
L filters with N_g coefficients each. Executes at symbol rate.

MACs/s : $L N_g f_{sym}$

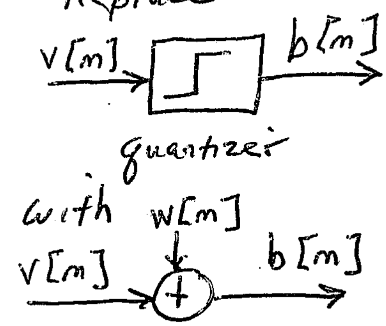
Computational savings over direct implementation by a factor of L.

Problem 2.4. *Noise Shaping.* 19 points.

Here is a block diagram of a noise-shaping feedback coder used in data conversion.



Replace



$h[m]$ 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 $x[m] = 0$.

$b[m] = v[m] + w[m]$

$v[m] = \cancel{x[m]}^{\,0} - h[m] * e[m]$

$e[m] = w[m]$

$B(\omega) = V(\omega) - W(\omega)$

$V(\omega) = -H(\omega)W(\omega)$

$\Downarrow$

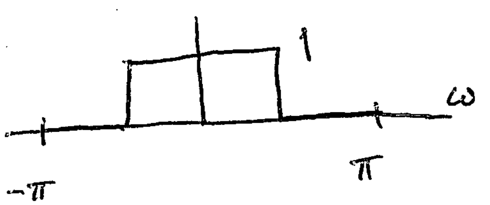
$B(\omega) = -H(\omega)W(\omega) + W(\omega)$

$B(\omega) = (1 - H(\omega))W(\omega)$

$\dfrac{B(\omega)}{W(\omega)} = 1 - H(\omega) \quad \Leftarrow$

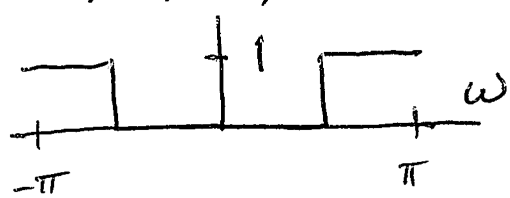
(b) If the frequency selectivity of $h[m]$ were lowpass, what is the frequency selectivity of the noise transfer function? Sketch example frequency responses for both to help justify your answers. *9 points.*

$H(\omega)$



Lowpass

$1 - H(\omega)$



Highpass

$\dfrac{B(\omega)}{W(\omega)} = 1 - H(\omega)$ 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

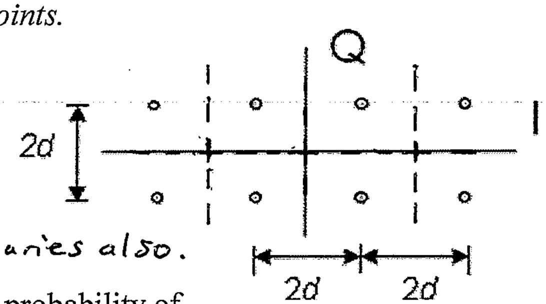
- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected 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.

Bot
Milli
Umifriend
Geo

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		

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 (b), 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.

$$P(c) = \frac{4}{8}\left(1 - Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right)\left(1 - 2Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right) + \frac{4}{8}\left(1 - Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)\right)^2$$

$$P(e) = 1 - P(c) = \frac{5}{2} Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right) - \frac{3}{2} Q^2\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right)$$

(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 amplifier, because it has to operate in a linear region over a larger voltage range.

(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? $\uparrow T_{sym}$ $\downarrow P(e)$ because $\uparrow T_{sym}$ causes increase in argument of Q function

symbol rate? Decreases because $f_{sym} = \frac{1}{T_{sym}}$

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.

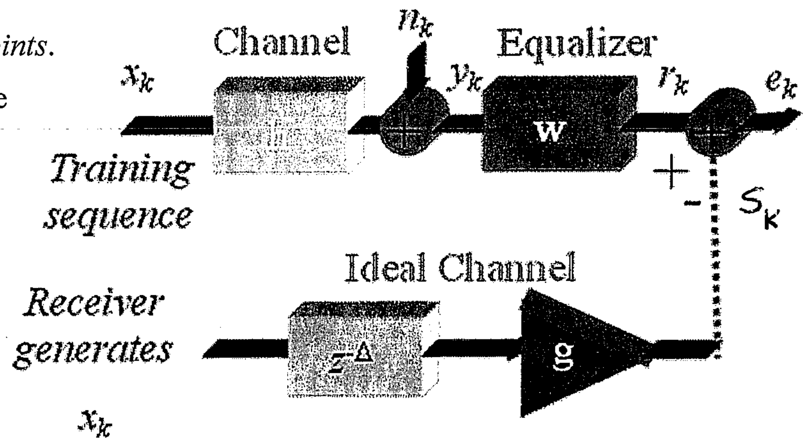
Problem 2.2. *Channel Equalization. 23 points.*

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:

$w[k] = \delta[k] + w_1\, \delta[k-1]$ 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.*

$$w_1[k+1] = w_1[k] - \mu \left.\frac{\partial J_{LMS}[k]}{\partial w_1}\right|_{w_1 = w_1[k]}$$

$$J_{LMS}[k] = \frac{1}{2} e^2[k]$$

$$e[k] = r[k] - s[k] \quad \text{and} \quad s[k] = g\, x[k-\Delta]$$

$$r[k] = y[k] + w_1[k]\, y[k-1]$$

$$w_1[k+1] = w_1[k] - \mu\, e[k]\, y[k-1]$$

(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 the complexity of the adaptive update equation and allows exact recovery of transmitted signal without having to multiply by $\frac{1}{g}$.

(c) What value of mu would you advocate? Why? *4 points.*

To minimize the objective function $J_{LMS}[k]$, $\mu > 0$.

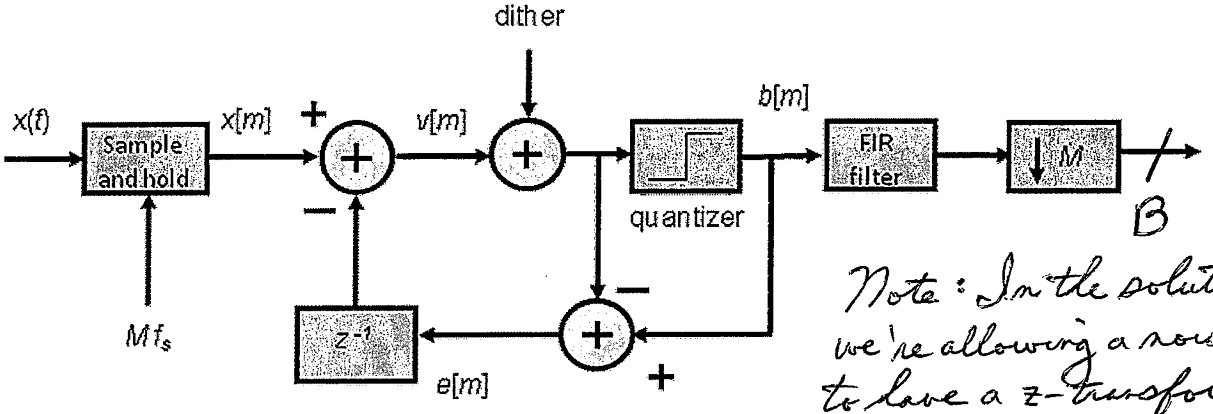
For stability, $0 < \mu < 1$.

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.

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:



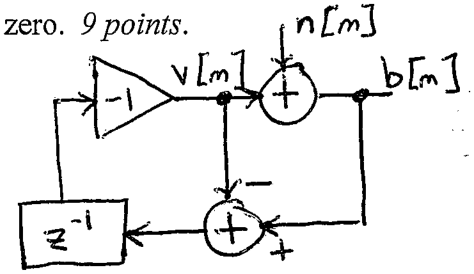
Note: In the solution for (b), we're allowing a noise signal to have a z-transform. This is only true for a finite length noise sequence.

The internal clock runs at $M f_s$.

(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 the two pdfs p_1 and p_2.

(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.*



$$B(z) = V(z) + N(z)$$

$$V(z) = \left(B(z) - V(z)\right) z^{-1}(-1)$$

$$= -N(z) z^{-1}$$

$$B(z) = -N(z) z^{-1} + N(z) \Rightarrow \frac{B(z)}{N(z)} = 1 - z^{-1}$$

(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)

$2^4 = 16$ 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 worst case...

Polyphase filter bank implementation of cascade of FIR filter and downsampler

$16M$ A polyphase filter bank would have M polyphase FIR filters. Each filter can be $2^4 = 16$ coefficients long.

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 receiver.

(c) Fading. *4 points.* Automatic gain control.

Method #1: Squared difference adaptive element (JSK p. 122)

Method #2: Use counters of occurence of A/D output values of maximum integer, 0, and minimum integer and adapt gain. (Midterm #2 from fall 2011.)

(d) Carrier mismatch. *4 points.* Carrier recovery.

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 $w_c + 0.5\, w_{sym}$ and $w_c - 0.5\, w_{sym}$ 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: Diego
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.*

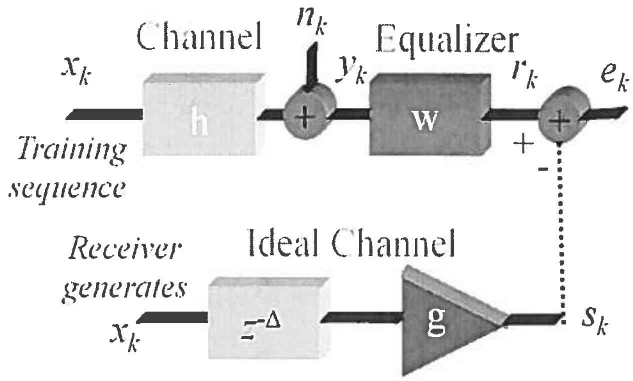
In the discrete-time system on the right, the equalizer operates at the sampling rate.

Equalizer has real coefficients w_0 and w_1:

$r[k] = w_0\, y[k] + w_1\, y[k\text{-}1]$

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.*

$$e[k] = r[k] - s[k] = w_0\, y[k] + w_1\, y[k-1] - g\, x[k-\Delta]$$

$$w_0[k+1] = w_0[k] - \mu \frac{d}{dw_0} J_{LMS}[k] \Big|_{w_0 = w_0[k]}$$

$$\frac{d}{dw_0} J_{LMS}[k] = e[k]\, y[k]$$

$$w_0[k+1] = w_0[k] - \mu\, e[k]\, y[k]$$

(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 $\mathbf{w} = [w_0\ w_1]$. Using the result from (a), write the update in one equation in vector form. Please define any new vectors that you introduce. *6 points.*

$$\vec{w}[k+1] = \vec{w}[k] - \mu\, e[k]\, \vec{y}[k]$$

where $\vec{y}[k] = [\, y[k] \quad y[k-1]\,]$

(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.

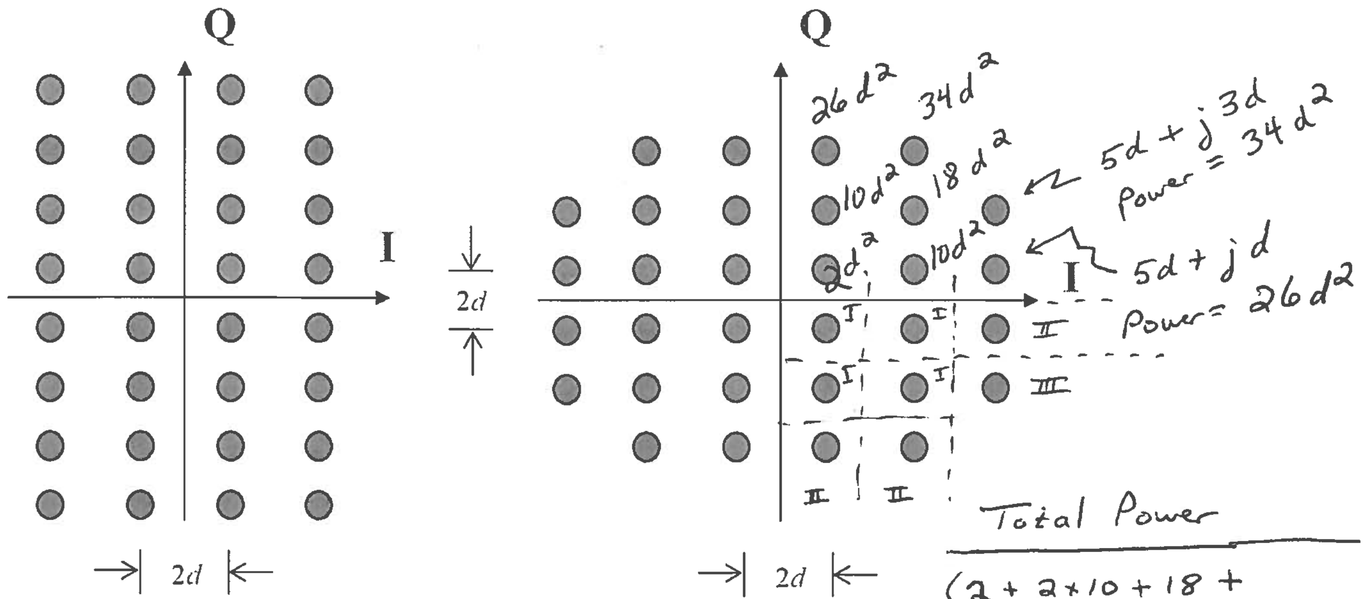
$e[k]$ takes $n+1$ multiplications to compute.

$\mu\, e[k]$ 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 $2d$.



Total Power

$(2 + 2\times 10 + 18 + 2\times 26 + 2\times 34)\,d^2 = 160\,d^2$

Energy in pulse shape is 1. Symbol time T_{sym} is 1s.

	Left Constellation	Right Constellation
(a) Peak power	$\cancel{56d^2}$ $58d^2$	$34d^2$
(b) Average power	$25.875\,d^2$	$20\,d^2$
(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*.

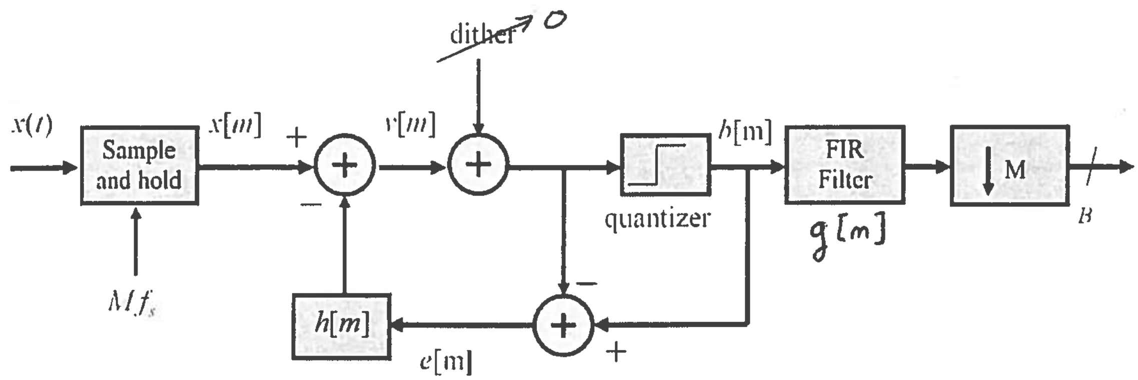
Due to quadrant symmetry, average power can be computed over one quadrant.

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 $Q\left(\frac{d}{\sigma}\right)$. 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.*

$$B(z) = K\,V(z)$$

$$E(z) = V(z) - B(z) = \frac{1}{K}B(z) - B(z) = \frac{K-1}{K}B(z)$$

$$V(z) = X(z) - H(z)\,E(z)$$

$$\Rightarrow \frac{1}{K}B(z) = X(z) - H(z)\frac{K-1}{K}B(z) \Rightarrow \frac{B(z)}{X(z)} = \frac{K}{1+(K-1)H(z)}$$

(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.*

$$\omega_{stop} = \frac{\pi}{M}$$

$$\omega_{pass} = 0.9\,\omega_{stop}$$

$$A_{pass} = 20\log_{10}\Delta \quad \text{where } \Delta = \frac{1}{2^B - 1}$$

$$A_{stop} = 6B + \underbrace{C_0}_{\text{constant that depends on the application}}$$

Note: An FIR filter of length M+1 coefficients that performs averaging would give 13.5 dB of stopband attenuation (not much).

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 detected to then run the correlator. This saves power/energy.

(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 frequency. That is why the method locks onto frequencies $\omega_c - \omega_{sym}$ and $\omega_c + \omega_{sym}$ for QAM 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 $w_c - \frac{1}{2} w_{sym}$ and $w_c + \frac{1}{2} w_{sym}$.

(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

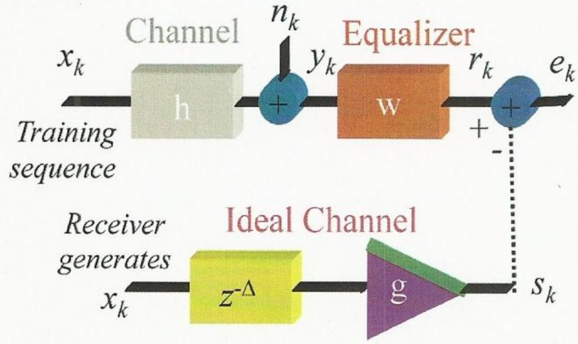
- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets. You may not share materials with other students.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected 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.

Johnny
Susan
Mary
Dukey

Problem	Point Value	Your score	Topic
1	25		Channel Equalization
2	27		Receiver Design
3	30		Pre-emphasis
4	18		Potpourri
Total	100		

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:

$r[k] = w_0\, y[k] + w_1\, y[k\text{-}1]$

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.*

$$J(e[k]) = \frac{1}{4}e^4[k]$$

$$e[k] = r[k] - s[k]$$

$$= w_0\, y[k] + w_1\, y[k-1] - s[k]$$

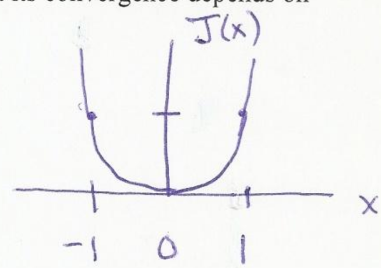
$$w_1[k+1] = w_1[k] - \mu \left.\frac{\partial J(e[k])}{\partial w_1}\right|_{w_1 = w_1[k]}$$

$$w_1[k+1] = w_1[k] - \mu\, e^3[k]\, y[k-1]$$

(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 $|f'(x)| < 1$.

$$f'(x) = 1 - 3\mu x^2$$

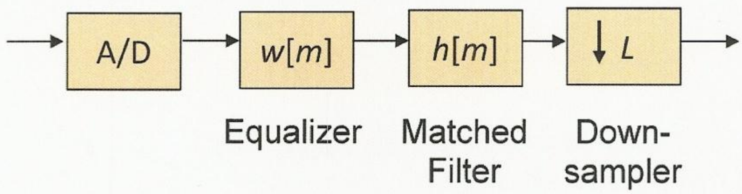
$$-1 < 1 - 3\mu x^2 < 1 \Rightarrow -2 < -3\mu x^2 < 0$$

$$|x| < \sqrt{\frac{2}{3\mu}} \Leftarrow 0 < 3\mu x^2 < 2$$

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 $[0, 3 f_{sym}]$ and operates at the sampling rate of $6 f_{sym}$. 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 precision calculations.

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, ...

$6,5,4,3,2,1 \rightarrow \boxed{\downarrow 6} \rightarrow 1$ $\qquad$ $6,5,4,3,2,1 \rightarrow \boxed{\downarrow 3} \xrightarrow{4,1} \boxed{\downarrow 2} \rightarrow 1$

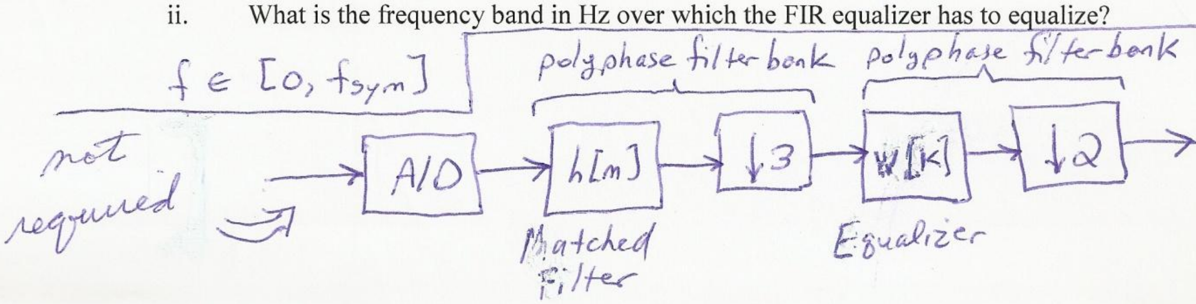
d) The second step to reduce the inefficiency is to exchange the FIR equalizer with the downsampling by 3. *9 points.*

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.

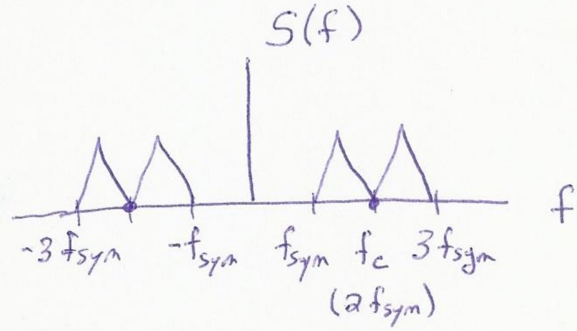
ii. What is the frequency band in Hz over which the FIR equalizer has to equalize?

$f \in [0, f_{sym}]$



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	$\alpha = 1$

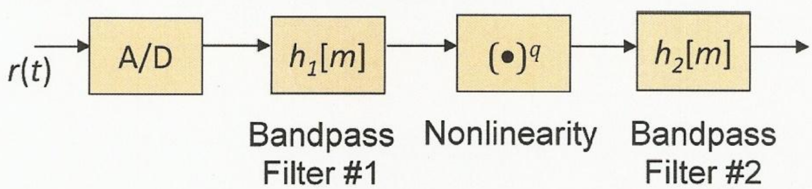


and where

Carrier frequency	$f_c = 2 f_{sym}$
Transmission bandwidth	$B = 2 f_{sym}$

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 qth 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 $3 q f_{sym}$. Aliasing if $q > 3$.

Carrier frequency becomes $2 q f_{sym}$. Aliasing if $q > 4$. q is 2 or 4.

ii. which value of q would you use and why?

$q = 2$ for computational efficiency and less word length expansion vs. $q = 4$.

iii. give the center frequency for BPF #2

$2 q f_{sym}$ or $4 f_{sym}$ for $q = 2$.

(c) Pre-emphasis of symbol clock f_{sym}. *12 points.*

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 $f_c - \frac{1}{2} f_{sym} = \frac{3}{2} f_{sym}$

i. give all possible values for q

q must be even. Symbol clock at $\frac{3}{2} q f_{sym}$. Aliasing if $q > 6$. q is 2, 4 or 6.

ii. which value of q would you use and why?

$q = 2$ for computational efficiency

iii. give the center frequency for BPF #2

$\frac{3}{2} q f_{sym}$ or $3 f_{sym}$ for $q = 2$.

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).

Noise power increases only when $B > \frac{1}{2}$.

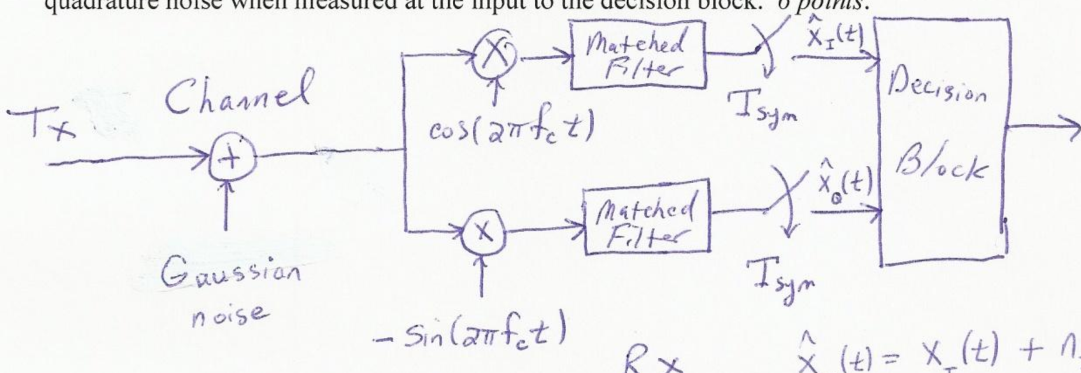
(b) In discrete time, an ideal channel can be modeled as

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.*



$\hat{x}_I(t) = x_I(t) + n_I(t)$

$\hat{x}_Q(t) = x_Q(t) + n_Q(t)$

where $t = n T_{sym}$.

In-phase noise $n_I(t)$ and quadrature noise $n_Q(t)$ have the same source — Gaussian channel noise. They cannot be statistically independent.

Direct Sequence Spreading

Gene W. Marsh

I. A General Description of Direct Sequence Spreading

A. The standard view of a communication system

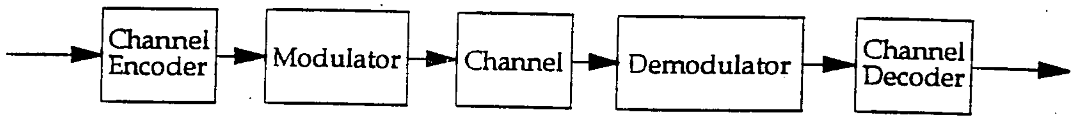


Fig. 1. A block diagram for a Standard Communication System.

B. A PN-Spread Communication System

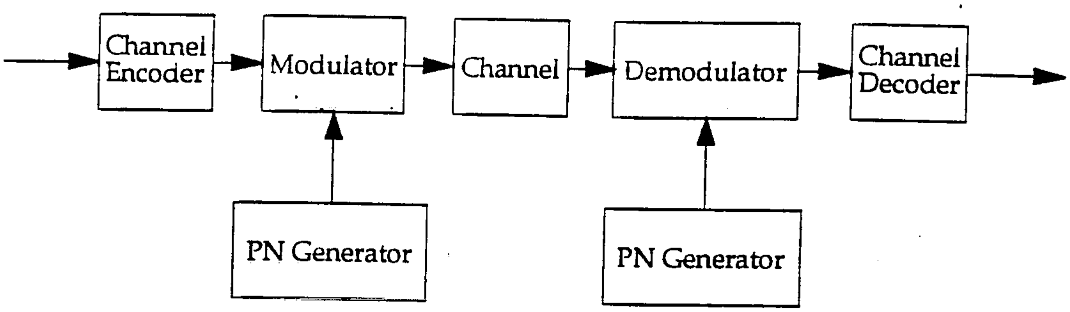


Fig. 2. A block diagram for a PN-spread Communication System.

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]

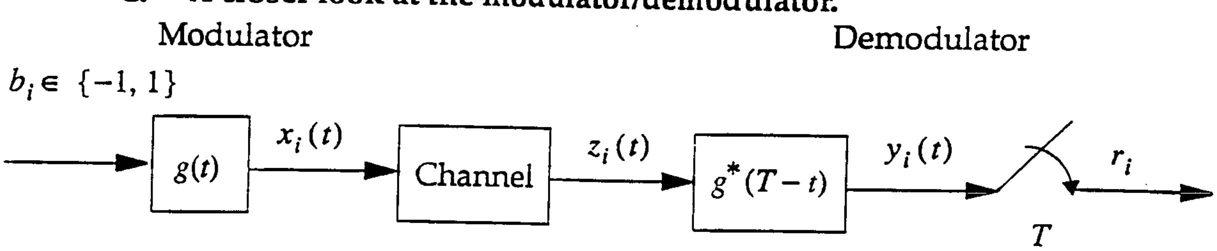


Fig. 3. The Standard Model

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.

L - 1

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.**

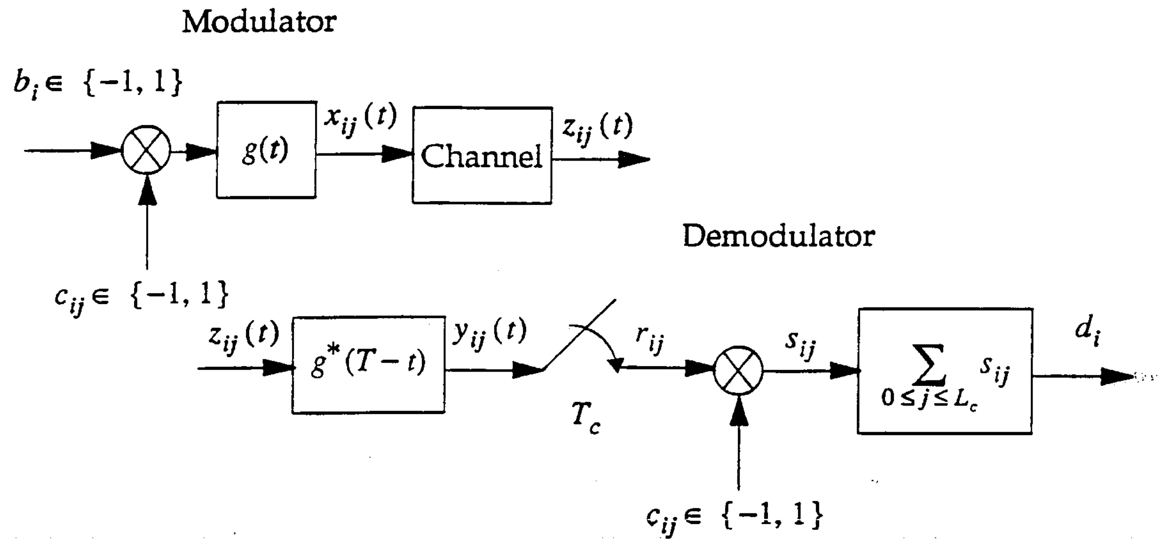


Fig. 4. The CDMA Model

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 $\frac{1}{2}$.

L-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?

1. 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.

   a) 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:

$$P(\mathrm{e}|b_i=-1) = P(d_i>0|b_i=-1) = Q\left(\frac{E_b}{\sqrt{\frac{N_0}{2}E_b}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P(\mathrm{e}|b_i=1) = P(d_i<0|b_i=1) = 1-Q\left(\frac{-E_b}{\sqrt{\frac{N_0}{2}E_b}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P(\mathrm{e}) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \qquad (1)$$

   b) In the CDMA system, for each chip, we have

   $$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + n(t) * g^*(T-t)$$

The r_{ij} are then independent, with a Gaussian distribution such that $E[r_{ij}|b_i, c_j] = b_i c_{ij} E_c$ and

$\sigma^2(r_{ij}|b_i, c_{ij}) = \frac{N_0}{2}\int_{-\infty}^{\infty}|G(f)|^2 df = \frac{N_0}{2}E_c$ where $E_c = \int_0^T |g(t)|^2 dt$.

After the multiplier, the s_{ij} clearly have mean $E[s_{ij}|b_i, c_{ij}] = b_i E_c$ and variance $\sigma^2(s_{ij}|b_i, c_{ij}) = \sigma^2(r_{ij}|b_i, c_{ij})$. 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

$$P(e) = Q\left(\sqrt{\frac{2L_c E_c}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \qquad (2)$$

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

L-4

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 $0 \le k < 2^n$. 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

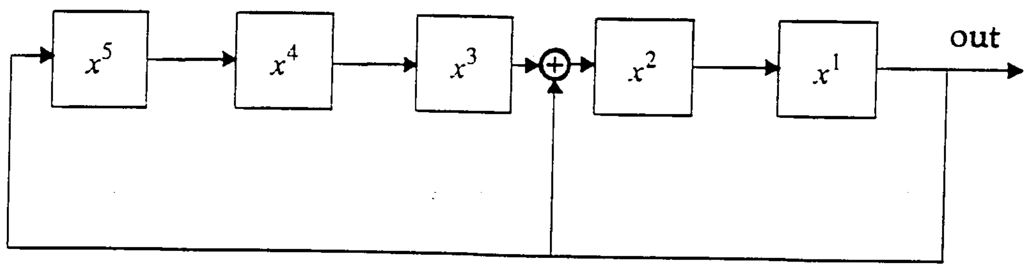


Fig. 5. A Galois Configuration

general, this is the easiest way to implement an M-sequence in software. The sequence produced by this machine is shown in Table 1.

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.

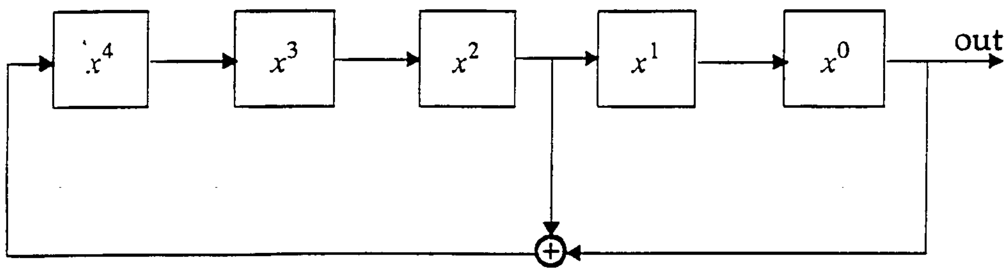


Fig. 6. A Fibonacci Configuration

L-6

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 hyper-space.

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. Buy 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 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

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + b_k c_{kl} g(t) * g^*(T-t) + n(t) * g^*(T-t) \tag{3}$$

This will in turn give:

$$E[r_{ij}| b_i, c_j] = b_i c_{ij} E_c$$

$$\sigma^2(r_{ij}| b_i, c_{ij}) = \frac{N_0}{2}E_c + E_c^2$$

$$E[s_{ij}| b_i, c_{ij}] = b_i E_c$$

$$\sigma^2(s_{ij}| b_i, c_{ij}) = \sigma^2(r_{ij}| b_i, c_{ij})$$

$$E[d_i| b_i] = b_i L_c E_c$$

$$\sigma^2(d_i| b_i) = \frac{N_0}{2}L_c E_c + L_c E_c^2$$

$$P(e) = Q\left(\sqrt{\frac{(L_cE_c)^2}{\frac{N_0}{2}L_cE_c + L_cE_c^2}}\right) = Q\left(\sqrt{\frac{E_b}{\frac{N_0}{2} + E_c}}\right) \tag{4}$$

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

$$P(e) \le Q\left(\sqrt{\frac{E_b}{\frac{N_0}{2} + (N-1)E_c}}\right) \tag{5}$$

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 $TM = \left(\sum_{0 \le j < L_c} s_{ij}^e\right)^2 - \left(\sum_{0 \le j < L_c} s_{ij}^l\right)^2$. 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.

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

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + b_k c_{kl} g(t+\tau) * g^*(T-t) + n(t) * g^*(T-t) \tag{6}$$

This will in turn give:

$$E[r_{ij} | b_i, c_j] = b_i c_{ij} E_c$$

$$\sigma^2(r_{ij} | b_i, c_{ij}) = \frac{N_0}{2} E_c + E'^2$$

$$E[s_{ij} | b_i, c_{ij}] = b_i E_c$$

$$\sigma^2(s_{ij} | b_i, c_{ij}) = \sigma^2(r_{ij} | b_i, c_{ij})$$

$$E[d_i | b_i] = b_i L_c E_c$$

$$\sigma^2(d_i | b_i) = \frac{N_0}{2} L_c E_c + L_c E'^2$$

$$P(\mathrm{e}) = Q\left(\sqrt{\frac{(L_c E_c)^2}{\frac{N_0}{2} L_c E_c + L_c E'^2}}\right)$$

Now, we know that $E' \leq E_c$, so that

$$P(e) \le Q\left(\sqrt{\frac{L_c E_c}{\frac{N_0}{2}+E_c}}\right) = Q\left(\sqrt{\frac{E_b}{\frac{N_0}{2}+E_c}}\right) \tag{7}$$

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 System[2]

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 $\left.\left(\frac{E_b}{N_0}\right)\right|_{\text{desired}} = 7\text{ dB}$.
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 the 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.

a) We need this to be equal to $\left.\left(\frac{E_b}{N_0}\right)\right|_{\text{desired}}$.

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

$$N \approx \frac{R_c}{R} \cdot \frac{1}{\left.\left(\frac{E_b}{N_0}\right)\right|_{\text{desired}}} \cdot \frac{1}{d} \cdot F \cdot G \tag{8}$$

8. For the CDMA system, $R = 9600\ \text{Hz}$ and $R_c = 1.2288\ \text{MHz}$.
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

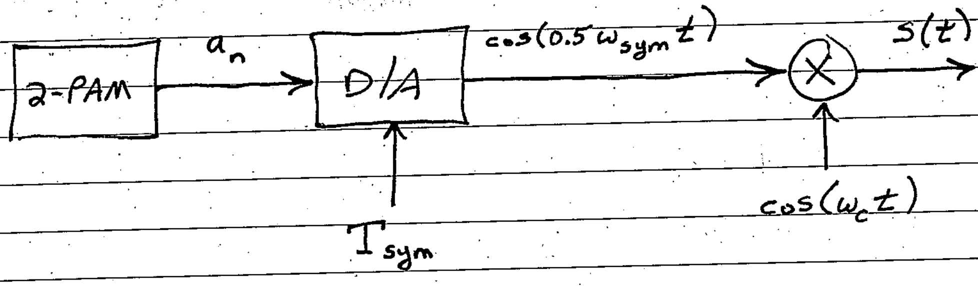
[4] *An Overview of the Application of Code Division Multiple Access (CDMA) to Digital Cellular Systems and Personal Cellular Networks.* Document Number EX60-10010. Qualcomm, Inc., San Diego, CA, May 21, 1992.

Symbol Recovery Simplified

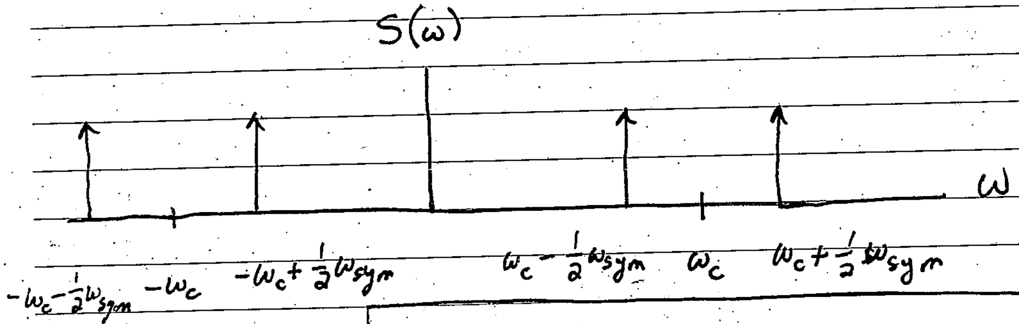
EE345S
Evans

Transmit 2-PAM Symbols

$$a_n = (-1)^n = \cos(n\pi) = \cos(0.5\,\omega_{sym} T_{sym}\, n)$$



$$s(t) = \cos(0.5\,\omega_{sym} t)\cos(\omega_c t)$$



Receiver

$$f[n] = e^{j\omega_{sym}(n T_{sym} + \tau)}$$
$$= e^{j(\omega_{sym} T_{sym} n + \omega_{sym}\tau)}$$
$$= e^{j(2\pi n + \omega_{sym}\tau)}$$
$$f[n] = e^{j\omega_{sym}\tau}$$

$$p[n] = \alpha v[n] + \beta \gamma[n]$$

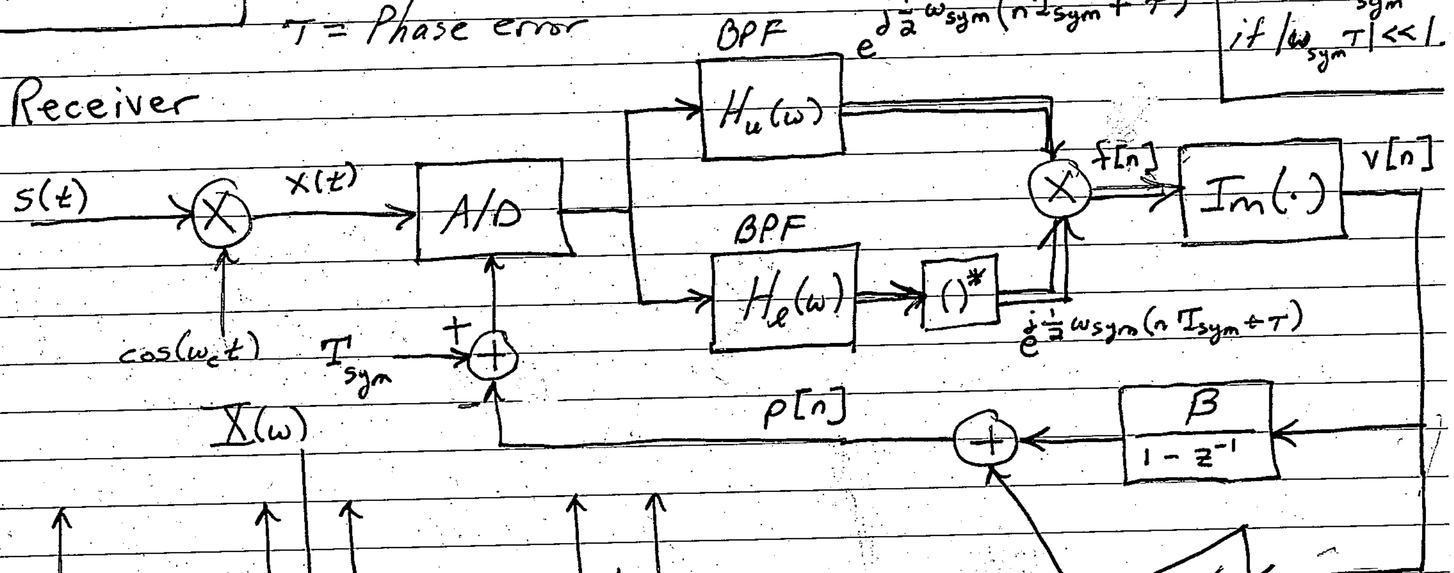
$$\gamma[n] = v[n] + \gamma[n-1]$$
with $\gamma[-1] = 0$.

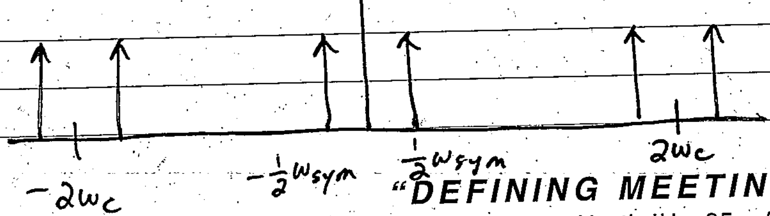
$$v[n] = Im(e^{j\omega_{sym}\tau}) = \sin(\omega_{sym}\tau)$$

$v[n] \approx \omega_{sym}\tau$
if $|\omega_{sym}\tau| \ll 1$

τ = Phase error

Receiver





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, ... h_{N-1}) given one input sample $x[n]$ takes N multiplication and N-1 addition operations:

$$y[n] = h_0\, x[n] + h_1\, x[n-1] + \ldots + h_{N-1}\, x[n - (N-1)]$$

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).

Cycle	Instruction
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, product0**
7	LDW $x[n-6]$ || LDW h_6 || MPYSP $x[n-1]$, h_1, product1
8	LDW $x[n-7]$ || LDW h_7 || MPYSP $x[n-2]$, h_2, product2
9	LDW $x[n-8]$ || LDW h_8 || MPYSP $x[n-3]$, h_3, product3
10	LDW $x[n-9]$ || LDW h_9 || MPYSP $x[n-4]$, h_4, product4 || **ADDSP product0, accumulator0, accumulator0**
11	LDW $x[n-10]$ || LDW h_{10} || MPYSP $x[n-5]$, h_5, product5 || ADDSP product1, accumulator1, accumulator1
12	LDW $x[n-11]$ || LDW h_{11} || MPYSP $x[n-6]$, h_6, product6 || ADDSP product2, accumulator2, accumulator2
13	LDW $x[n-12]$ || LDW h_{12} || MPYSP $x[n-7]$, h_7, product7 || ADDSP product3, accumulator3, accumulator3
14	LDW $x[n-13]$ || LDW h_{13} || MPYSP $x[n-8]$, h_8, 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 345S 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. The 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 $H_{freq}(f) = e^{-j 2\pi f t_0}$ or $H_{freq}(\omega) = e^{-j\omega n_0}$. 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:

$$H(z) = \frac{z - \frac{1}{r}}{z - r} \longrightarrow H_{freq}(\omega) = \frac{e^{j\omega} - \frac{1}{r}}{e^{j\omega} - r}$$

assuming that $|r| < 1$ for asymptotic stability. Magnitude response is

$$|H_{freq}(\omega)| = \left| \frac{e^{j\omega} - \frac{1}{r}}{e^{j\omega} - r} \right| = \frac{|e^{j\omega} - \frac{1}{r}|}{|e^{j\omega} - r|}$$

Here, $|e^{j\omega} - a| = |(\cos\omega - a) + j \sin\omega| = \sqrt{(\cos\omega - a)^2 + \sin^2\omega}$.

$$= \sqrt{\cos^2\omega - 2a\cos\omega + a^2 + \sin^2\omega} = \sqrt{a^2 - 2a\cos\omega + 1}$$

$$|H_{freq}(\omega)| = \frac{\sqrt{\frac{1}{r^2} - \frac{2}{r}\cos\omega + 1}}{\sqrt{r^2 - 2r\cos\omega + 1}} = \frac{\sqrt{\frac{1}{r^2}(r^2 - 2r\cos\omega + 1)}}{\sqrt{r^2 - 2r\cos\omega + 1}} = \frac{1}{|r|}$$

- 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:

$$H(z) = \frac{z - \frac{1}{r_0} e^{j\omega_0}}{z - r_0 e^{j\omega_0}} \Rightarrow H_{freq}(\omega) = \frac{e^{j\omega} - \frac{1}{r_0} e^{j\omega_0}}{e^{j\omega} - r_0 e^{j\omega_0}}$$

assuming that $r_0 < 1$ for asymptotic stability. The magnitude response is

$$|H_{freq}(\omega)| = \frac{|e^{j\omega} - \frac{1}{r_0} e^{j\omega_0}|}{|e^{j\omega} - r_0 e^{j\omega_0}|}$$

Here,

$$\begin{aligned} |e^{j\omega} - (a+jb)| &= |(\cos\omega - a) + j(\sin\omega - b)| \\ &= \sqrt{(\cos\omega - a)^2 + (\sin\omega - b)^2} \\ &= \sqrt{\cos^2\omega - 2a\cos\omega + a^2 + \sin^2\omega - 2b\sin\omega + b^2} \\ &= \sqrt{(a^2+b^2) - 2\sqrt{a^2+b^2}\cos(\omega+\theta) + 1} \end{aligned}$$

where $\theta = \arctan\left(-\frac{b}{a}\right)$.

$$|H_{freq}(\omega)| = \frac{\sqrt{\frac{1}{r_0^2} - \frac{2}{r_0}\cos(\omega+\theta) + 1}}{\sqrt{r_0^2 - 2r_0\cos(\omega+\phi) + 1}}$$

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,

$$|H_{freq}(\omega)| = \frac{\sqrt{\frac{1}{r_0^2} - \frac{2}{r_0}\cos(\omega-\omega_0) + 1}}{\sqrt{r_0^2 - 2r_0\cos(\omega-\omega_0) + 1}}$$

$$= \frac{\sqrt{\frac{1}{r_0^2}(r_0^2 - 2r_0\cos(\omega-\omega_0) + 1)}}{\sqrt{r_0^2 - 2r_0\cos(\omega-\omega_0) + 1}} = \frac{1}{r_0}$$

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

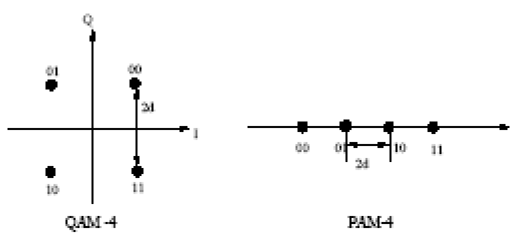
In the receiver,

- 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,



(a) Derive the symbol error probability formula for 4-QAM, also known as Quadrature Phase Shift Keying (QPSK), shown in Figure 1.

(b) Calculate the average power of the QPSK signal given d.

(c) 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.

(d) 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 by $P_3(c) = \left(1 - Q\left(\frac{d}{\sigma}\right)\right)^2$. Thus the probability of error is given 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 $\left(\sqrt{E}\cos[(2i-1)\frac{\pi}{4}], \sqrt{E}\sin[(2i-1)\frac{\pi}{4}]\right)$ in the $(\phi_1(t) - \phi_2(t))$ coordinate system. Thus, it is immediate that $E\left(\cos^2[(2i-1)\frac{\pi}{4}] + \sin^2[(2i-1)\frac{\pi}{4}]\right) = E$. This implies that $P = \frac{E}{T}; T = 1 \Rightarrow P = E$.

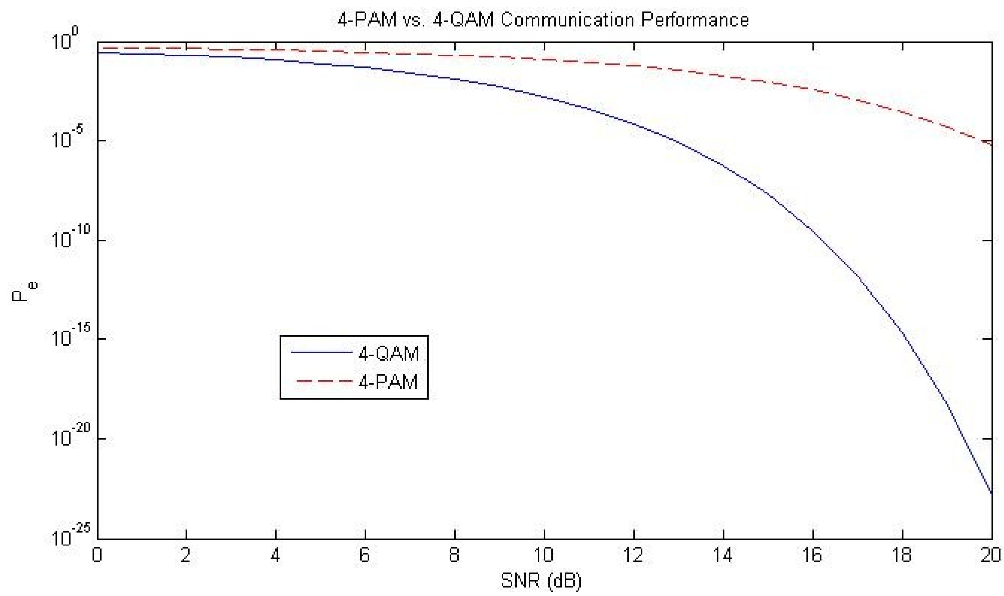
$P_{AVG} = \frac{1}{4}\left(4 \times 2d^2\right) = 2d^2$.

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-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:

$$P_{e-QAM} = 2Q\left(\frac{d}{\sigma}\right) - Q^2\left(\frac{d}{\sigma}\right) = 2Q\left(\sqrt{SNR}\right) - Q^2\left(\sqrt{SNR}\right)$$

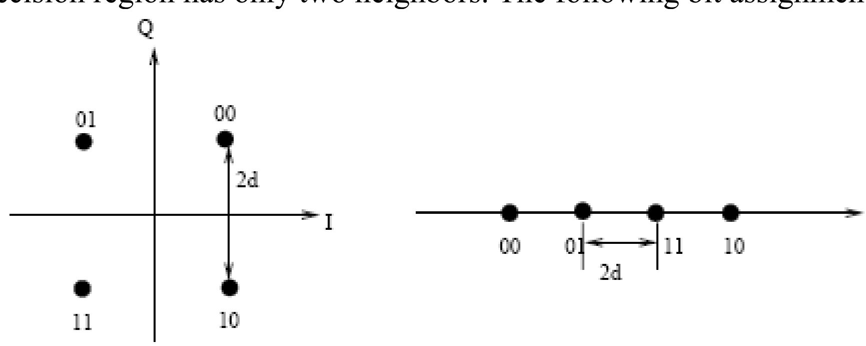
$$P_{e-PAM} = \frac{3}{2}Q\left(\sqrt{\frac{SNR}{5}}\right)$$

```
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.



QAM -4 PAM-4

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

$$\mathbf{x} = [-7\ 0\ 0\ 0\ \ -5\ 0\ 0\ 0\ \ -3\ 0\ 0\ 0\ \ -1\ 0\ 0\ 0\ \ 1\ 0\ 0\ 0\ \ 3\ 0\ 0\ 0\ \ 5\ 0\ 0\ 0\ \ 7\ 0\ 0\ 0]$$

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.

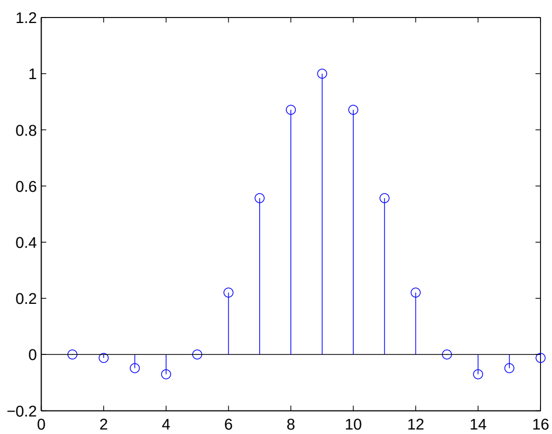
Hints: To compute the values of h, please use the "rcosine" command in Matlab and not the "SRRC" command. The length of h should be 16. The syntax of the "rcosine" command is

```
rcosine(Fd, Fs, TYPE_FLAG, beta)
```

The ratio Fs/Fd must be a positive integer. Since the the number of samples per symbol is 4, Fs/Fd 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

$$y_{circular}[n] = x[n] \otimes_N h[n] = \sum_{m=0}^{N-1} h[((m))_N] \; x[((n-m))_N]$$

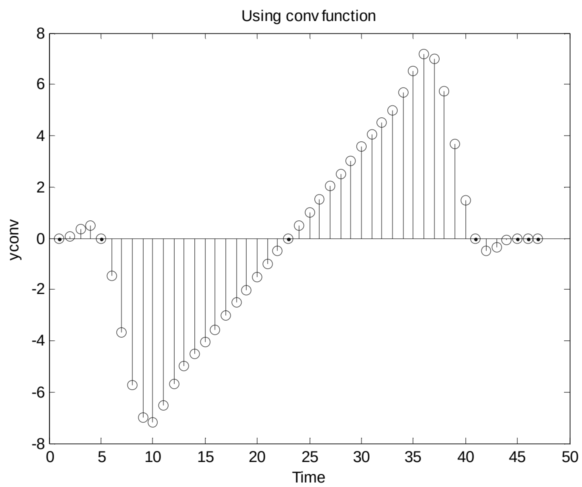
where $((\bullet))_N$ means that the argument is taken modulo N. We will henceforth refer to the circular convolution between periodic signals of length N as circular convolution of length N. On a programmable digital signal processor, we would use the modulo addressing mode to accelerate the computation of circular convolution.

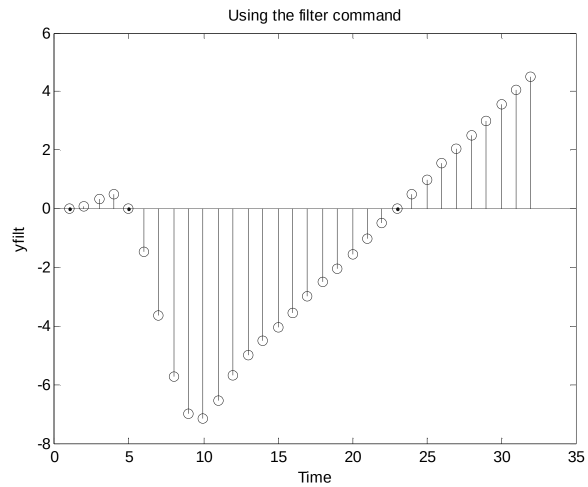
Circular convolution of two finite-length sequences $x[n]$ and $h[n]$ is equivalent to linear convolution of those sequences by padding (appending) L_x-1 zeros to $h[n]$ and L_h-1 zeros to $x[n]$ so that both of them are of the same length and using a circular convolution length of L_x+L_h-1 samples. This is the approach used in the FFT-based method in this problem.

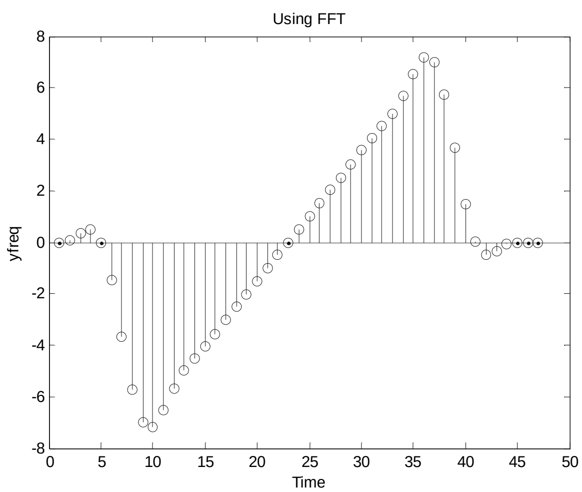
The FFT-based method to compute the linear convolution uses an FFT length of N of L_x+L_h-1. First, the FFT of length N of the zero-padded $x[n]$ is computed to give $X[k]$, and the FFT of length N of the zero-padded $h[n]$ is computed to give $H[k]$. Second, the product $Y_{circular}[k] = X[k]\, H[k]$ for $k = 0,\ldots, N-1$ is computed. Then, the inverse FFT of length N of $Y_{circular}[k]$ is computed to find $y_{circular}[n]$. This ***third way*** results in an output signal of $L_x+L_h-1 = 47$ samples.

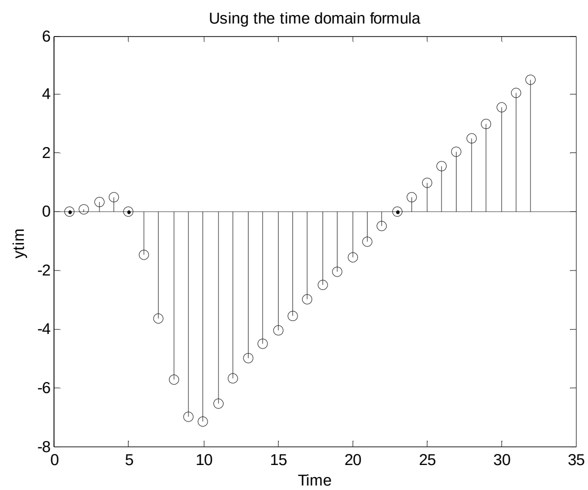
The ***fourth way*** to filter a signal uses a time-domain formula. It is an alternate implementation of the same approach used by the filter command. Hence, this approach gives an output of length $L_x = 32$ samples.

```
% 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
```


Using conv function
yconv
Time
8
6
4
2
0
-2
-4
-6
-8
0 5 10 15 20 25 30 35 40 45 50


Using the filter command
yfilt
Time
6
4
2
0
-2
-4
-6
-8
0 5 10 15 20 25 30 35


Using FFT
yfreq
Time
8
6
4
2
0
-2
-4
-6
-8
0 5 10 15 20 25 30 35 40 45 50


Using the time domain formula
ytim
Time
6
4
2
0
-2
-4
-6
-8
0 5 10 15 20 25 30 35

Quick Introduction to the Fourier Transform

by Prof. Brian Evans
and Mr. Jean Farias
The University of Texas
at Austin Spring 2013

Fourier Transform (radian, Hz, seconds)

$$\underset{\substack{\uparrow\\ \text{real variable } f}}{X(f)} = \int_{-\infty}^{\infty} \underset{\substack{\uparrow\\ \text{Amplitude}\\ \text{real or complex}}}{x(t)}\, e^{-j 2\pi f t}\, dt$$

Real variable t

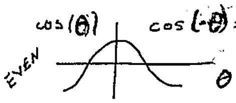
Amplitude is usually complex

$$\underbrace{e^{-j2\pi f t}}_{\text{complex-valued sinusoid}} = \overset{\text{real}}{\cos(-2\pi f t)} + \overset{\text{imag}}{j\sin(-2\pi f t)}$$

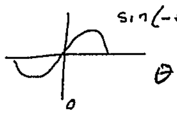
$$= \cos(2\pi f t) - j\sin(2\pi f t)$$

$$x(t) \xrightarrow{\mathcal{F}} X(f)$$

$$x_1(t) + x_2(t) \xrightarrow{\mathcal{F}} X_1(f) + X_2(f)$$



$\cos(-\theta) = \cos(\theta)$



$\sin(-\theta) = -\sin\theta$ odd

$$\boxed{X(s) = \mathcal{L}\{x(t)\} \quad \text{region of convergence (ROC)}}$$

Laplace Transform $\quad X(s) = \int_{-\infty}^{\infty} x(t)\, e^{-st}\, dt$

If the ROC includes the imaginary axis

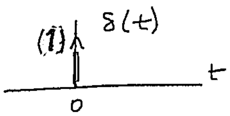
$$X(f) = X(s)\Big|_{s = j2\pi f}$$

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

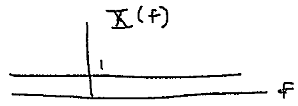
$$\int_{-\infty}^{\infty} g(t)\,\delta(t)\, dt = g(0)$$

Unit area concentrated at origin — area denoted by parentheses

sifting Property

If $g(t)$ is defined at origin

$$\mathbb{X}(f) = \int_{-\infty}^{\infty} \delta(t) \underbrace{e^{-j2\pi f t}}_{g(t)}\, dt = 1 \qquad \text{when } t=0$$



contains all frequencies

How to get direct impulse response

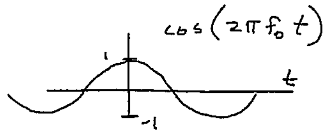
$$\underset{\text{impulse}}{\xrightarrow{\delta(t)}} \boxed{\text{LTI System}} \underset{\text{impulse response}}{\xrightarrow{h(t)}}$$

$$\delta(t) \xrightarrow{\mathcal{F}} 1$$

Example

two sided cosine

this does not have laplace
it have fourier transform
can't build this in lab



Euler's Formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

$$\cos(2\pi f_0 t) = \frac{1}{2}e^{-j2\pi f_0 t} + \frac{1}{2}e^{j2\pi f_0 t}$$

$$= \frac{1}{2}\left(\cos(-2\pi f_0 t) + j\sin(\cancel{-2\pi f_0 t})\right) + \frac{1}{2}\left(\cos(2\pi f_0 t) + j\sin(\cancel{2\pi f_0 t})\right)$$

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

Inverse Fourier Transform

$$x(t) = \int_{-\infty}^{\infty} \mathbb{X}(f) e^{j2\pi f t}\, df$$

$$\int_{-\infty}^{\infty} e^{-j2\pi f_0 t} e^{-j2\pi f t}\, dt$$

$$= \int_{-\infty}^{\infty} e^{-j2\pi(f+f_0)t}\, dt = \delta(f+f_0)$$

For $\mathbb{X}(f) = \delta(f)$,

$$x(t) = \int_{-\infty}^{\infty} \delta(f) e^{j2\pi f t}\, df$$

$$= 1$$

example continued on next page

$$1 \xrightarrow{\mathcal{F}} \delta(f)$$

$$X(f+f_0) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi(f+f_0)t}\, dt$$

$$= \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi f_0 t}\, e^{-j2\pi f t}\, dt$$

$$x(t)\, e^{-j2\pi f_0 t} \xrightarrow{\mathcal{F}} X(f+f_0)$$

$$1 \xrightarrow{\mathcal{F}} \delta(f)$$

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{j2\pi f t}\, df \qquad \text{Inverse Fourier}$$

$$X(f) = \delta(f), \quad x(t) = \int_{-\infty}^{\infty} \delta(f) \underbrace{e^{j2\pi f t}}_{g(f)} df = g(0) = 1$$

always ~~evaluate~~ this ~~at the~~ origin in the above case is when $f = 0$

$$x(t)\, e^{-j2\pi f_0 t} \xrightarrow{\mathcal{F}} X(f+f_0)$$

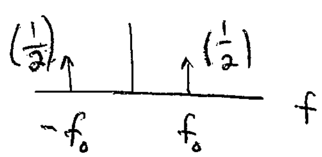
$$1 \xrightarrow{\mathcal{F}} \delta(f)$$

$$1 \cdot e^{-j2\pi f_0 t} \xrightarrow{\mathcal{F}} \delta(f+f_0)$$

$$e^{j2\pi f_0 t} \xrightarrow{\mathcal{F}} \delta(f-f_0)$$

$$\mathcal{F}\{\cos(2\pi f_0 t)\} = \mathcal{F}\left\{ \tfrac{1}{2} e^{-j2\pi f_0 t} + \tfrac{1}{2} e^{j2\pi f_0 t} \right\}$$

$$= \underbrace{\tfrac{1}{2}}_{\text{area of } \frac{1}{2}} \delta(f+f_0) + \underbrace{\tfrac{1}{2}}_{\text{area of } \frac{1}{2}} \delta(f-f_0)$$

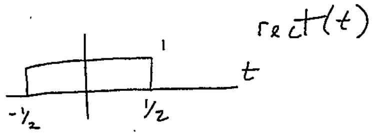


1st find the origin

$f + f_0 = 0 \qquad f - f_0 = 0$

$f = -f_0 \qquad f = f_0$

Example using the rectangular pulse



Has unit area.

Taking the Fourier transform,

$$\int_{-1/2}^{1/2} \overset{1}{\cancel{x(t)}}\, e^{-j2\pi f t}\, dt$$

$$\frac{1}{-j2\pi f} e^{-j2\pi f t} \Big|_{-1/2}^{1/2}$$

$$\frac{1}{-j2\pi f} e^{-j\pi f} + \frac{1}{j2\pi f} e^{j\pi f}$$

$$\frac{j}{j(-j)2\pi f} e^{-j\pi f} + \frac{j}{j(j)2\pi f} e^{j\pi f}$$

$$\frac{j}{2\pi f} e^{-j\pi f} - \frac{j}{2\pi f} e^{j\pi f}$$

$$\frac{j}{2\pi f}\left(e^{-j\pi f} - e^{j\pi f}\right)$$

$$\frac{j}{2\pi f}\left(-2j\sin(\pi f)\right) = \frac{\sin(\pi f)}{\pi f} = \operatorname{sinc}(f)$$

Spring 2014 EE 445S Real-Time Digital Signal Processing Laboratory Prof. Evans

Discussion of handout on YouTube: http://www.youtube.com/watch?v=7E8_EBd3xK8

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

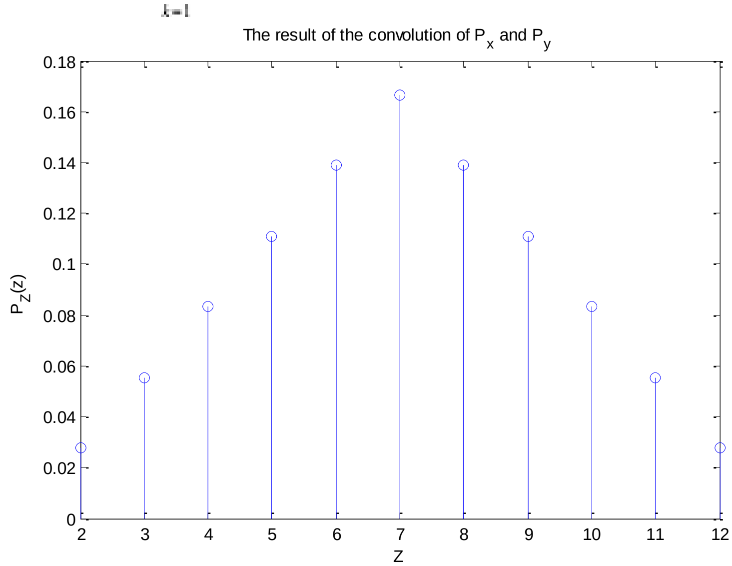
Let Xbe the number generated when the first die is rolled and Y be the number generated when the second die is rolled. Since each outcome is uniformly distributed for each die, $P(X = x) = 1/6$ where $x \in \{1, 2, 3, 4, 5, 6\}$ and $P(Y = y) = 1/6$ where $y \in \{1, 2, 3, 4, 5, 6\}$:

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:

$$p_z(2) = p_x(1)p_y(1) = \frac{1}{36}$$

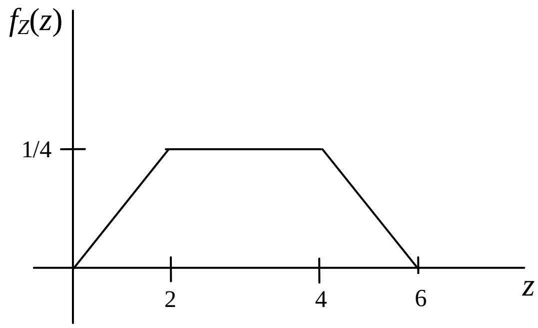
$$p_z(3) = p_x(1)p_y(2) + p_x(2)p_y(1) = \frac{2}{36}$$

$$p_z(4) = p_x(1)p_y(3) + p_x(2)p_y(2) + p_x(3)p_y(1) = \frac{3}{36}$$

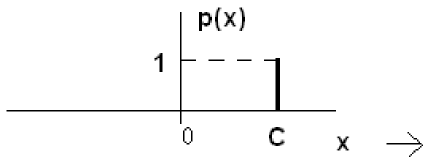
Evaluating the above convolution, we get the same pdf as obtained in the table. The output of the Matlab simulation of the convolution is displayed in the above graph. The *conv* method was used for the convolution. The *stem* method was used for plotting.

(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 $y \in [0, 4]$.

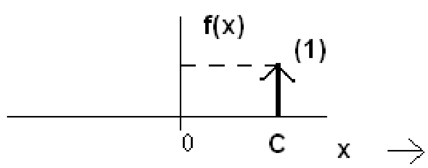
$$f_{X+Y}(z) = f_X(z) * f_Y(z) = \begin{cases} \int_0^z f_X(\lambda) - f_Y(z-\lambda)d\lambda = \frac{z}{8} & 0 \le z \le 2 \\ \int_0^2 f_X(\lambda) - f_Y(z-\lambda)d\lambda = \frac{1}{4} & 2 \le z \le 4 \\ \int_{z-4}^2 f_X(\lambda) - f_Y(z-\lambda)d\lambda = \frac{6}{8} - \frac{z}{8} & 0 \le z \le 2 \end{cases}$$



(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

$$f_{X+Y}(z) = f_X(z) * \delta(z-2) = \begin{cases} \frac{1}{3} & 2 \le z \le 5 \\ 0 & otherwise \end{cases}$$

This follows from the fact that convolution by $\delta(z-2)$ shifts $f_X(z)$ by 2.