Contents

QUESTION 2

DO NOT REMOVE THE LINE BELOW MAKE SURE 'eel3135_lab07_comment.m' IS IN THE SAME DIRECTORY AS THIS FILE

```
clear; close all; clc;
type('eel3135_lab07_comment.m')
```

```
%% QUESTION #1 COMMENTING
clear
close all
clc

%% DEFINE FILTER AND INPUT
N = 100;
n = 0:(N-1);

% FILTER
b = (1/4)*[1 1 1]; % Numerator Coefficients of filter's transfer function, which is the coefficients of the input sequence in the difference equation
a = [1 -1 0.5 0.25]; % Denominator coefficients of filter's transfer function, which is the coefficients of output sequence in difference equation
% <-- Answer: What is the numerator of this filter's transfer function?
% The numerator of the filter's transfer function is (1/4) * [1 1 1] or
% (1/4) + (1/4)*z^-1 + (1/4)*z^-2

% <-- Answer: What is the denominator of this filter's transfer function?
% The denominator fo the filter transfer function is [1 -1 0.5 0.25] or 1 -
% z^-1 + 0.5*z^-2 + 0.25*z^-3

% Hint: look a later when we implement the filter with the "filter" command
%  (use "help filter" to see what it does)
%

% INPUT 1
x1 = zeros(N,1);
x1(1) = 1; % Creates impulse input at n = 0
% <-- Answer: (True or False) x1 is an impulse input? If false,
% describe the input.
% True. x1 is an impulse input becuase it is a sequence of zeros with a
% single one at the first sample, definition of impulse

% INPUT 2
x2 = zeros(N,1);
x2(1:12)  = cos(3*pi/2*   n(1:12)); % First segment of x2
x2(13:24) = cos(pi/4*   n(13:24)); % Second segment of x2
x2(25:36) = cos(2*pi*   n(25:36)); % Third segment of x2
x2(37:48) = cos(3*pi/4* n(37:48)); % Fourth segment of x2
x2(49:60) = cos(pi*     n(49:60)); % Fifth segment of x2
x2(61:72) = cos(pi/8* n(61:72)); % Sixth segment of x2
```

```
x2(73:84) = cos(pi/2*   n(73:84)); % Seventh segment of x2
% <-- Answer: (True or False) x2 is an single frequency input? If
% false, describe the input.
% False. x2 is not a signle frequency input, it is a cobination of multiple
% cosine signals at different frequencies over time where each segment of
% x2 has a different frequency.

%% DEFINE AND PLOT OUTPUT

% OUTPUT 1
y1 = filter(b,a,x1);

% OUTPUT 2
y2 = filter(b,a,x2);


% PLOT THE IMPULSE RESPONSE AND DTFT
figure(1)
subplot(311)
stem(n,x1)
xlabel('Time (Samples)')
ylabel('x_1[n]')
subplot(312)
stem(n,y1)
xlabel('Time (Samples)')
ylabel('y_1[n]')
subplot(313)
pzplot(b,a)
axis equal

% PLOT THE IMPULSE RESPONSE AND DTFT
figure(2)
subplot(311)
stem(n,x2)
xlabel('Time (Samples)')
ylabel('x_2[n]')
subplot(312)
stem(n,y2)
xlabel('Time (Samples)')
ylabel('y_2[n]')
subplot(313)
pzplot(b,a)
axis equal
```

```
% ================================================================
% NOTE: YOU DO NOT NEED TO ADD COMMENTS IN THE CODE BELOW. WE JUST
% NEEDED POLE-ZERO PLOTTING CODE AND THUS WROTE IT.
% ================================================================
function pzplot(b,a)
% PZPLOT(B,A)  plots the pole-zero plot for the filter described by
% vectors A and B.  The filter is a "Direct Form II Transposed"
% implementation of the standard difference equation:
%
%    a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
%                          - a(2)*y(n-1) - ... - a(na+1)*y(n-na)
%

    % MODIFY THE POLYNOMIALS TO FIND THE ROOTS
    b1 = zeros(max(length(a),length(b)),1); % Need to add zeros to get the right roots
    a1 = zeros(max(length(a),length(b)),1); % Need to add zeros to get the right roots
    b1(1:length(b)) = b;    % New a with all values
    a1(1:length(a)) = a;    % New a with all values

    % FIND THE ROOTS OF EACH POLYNOMIAL AND PLOT THE LOCATIONS OF THE ROOTS
    h1 = plot(real(roots(a1)), imag(roots(a1)));
    hold on;
    h2 = plot(real(roots(b1)), imag(roots(b1)));
    hold off;

    % DRAW THE UNIT CIRCLE
    circle(0,0,1)

    % MAKE THE POLES AND ZEROS X's AND O's
    set(h1, 'LineStyle', 'none', 'Marker', 'x', 'MarkerFaceColor','none', 'linewidth', 1.5, 'markersize', 8);
    set(h2, 'LineStyle', 'none', 'Marker', 'o', 'MarkerFaceColor','none', 'linewidth', 1.5, 'markersize', 8);
    axis equal;

    % DRAW VERTICAL AND HORIZONTAL LINES
    xminmax = xlim();
    yminmax = ylim();
    line([xminmax(1) xminmax(2)],[0 0], 'linestyle', ':', 'linewidth', 0.5, 'color', [1 1 1]*.1)
    line([0 0],[yminmax(1) yminmax(2)], 'linestyle', ':', 'linewidth', 0.5, 'color', [1 1 1]*.1)

    % ADD LABELS AND TITLE
    xlabel('Real Part')
    ylabel('Imaginary Part')
    title('Pole-Zero Plot')
```

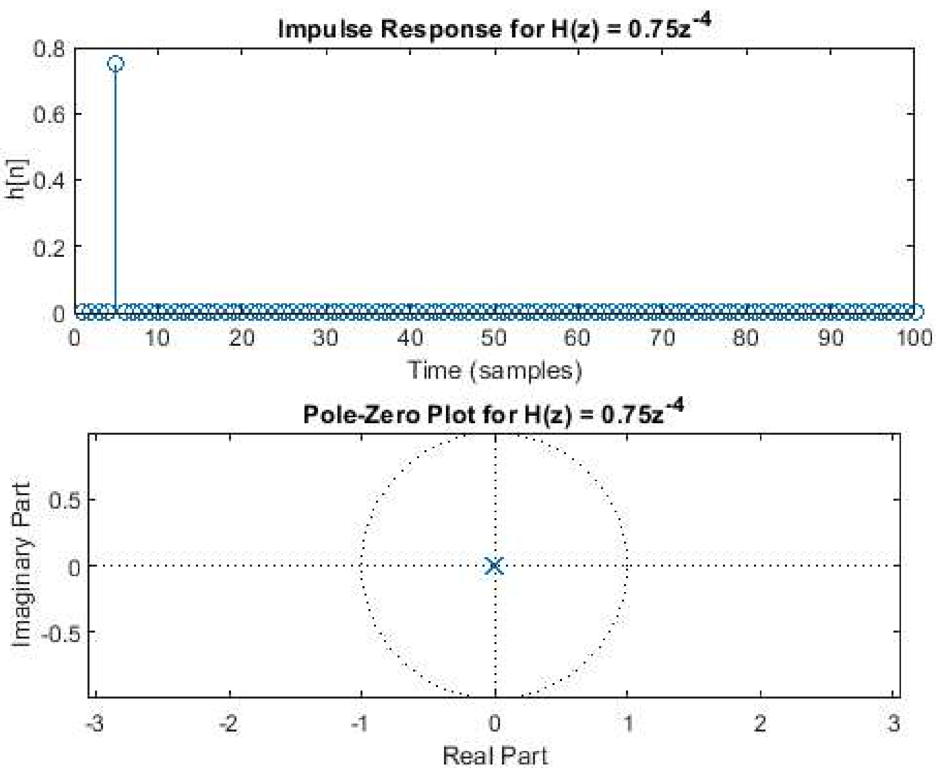
```
end


function circle(x,y,r)
% CIRCLE(X,Y,R)  draws a circle with horizontal center X, vertical center
% Y, and radius R.
%

    % ANGLES TO DRAW
    ang=0:0.01:2*pi;

    % DEFINE LOCATIONS OF CIRCLE
    xp=r*cos(ang);
    yp=r*sin(ang);

    % PLOT CIRCLE
    hold on;
    plot(x+xp,y+yp, ':', 'linewidth', 0.5, 'color', [1 1 1]*.1);
    hold off;

end
```

QUESTION 2: Z-TRANSFORM

2 (a) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define transfer function H(z) = 0.75 * z^(-4)
b1 = [0 0 0 0 0.75]; % Numerator (delayed by 4 samples)
a1 = [1]; % Denominator

% Generate impulse response
N = 100;
n = 0:(N-1);
x1 = zeros(N,1);
x1(1) = 1; % Impulse response
impulse_response_a = filter(b1, a1, x1); % Output using filter function

% Plot impulse response
figure;
subplot(2, 1, 1);
stem(impulse_response_a);
title('Impulse Response for H(z) = 0.75z^{-4}');
xlabel('Time (samples)');
ylabel('h[n]');
```

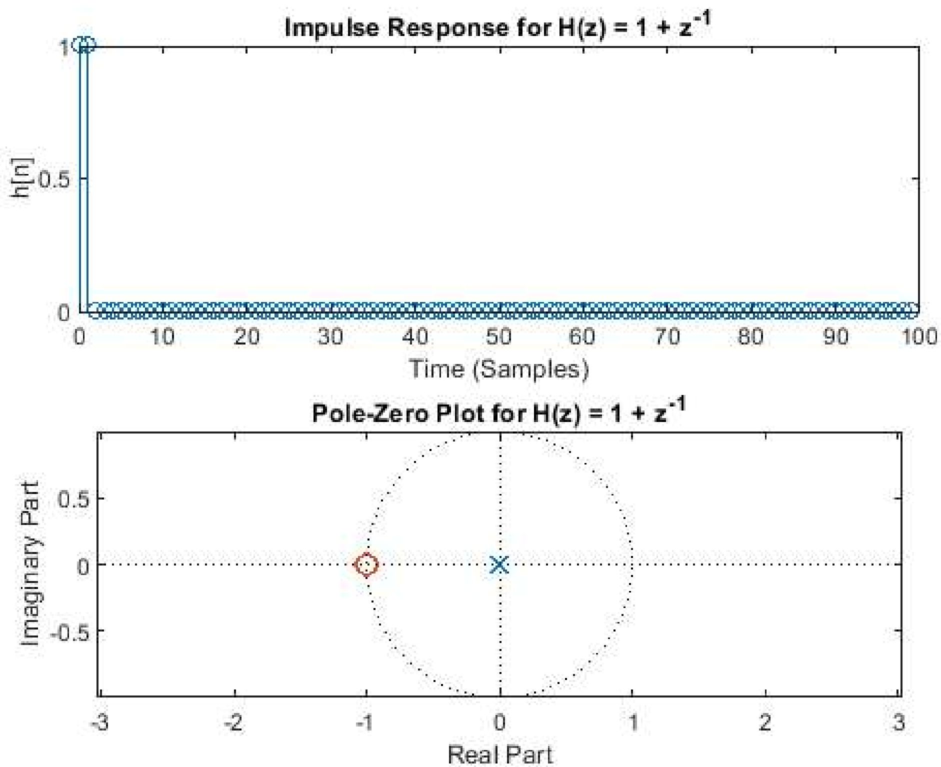
```
% Plot Pole-Zero plot
subplot(2, 1, 2);
pzplot(b1, a1);
title('Pole-Zero Plot for H(z) = 0.75z^{-4}');
```



2 (b) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define transfer function H(z) = 1 + z^(-1)
b2 = [1 1]; % Numerator (1 + z^-1)
a2 = [1]; % Denominator
impulse_response_b = filter(b2, a2, x1); % Generate impulse response

% Plot impulse response
figure(2);
subplot(2,1,1);
stem(n, impulse_response_b);
xlabel('Time (Samples)');
```

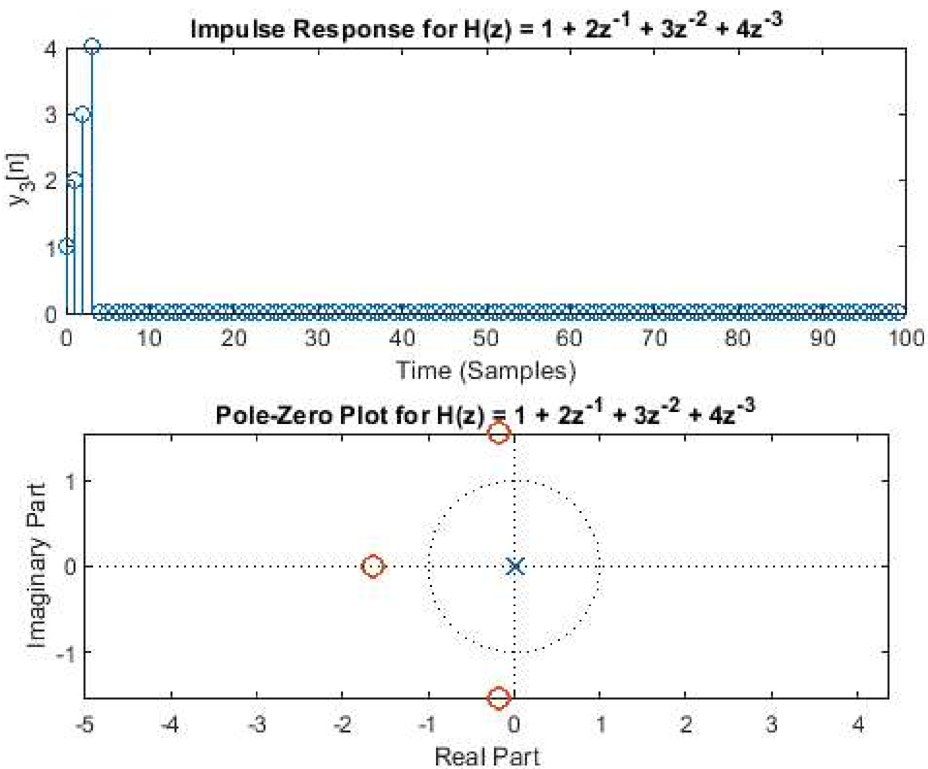
```
ylabel('h[n]');
title('Impulse Response for H(z) = 1 + z^{-1}');

% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b2, a2);
title('Pole-Zero Plot for H(z) = 1 + z^{-1}');
```



2 (c) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define the transfer function H(z) = 1 + 2z^(-1) + 3z^(-2) + 4z^(-3)
b3 = [1 2 3 4];  % Numerator
a3 = [1];        % Denominator (no feedback)
y3 = filter(b3, a3, x1); % Impulse response

% Plot impulse response
figure(3);
subplot(2,1,1);
```

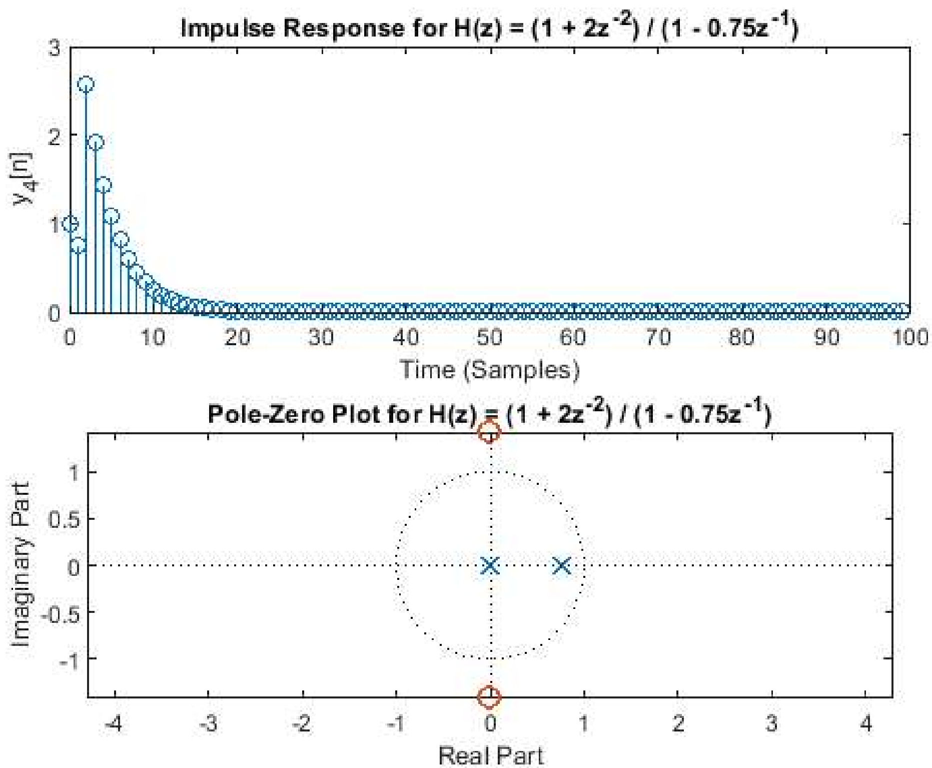
```
stem(n, y3);
xlabel('Time (Samples)');
ylabel('y_3[n]');
title('Impulse Response for H(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3}');

% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b3, a3);
title('Pole-Zero Plot for H(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3}');
```



2 (d) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define the transfer function H(z) = (1 + 2z^(-2)) / (1 - 0.75z^(-1))
b4 = [1 0 2];  % Numerator (1 + 2z^-2)
a4 = [1 -0.75];  % Denominator (1 - 0.75z^-1)
y4 = filter(b4, a4, x1); % Impulse Response

% Plot impulse response
```

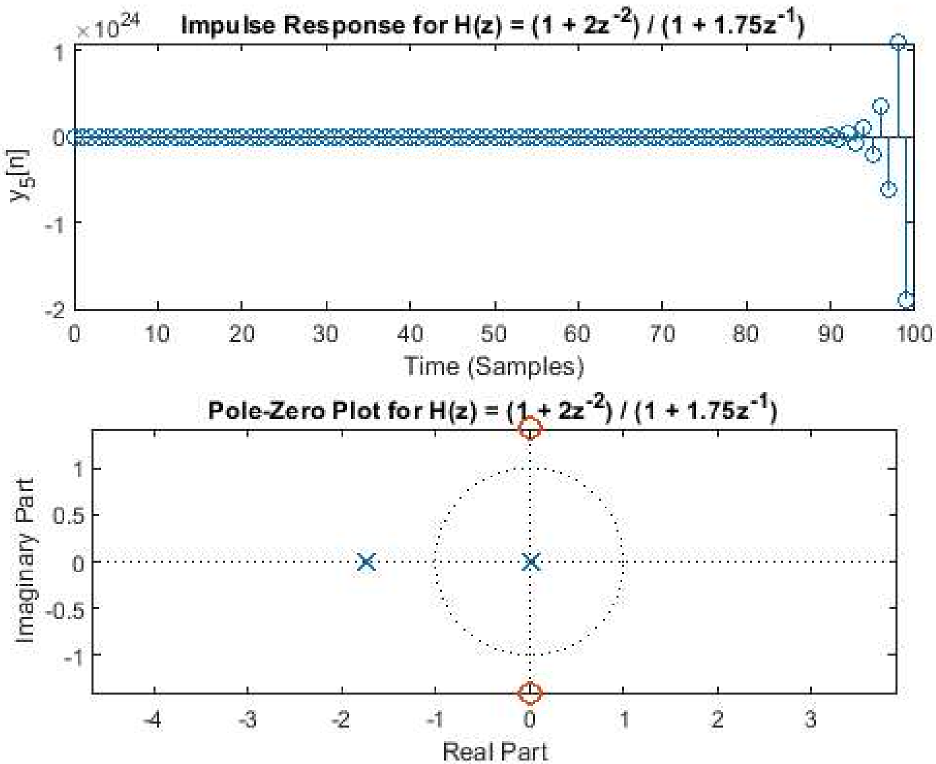
```
figure(4);
subplot(2,1,1);
stem(n, y4);
xlabel('Time (Samples)');
ylabel('y_4[n]');
title('Impulse Response for H(z) = (1 + 2z^{-2}) / (1 - 0.75z^{-1})');

% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b4, a4);
title('Pole-Zero Plot for H(z) = (1 + 2z^{-2}) / (1 - 0.75z^{-1})');
```



2 (e) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define the transfer function H(z) = (1 + 2z^(-2)) / (1 + 1.75z^(-1))
b5 = [1 0 2];  % Numerator
a5 = [1 1.75];  % Denominator
y5 = filter(b5, a5, x1); % Impulse Response
```

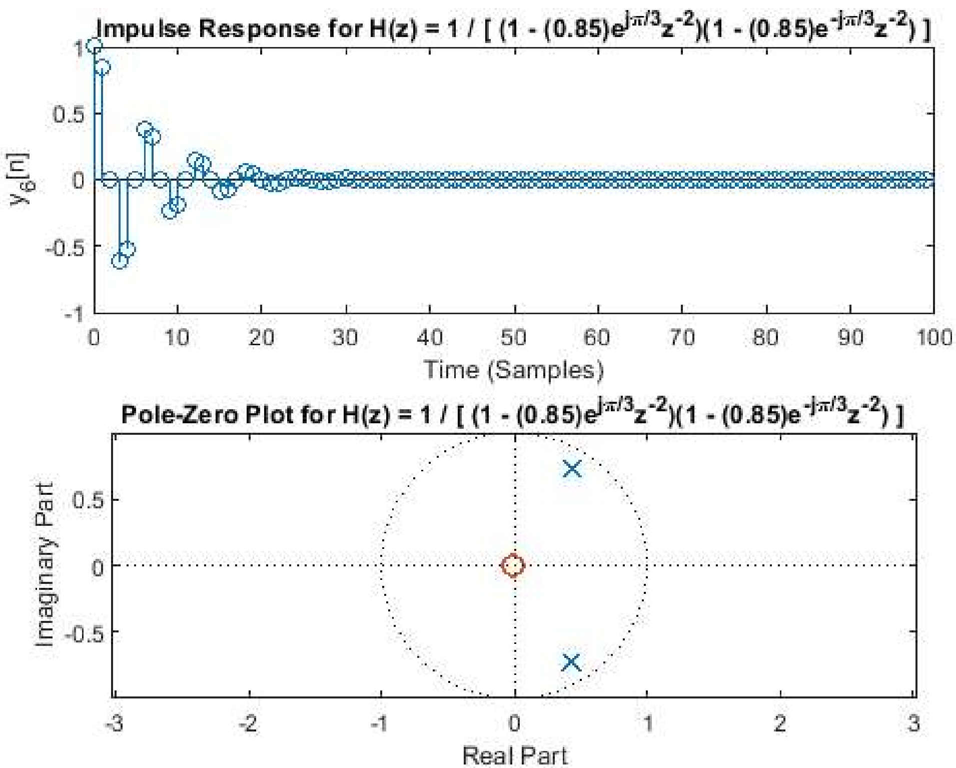
```
% Plot impulse response
figure(5);
subplot(2,1,1);
stem(n, y5);
xlabel('Time (Samples)');
ylabel('y_5[n]');
title('Impulse Response for H(z) = (1 + 2z^{-2}) / (1 + 1.75z^{-1})');

% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b5, a5);
title('Pole-Zero Plot for H(z) = (1 + 2z^{-2}) / (1 + 1.75z^{-1})');
```



2 (f) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

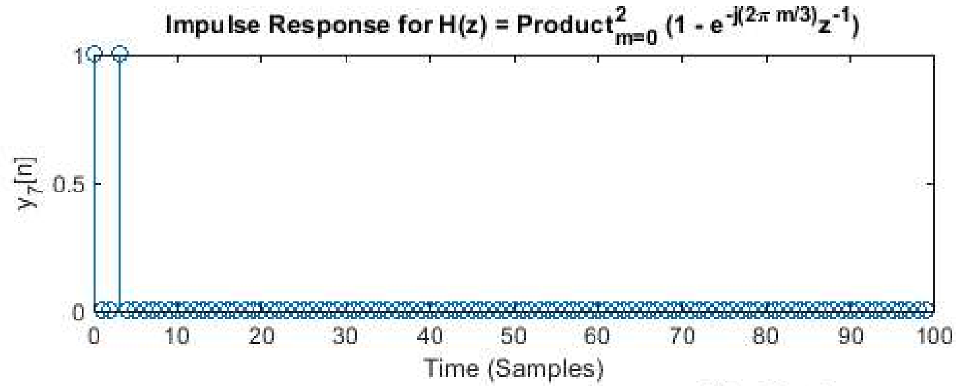
```
% Define the transfer function H(z) = (1) / [ (1 - (0.85)*exp(j*pi/3)*z^(-2)) * (1 - (0.85)*exp(-j*pi/3)*z^(-2)) ]
b6 = [1];  % Numerator
```

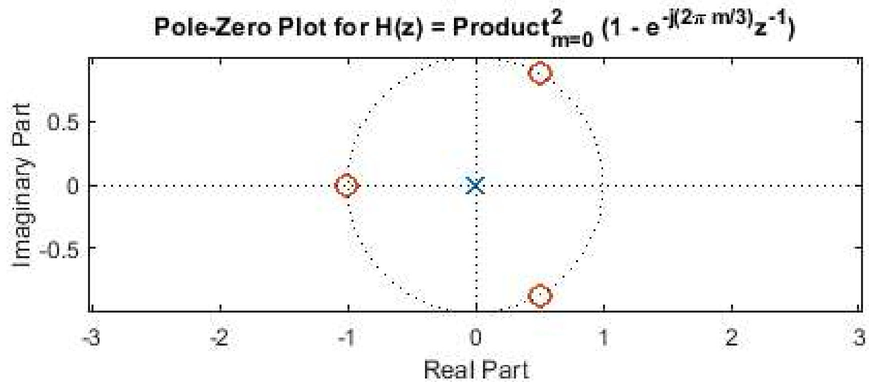
```
a6 = [1 -0.85 0.85^2];  % Denominator for second-order system with complex conjugate poles
y6 = filter(b6, a6, x1); % Impulse Response

% Plot impulse response
figure(6);
subplot(2,1,1);
stem(n, y6);
xlabel('Time (Samples)');
ylabel('y_6[n]');
title('Impulse Response for H(z) = 1 / [ (1 - (0.85)e^{j\pi/3}z^{-2})(1 - (0.85)e^{-j\pi/3}z^{-2}) ]');

% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b6, a6);
title('Pole-Zero Plot for H(z) = 1 / [ (1 - (0.85)e^{j\pi/3}z^{-2})(1 - (0.85)e^{-j\pi/3}z^{-2}) ]');
```



2 (g) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define the transfer function H(z) = Product_{m=0}^{2} (1 - e^{-j(2*pi*m/3)}z^{-1})
b7 = 1;

for m = 0:2
    b7 = conv(b7, [1 exp(-1j*2*pi*(m-1) / 3)]);
end

a7 = [1];  % Denominator (no feedback)
y7 = filter(b7, a7, x1); % Impulse Response

% Plot impulse response
figure(7);
subplot(2,1,1);
stem(n, y7);
xlabel('Time (Samples)');
ylabel('y_7[n]');
title('Impulse Response for H(z) = Product_{m=0}^{2} (1 - e^{-j(2\pi m/3)}z^{-1})');

% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b7, a7);
title('Pole-Zero Plot for H(z) = Product_{m=0}^{2} (1 - e^{-j(2\pi m/3)}z^{-1})');
```



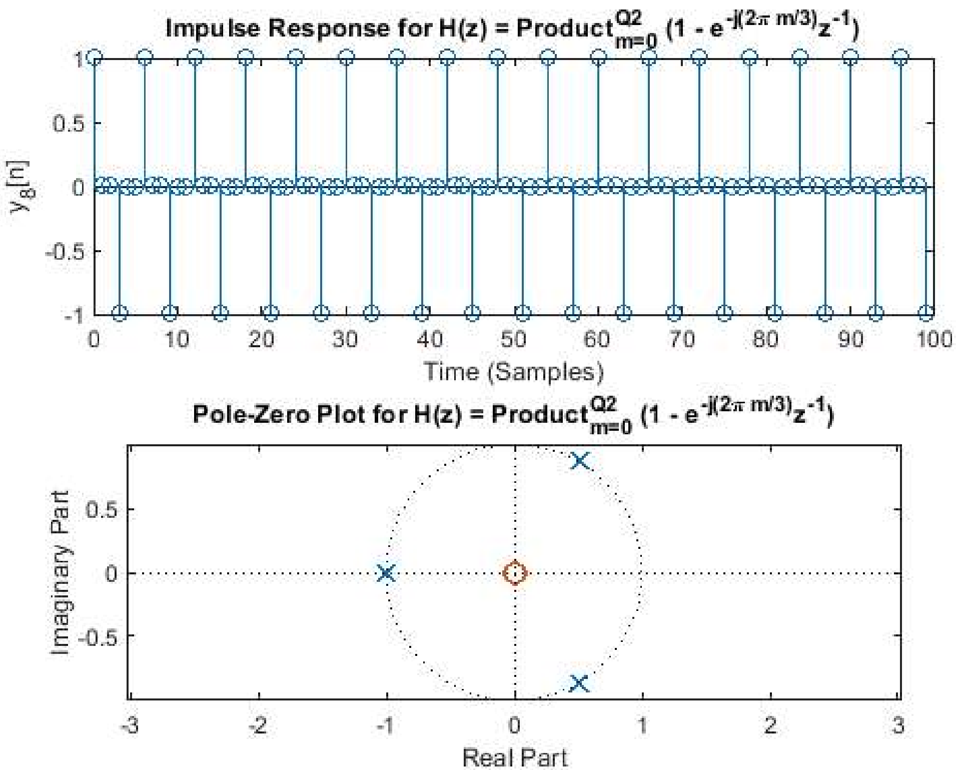


2 (h) PLOT IMPULSE RESPONSE AND POLE-ZERO PLOT

```
% Define the transfer function H(z) = Product_{m=0}^{Q2} (1 - e^{-j(2*pi*m/3)}z^{-1})
Q1 = 3;
b8 = 1;
a8 = 1;
for m = 1:Q1
    a8 = conv(a8, [1 exp(-1j*2*pi*(m-1)/3)]);
end
y8 = filter(b8, a8, x1); % Impulse Response

% Plot impulse response
figure(8);
subplot(2,1,1);
stem(n, y8);
xlabel('Time (Samples)');
ylabel('y_8[n]');
title('Impulse Response for H(z) = Product_{m=0}^{Q2} (1 - e^{-j(2\pi m/3)}z^{-1})');
```

```
% Plot Pole-Zero plot
subplot(2,1,2);
pzplot(b8, a8);
title('Pole-Zero Plot for H(z) = Product_{m=0}^{Q2} (1 - e^{-j(2\pi m/3)}z^{-1})');
```

```
Warning: Using only the real component of complex data.
```



QUESTION 3: MORE Z-TRANSFORM

3 (a) ANSWER QUESTION

For what pole-zero conditions is the impulse response unstable (i.e., goes to ∞ as □ → ∞)?

```
% The impulse response is unstable if the system has a pole that is on or outside the unit
% circle in the z-plane, or when the magnitude of the pole is greater than
% or equal to 1 bcecause it causes exponential growth in the impulse response over
```

```
% time, causing it to go to infinity.

% An example of this would be in parts 2(e) since one of the poles are outside the unit circle.
```

3 (b) ANSWER QUESTION

```
% For what pole-zero conditions is the impulse response stable (i.e., goes to 0 as n→∞)?

% The impulse response is stable if all poles are inside the unit circle
% because this means the pole magnitudes are less than one to ensure that
% the response decays exponentially to approach zero over time.

% An example would be part 2(d) where the poles from the pole-zero graph are inside the unit circle
```

3 (c) ANSWER QUESTION

For what pole-zero conditions is the impulse response critically stable (i.e., steady amplitude as n→∞)?

```
% The impulse response is critically stable if the poles are exactly on the
% unit circle, basically if their magnitude is 1 because this means there
% is a steady oscillation with constant amplitude

% An example of this would be 2(h) because the graph is not showing decay
% or growth, it remains stable and all the poles are exactly on the unit
% circle in the z plane
```

3 (d) ANSWER QUESTION

For what pole-zero conditions is the impulse response finite in length?

```
% The impulse response is finite if the system is a finite impulse response
% (FIR) system when there are no poles or zeros in the system or all poles are at
% the origin to indicate the the termination after a finite number of
% samples.

% An example of this would be 2(a) because it has a finite number of inputs since
% the only pole it has is at the origin of the unit circle.
```

3 (e) ANSWER QUESTION

For what pole-zero conditions is the impulse response infinite in length?

```
% Impulse is infinite if the length of the systme is an infinite impulse
% response (IIR) system where all poles are inside the unit circle and none at the origin.

% An example of this would be 2(f) because the poles lie inside the unit
% circle but none at the origin.
```

3 (f) ANSWER QUESTION

For what pole-zero conditions is the impulse response periodic (with a frequency > 0)?

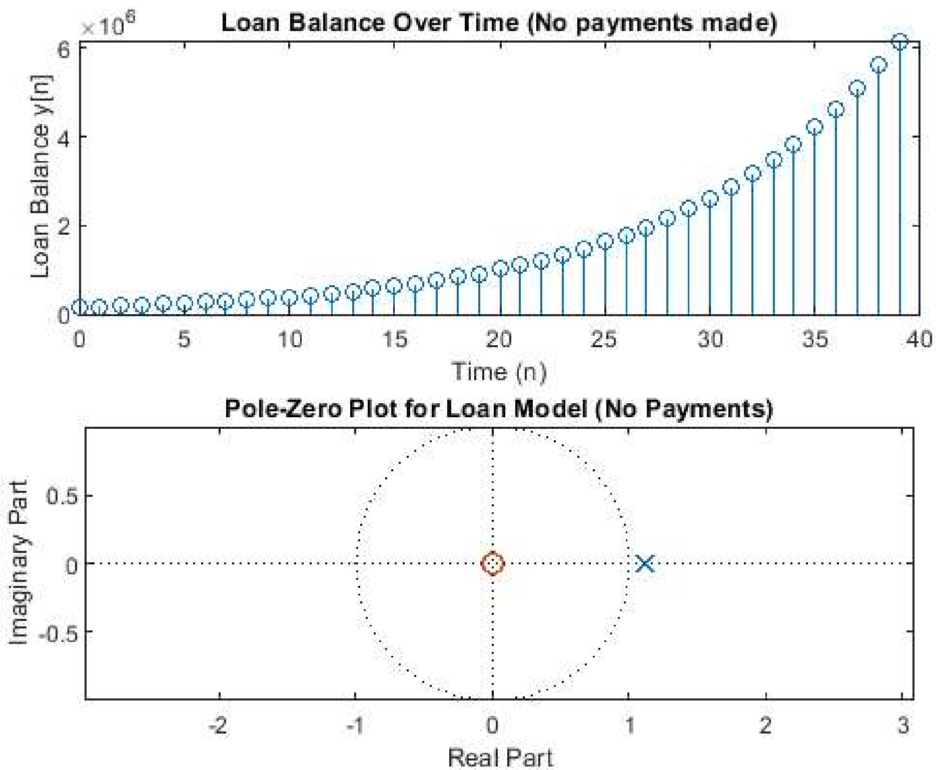
```
% The impulse response is periodic if the poles are on the unit circle and
% the magnitude is contant so the system has complex conjugate poles since this causes an oscillatory
% response with frequency corresponding to angular separation of poles.

% An example of this would be 2(h) since the poles are on the unit circle
```

QUESTION 4: LOAN DIFFERENCE EQUATION

4 (a) PLOT OUTPUT AND POLE-ZERO PLOT

```
% Loan parameters
N = 40; % 40 years
n = 0:(N-1); % Time vector
x1 = zeros(N, 1);
x1(1) = 150000; % Initial loan amount at n=0
alpha = 0.1;  % interest rate
b1 = 1;  % Numerator (no FIR part)
a1 = [1 -(1 + alpha)];  % Denominator for IIR system

y1 = filter(b1, a1, x1);

% Plot the loan balance over time
figure;
subplot(2,1,1);
stem(n, y1);
xlabel('Time (n)');
ylabel('Loan Balance y[n]');
title('Loan Balance Over Time (No payments made)');

% Pole-Zero plot
```

```
subplot(2,1,2);
pzplot(b1, a1);
title('Pole-Zero Plot for Loan Model (No Payments)');
```



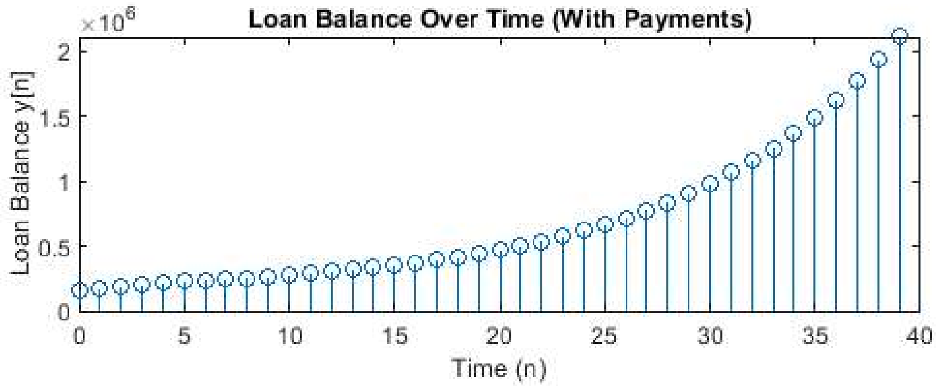
4 (b) ANSWER QUESTION

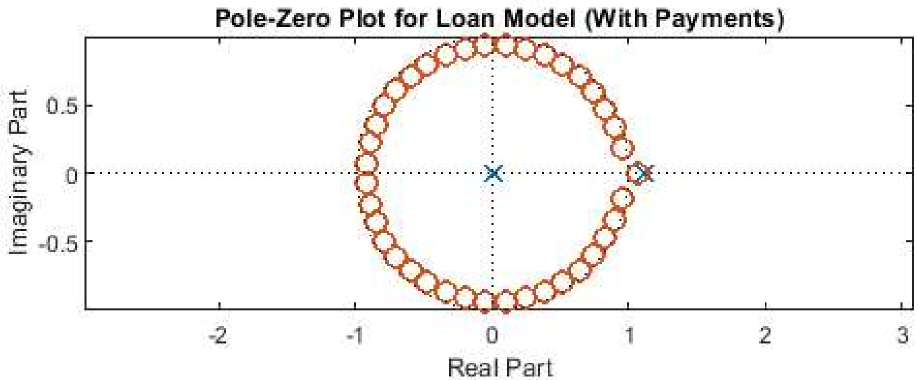
```
% The loan will exceed 1 million dollars after some time between 19 - 20 years this can be shown
% from the plot where the loan value is over the 1 million mark.
```

4 (c) PLOT OUTPUT AND POLE-ZERO PLOT

```
% Loan parameters
N = 40; % 40 years
n = 0:(N-1); % Time vector
R0 = 5; % starts payments after 5 years
x2 = zeros(N, 1);
```

```
x2(1) = 150000; % Initial loan amount at n=0
alpha = 0.1;  % interest rate
beta = 0.1;
b2 = [1 zeros(1,R0-1) (-beta*ones(1, N-R0))];  % Numerator (no FIR part)
a2 = [1 -(1 + alpha)];  % Denominator for IIR system

y2 = filter(b2, a2, x2);

% Plot the loan balance over time
figure;
subplot(2,1,1);
stem(n, y2);
xlabel('Time (n)');
ylabel('Loan Balance y[n]');
title('Loan Balance Over Time (With Payments)');

% Pole-Zero plot
subplot(2,1,2);
pzplot(b2, a2);
title('Pole-Zero Plot for Loan Model (With Payments)');
```





4 (d) ANSWER QUESTION

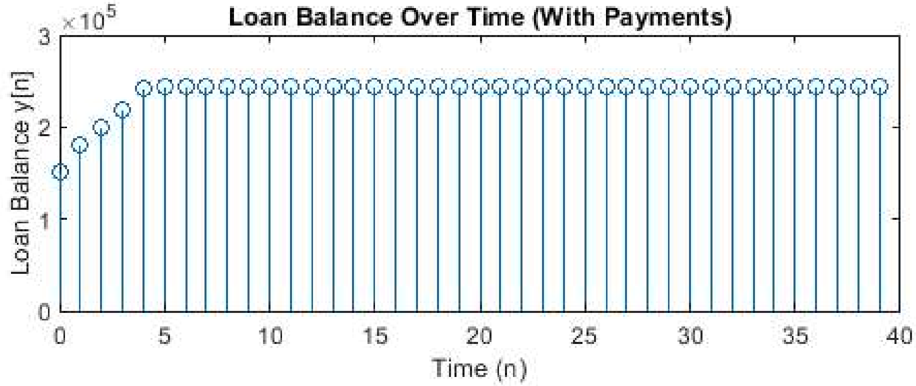
```
% The loan goes towards infinity where they would owe 1 million dollars
% sometime between 30 to 31 years.
```

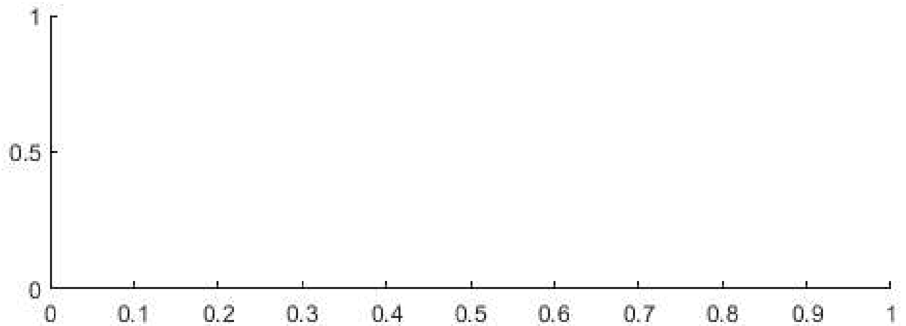
4 (e) ANSWER QUESTION

```
% The percentage for the loan to level out would be about 14.641% you get
% this by taking the current loan amount after the five years and
% multiplying it by the alpha value to see how much interest is needed.
% Then you take that value and divide it by the original loan amount that
% beta is based on since you just need to pay the interest to level out.

% If you start paying at the start in order to level out you just need to
% match the percentage of interest they would be the same, 10%. If you delayed
% payment by 4 years you would need to pay about 13.31% of the original loan to level out, so you
% would be paying about 3.31% more.
```

4 (f) PLOT OUTPUT AND POLE-ZERO PLOT

```
% Loan parameters
N = 40; % 40 years
n = 0:(N-1); % Time vector
Q = 5; % starts payments after 5 years
x3 = zeros(N, 1);
x3(1) = 150000; % Initial loan amount at n=0
alpha = 0.1;  % interest rate
gamma = 0.1; % payment rate

% Filter coefficients
b3 = [1];

for m = 1:Q-1
    b3 = [b3 gamma*(1+alpha)^(m-1)]; % Coefficients from recuuring gamma * (a+alpha)^(m-1)
end

a3 = [1 -(1 + alpha) gamma];  % Denominator for IIR system

y3 = filter(b3, a3, x3);

% Plot the loan balance over time
figure;
subplot(2,1,1);
stem(n, y3);
xlabel('Time (n)');
ylabel('Loan Balance y[n]');
title('Loan Balance Over Time (With Payments)');

% Pole-Zero plot
subplot(2,1,2);
pzplot(b3, a3);
title('Pole-Zero Plot for Loan Model (With Payments)');
```





4 (g) ANSWER QUESTION

```
% The loan levels out at the amount $244,017 at year 9.
```

ALL FUNCTIONS SUPPORTING THIS CODE %% % ==

NOTE: YOU DO NOT NEED TO ADD COMMENTS IN THE CODE BELOW. WE JUST NEEDED POLE-ZERO PLOTTING CODE AND THUS WROTE IT. ==

```
function pzplot(b,a)
% PZPLOT(B,A)  plots the pole-zero plot for the filter described by
% vectors A and B.  The filter is a "Direct Form II Transposed"
% implementation of the standard difference equation:
%
%    a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
%                          - a(2)*y(n-1) - ... - a(na+1)*y(n-na)
%
```

```
    % MODIFY THE POLYNOMIALS TO FIND THE ROOTS
    b1 = zeros(max(length(a),length(b)),1); % Need to add zeros to get the right roots
    a1 = zeros(max(length(a),length(b)),1); % Need to add zeros to get the right roots
    b1(1:length(b)) = b;    % New a with all values
    a1(1:length(a)) = a;    % New a with all values

    % FIND THE ROOTS OF EACH POLYNOMIAL AND PLOT THE LOCATIONS OF THE ROOTS
    h1 = plot(real(roots(a1)), imag(roots(a1)));
    hold on;
    h2 = plot(real(roots(b1)), imag(roots(b1)));
    hold off;

    % DRAW THE UNIT CIRCLE
    circle(0,0,1)

    % MAKE THE POLES AND ZEROS X's AND O's
    set(h1, 'LineStyle', 'none', 'Marker', 'x', 'MarkerFaceColor','none', 'linewidth', 1.5, 'markersize', 8);
    set(h2, 'LineStyle', 'none', 'Marker', 'o', 'MarkerFaceColor','none', 'linewidth', 1.5, 'markersize', 8);
    axis equal;

    % DRAW VERTICAL AND HORIZONTAL LINES
    xminmax = xlim();
    yminmax = ylim();
    line([xminmax(1) xminmax(2)],[0 0], 'linestyle', ':', 'linewidth', 0.5, 'color', [1 1 1]*.1)
    line([0 0],[yminmax(1) yminmax(2)], 'linestyle', ':', 'linewidth', 0.5, 'color', [1 1 1]*.1)

    % ADD LABELS AND TITLE
    xlabel('Real Part')
    ylabel('Imaginary Part')
    title('Pole-Zero Plot')

end


function circle(x,y,r)
% CIRCLE(X,Y,R)  draws a circle with horizontal center X, vertical center
% Y, and radius R.
%

    % ANGLES TO DRAW
    ang=0:0.01:2*pi;

    % DEFINE LOCATIONS OF CIRCLE
    xp=r*cos(ang);
```

```
    yp=r*sin(ang);

    % PLOT CIRCLE
    hold on;
    plot(x+xp,y+yp, ':', 'linewidth', 0.5, 'color', [1 1 1]*.1);
    hold off;

end
```

Published with MATLAB® R2024a