The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 ***Solutions 2.0***

Date: March 24, 2021 Course: EE 445S Evans

Name: **Eilish**, **Billie**
Last, First

Please sign your name below to certify that you did not receive any help, directly or indirectly, on this test from another human other your instructor, Prof. Brian L. Evans, and to certify that you did not provide help, directly or indirectly, to another student taking this exam.

(please sign here) ______________________________

- **Take-home exam** is scheduled for Wednesday, Mar. 24, 2021, 10:30am to 11:59pm.
  - The exam will be available on the course Canvas page at 10:30am on Mar. 24, 2021.
  - Your solutions can be on notebook paper, or on the test and your own paper, or whatever. This means that you won't have to print the test to complete the test.
  - Please include this cover page signed by you with your solution and upload your solution as a single PDF file to the course Canvas page by 11:59pm on Mar. 24, 2021.
- **Fully justify your answers**. When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. Sources can include course lecture slides, handouts, homework solutions, etc.
- **Matlab**. No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Internet access.** Yes, you may fully access the Internet when answering exam questions provided that you comply with the other instructions on this page.
- **Academic integrity.** You shall not receive help directly or indirectly on this test from another human except your instructor, Prof. Evans. You shall not provide help, directly or indirectly, to another student taking this exam.
- **Send questions to Prof. Evans.** You may send questions or concerns about this midterm exam during the test to Prof. Evans via Canvas or by e-mail at bevans@ece.utexas.edu.
- **Contact by Prof. Evans.** Prof. Evans might contact all students in the class during the exam through Canvas announcements. Please periodically monitor those announcements.

Problem	*Point Value*	*Your score*	*Topic*
1	24		IIR Filter Design
2	24		Filter Design Tradeoffs
3	28		FIR Filter Bank Design
4	24		Mystery Nonlinearities
Total	100		

Similar to Fall 2014 Midterm 1.3 & Fall 2020 Midterm 1.2

Lecture Slides 6-5 to 6-10 | In-Lecture #2 Assignment

Problem 1.1. *IIR Filter Design*. 24 points.

Consider a second-order discrete-time linear time-invariant (LTI) infinite impulse response (IIR) filter.

A second-order section, also known as a biquad, has <u>two</u> zeros z_0 and z_1 and <u>two</u> poles p_0 and p_1. Its transfer function in the *z*-domain is

HW 0.4 1.1 2.1 3.1 & 3.3

Labs 2 & 3

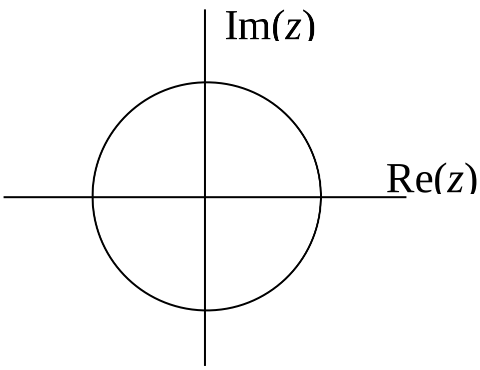
$$H(z) = C\,\frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)}$$

In this problem, **all poles and zeros will be complex-valued but <u>not real-valued</u>.** The imaginary part of the complex number cannot be zero, and the real part of the complex number can be anything.

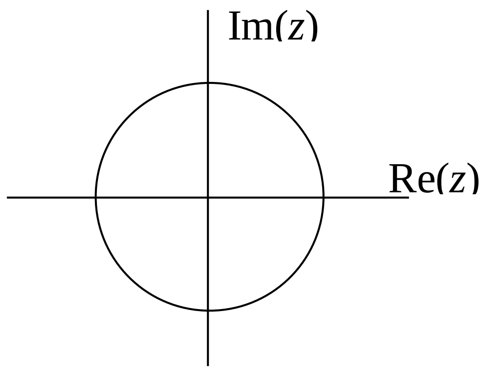
In each part below, design a biquad by placing complex non-real-valued poles and zeros to achieve the indicated frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch) or indicate that no such biquad could be designed. For each filter,

- Please use O to indicate the zero locations and X to indicate pole locations
- Give numeric values for the two poles and two zeros in polar form (i.e. magnitude and phase form)

(a) Lowpass filter



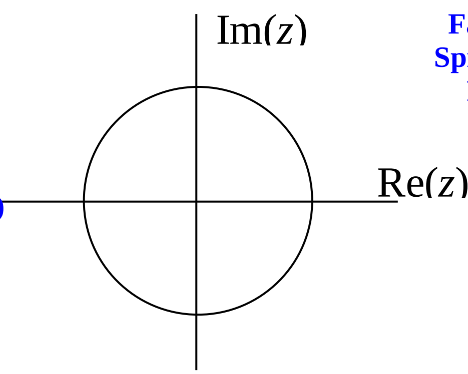
(b) Highpass filter



Please see the next two pages for solutions.

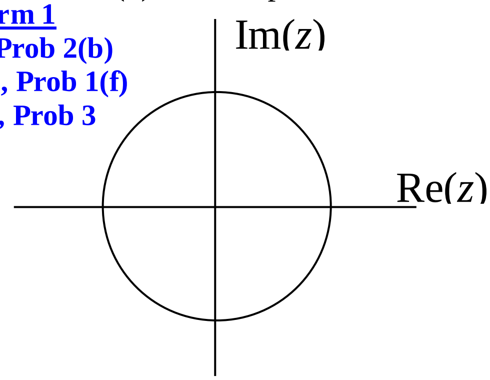
<u>Midterm 1</u>
Fall 2008, Prob 2(a)
Spring 2017, Prob 3
Fall 2017, Prob 3
Fall 2019, Prob 1(f)
Spring 2020, Prob 1(f)
Fall 2020, Prob 2(c)

(c) Bandpass filter

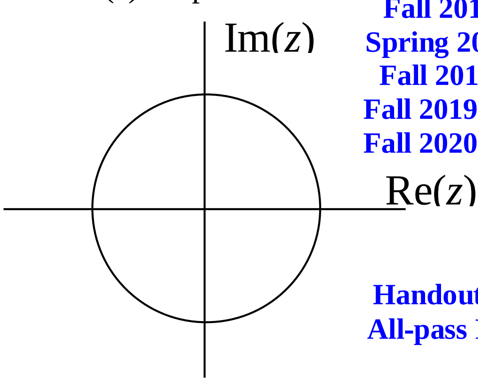


<u>Midterm 1</u>
Fall 2008, Prob 2(b)
Spring 2009, Prob 1(f)
Fall 2016, Prob 3

(d) Bandstop filter



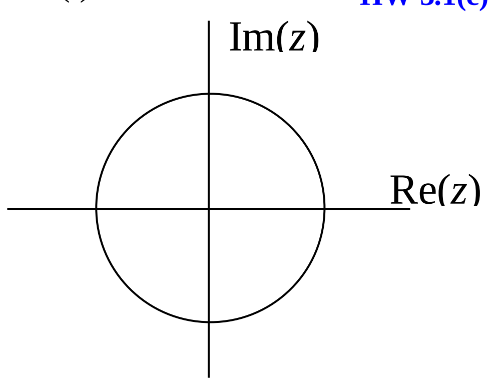
(e) Allpass filter



<u>Midterm 1</u>
Fall 2010 Prob 1
Spring 2012 Prob 1
Fall 2014, Prob 3
Fall 2019, Prob 2(c)
Fall 2020, Prob 2(e)

Handout O on All-pass Filters

(f) Notch filter



HW 3.1(c)

<u>Midterm 1</u>
Spring 2007, Prob 1
Fall 2010, Prob 4(b)
Spring 2011, Prob 2
Fall 2011, Prob 1
Spring 2012, Prob 2
Fall 2016, Prob 1
Spring 2019, Prob 3
Fall 2019, Prob 2(b)
Fall 2020, Prob 2(f)

Lecture slide 6-9

```
%%% Lowpass filter example
zeroAngle = 15*pi/16;
z0 = exp(j*zeroAngle);
z1 = exp(-j*zeroAngle);
feedforwardcoeffs = [1 -(z0+z1) z0*z1];

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

%%% Normalize frequency response
%%% to 1 at center of passband
z = 1; zvec = [1 z^(-1) z^(-2)]';
C = (feedbackcoeffs * zvec) /
(feedforwardcoeffs * zvec);

figure; zplane(C*feedforwardcoeffs,
feedbackcoeffs);
figure; freqz(C* feedforwardcoeffs,
feedbackcoeffs);
```

Per lecture slide 6-8,

- **Angle of pole near unit circle indicates frequency at which peak occurs in magnitude response**
- **Angle of zero on or near unit circle indicates frequency at which valley occurs in mag. response.**

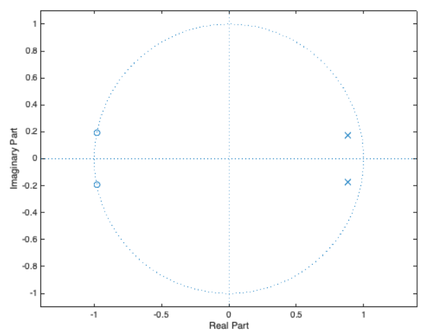
Although not explicitly requested, we will choose poles that are *conjugate symmetric* to give *real-valued* feedback coefficients, and zeros that are *conjugate symmetric* to give *real-valued* feedforward coefficients, as we've been doing throughout the semester to reduce run-time complexity. Using complex coefficients in the difference equation to implement the filter requires 4x complexity for multiplication-addition operations and 2x storage for input $x[n]$ and output $y[n]$:

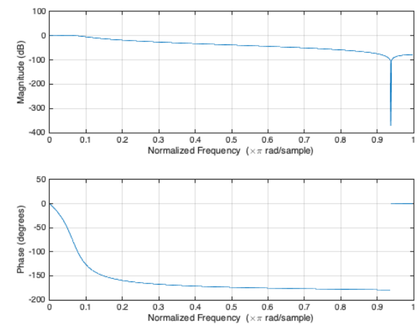
$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

Lecture slide 6-7

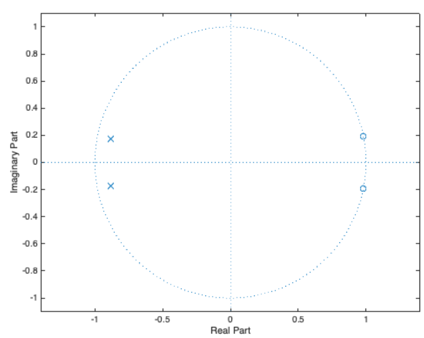
For lowpass, highpass, bandpass, and bandstop filters, we'll place each pole at the center of each passband as per lecture slide 6-9 (left pole-zero plot) and demos in lecture slide 6-10.

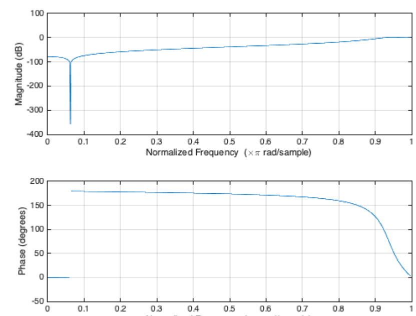
(a) Lowpass: zeros at
$\boldsymbol{z_0 = e^{j\theta_1}}$ **and**
$\boldsymbol{z_1 = e^{-j\theta_1}}$
with $\boldsymbol{\theta_1 = \frac{15}{16}\pi}$ **and poles at**
$\boldsymbol{p_0 = 0.9\, e^{j\theta_2}}$ **and**
$\boldsymbol{p_1 = 0.9\, e^{-j\theta_2}}$
with $\boldsymbol{\theta_2 = \frac{1}{16}\pi}$



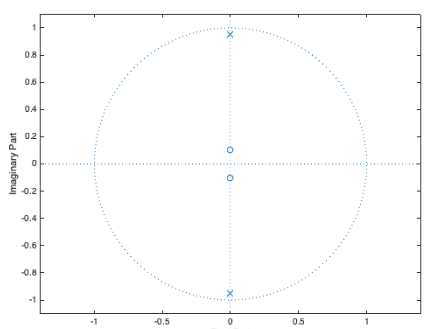


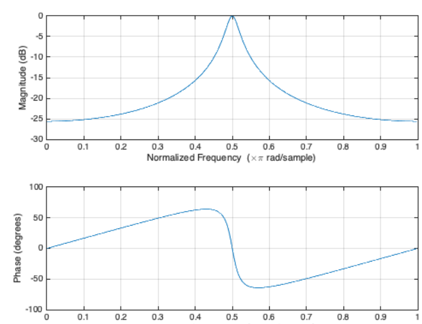
(b) Highpass: zeros at
$\boldsymbol{z_0 = e^{j\theta_2}}$ **and**
$\boldsymbol{z_1 = e^{-j\theta_2}}$
with $\boldsymbol{\theta_2 = \frac{1}{16}\pi}$ **and poles at**
$\boldsymbol{p_0 = 0.9\, e^{j\theta_1}}$ **and**
$\boldsymbol{p_1 = 0.9\, e^{-j\theta_1}}$
with $\boldsymbol{\theta_1 = \frac{15}{16}\pi}$



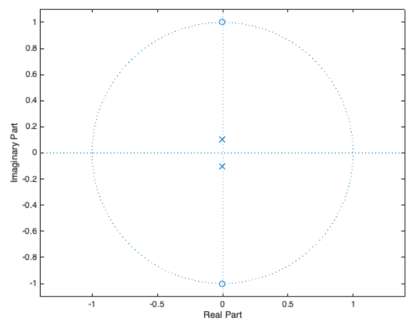


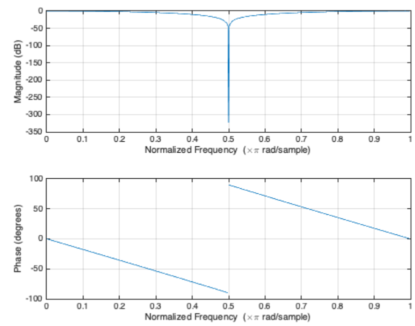
(c) Bandpass: zeros at
$\boldsymbol{z_0 = 0.1\, e^{j\theta_3}}$ **and**
$\boldsymbol{z_1 = 0.1\, e^{-j\theta_3}}$
with $\boldsymbol{\theta_3 = \frac{\pi}{2}}$ **and poles at**
$\boldsymbol{p_0 = 0.95\, e^{j\theta_3}}$ **and**
$\boldsymbol{p_1 = 0.95\, e^{-j\theta_3}}$





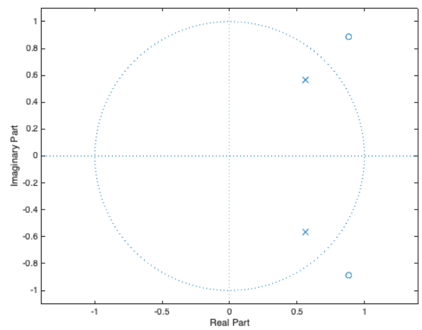
(d) Bandpass: zeros at
$z_0 = e^{j\theta_3}$ and
$z_1 = e^{-j\theta_3}$
with $\theta_3 = \frac{\pi}{2}$ and
poles at
$p_0 = 0.1\, e^{j\theta_3}$ and
$p_1 = 0.1\, e^{-j\theta_3}$

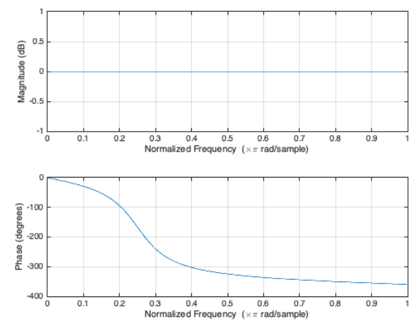




For allpass filters, we'll follow [lecture slide 6-9] (right pole-zero plot) and the [all-pass filter handout] to place pole-zero pairs at the same angle and reciprocal magnitudes.

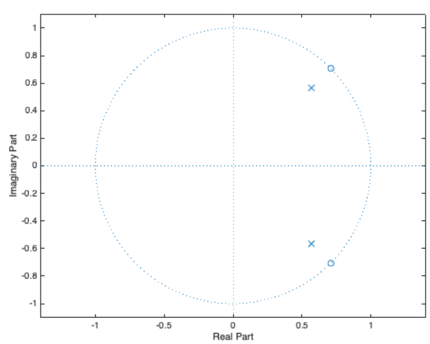
(e) Allpass: zeros at
$z_0 = 1.25\, e^{j\theta_4}$ and
$z_1 = 1.25\, e^{-j\theta_4}$
with $\theta_4 = \frac{\pi}{4}$
poles at
$p_0 = 0.8\, e^{j\theta_4}$ and
$p_1 = 0.8\, e^{-j\theta_4}$

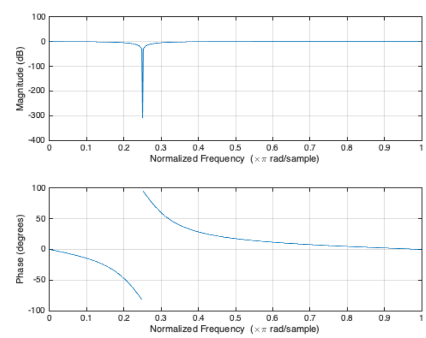




For notch filters, we'll follow [lecture slide 6-9] (middle pole-zero plot), demos in [lecture slide 6-10], and homework 3.1 on designing a notch filter to remove narrowband interference.

(f) Notch: zeros at
$z_0 = e^{j\theta_4}$ and
$z_1 = e^{-j\theta_4}$
with $\theta_4 = \frac{\pi}{4}$
poles at
$p_0 = 0.8\, e^{j\theta_4}$ and
$p_1 = 0.8\, e^{-j\theta_4}$





For reference, and not asked, we obtain the difference equation with feedback coefficiens a_1 and a_2 and feedforward coefficients b_0, b_1, and b_2 from the transfer function:

$$H(z) = C\frac{(z-z_0)(z-z_1)}{(z-p_0)(z-p_1)} = C\frac{z^2-(z_0+z_1)z+z_0z_1}{z^2-(p_0+p_1)z+p_0p_1} = C\frac{1-(z_0+z_1)z^{-1}+z_0z_1z^{-2}}{1-(p_0+p_1)z^{-1}+p_0p_1z^{-2}}$$

$$H(z) = C\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} = \frac{Y(z)}{X(z)}$$

$$(1+a_1z^{-1}+a_2z^{-2})\,Y(z) = (b_0+b_1z^{-1}+b_2z^{-2})\,X(z)$$

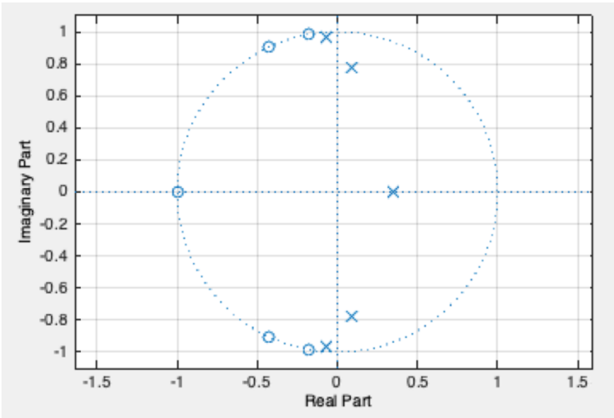
$$y[n] = a_1y[n-1] + a_2y[n-2] + b_0x[n] + b_1x[n-1] + b_2x[n-2]$$

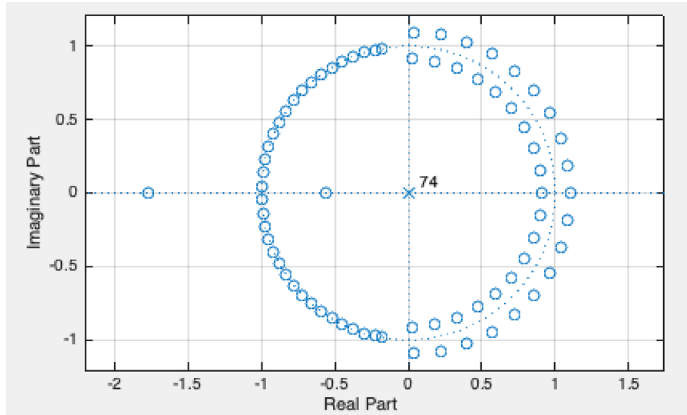
Selectivity	**Example Application(s)**	**Selectivity**	**Example Application**
Lowpass	Anti-aliasing filter before sampler in ADC; demodulation	Bandstop	Alleviate ringing of the ear symptoms (HW 2.3 & 3.3)
Highpass	Enhance edges/texture in images	Allpass	Phase correction after ADC
Bandpass	Reject out-of-band interference and noise (HW 3.1); modulation	Notch	Remove in-band narrowband interference (HW 3.1)

Midterm: 1.2 Sp10, 1.4(b) F15, 1.3 Sp16, 1.3 F18, 1.4 F18 & 1.3 F20	
Lecture 5 & 6 Slides	In-Lecture #2 Assignment
HW 0.4 1.1 2.1 2.2 2.3 3.1 3.2 & 3.3	Labs 2 & 3

Problem 1.2 *Filter Design Tradeoffs.* 24 points.

Both discrete-time linear time-invariant (LTI) filters below meet the same filter design specifications based on the magnitude response for an audio application.





Design #1
Number of complex/real poles: 5
Number of complex/real zeros: 5
Passband behavior: Rippling
Stopband behavior: Rippling
Design method: Elliptic
Maximum Group Delay: 33 samples

Design #2
Number of complex/real poles: 74
Number of complex/real zeros: 74
Passband behavior: Rippling
Stopband behavior: Rippling
Design method: Parks-McClellan
Maximum Group Delay: 37 samples

Please **answer** the following questions about the filter designs with **justification**. *3 points each.*

	Design #1	**Design #2**
(a) Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) Filter?	**IIR filter because it has at least one non-trivial (non-zero) pole and was designed using the elliptic method.**	**FIR filter because all poles are zero and it was designed using Parks-McClellan method.**
(b) Filter Order	**#poles = 5 (Homework 3.3)**	**#zeros = 74 (Homework 2.3)**
(c) Bounded-Input Bounded Output Stable?	**Yes, all poles inside unit circle (Lecture Slides 6-13 & 6-14; Homework 2.1 & 3.3)**	**Yes, all FIR filters are BIBO stable (Lecture Slides 6-13 & 6-14 and BIBO Stability Handout)**
(d) Approximate range of discrete-time frequencies in the passband in rad/sample	**Pole angles from 0 to 0.52π approx. and zero angles 0.56π approx. to π (Lecture Slide 6-8)**	**Pole angles from 0 to 0.5π and zero angles 0.58π approx. to π (Lecture Slide 6-8)**
(e) Frequency selectivity (lowpass, highpass, bandpass, bandstop, allpass or notch)	**Lowpass due to (d)**	**Lowpass due to (d)**
(f) Number of multiplication operations (use cascade of biquads structure if IIR filter)	**3 + 2 biquads x 5 = 13 mults Fall 2018, Problem 1.4(a) *-OR-*** ***3 + 2 biquads x 4 = 11 mults each pair of zeros on unit circle give 2 of 3 feedforward coefficients with value 1* (Homework 3.3(b)(c))**	**Number of FIR coefficients is filter order + 1 = 75.** **75 mults**
(g) Amount of storage (use cascade of biquads structure if IIR filter)	***Data*: 3 + 2 biquads x 6 = 15 words** ***Coeffs*: 3 + 2 biquads x 5 = 13 words Fall 2018, Problem 1.4(a) *-OR-* *3 + 2 biquads x 4 = 11 words per (f)***	***Data*: 75 words** ***Coeffs*: 75 words**
(h) Give an advantage of each design, and indicate which design you would choose.	**5-6x lower complexity** **Mildly lower maximum group delay**	**Linear phase over all freq.** **BIBO stability regardless of implementation**

See work on next page

Include storage of input & output samples

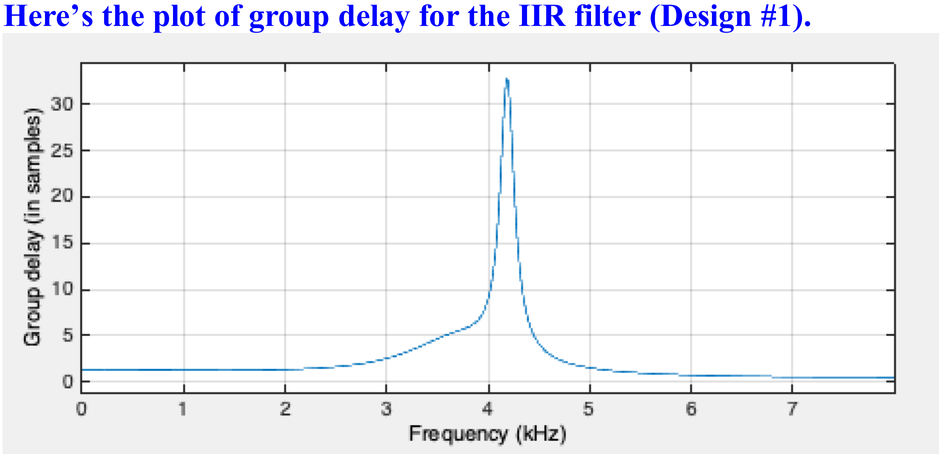
(c) Poles are separated from the zeros in angles, and the poles are close to the unit circle. The pole angles indicate the passband frequencies, and the zero locations on the unit circle indicate the stopband frequencies. I estimated the angle of the pole with greatest positive angle by zooming into the pole-zero plot in the PDF file, measuring their Cartesian in the complex plane with a ruler, and computing the phase. I followed the same approach for the zero with smallest positive angle.

The lowpass filter design specification was to attenuate frequencies above the highest note on an 88-key piano, which is at 4186 Hz (C8). The next note above 4186 Hz is 4434 Hz which will be in the stopband.

Filter specifications
- **Fpass = 4186 Hz**
- **Fstop = 4434 Hz**
- **Apass = 1 dB**
- **Astop = 30 dB**
- **Fs = 16000 Hz**

I chose a low stopband attentuation so the FIR filter order wouldn't be so large which in turn would allow one to see all of the zeros on the pole-zero plot. In audio, 80 dB is a common value for stopband attenuation. We'll learn in Lecture 8 on Quantization that 80 dB is equivalent to 13 bits, so 80 dB of stopband attention means that high frequencies will lose the upper 13 bits per sample in strength. The conversion is $SNR_{dB} = 2 + 6B$ where B is the number of bits. When using 80 dB for the stopband attention, the filter orders increase to 11 for the elliptic IIR design method and 164 for the Parks-McClellan FIR design method.

Here's the plot of group delay for the IIR filter (Design #1).



The lowest note on an 88-key piano is 27.5 Hz (A0). If one wanted to also remove frequencies below the lowest note, then one could either add a DC notch filter in cascade with the lowpass filter above, or design a bandpass filter from scratch.

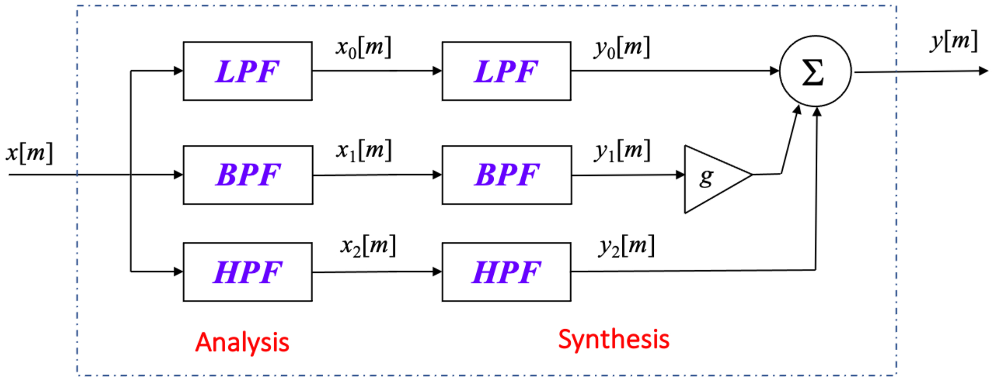
Midterm 1: 1.3 Sp11, 1.3 Sp14, 1.2 Sp15, 1.3 Sp16 & 1.3 F20	
Lecture Slides 6-5 to 6-13	In-Lecture #2 Assignment
HW 1.1 2.1 2.2 2.3 3.1 & 3.2	Lab 3

Problem 1.3 *FIR Filter Bank Design.* 28 points.

A bank of analysis filters divides a signal $x[m]$ into frequency bands for subsequent processing.

A bank of synthesis filters combines the analysis frequency bands into a single signal $y[m]$.

The analysis-synthesis filter bank below has three linear time-invariant (LTI) filters in each bank:



The LPF, BPF, and HPF filters are finite impulse response (FIR) filters with three coefficients each.

Both lowpass filters (LPFs) have impulse response of $h_0[m] = \delta[m] + \delta[m-1] + \delta[m-2]$. The LPF coefficients are [1, 1, 1].

(a) Both bandpass filters (BPFs) are designed by modulating the LPF to shift its frequency response by π/2 to the right and left: $h_1[m] = \cos\left(\frac{\pi}{2} m\right) \; h_0[m]$. Give the coefficients for $h_1[m]$. Does the BPF have linear phase? Why or why not? *6 points.*

[1, 0, -1]. Yes, BPF has (generalized) linear phase due to odd symmetry about midpoint of impulse response. Odd symmetry: $h[n] = -h[N-1-n]$ for $n = 0, 1, \ldots, N-1$. Here, $h[0]$ = -$h[2]$ and $h[1]$ = -$h[1]$ = 0. (Modulation is used during the offline filter design procedure.)

(b) Both highpass filters (HPFs) are designed by modulating the LPF to shift its frequency response by π to the right and left: $h_2[m] = \cos(\pi\, m) \; h_0[m]$. Give the coefficients for $h_2[m]$. Does the HPF have linear phase? Why or why not? *6 points.*

[1, -1, 1]. Yes, HPF has linear phase due to even symmetry about midpoint of impulse response. Even symmetry: $h[n] = h[N-1-n]$ for $n = 0, 1, \ldots, N-1$. Here, $h[0]$ = $h[2]$ and $h[1]$ = $h[1]$. (Modulation is used during the filter design procedure, not implementation.)

(c) Compute the impulse response $h[m]$ for the overall system with input $x[m]$ and output $y[m]$. The impulse response will include the real-valued gain g. **Hint:** $y[m] = y_0[m] + g\, y_1[m] + y_2[m]$. *9 points.*

Overall impulse response: $h[m] = h_0[m] * h_0[m] + g\, h_1[m] * h_1[m] + h_2[m] * h_2[m]$

$h_0[m] * h_0[m]$:	conv([1, 1, 1], [1, 1, 1])	=	[1,	2,	3,	2,	1]
$g\, h_1[m] * h_1[m]$:	*g* conv([1, 0, -1], [1, 0, -1])	=	[*g*,	0,	-2*g*,	0,	*g*]
$h_2[m] * h_2[m]$:	conv([1, -1, 1], [1, -1, 1])	=	[1,	-2,	3,	-2,	1]
Add all term:			[*g*+2,	0,	6-2*g*,	0,	*g*+2]

(d) Does the overall system have linear phase for all possible values of g? Why or why not? *3 points.*

Yes, because its impulse response is even symmetric about its midpoint for all values of *g*.

(e) Compute the value of g that causes the overall system to act like an ideal delay with gain C, i.e. $y[m] = C\, x[m - m_0]$. Please give the values of C and m_0. *4 points.*

When *g* = -2, the overall impulse response becomes [0, 0, 10, 0, 0]. This equivalent to an ideal delay of m_0 = 2 samples with a gain of *C* = 10.

Midterm 1: 1.4 Sp11, 1.3 Sp15, 1.5 Sp19, 1.4 F19 & 1.4 Sp20
Lecture 4 | Handout Common Signals in Matlab
HW 1.2 1.3 & 2.2 | In-Lecture #1 Assignment

Problem 1.4. *Mystery Nonlinearities.* 24 points.

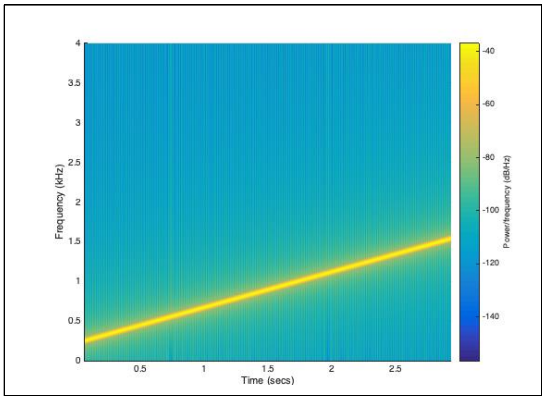
You're trying to determine input-output relationships for discrete-time pointwise nonlinear systems.

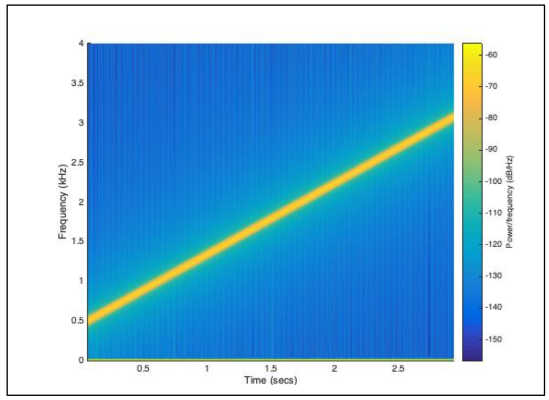
For discrete-time pointwise systems, output at discrete-time *n* only depends on input at discrete-time *n*.

You input a chirp signal and look at the resulting output signal to figure out what the system is doing.

For the analysis, you decide to use the chirp signal from in-lecture assignment #1, which starts around 240 Hz and ends around 1520 Hz, and uses a sampling rate of 8000 Hz.

(a) Give a formula for output $y[n]$ in terms of input $x[n]$ by looking at the spectrogram for a chirp input signal (left) and the spectrogram of the output signal (right). *12 points*.





The spectrogram of the output shows a strong DC component at all times, and another component shows a linear increase from around 480 Hz to around 3040 Hz over time.

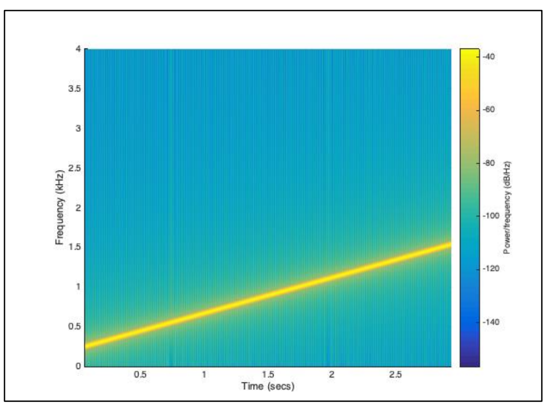
A spectrogram is a plot of the frequency content (y-axis) vs. time (x-axis) of a signal.

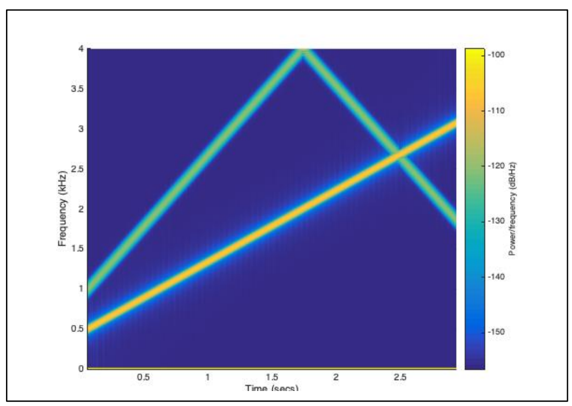
At any point in time, the chirp input signal has one principal frequency and the output signal contains double the principal frequency plus a DC component.

$$\cos^2(\omega_0 n) = \frac{1}{2} + \frac{1}{2}\cos(2\,\omega_0 n)$$

This is a squaring block: $y[n] = x^2[n]$.

(b) Give a formula for output $y[n]$ given the input $x[n]$ by looking at the spectrogram for a chirp input signal (left) and the spectrogram of the output signal (right). *12 points*.





The spectrogram of the output shows a strong DC component at all times; a second component that shows a linear increase from around 480 Hz to around 3040 Hz; and a third component that starts around 960 Hz, linearly increases to 4000 Hz, and then linearly decreases to around 1880 Hz.

See next page.

At any point in time, the input chirp signal has one principal frequency, and the output contains quadruple and double the input principal frequency as well as a DC component from 0 to 1.8 s. The quadruple component is twice the doubled input principal frequency, which would be consistent with a second squaring block in cascade with the first: $y[n] = x^4[n]$.

For input $\cos(\omega_0 n)$, the first squaring block would give as output

$$\frac{1}{2} + \frac{1}{2}\cos(2\,\omega_0 n)$$

The second squaring block would give

$$\left(\frac{1}{2} + \frac{1}{2}\cos(2\,\omega_0 n)\right)^2 = \frac{1}{4} + \frac{1}{2}\cos(2\,\omega_0 n) + \frac{1}{4}\cos^2(2\,\omega_0 n)$$

where

$$\cos^2(2\,\omega_0 n) = \frac{1}{2} + \frac{1}{2}\cos(4\,\omega_0 n)$$

and hence

$$\left(\frac{1}{2} + \frac{1}{2}\cos(2\,\omega_0 n)\right)^2 = \frac{3}{8} + \frac{1}{2}\cos(2\,\omega_0 n) + \frac{1}{8}\cos(4\,\omega_0 n)$$

Among the three terms, the $\cos(2\,\omega_0 n)$ term is strongest, followed by the DC term of 3/8, and finally by the $\cos(4\,\omega_0 n)$ term. This is reflected in the color ascribed to these three terms according to the color map show at the right of the spectrogram plot.

We can track the $\cos(4\,\omega_0 n)$ component in green from 0 to 3s to see what happens. Let f_0 be the principal input frequency at any point in time. From 0 to 1.8s, $4f_0 < ½f_s$ which is 4000 Hz, or equivalently $4\omega_0 < \pi$ rad/sample, and we see a linear increase. From 1.8s to 3s, the $\cos(4\,\omega_0 n)$ term aliases because $4f_0 > ½f_s$ which accounts for the downward linear trajectory.

Here's the Matlab code to generate the above plots:

```
fs = 8000;                  % Samples/s
n = 0 : 3*fs;               % There are fs samples in 1s
f0 = 220;                   % A3 (A note at 220 Hz in third octave on Western scale)
w0 = 2*pi*f0/fs;
x = 0.1*cos(w0*n + pi*(0.7*10^(-5))*(n.^2));

blockSize = 1024;
overlap = 1023;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');

y = x .^ 2;

figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');

y = x .^ 4;

figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```