EECE-5666 : Midterm-1 Exam : 2022-SPRG

130 Minutes between 3:30 pm and 6:30 pm on Friday

[40-Points]

Instructions:

1. You are required to read the NU Academic Integrity policy (given below) and sign below that the submitted work is your own work. (This is a COE requirement.)
2. You are required to complete this exam using Live Editor.
3. Use of the equation editor to typeset mathematical material such as variables, equations, etc., is strongly recommended. However, in the interest of time, a properly scanned image of a neatly hand-written fragment can be inserted as your work in the required space.
4. After completing this exam, export this Live script to PDF. If you encounter problems in the direct export to PDF, export to Microsoft Word and then print to PDF
5. Submit the PDF file only. **DO NOT SUBMIT YOUR LIVE EDITOR (.MLX) FILE, IT WILL NOT BE GRADED**.
6. You should complete this exam in two hours with additional 10 minutes for submission activities.
7. The exam is made available for only one attempt.

Academic Integrity Policy

A commitment to the principles of academic integrity is essential to the mission of Northeastern University. The promotion of independent and original scholarship ensures that students derive the most from their educational experience and their pursuit of knowledge. Academic dishonesty violates the most fundamental values of an intellectual community and undermines the achievements of the entire University.

As members of the academic community, students must become familiar with their rights and responsibilities. In each course, they are responsible for knowing the requirements and restrictions regarding research and writing, examinations of whatever kind, collaborative work, the use of study aids, the appropriateness of assistance, and other issues. Students are responsible for learning the conventions of documentation and acknowledgment of sources in their fields. Northeastern University expects students to complete all examinations, tests, papers, creative projects, and assignments of any kind according to the highest ethical standards, as set forth either explicitly or implicitly in this Code or by the direction of instructors. The full academic integrity policy is available at

http://www.northeastern.edu/osccr/academic-integrity-policy/

Declaration

By signing (i.e., entering my name below) and submitting this exam through the submission portal, I declare that I have read the Academic Integrity Policy and that the submitted work is my own work.

Enter your name (Firstname MI Lastname): Tyler B McKean

Default Plot Parameters: Execute the following script in the beginning or when you restart this file.

```
set(0,'defaultfigurepaperunits','points','defaultfigureunits','points');
set(0,'defaultaxesfontsize',10);
set(0,'defaultaxestitlefontsize',1.4,'defaultaxeslabelfontsize',1.2);
```

Problem-1 (10-points) *LTI System and Convolution*

A causal LTI system is described by the impulse response

$$h[n] = 0.4(0.8)^n\big(u[n] - u[n-16]\big).$$

The input to this system is given by

$$x[n] = 2.5(0.8)^n\big(u[n+15] - u[n-1]\big).$$

Let $y[n]$ be the system output sequence.

(a) [5-Points] Using the ***convolution summation*** $y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$, determine analytically the output sequence $y[n]$. Do not use the other summation $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$ or any other method.

Solution:

We can simplify both $h[n]$ and $x[n]$ by computing

$$y[n] = (2.5)(0.4)\sum_{n=0}^{15}(0.8)^n u[n]$$

$$\sum_{n=-15}^{0} 2.5(0.8)^n = (2.5)\frac{0.8^{-15}-1}{1-0.8}$$

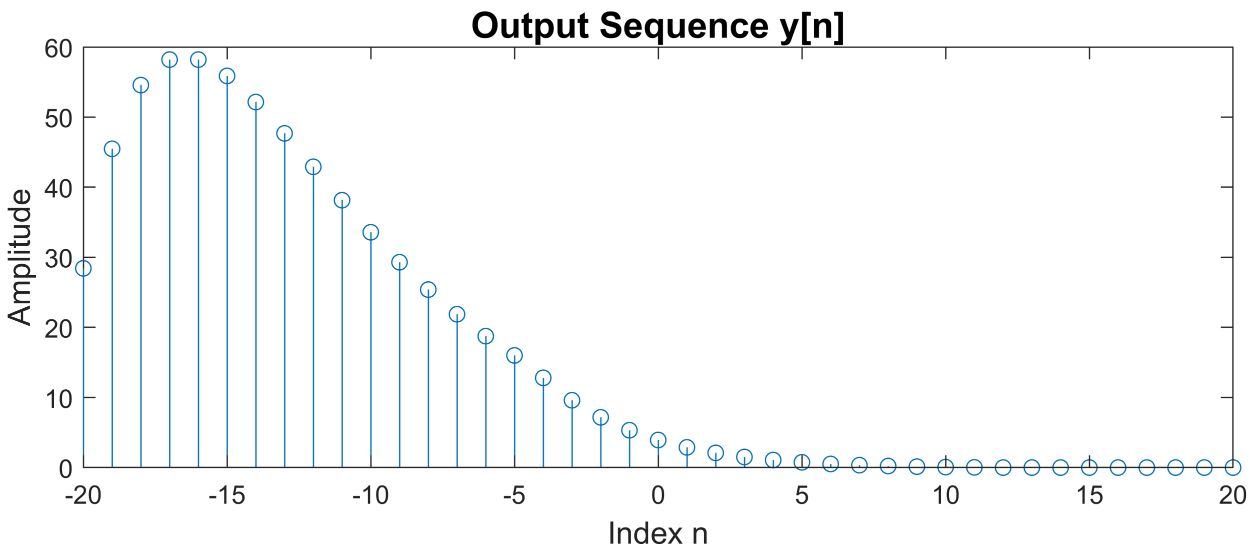
(b) [3-Points] Numerically compute and **`stem`** plot $y[n]$ over $-20 \le n \le 20$ range.

MATLAB script: Use the folowing fragments for your plot.

```
figure('Position',[0,0,8,3]*72);

xlabel('Index n'); ylabel('Amplitude'); title('Output Sequence y[n]');
```

```
clc; close all; clear;
% Enter your code below
n = -20:20; u1 = [zeros(1,20) ones(1,16) 0 0 0 0 0]; u2 = [0 0 0 0 0 ones(1,17) zeros(1, 19)];
```

```
h = 0.4.*(0.8.^n).*u1;
x = 2.5.*(0.8.^n).*u2;
y = conv(x,h);
figure('Position',[0,0,8,3]*72);
stem(n,y(26:66))
xlabel('Index n'); ylabel('Amplitude'); title('Output Sequence y[n]');
```



(c) [2-Points] Provide a **`stem`** plot of your output sequence $y[n]$ in part (a) over the $-20 \le n \le 20$ range and verify that it agrees with the graph in part (b).

MATLAB script: Use the folowing fragments for your plot.

```
figure('Position',[0,0,8,3]*72);

xlabel('Index n'); ylabel('Amplitude'); title('Output Sequence y[n]');
```

```
% Enter your code below

figure('Position',[0,0,8,3]*72);


xlabel('Index n'); ylabel('Amplitude'); title('Output Sequence y[n]');
```

Verification:

Problem-2 (10-Points) *The z-Transform*

Consider the z-transform expression: $X(z) = \dfrac{(z+0.4)\left(z^2+0.3z+0.4\right)(z-0.9)}{\left(z^2+3z+5\right)\left(z^2-0.6z+0.6\right)}$

(a) [2-Points] Determine and plot the ***pole-zero pattern*** of $X(z)$.

MATLAB script: Use the following fragments in your script for figure size and title.

```
figure('position',[1,1,5,4]*72);


title('Pole-Zero Plot');
```

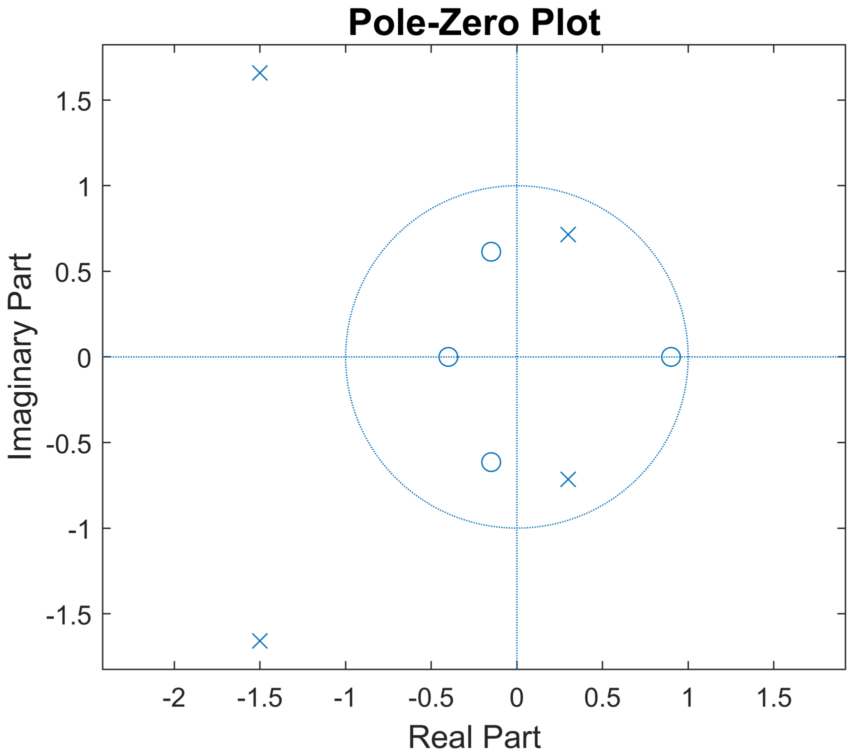
```
clc; close all; clear;
% Enter your code below
b1 = conv([1,0.4],[1,0.3,0.4]); b = conv(b1,[1,-0.9]);
a = conv([1,3,5],[1,-0.6,0.6]);
Z = roots(b).'; P = roots(a).';
Zmag = abs(Z), Zpharad = angle(Z)/pi, Zphadeg = angle(Z)*180/pi
```

```
Zmag = 1×4
    0.9000    0.6325    0.6325    0.4000
Zpharad = 1×4
         0    0.5762   -0.5762    1.0000
Zphadeg = 1×4
         0  103.7196 -103.7196  180.0000
```

```
Pmag = abs(P), Ppharad = angle(P)/pi, Pphadeg = angle(P)*180/pi
```

```
Pmag = 1×4
    2.2361    2.2361    0.7746    0.7746
Ppharad = 1×4
    0.7341   -0.7341    0.3734   -0.3734
Pphadeg = 1×4
  132.1304 -132.1304   67.2135  -67.2135
```

```
figure('position',[1,1,5,4]*72);
zplane(b,a)
title('Pole-Zero Plot');
```



(b) [3-Points] List all possible ***regions of convergence*** (ROCs) for this z-transform. Do not provide plots of these ROCs.

Answer:

$$ROC_1 = |z| > 2.2361$$

$$ROC_2 = 0.7746 < |z| < 2.2361$$

$$ROC_3 = |z| < 0.7746$$

(c) [3-Points] Determine the inverse z-transform so that the resulting sequence $x[n]$ is bounded. Your sequence $x[n]$ should not contain any complex numbers.

Solution:

```
[A, p, C] = residuez(b,a), magA = abs(A)', phaA = (angle(A)*180/pi)',
```

```
A = 4×1 complex
   0.5038 + 0.2941i
   0.5038 - 0.2941i
   0.0202 + 0.0878i
   0.0202 - 0.0878i
p = 4×1 complex
  -1.5000 + 1.6583i
  -1.5000 - 1.6583i
   0.3000 + 0.7141i
```

```
   0.3000 - 0.7141i
C = -0.0480
magA = 1×4
    0.5834    0.5834    0.0900    0.0900
phaA = 1×4
   30.2717  -30.2717   77.0464  -77.0464
```

```
magP = abs(p)', phaP = (angle(p)/pi)'
```

```
magP = 1×4
    2.2361    2.2361    0.7746    0.7746
phaP = 1×4
    0.7341   -0.7341    0.3734   -0.3734
```

The inverse transform of $X(z)$ would be:

$$x[n] = -0.05\delta[n] + (2)(0.583)(2.236)^n \cos(0.7341\pi n + 30.27^\circ)u[-n-1] + (2)(0.09)(0.7746)^n \cos(0.3734\pi n + 77.05^\circ)u[n]$$

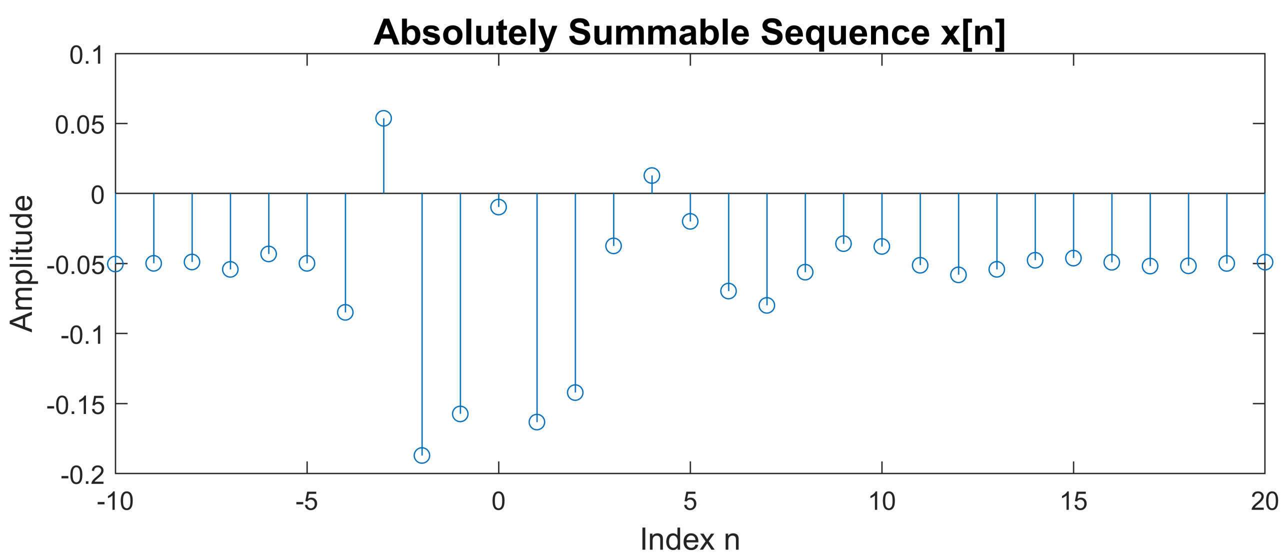
$$x[n] = -0.05\delta[n] + (1.166)(2.236)^n \cos(0.7341\pi n + 30.27^\circ)u[-n-1] + (0.18)(0.7746)^n \cos(0.3734\pi n + 77.05^\circ)u[n]$$

(d) [2-points] Provide a **`stem`** plot of $x[n]$ over $-10 \le n \le 20$.

MATLAB script: Use the following fragments in your script for figure size and title.

```
figure('position',[1,1,8,3]*72);


xlabel('Index n'); ylabel('Amplitude');title('Absolutely Summable Sequence x[n]');
```

```
% Enter your code below
n = -10:20;
u1 = [ones(1,10) zeros(1,21)]; u2 = [zeros(1,10) ones(1,21)];
x = -0.05 + (1.166).*(2.236.^n).*cos(0.7341.*pi.*n + 0.1681.*pi).*u1 + (0.18).*(0.7746.^n).*cos
figure('position',[1,1,8,3]*72);
stem(n,x)

xlabel('Index n'); ylabel('Amplitude');title('Absolutely Summable Sequence x[n]');
```



Problem-3 (9-Points) *The Discrete-Time Fourier Transform*

A causal LTI system is described by the following system function

$$H(z) = \frac{1 + z^{-4}}{1 - 0.8145z^{-4}}. \qquad (3.1)$$

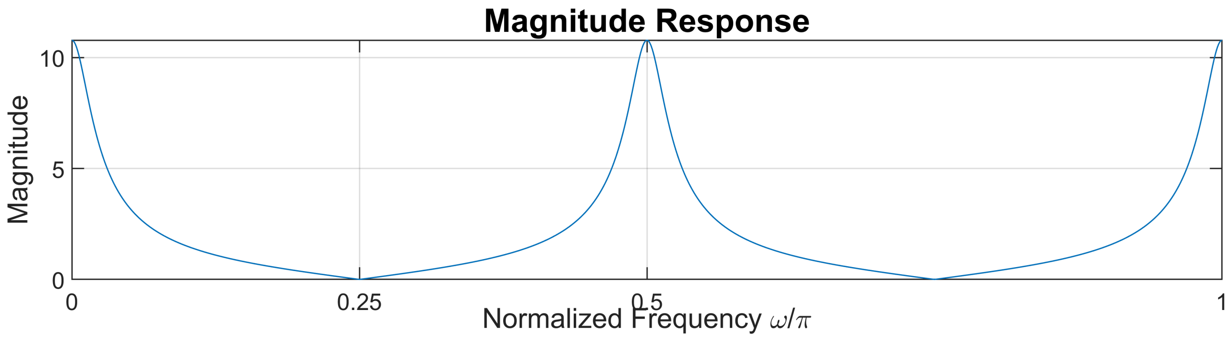
Let $x[n]$ be the input to the above system and let $y[n]$ be the corresponding output.

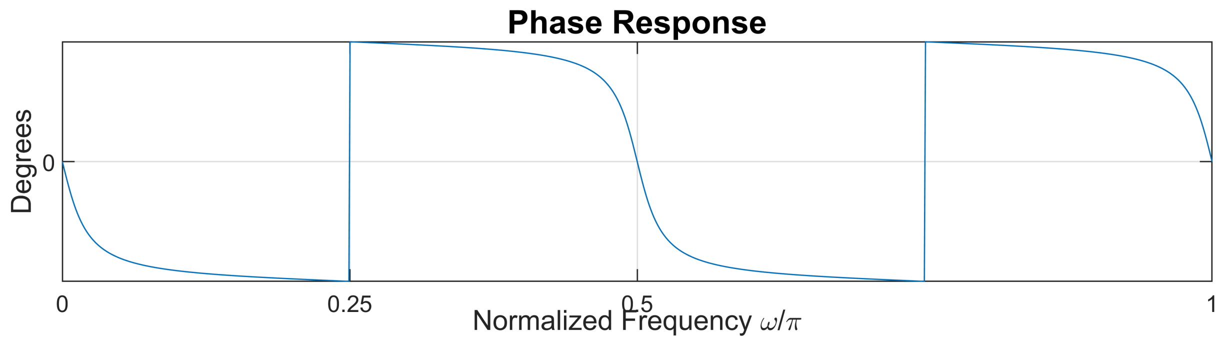
(a) (3-points) Using MATLAB compute and plot the magnitude and phase of the frequency response between $0 \le \omega \le \pi$ interval.

MATLAB script and plots: Use the following fragments in your script for plotting.

```
figure('position',[0,0,8,4]*72,'paperposition',[0,0,8,4]*72);
subplot(2,1,1); % Plot for magnitude response

ylabel('Magnitude'); title('Magnitude Response'); grid;
xlabel('Normalized Frequency \omega/\pi','VerticalAlignment','middle');
set(gca,'xtick',[0,0.25,0.5,1]); title('Magnitude Response');
subplot(2,1,2); % Plot for phase-delay response

ylabel('Degrees'); title('Phase Response'); grid;
xlabel('Normalized Frequency \omega/\pi','VerticalAlignment','middle');
set(gca,'xtick',[0,0.25,0.5,1],'ytick',(-180:90:180));
title('Phase Response');
```

```
% Insert your code below for the plot
b = [1 0 0 0 1]; a = [1 0 0 0 -0.8145]; omega = linspace(0,pi,1000);
H = freqz(b,a,omega); Hmag = abs(H); Hpha = angle(H)/pi;

figure('position',[0,0,8,4]*72,'paperposition',[0,0,8,4]*72);
subplot(2,1,1); % Plot for magnitude response
plot(omega/pi, Hmag)
ylabel('Magnitude'); title('Magnitude Response'); grid;
xlabel('Normalized Frequency \omega/\pi','VerticalAlignment','middle');
set(gca,'xtick',[0,0.25,0.5,1]); title('Magnitude Response');
subplot(2,1,2); % Plot for phase-delay response
plot(omega/pi,Hpha)
ylabel('Degrees'); title('Phase Response'); grid;
xlabel('Normalized Frequency \omega/\pi','VerticalAlignment','middle');
set(gca,'xtick',[0,0.25,0.5,1],'ytick',(-180:90:180));
title('Phase Response');
```





(b) (3-points) Let $x[n] = [\sin(\pi n/4) + 5\cos(\pi n/2)]u[n]$. Determine, analytically, the ***steady-state response*** $y_{ss}[n]$.

Solution:

```
om = [0.25,0.5]*pi;
H = freqz(b,a,om); Hmag = abs(H); Hpha = angle(H)/pi; Hmax = max(Hmag)
```

```
Hmax = 10.7817
```

```
H1 = [Hmag(1), Hpha(1)], H2 = [Hmag(2), Hpha(2)]
```

```
H1 = 1×2
    0.0000   -0.5000
H2 = 1×2
   10.7817    0.0000
```

Solving for $H(e^{j\pi/4})$ and $H(e^{j\pi/2})$, we see from the calculations that:

$H(e^{j\pi/4}) = 0$

$H(e^{j\pi/2}) = 10.78\angle 0^\circ$

Thus, the first component of the input will be filtered out because of the magnitude response and the second component's amplitude will only be scaled by a factor of $-\frac{1}{2}$

So, $y_{ss}[n] = (10.78)5\cos(\pi n/2) = 53.9\cos(\pi n/2)$

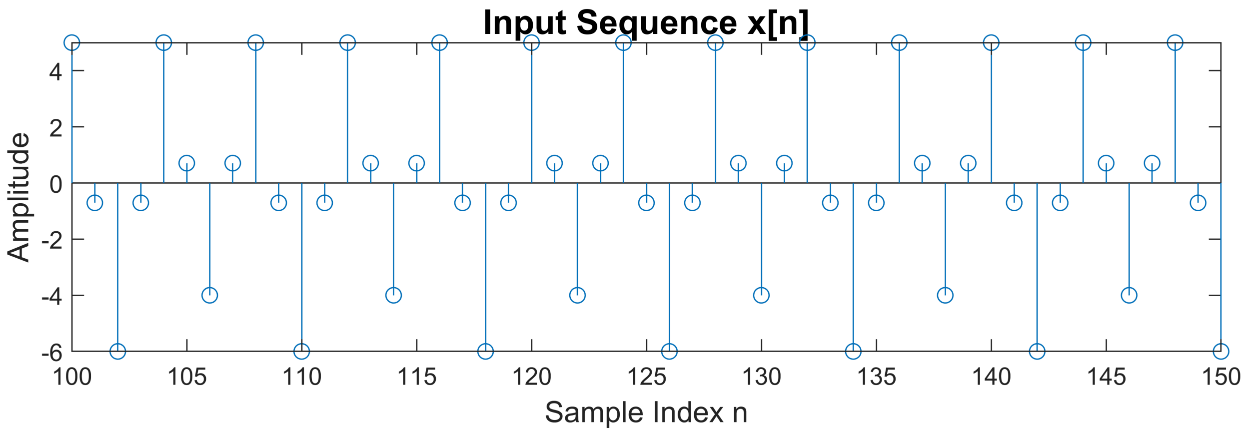
$y_{ss}[n] = 54\cos(\pi n/2)$

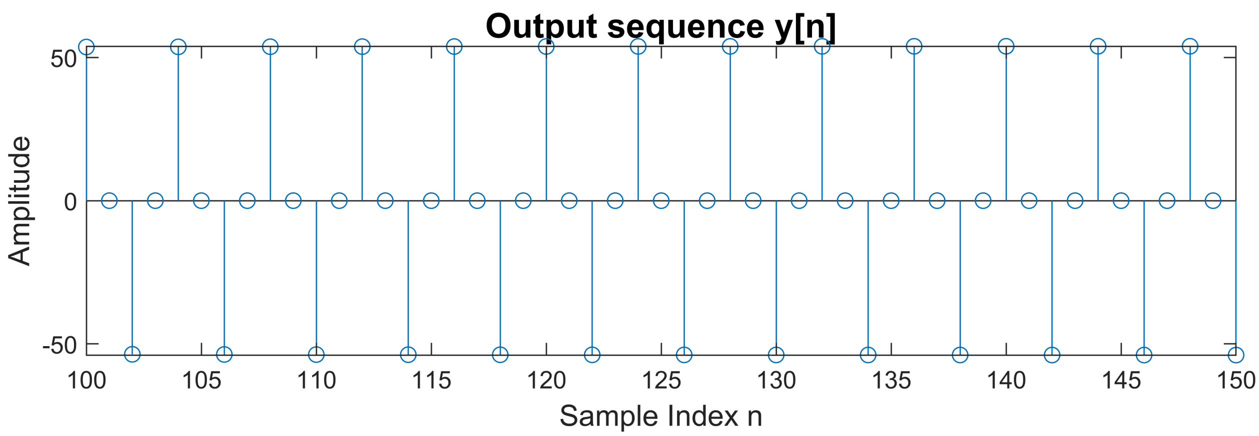
(c) (3-points) Generate 151 samples of the sequence $x[n]$ over $0 \le n \le 150$ in part (c) and process them through the filter in (3.1) to obtain the output $y[n]$. In one figure with two subplot rows, graph **`stem`** plot of $x[n]$ in the top subplot and **`stem`** plot $y[n]$ in the bottom subplot. To avoid transient response, graph both of these plots over $100 \le n \le 150$. Verify whether your graph of the output agrees with your answer $y_{ss}[n]$ in part (c).

MATLAB script and plots: Use the following fragments in your script for plotting.

```
figure('position',[0,0,8,5]*72,'paperposition',[0,0,8,5]*72);
subplot(2,1,1); % plot for x[n], 100 <= n <= 150

xlabel('Sample Index n'); ylabel('Amplitude'); title('Input Sequence x[n]');
subplot(2,1,2); % plot for y[n], 100 <= n <= 150

xlabel('Sample Index n'); ylabel('Amplitude'); title('Output sequence y[n]');
```

```
% Insert your code below for the plots
n = 0:150; u = (n>=0); x = (sin(.25*pi.*n) + 5*cos(0.5*pi.*n)).*u;
y = filter(b,a,x);
figure('position',[0,0,8,5]*72,'paperposition',[0,0,8,5]*72);
subplot(2,1,1); % plot for x[n], 100 <= n <= 150
stem(n(101:end),x(101:end))
xlabel('Sample Index n'); ylabel('Amplitude'); title('Input Sequence x[n]');
subplot(2,1,2); % plot for y[n], 100 <= n <= 150
stem(n(101:end),y(101:end))
xlabel('Sample Index n'); ylabel('Amplitude'); title('Output sequence y[n]');
```





Verification:

Looking at our steady state reponse of the output, we observe that the amplitude has been increased to just over ten-times the original amplitude of the second component. Thus, our steady- state response matches the frequency responses values calculated in the previous part.

Problem-4 (11-points) *Sampling and Reconstruction*

A CT signal $x_c(t) = 5 + 3\sin(25\pi t) + 2\cos(200\pi t + \pi/3)$ is sampled at $t = 0.01n$ to obtain the DT signal $x[n]$. It is then processed through a causal FIR filter given by the impulse response

$$h[n] = \frac{1}{8}\left(u[n] - u[n-8]\right)$$

to obtain the output DT signal $y[n]$. Finally, the output sequence $y[n]$ is applied to a ***sample-and-hold*** digital-to-analog converter (SH-DAC) to obtain the CT signal $y_{\mathrm{SH}}(t)$.

(a) (3-points) Determine $x[n]$. Your answer must be in the ***most possible compact form*** to get maximum points.

Solution: Substituting $t = 0.01n$ in $x_c(t)$ above, we obtain

$$
\begin{aligned}
x[n] &= x_c(0.01n) = 5 + 3\sin(.25\pi n) + 2\cos(2\pi n + \pi/3) \\
&= 5 + 3\sin(\pi n/4) + 2\cos(\pi/3) = 6 + 3\sin(\pi n/4).
\end{aligned}
$$

(b) (3-points) Determine $y[n]$. Your answer must be in the ***most possible compact form*** to get maximum points.

Solution:

```
h = (1/8)*ones(1,8); om = [0 0.25 .5 .75 1]*pi; H = freqz(h,1,om);
Hmag = abs(H), Hpha = angle(H)/pi;
```

```
Hmag = 1×5
    1.0000    0.0000    0.0000    0.0000    0.0000
```

Analyzing the frequency response, we see that at $\omega = \frac{\pi}{4}$ the magnitude response zeros that frequency, which eliminates the sinusoid component and only our DC component of $x[n]$ remains

So, $y[n] = 6 + (0)\sin(\pi n/4) \rightarrow y[n] = 6$

(c) (4-points) Provide the following graphs in the same figure.

1. CT signal $x_c(t)$ over $0 \le t \le 300$ ms (use the **`plot`** function),
2. DT signal $x[n]$ over $0 \le t \le 300$ ms (use the **`plot`** function and **`'o'`** marker),
3. DT signal $y[n]$ over $0 \le t \le 300$ ms (use the **`plot`** function and **`'x'`** marker),
4. CT staircase signal $y_r(t)$ over $0 \le t \le 300$ ms. For this graph use the **`stairs`** function. This is a MATLAB function and renders $y_r(t)$ as a SH-DAC reconstruction.

Note that the above graphs are over natural time t in miliseconds and not over index n. Use the **`hold`** function to display all graphs in one figure and use the following fragments in your script.

```
figure('PaperPosition',[0,0,8,4]*72,'position',[0,0,8,4]*72);


xlabel('Time in Second'); ylabel('Amplitude');
title('Ideal Sampling and SH-DAC Reconstruction');
set(gca,'ytick',(0:2:10)); grid; hold off;
```

MATLAB script:

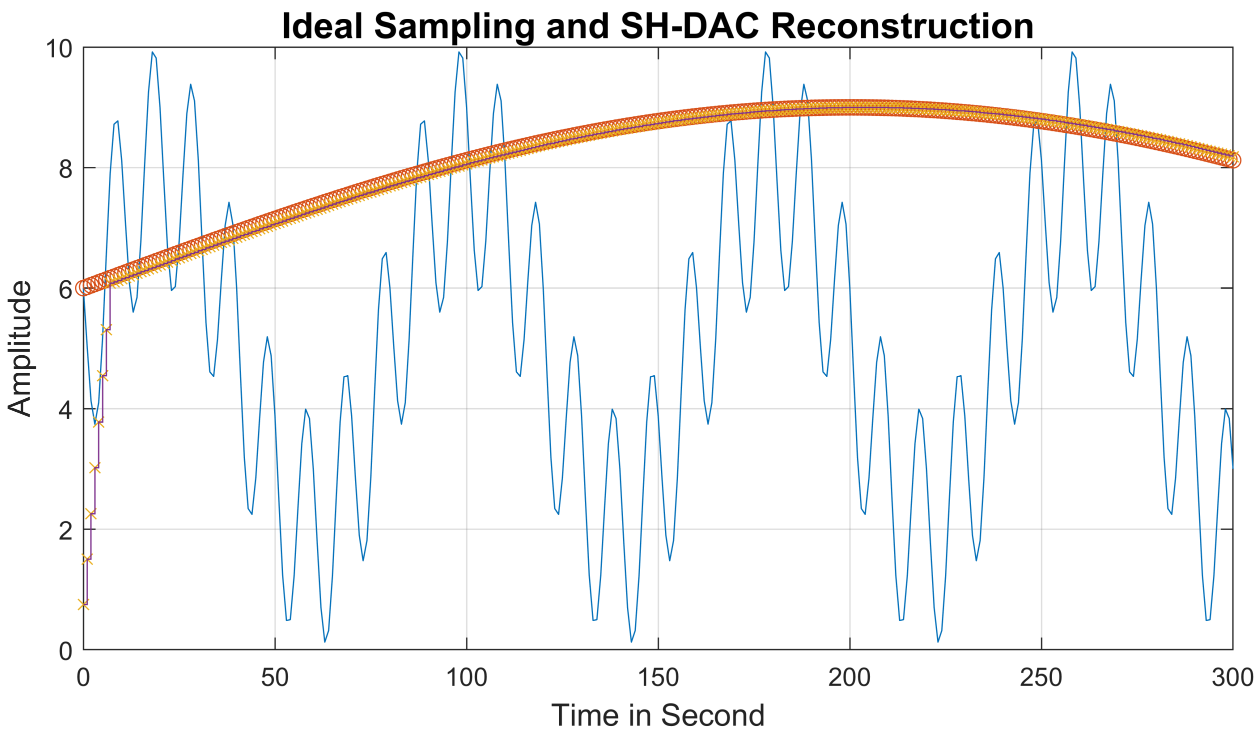
```
%clc; close all;
% Enter your code below
t = 0:1/1000:0.3; xc = 5 + 3*sin(25*pi*t) + 2*cos(200*pi*t + pi/3);
T = 0.01; n = 0:300; nTs = n.*T; y
```

```
y = 1×301
    0.7500    1.5029    2.2588    3.0177    3.7794    4.5442    5.3118    6.0824 ···
```

```
xn = 6 + 3*sin(0.25*pi*n*T); y = filter(h,1,xn)
```

```
y = 1×301
    0.7500    1.5029    2.2588    3.0177    3.7794    4.5442    5.3118    6.0824 ···
```

```
figure('PaperPosition',[0,0,8,4]*72,'position',[0,0,8,4]*72);
plot(1000*t,xc), hold on
plot(n,xn,'o'), plot(n,y,'x'), stairs(n,y)

xlabel('Time in Second'); ylabel('Amplitude');
title('Ideal Sampling and SH-DAC Reconstruction');
set(gca,'ytick',(0:2:10)); grid; hold off;
```



(d) (1-point) After observing the above plots, explain why it is not necessary to use the ideal DAC for this signal.

Answer:

It was not necessary to use the Ideal DAC because the resulting signal after being outputted from out impulse response was only a DC component. This filtered component does not need to be reconstructed using the Ideal DAC interpolation method, so these methods seem unnecessary.