The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #2 ***Version 3.0***

Date: November 13, 2025 Course: EE 313 Evans

Name: **Solutions**
Last, First

- **Exam duration**. The exam is scheduled to last 75 minutes.
- **Materials allowed**. You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks**. Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **No AI tools allowed**. As mentioned on the course syllabus, you may not use GPT or other AI tools during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers**. When justifying your answers, reference your source and page number as well as quote the content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab**. No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test**. All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

Problem	*Point Value*	*Your score*	*Topic*
1	27		System Properties
2	25		FIR Filter Analysis
3	24		System Identification
4	24		Upsampling, Downsampling & Filtering
Total	100		

SPFirst Sec. 5-4, 5-5, 8-2 & 8-4.2

HW 5.2 | Lecture Slides 8-3 to 8-6, 8-8 & 12-11 to 12-15

Problem 2.1. *System Properties.* 27 points. Midterms: F18 Prob 2.5, F21 Prob 2.1, F23 Prob. 2.1 & F24 Prob 2.1

Each discrete-time system has input $x[n]$ and output $y[n]$, and $x[n]$ and $y[n]$ might be complex-valued. Determine if each system is linear or nonlinear, time-invariant or time-varying, and causal or not causal.

You must either prove that the system property holds in the case of linearity, time-invariance, or causality, or provide a counter-example that the property does not hold. Providing an answer without any justification will earn 0 points.

Part	*System Name*	*System Formula*	*Linear?*	*Time-Invariant?*	*Causality?*
(a)	Averaging **Finite** Impulse Response Filter	$y[n] = x[n] + x[n+1]$ for $-\infty < n < \infty$	**YES**	**YES**	**NO**
(b)	Averaging **Infinite** Impulse Response Filter	$y[n] = 0.9\, y[n-1] + 0.1\, x[n]$ for $n \geq 0$	**NO**	**NO**	**YES**
(c)	Phase **Modulation**	$y[n] = \cos(\hat{\omega}_0\, n + x[n])$ where $\hat{\omega}_0$ is a constant for $-\infty < n < \infty$	**NO**	**NO**	**YES**

Linearity. **First, we'll apply the all-zero input test— input $x[n] = 0$ for all observed time n and if the output is not zero for all observed time n, then the system is not linear. Otherwise, we'll have to apply the definitions for homogeneity and additivity. All-zero input test is a special case of homogeneity $a\, x[n] \rightarrow a\, y[n]$ when the constant $a = 0$.**

Causality: **Current output only depends on current input, previous input and/or previous output.**

(a) Averaging Finite Impulse Response Filter: $y[n] = x[n] + x[n+1]$ for $-\infty < n < \infty$. *9 points.*

Linearity: **Passes all-zero input test. No initial conditions due to observing $-\infty < n < \infty$. YES,**

- ***Homogeneity***: **Input $a\, x[n]$. Output is** $y_{scaled}[n] = (a\, x[n]) + (a\, x[n])_{n \rightarrow n+1} = a\, x[n] + a\, x[n+1] = a\, y[n]$**? YES.** HW 5.2
- ***Additivity***. **Input $x_1[n] + x_2[n]$. Output is** $y_{additive}[n] = (x_1[n] + x_2[n]) + (x_1[n] + x_2[n])_{n \rightarrow n+1} = (x_1[n] + x_2[n]) + (x_1[n+1] + x_2[n+1]) = x_1[n] + x_1[n+1] + x_2[n] + x_2[n+1] = y_1[n] + y_2[n]$**? YES.**

T-I. **Input $x[n - n_0]$. Output $y_{shifted}[n] = x[n - n_0] + x[n - n_0 + 1] = y[n - n_0]$? YES**

Causality: **Current output depends on $x[n+1]$, which is the input one sample in the future. NO.**

(b) Averaging Infinite Impulse Response Filter: $y[n] = 0.9\, y[n-1] + 0.1\, x[n]$ for $n \geq 0$. Analyze the impact of the initial condition(s) on the system properties. *9 points.*

Linearity: **All-zero input test: $y[n] = 0.9\, y[n-1]$ for $n \geq 0$. First output is $y[0] = 0.9\, y[-1]$. YES if $y[-1] = 0$. Since $y[-1]$ is unspecified, NO.**

T-I. **The initial condition does not shift when the input shifts. Since $y[-1]$ is unspecified, NO.**

Causality: **Current output depends on previous output $y[n-1]$ and current input $x[n]$. YES.**

(c) Phase Modulation: $y[n] = \cos(\hat{\omega}_0\, n + x[n])$ where $\hat{\omega}_0$ is constant and $-\infty < n < \infty$ *9 points.*

Linearity: **All-zero input test: $y[n] = \cos(\hat{\omega}_0\, n)$. Output is not zero for all time. NO.**

Time-Invariance: **With input $x[n - n_0]$, $y_{shifted}[n] = \cos(\hat{\omega}_0\, n + x[n - n_0])$. For all values of of n_0, does $y_{shifted}[n] = y[n - n_0]$? Here, $y[n - n_0] = \cos(\hat{\omega}_0\, (n - n_0) + x[n - n_0])$. NO.**

Causality: **Current output only depends on current input. YES.**

Problem 2.2 *FIR Filter Analysis*. 25 points.

Consider the following causal finite impulse response (FIR) linear time-invariant (LTI) filter with input $x[n]$ and output $y[n]$ described by

$$y[n] = x[n] - a\, x[n-1]$$

for $n \geq 0$. Here a is a real-valued constant where $a \neq 0$.

(a) Give a formula for the impulse response $h[n]$. Plot $h[n]$. *3 points*.

The impulse response is the system response to an impulse, $\delta[n]$.
Let $x[n] = \delta[n]$ and the impulse response is $h[n] = \delta[n] - a\,\delta[n-1]$.

(b) What are the initial condition(s)? What are their value(s)? *3 points*.

The system must be at rest as a necessary condition for LTI to hold; i.e, the initial condition(s) must be zero. We can determine the initial conditions by starting at $n = 0$:
$y[0] = x[0] - a\,x[-1]$ which depends on an initial condition $x[-1]$. So, $x[-1] = 0$.
$y[1] = x[1] - a\,x[0]$ which does not depend on any initial conditions.

(c) Compute the transfer function $H(z)$ in the z-domain and give the region of convergence. *3 points*.

Take the z-transform of both sides of $y[n] = x[n] - a\,x[n-1]$ with $x[-1] = 0$:

$$Y(z) = X(z) - a\,z^{-1}\,X(z)$$

Divide each side by $X(z)$ gives

$$H(z) = \frac{Y(z)}{X(z)} = 1 - a\,z^{-1} = \frac{z - a}{z} \quad \text{for } z \neq 0$$

We have to exclude $z = 0$ to avoid division by zero in the expression z^{-1}.

(d) Give a formula for the discrete-time frequency response of the FIR filter. *4 points*.

Because the region of convergence $z \neq 0$ includes the unit circle, it's valid to substitute $z = e^{j\,\hat{\omega}}$ to convert the transfer function in the z-domain to a frequencey response:

$$H\left(e^{j\,\hat{\omega}}\right) = 1 - a\,e^{-j\,\hat{\omega}} = \frac{e^{j\,\hat{\omega}} - a}{e^{j\,\hat{\omega}}}$$

(e) Does the FIR filter have linear phase? If yes, then give the conditions on the coefficient a for the filter to have linear phase. If no, then show that the coefficients cannot meet the conditions for linear phase. *6 points*

An *N*-point FIR filter has linear phase when its impulse response is either even symmetric or odd symmetric about its midpoint. The midpoint is at $n = \frac{N-1}{2} = \frac{1}{2}$ sample. Here, $N = 2$.
***Even symmetry*. $h[n] = h[N-1-n]$ for $n = 0, 1, \ldots, N-1$. Here, $h[0] = h[1]$ or $a = -1$.**
***Odd symmetry*. $h[n] = -h[N-1-n]$ for $n = 0, 1, \ldots, N-1$. Here, $h[0] = -h[1]$ or $a = 1$.**

(f) What are all of the possible frequency selectivities that the FIR filter could provide: lowpass, highpass, bandpass, bandstop, or allpass? *6 points*

***Answer #1*: Transfer function has zero at $z = a$ and pole at $z = 0$. A pole at origin does not affect the magnitude response. When $a \approx 1$, low frequencies attenuated (highpass). Vice-versa when $a \approx -1$ (lowpass). As $a \to \infty$ or $a \to -\infty$ or $a \to 0$, filter is allpass.**

***Answer #2:* When *a* = *1*, $y[n] = x[n] - x[n-1]$ is a first-order difference (highpass) filter. When *a* = *-1*, $y[n] = x[n] + x[n-1]$ is a two-point averaging (lowpass) filter.**

***Answer #3:* The magnitude response is lowpass when $a \approx -1$ and highpass when $a \approx 1$:**

$$\left|H\left(e^{j\,\hat{\omega}}\right)\right| = \left|\frac{e^{j\,\hat{\omega}} - a}{e^{j\,\hat{\omega}}}\right| = \frac{\left|e^{j\,\hat{\omega}} - a\right|}{\left|e^{j\,\hat{\omega}}\right|} = \left|e^{j\,\hat{\omega}} - a\right|$$

SPFirst Sec.3-7 to 3-8 and 4-1 to 4-5 and 5-5 to 5-8 and 6-1 to 6-6 | *HW 4.1, 4.2, 4.3, 5.1, 5.4, 6.1, 7.2*

Lectures Slides 4-4 to 4-12 and 5-9 to 5-13 and 6-1 to 6-7 and 8-6 to 8-9 and 9-3 to 9-10

Mini-Project #2 | *Tuneup #7*

Problem 2.3 *System Identification.* 24 points.

You're trying to identify unknown discrete-time systems.

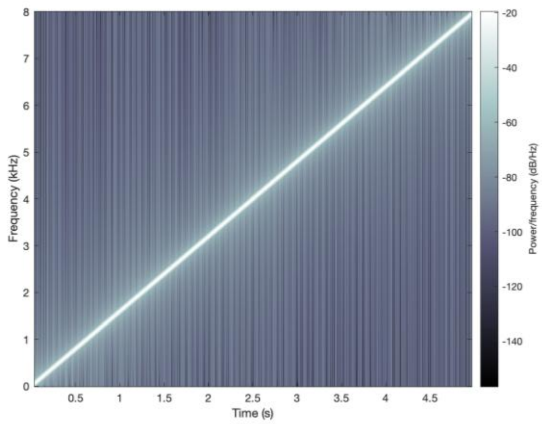
You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 8000 Hz over 0 to 5s

$$x(t) = \cos(2\pi f_1 t + 2\pi \mu t^2)$$

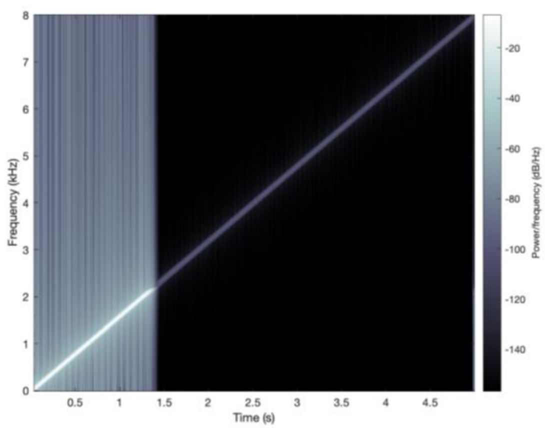
where $f_1 = 0$ Hz, $f_2 = 8000$ Hz, and $\mu = \frac{f_2 - f_1}{2\, t_{\max}} = \frac{8000 \text{ Hz}}{10\ s} = 800 \text{ Hz}^2$. Sampling rate f_s is 16000 Hz.

Spectrogram for chirp signal $x[n]$



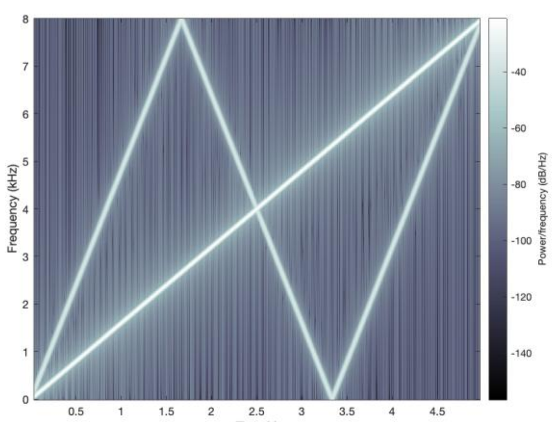
In each part below, identify the unknown system as one of the following **with justification**:

1. filter – give the frequency selectivity (lowpass, highpass, bandpass, bandstop) as well as the passband and stopband frequencies
2. pointwise nonlinearity – give the integer exponent k to produce output $y[n] = x^k[n]$
3. amplitude modulation – give the amplitude modulation frequency f_0 to produce output $y[n] = \cos(\omega_0\, n)\, x[n]$ where $\omega_0 = 2\pi\, f_0 \,/\, f_s$.



(a) When the chirp signal $x[n]$ is input, a system gives the output signal $y[n]$ whose spectrogram is plotted on the left. *12 points*.

No new frequencies are being created— hence, this is an LTI system. The output shows that the LTI system passes frequencies from 0 to ~2 kHz and attenuates higher frequencies. Lowpass filter with passband frequencies 0-2 kHz and stopband frequencies 2-8 kHz.



(b) When the chirp signal $x[n]$ is input, another system gives the output signal $y[n]$ whose spectrogram is plotted on the left. *12 points*

From 0s to 1s, output has frequencies not previously seen on the input. The system cannot be LTI.

The output contains the input chirp signal plus another frequency component that appears like an italic letter *N*. From 0s to 1.67s, this additional component has 3x the slope of the input chirp. Output is $y[n] = x^3[n]$, e.g.

$$y(t) = \cos^3(f_1\, t) = \frac{1}{4}\cos(3\, f_1\, t) + \frac{3}{4}\cos(f_1\, t)$$

Aliasing occurs from 1.67s to 3.33s and from 3.33s to 5s.

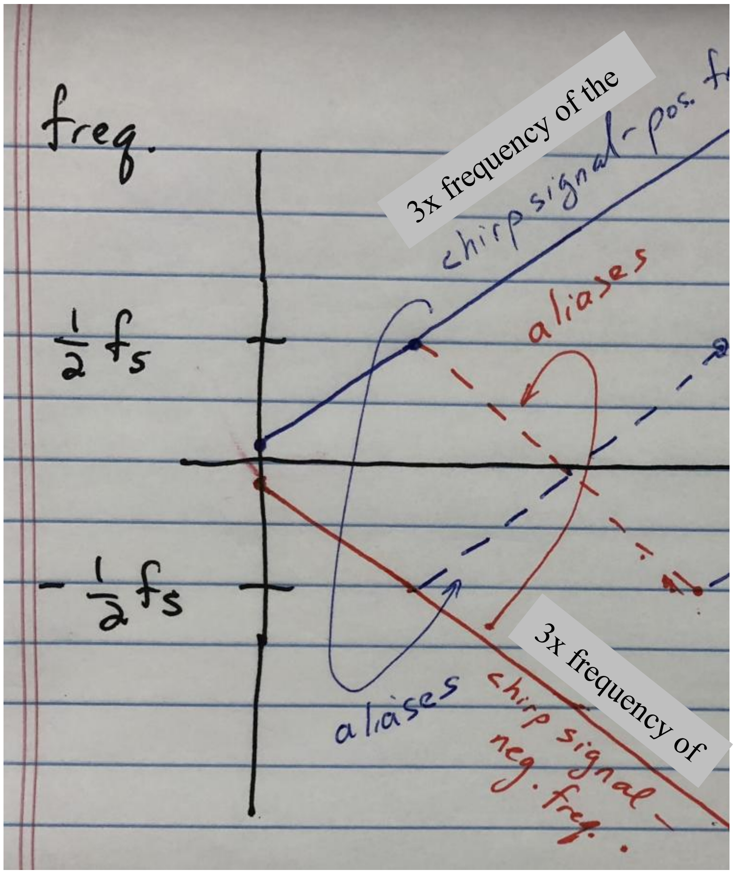
Hint: On the right of each spectrogram plot is an intensity map to decibels (dB). All values are negative.

Part (b) is related to Fall 2017 Midterm Problem 1.4(c).

In our case, $y[n] = x^3[n]$. For every chirp frequency f_1 on the input

$$y(t) = \cos^3(f_1\, t) = \frac{1}{4}\cos(3\, f_1\, t) + \frac{3}{4}\cos(f_1\, t)$$

The spectrogram below tracks the $\cos(3\, f_1\, t)$ component.



Spectrogram is showing positive and negative frequencies.

Sampling theorem: $f_s > 2\, f_{max}$ which means $f_{max} < \frac{1}{2}\, f_s$.

Frequencies at or above $\frac{1}{2}\, f_s$ will alias and frequencies at or below $-\frac{1}{2}\, f_s$ will alias.

```matlab
%% Matlab code to generate the spectrograms for Problem 2.3
fs = 16000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% (a) Lowpass Filter
fnyquist = fs/2;
fpass = 2000;
fstop = 2200;
ctfrequencies = [0 fpass fstop fnyquist];
idealAmplitudes = [1 1 0 0];
pmfrequencies = ctfrequencies / fnyquist;
filterOrder = 400;
h = firpm( filterOrder, pmfrequencies, idealAmplitudes );
h = h / sum(h .^ 2);

y = conv(x, h);


%%% Spectrogram parameters
blockSize = 1024;
overlap = 1023;

%%% Plot spectrogram of input signal
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
colormap bone;

%%% Plot spectrogram of output signal
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
colormap bone;

%% (b) Cubic nonlinearity
y = x .^ 3;

%%% Spectrogram parameters
blockSize = 1024;
overlap = 1023;

%%% Plot spectrogram of input signal
figure;
spectrogram(x, blockSize, overlap, blockSize, fs, 'yaxis');
colormap bone;

%%% Plot spectrogram of output signal
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
colormap bone;
```

SPFirst Sec.4-1 to 4-5 and 5-5 to 5-8 and 6-1 to 6-6 | HW 4.1, 4.2, 5.1, 5.4, 6.1, 7.2

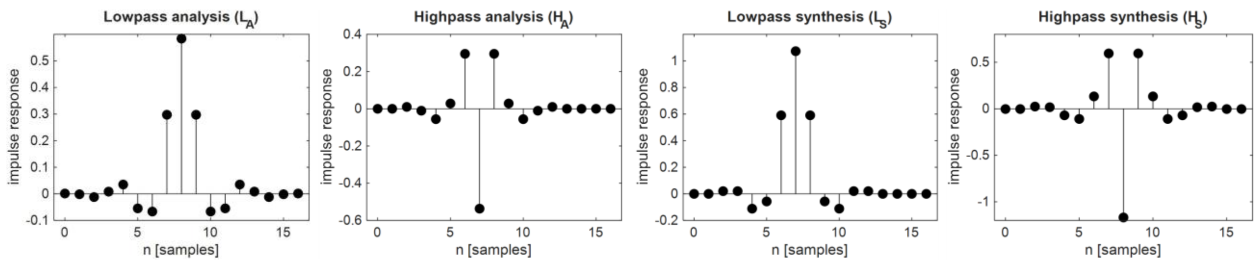
Lectures Slides 5-9 to 5-13 and 6-1 to 6-7 and 8-6 to 8-9 and 9-3 to 9-10

Mini-Project #2 | Tuneup #7

Problem 2.4. *Upsampling, Downsampling, and Filtering.* 24 points.

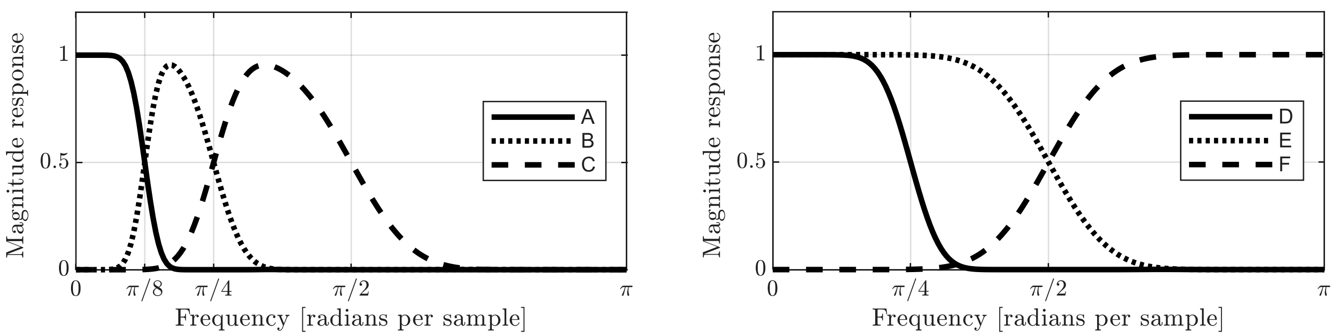
This problem is related to mini-project #2. Please justify your answers.

The impulse responses for four LTI filters L_A, H_A, L_S, and H_S are shown below. These filters are cascaded with downsampling ($\downarrow_2$) and upsampling ($\uparrow_2$) operations as shown in the block diagrams below.



i. Match each of the six system block diagrams with the magnitude responses (A-F) shown below. The magnitude response plot represents the change in magnitude when a complex sinusoid is input to the system, disregarding other frequencies that are created by upsampling.

ii. For each system, state whether or not the passband width is an octave. Assume that the passband is the set of frequencies where the magnitude response is greater than or equal to 0.5.



System Block Diagram	Match (A-F)	Is an octave?
$x[n] \to L_A \to \downarrow_2 \to \uparrow_2 \to L_S \to \hat{x}[n]$	**E**	**No**
$x[n] \to H_A \to \downarrow_2 \to \uparrow_2 \to H_S \to \hat{x}[n]$	**F**	**Yes**
$x[n] \to L_A \to \downarrow_2 \to L_A \to \downarrow_2 \to \uparrow_2 \to L_S \to \uparrow_2 \to L_S \to \hat{x}[n]$	**D**	**No**
$x[n] \to L_A \to \downarrow_2 \to H_A \to \downarrow_2 \to \uparrow_2 \to H_S \to \uparrow_2 \to L_S \to \hat{x}[n]$	**C**	**Yes**
$x[n] \to L_A \to \downarrow_2 \to L_A \to \downarrow_2 \to L_A \to \downarrow_2 \to \uparrow_2 \to L_S \to \uparrow_2 \to L_S \to \uparrow_2 \to L_S \to \hat{x}[n]$	**A**	**No**
$x[n] \to L_A \to \downarrow_2 \to L_A \to \downarrow_2 \to H_A \to \downarrow_2 \to \uparrow_2 \to H_S \to \uparrow_2 \to L_S \to \uparrow_2 \to L_S \to \hat{x}[n]$	**B**	**Yes**

Lowpass filters L_A and L_S are halfband filters with passbands from 0 to $\pi/2$.
Highpass filters H_A and H_S are halfband filters with passbands from $\pi/2$ to π.
Each system block diagram in linear time-varying which we will model as linear time-invariant.

Approach #1: Consider a discrete-time complex sinusoid $x[n] = e^{j\hat{\omega}n}$. Each downsampling stage doubles the discrete-time frequency. Each upsampling stage halves the frequency, scales the amplitude, and creates another frequency component shifted by π:

$$\downarrow_2 \left\{e^{j\hat{\omega}n}\right\} = e^{j\hat{\omega}(2n)} = e^{j(2\hat{\omega})n}, \qquad \uparrow_2 \left\{e^{j\hat{\omega}n}\right\} = \frac{1}{2}e^{j\frac{\hat{\omega}}{2}n} + \frac{1}{2}e^{j\left(\frac{\hat{\omega}}{2}-\pi\right)n}$$

For a cascade of N filters with equal downsampling and upsampling stages, the output will include a component at the original frequency. At this frequency, the effective frequency response is:

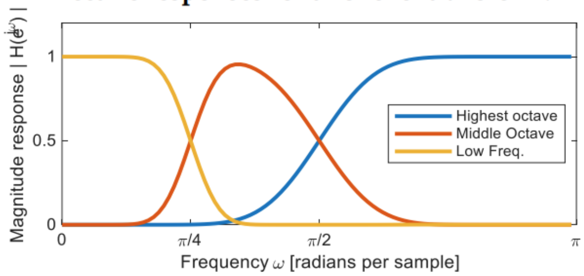
$$H_{\text{eff}}(\hat{\omega}) = \frac{1}{2^{(N/2)}} \prod_{k=1}^{N-1} H_k(e^{j\omega_k}), \qquad \hat{\omega}_k = 2^{D_k - U_k}\hat{\omega}$$

Here D_k is the number of downsampling stages applied prior to applying the kth filter and U_k is the number of upsampling stages applied prior to applying the kth filter. This is an LTI model.

Is it an octave? Lowpass and highpass filters L_A and H_A have passband widths of $\pi/2$. Each lowpass filter + downsampling extracts the lower half of the frequency band. If the final downsampling filter is highpass, it extracts the upper half of the frequency band, yielding one octave, as in **rows 2, 4 & 6.**

Approach #2: From the plot below in Section 3.0 of the mini-project #2 solution,

Effective responses for two level transform.



a. Upper octave branch: $(H_A \circ \downarrow_2 \circ \uparrow_2 \circ H_S)$

$$H_1(e^{j\omega}) = \frac{1}{2}H_{HA}(e^{j\hat{\omega}})H_{HS}(e^{j\hat{\omega}})$$

b. Middle octave branch: $(L_A \circ \downarrow_2 \circ H_A \circ \downarrow_2 \circ \uparrow_2 \circ H_S \circ \uparrow_2 \circ L_S)$

$$H_2(e^{j\hat{\omega}}) = \frac{1}{4}H_{LA}(e^{j\hat{\omega}})H_{HA}(e^{2j\hat{\omega}})H_{HS}(e^{j2\hat{\omega}})H_{LS}(e^{j\hat{\omega}})$$

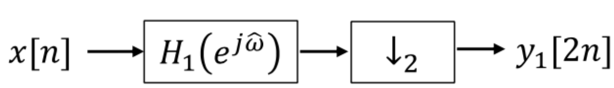
c. Low frequency branch: $(L_A \circ \downarrow_2 \circ L_A \circ \downarrow_2 \circ \uparrow_2 \circ L_S \circ \uparrow_2 \circ L_S)$

$$H_3(e^{j\hat{\omega}}) = \frac{1}{4}H_{LA}(e^{j\hat{\omega}})H_{LA}(e^{2j\hat{\omega}})H_{LS}(e^{j2\hat{\omega}})H_{LS}(e^{j\hat{\omega}})$$

In the table: *Second row* <u>matches F</u>. *Third row* <u>matches D</u>. *Fourth row* <u>matches C</u>.
Need to use one of the other approaches to find the remaining matches.

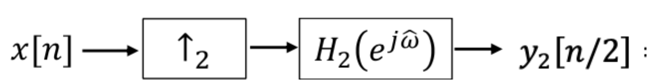
Approach #3: We'll directly reuse results from the mini-project #2 assignment and solution.

Downsampling stage: Filtering followed by downsampling by 2.



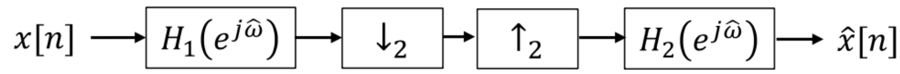
$$x[n] = e^{j\hat{\omega}n} \qquad\qquad y_1[2n] = H_1(e^{j\hat{\omega}})e^{j(2\hat{\omega})n}$$

Upsampling stage: Upsampling by 2 followed by filtering.



$$x[n] = e^{j\hat{\omega}n} \qquad y_2[n/2] = \frac{1}{2}H_2(e^{j\hat{\omega}/2})e^{j\left(\frac{\hat{\omega}}{2}\right)n} + \frac{1}{2}H_2(e^{j(\hat{\omega}/2-\pi)})e^{j\left(\frac{\hat{\omega}}{2}-\pi\right)n}$$

Single cascade: a downsampling stage followed by an upsampling stage.



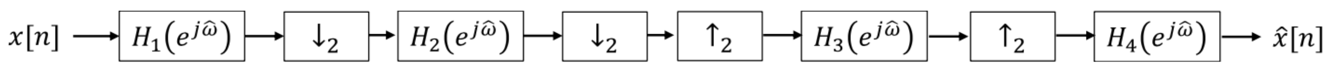
For $x[n] = e^{j\,\hat{\omega}\,n}$, the effective frequency response from input $x[n]$ to output $\hat{x}[n]$ for the input frequency $\hat{\omega}$, which is obtained by ignoring the second term in $y_2[n/2]$ for the output of upsampling by 2, is

$$\mathrm{H}_{\mathrm{eff}}(e^{j\hat{\omega}}) = \frac{1}{2} H_1(e^{j\hat{\omega}}) H_2(e^{j\hat{\omega}})$$

***First row*: Using this result, the first row in the table has an effective frequency response that is the product of two lowpass halfband filters $L_A\, L_S$, which is a lowpass halfband filter with passband from 0 to $\pi/2$. Match: E.**

***Second row*: Likewise, the second row in the table has an effective frequency response that is the product of two highpass halfband filters $H_A\, H_S$, which is a highpass halfband filter with passband from $\pi/2$ to π. Match: F.**

Double cascade: two downsampling stages in cas cade followed by two upsampling stages in cascade.



For $x[n] = e^{j\,\hat{\omega}\,n}$, the effective frequency response from input $x[n]$ to output $\hat{x}[n]$ for the input frequency $\hat{\omega}$, which is obtained by ignoring the second term in $y_2[n/2]$ for the output of each upsampling by 2, is

$$H_{\mathrm{eff}}(e^{j\hat{\omega}}) = \frac{1}{4} H_1(e^{j\hat{\omega}}) H_2(e^{2j\hat{\omega}}) H_3(e^{j2\hat{\omega}}) H_4(e^{j\hat{\omega}})$$

***Third row:* Using this result, the third row in the table has filters L_A, L_A, L_S, and L_S, and their effective frequency response matches D.**

***Fourth row*: Likewise, the fourth row in the table has filters L_A, H_A, H_S, and L_S, and their effective frequency response matches C.**

***Fifth row:* All the filters in the fifth row are lowpass, which will yield an effective lowpass frequency response equal to the narrowest bandwidth among the six lowpass filters. Matches A.**

***Sixth row:* By process of elimination, the sixth row in the table matches B.**

The Matlab code to generate the figures is provided below.

```
% Coefficients for a dyadic perfect reconstruction filterbank
% Based on the Cohen-Daubechies-Feauveau wavelet (bior6.8)

% If the wavelet toolbox is installed, you can also use the following code:
% [LA, HA, LS, HS] = wfilters('bior6.8');
% LA = LA(2:end)/sqrt(2); HA(2:end) = HA/sqrt(2);
% LS= sqrt(2)*LS(2:end); HS = sqrt(2)*HS(2:end);

coeffs = [
       0.00134974786501001                     0                     0    -0.00269949573002003
      -0.00135360470301001                     0                     0    -0.00270720940602003
       -0.0120141966670801    0.0102009221870399    0.0204018443740798      0.0240283933341602
       0.00843901203981008   -0.0102300708193699    0.0204601416387398      0.0168780240796202
        0.0351664733065404   -0.0556648607799594    -0.111329721559919     -0.0703329466130807
       -0.0546333136825205    0.0285444717151497   -0.0570889434302994      -0.109266627365041
       -0.0665099006248407     0.295463938592917     0.590927877185834       0.133019801249681
         0.297547906345713    -0.536628801791565      1.07325760358313       0.595095812691426
```

```
          0.584015752240756       0.295463938592917       0.590927877185834       -1.16803150448151
          0.297547906345713      0.0285444717151497     -0.0570889434302994       0.595095812691426
        -0.0665099006248407     -0.0556648607799594      -0.111329721559919       0.133019801249681
        -0.0546333136825205     -0.0102300708193699      0.0204601416387398      -0.109266627365041
         0.0351664733065404      0.0102009221870399      0.0204018443740798     -0.0703329466130807
        0.00843901203981008                       0                       0      0.0168780240796202
        -0.0120141966670801                       0                       0      0.0240283933341602
       -0.00135360470301001                       0                       0    -0.00270720940602003
        0.00134974786501001                       0                       0    -0.00269949573002003
];

LA = coeffs(:,1); % Lowpass Analysis
HA = coeffs(:,2); % Highpass Analysis
LS = coeffs(:,3); % Lowpass Synthesis
HS = coeffs(:,4); % Highpass Synthesis

w = linspace(0,pi,201);
[HLA, ~] = freqz(LA,1,w);
[HLA2, ~] = freqz(LA,1,2*w,'whole');
[HLA4, ~] = freqz(LA,1,4*w,'whole');
[HHA, ~] = freqz(HA,1,w);
[HHA2, ~] = freqz(HA,1,2*w,'whole');
[HHA4, ~] = freqz(HA,1,4*w,'whole');
[HLS, ~] = freqz(LS,1,w);
[HLS2, ~] = freqz(LS,1,2*w,'whole');
[HLS4, ~] = freqz(LS,1,4*w,'whole');
[HHS, ~] = freqz(HS,1,w);
[HHS2, ~] = freqz(HS,1,2*w,'whole');
[HHS4, ~] = freqz(HS,1,4*w,'whole');

% 0 to pi/8
H_band1 = 0.125*HLA.*HLA2.*HLA4.*HLS4.*HLS2.*HLS;
figure; plot(w,abs(H_band1),'k',linewidth=2)

% pi/8 to pi/4
H_band2 = 0.125*HLA.*HLA2.*HHA4.*HHS4.*HLS2.*HLS;
hold on; plot(w,abs(H_band2),'k:',linewidth=2)

% pi/4 to 3*pi/8
H_band3 = 0.25*HLA.*HHA2.*HHS2.*HLS;
hold on; plot(w,abs(H_band3),'k--',linewidth=2)

xlim([0,pi]); set(gca,'XTick', ...
    [0,pi/8,pi/4,pi/2,pi])
set(gca,'TickLabelInterpreter','latex')
set(gca,'XTickLabels',["0", "$\pi/8$", "$\pi/4$","$\pi/2$","$\pi$"])
xlabel('Frequency [radians per sample]','Interpreter','latex')
ylabel('Magnitude response ','Interpreter','latex')
grid on;
legend("A","B","C",'location','east')

% 0 to pi/4
H_band1 = 0.25*HLA.*HLA2.*HLS2.*HLS;
figure; plot(w,abs(H_band1),'k-',linewidth=2)

% 0 to pi/2
H_band2 = 0.5*HLA.*HLS;
hold on; plot(w,abs(H_band2),'k:',linewidth=2)

% pi/2 to pi
H_band4 = 0.5*HHA.*HHS;
hold on; plot(w,abs(H_band4),'k--',linewidth=2)

xlim([0,pi]); set(gca,'XTick', ...
    [0,pi/4,pi/2,pi])
set(gca,'TickLabelInterpreter','latex')
set(gca,'XTickLabels',["0", "$\pi/4$","$\pi/2$","$\pi$"])
xlabel('Frequency [radians per sample]','Interpreter','latex')
ylabel('Magnitude response ','Interpreter','latex')
grid on;
legend("D","E","F",'location','east')
```