DIGITAL SIGNAL PROCESSING – EC5011

LABORATORY SESSION 2

DIGITAL SIGNAL PROCESSING THEORY AND APPLICATION

WIMALASOORIYA G.H.N.P.D.

2022/E/039

GROUP CG03

SEMESTER 05

28 APRIL 2025

PART 1: ANALYSIS OF A DIGITAL FILTER

$$H(Z) = \frac{Z^4 + 10Z^2 + 3Z + 28}{Z^4 + 0.35Z^2 - 0.35Z + 0.425}$$

Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Signal Processing\Labs\L...

```matlab
% WIMALASOORIYA G.H.N.P.D.
% 2022E039

% PART 01
% -------------------------

% Define numerator and denominator
num = [1 0 10 3 28];
den = [1 0 0.35 -0.35 0.425];

% Plot Pole-Zero Map
figure; % <-- ADD THIS to open a new figure window
zplane(num, den);
title('Pole-Zero Map of H(z)');
xlabel('Real part');
ylabel('Imaginary part');
grid on;

% Define coefficients
b = [1 0 10 3 28];       % Numerator coefficients (highest to lowest power)
a = [1 0 0.35 -0.35 0.425]; % Denominator coefficients

% Number of samples
N = 50; % length of impulse response

% Generate impulse input
x = zeros(1, N);
x(1) = 1; % impulse at n=0

% Initialize output
y = zeros(1, N);

% Manual filtering using the difference equation
for n = 1:N
    % Feedforward part (numerator)
    for k = 1:length(b)
        if n-k+1 > 0
            y(n) = y(n) + b(k) * x(n-k+1);
        end
    end
    % Feedback part (denominator)
    for k = 2:length(a)
        if n-k+1 > 0
            y(n) = y(n) - a(k) * y(n-k+1);
        end
    end
end

% Plot the impulse response
figure; % <-- ADD THIS also if you want impulse response in a new figure
stem(0:N-1, y, 'filled');
title('Impulse Response of H(Z)');
xlabel('n');
ylabel('h[n]');
grid on;

%-------------------------------------------------------
% Plot Magnitude and Phase Response using the Impulse Response
% Take FFT of the impulse response
Y = fft(y, 512); % 512-point FFT for better resolution

% Frequency axis
f = linspace(0, 1, 512); % Normalized frequency (0 to 1 corresponds to 0 to π rad/sample)

% Plot magnitude response
figure;
plot(f, abs(Y));
title('Magnitude Response from Impulse Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('|H(e^{j\omega})|');
grid on;

% Plot phase response
figure;
plot(f, angle(Y));
title('Phase Response from Impulse Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
grid on;

%-------------------------------------------------------

%-------------------------------------------------------
%Find Frequency Response Directly using freqz (without impulse response)

% Directly find the frequency response using freqz
[H, w] = freqz(b, a, 512); % 512 points

% Plot magnitude response
figure;
plot(w/pi, abs(H));
title('Magnitude Response using freqz');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('|H(e^{j\omega})|');
grid on;

% Plot phase response
figure;
plot(w/pi, angle(H));
title('Phase Response using freqz');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
grid on;

%-------------------------------------------------------
```

FIGURE 01: CODE

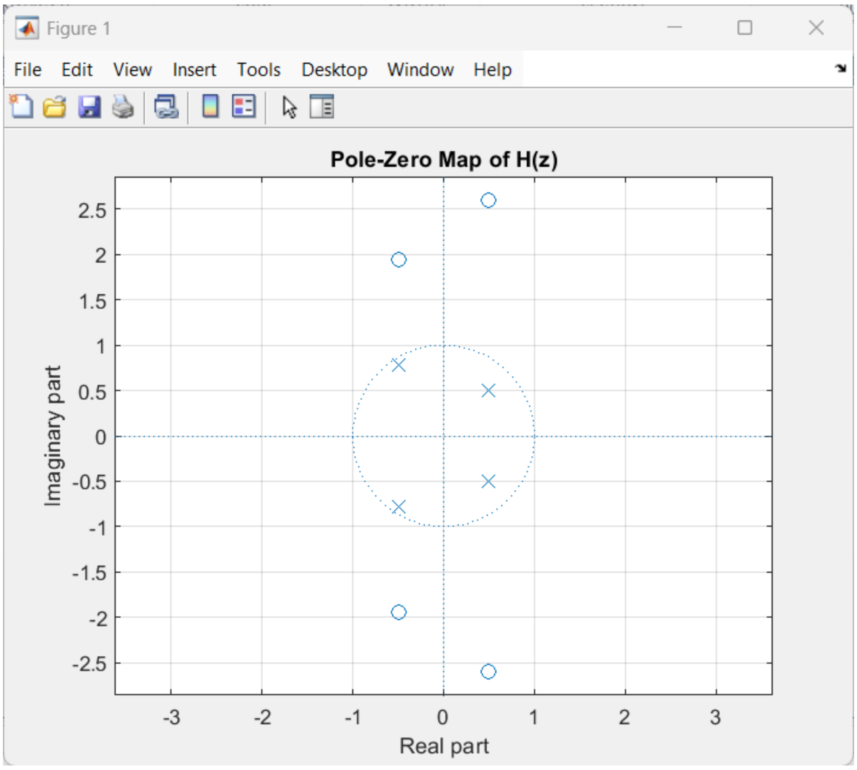

FIGURE 02: ZEROS & POLES

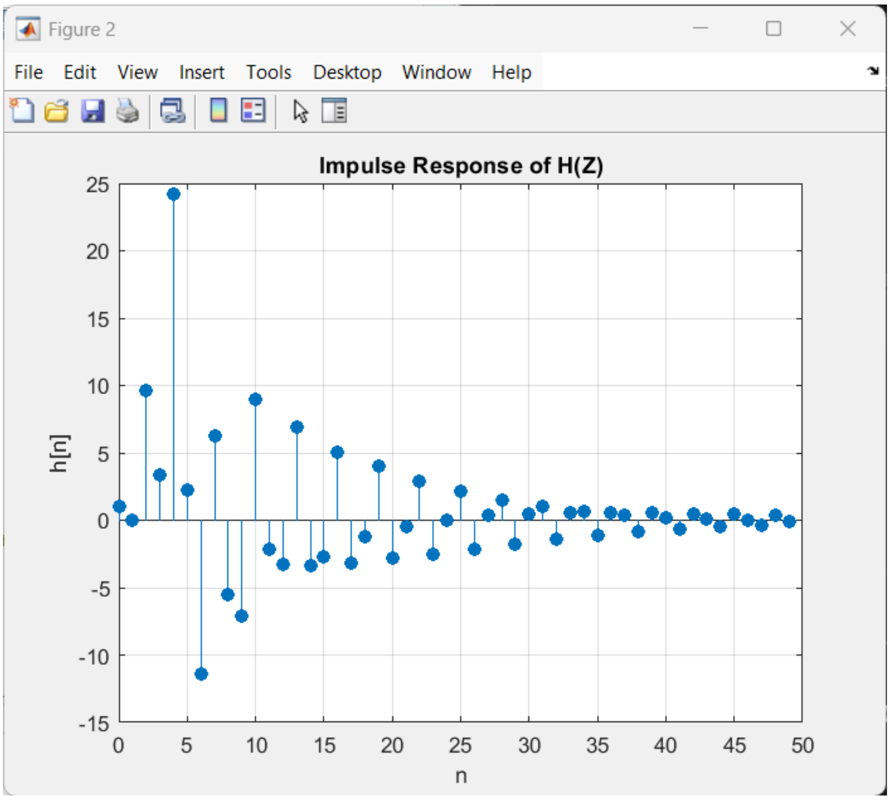

FIGURE 03: IMPULSE RESPONSE

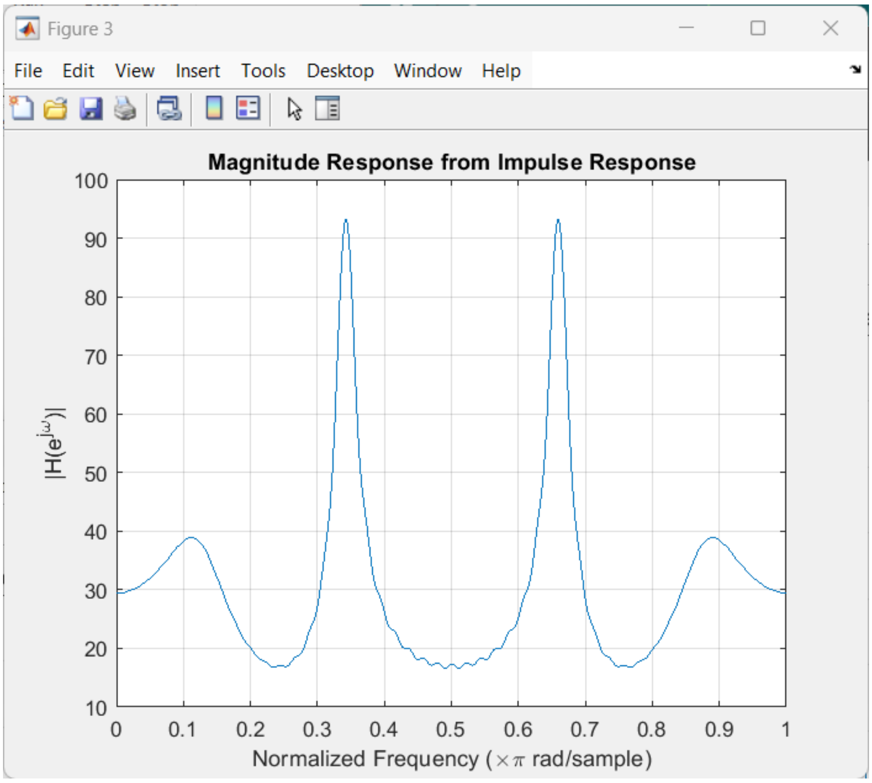

FIGURE 04: MAGNITUDE RESPONSE

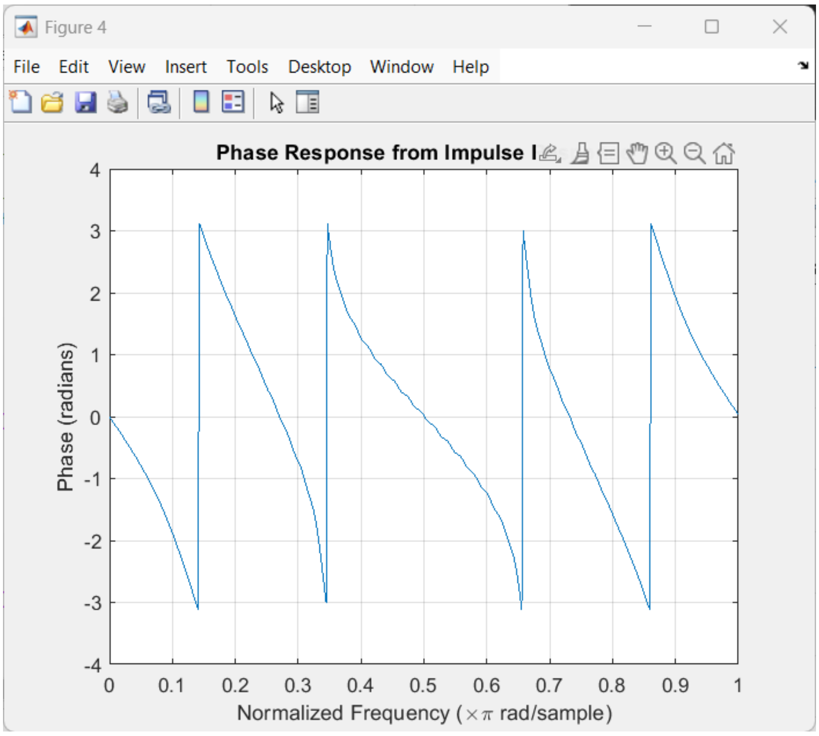

FIGURE 05: PHASE RESPONSE

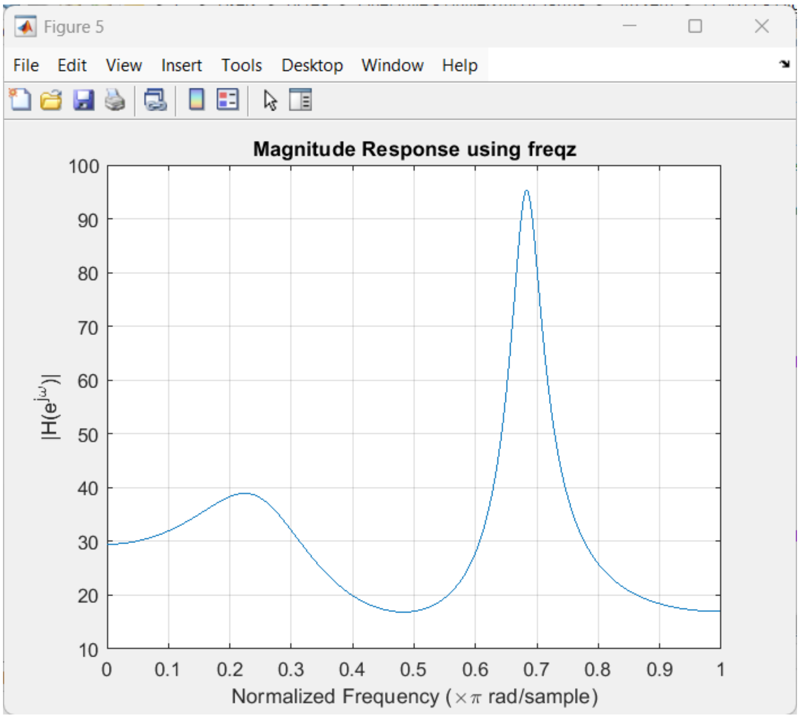

FIGURE 06: MAGNITUDE RESPONSE USING freqz

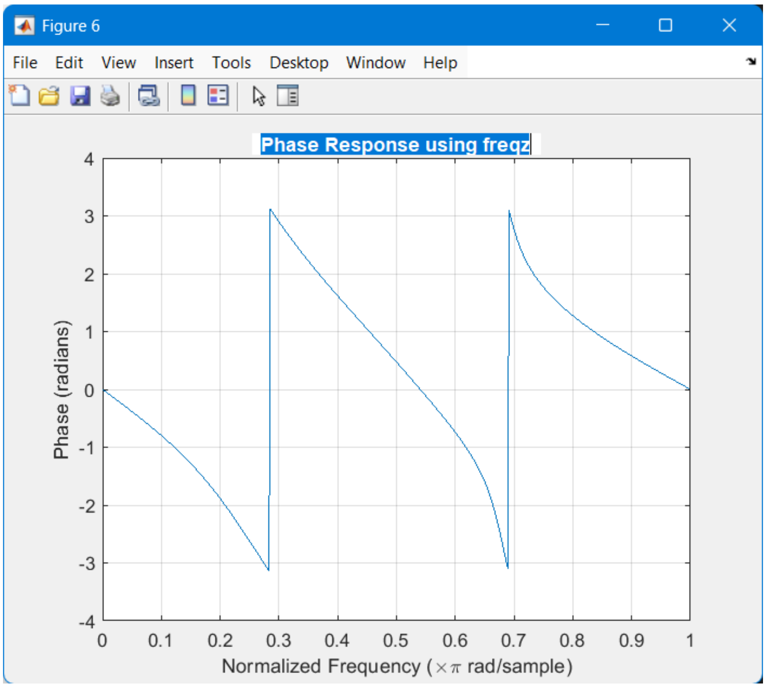

FIGURE 07: PHASE RESPONSE USING freqz

PART2: DIGITAL FILTER DESIGN IN MATLAB

Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Signal Processi

```matlab
% WIMALASOORIYA G.H.N.P.D.
% 2022E039

% Define values of 'a'
a_values = [0.9, -0.9];

% Frequency range
[W,~] = freqz(1,[1],512); % just to get W values for plotting

for i = 1:length(a_values)
    a = a_values(i);

    % Define H(z) and G(z)
    b_H = 1;          % Numerator of H(z)
    a_H = [1 a];      % Denominator of H(z)

    b_G = [1 -a];     % Numerator of G(z)
    a_G = 1;          % Denominator of G(z) (FIR)

    % Frequency response for H(z)
    [H,w] = freqz(b_H, a_H, 512);

    % Frequency response for G(z)
    [G,~] = freqz(b_G, a_G, 512);

    % Plot for H(z)
    figure;
    subplot(2,1,1);
    plot(w/pi, abs(H));
    title(['Magnitude Response of H(z) for a = ' num2str(a)]);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('|H(e^{j\omega})|');
    grid on;

    subplot(2,1,2);
    plot(w/pi, angle(H));
    title(['Phase Response of H(z) for a = ' num2str(a)]);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('Phase (radians)');
    grid on;

    % Plot for G(z)
    figure;
    subplot(2,1,1);
    plot(w/pi, abs(G));
    title(['Magnitude Response of G(z) for a = ' num2str(a)]);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('|G(e^{j\omega})|');
    grid on;

    subplot(2,1,2);
    plot(w/pi, angle(G));
    title(['Phase Response of G(z) for a = ' num2str(a)]);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('Phase (radians)');
    grid on;
end
```

FIGURE 08: CODE

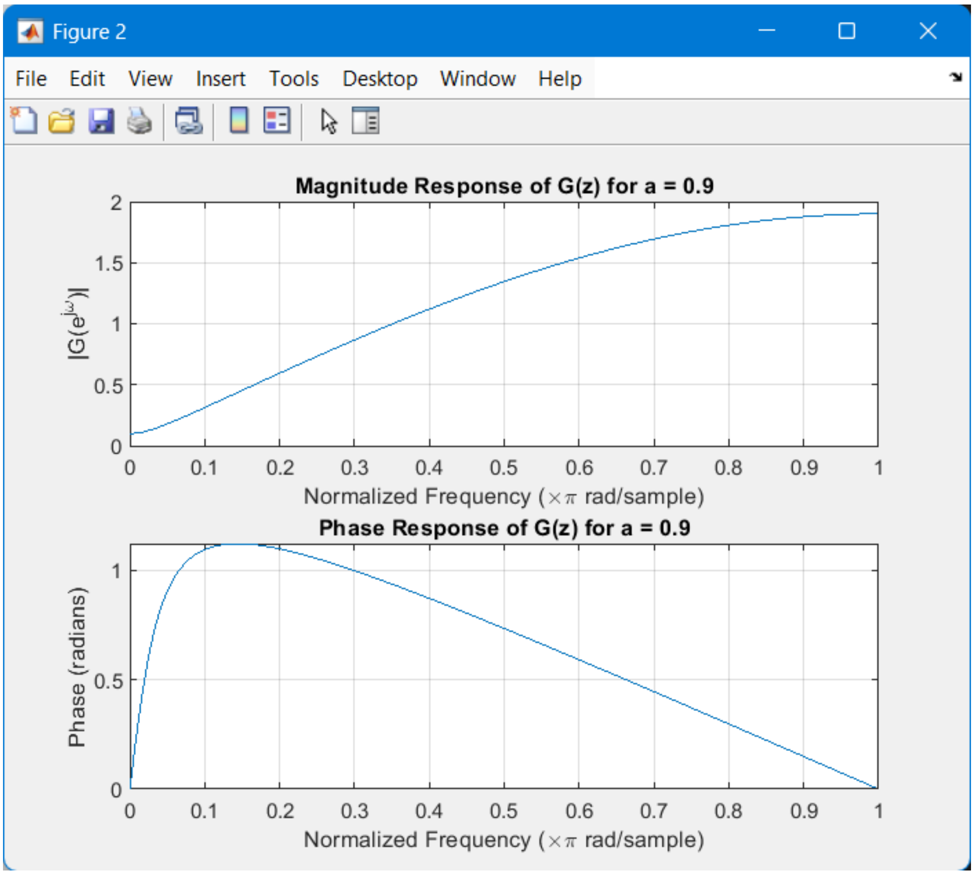

FIGURE 09: MAGNITUDE & PHASE RESPONSE OF G(Z) FOR A = 0.9

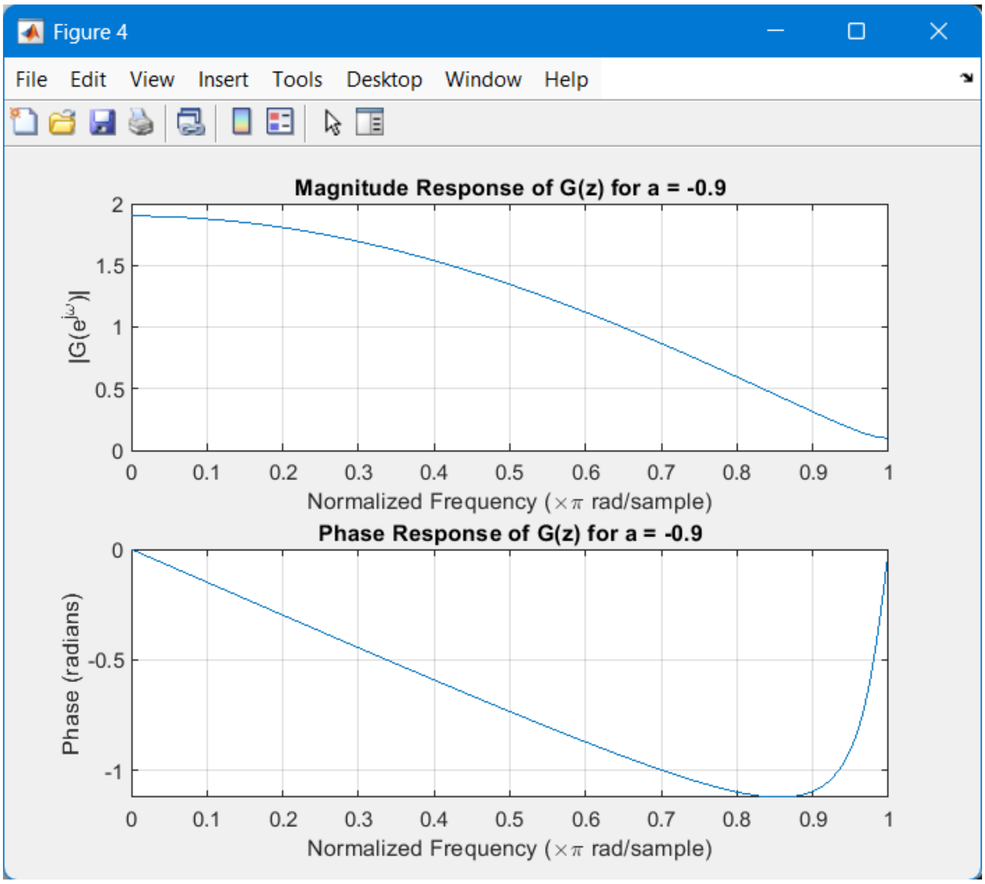

FIGURE 10: MAGNITUDE & PHASE RESPONSE OF G(Z) FOR A = -0.9

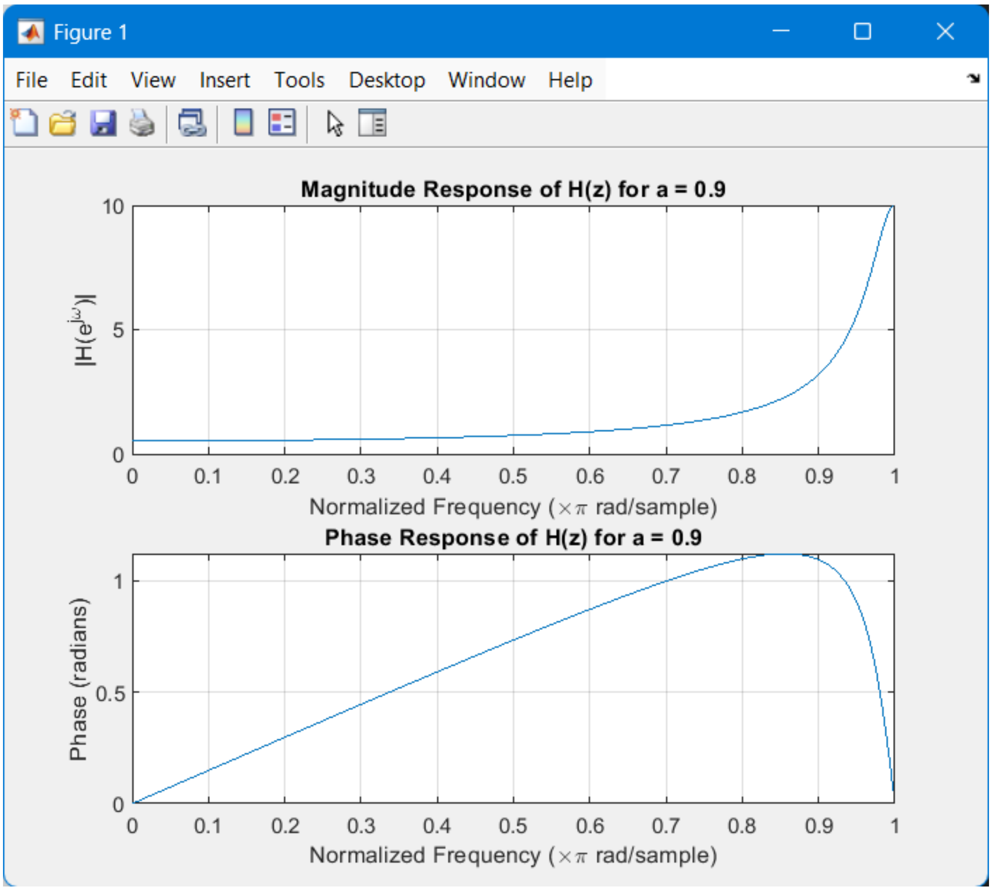

FIGURE 11: MAGNITUDE & PHASE RESPONSE OF H(Z) FOR A = 0.9

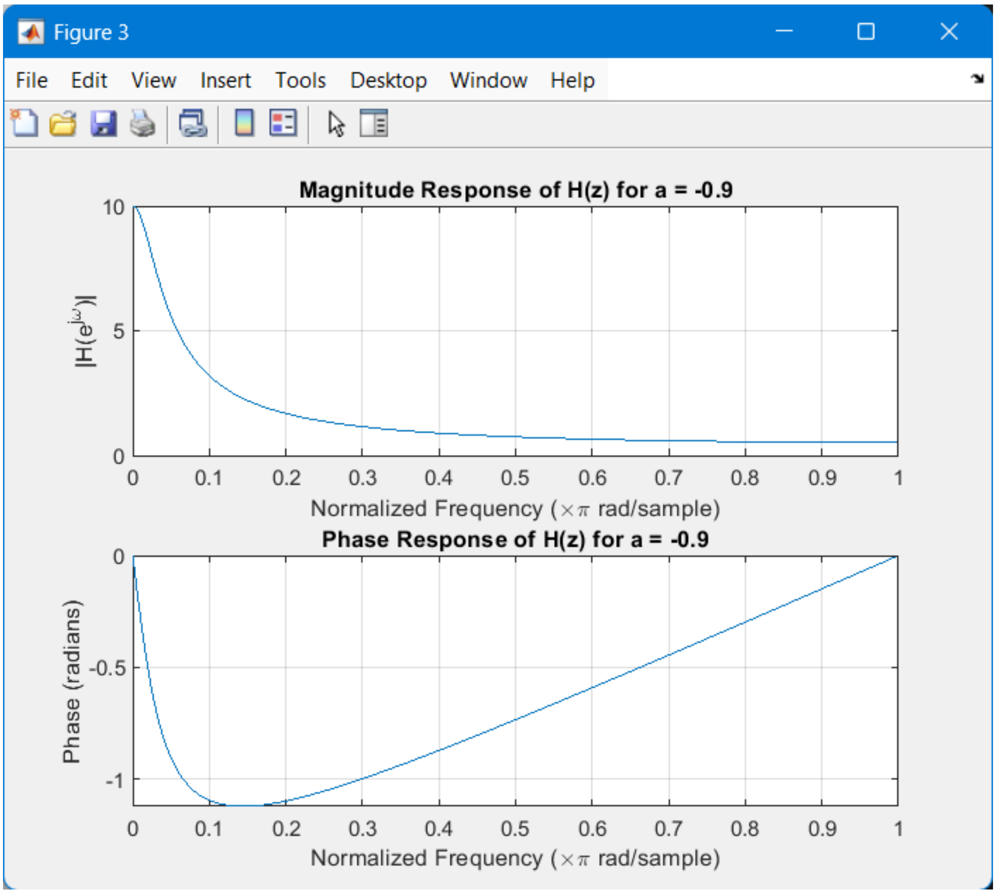

FIGURE 12: MAGNITUDE & PHASE RESPONSE OF H(Z) FOR A = -0.9

PART3: ALLPASS FILTER

```
% WIMALASOORIYA G.H.N.P.D.
% 2022E039

% Define numerator and denominator
b = [1 0 0 0 4];  % Z^4 + 4
a = [4 0 0 0 1];  % 4Z^4 + 1

% Plot pole-zero map
figure;
zplane(b, a);
title('Pole-Zero Map of H(z)');
xlabel('Real Part');
ylabel('Imaginary Part');
grid on;

% Find zeros (roots of numerator)
zeros_H = roots(b);

% Find poles (roots of denominator)
poles_H = roots(a);

% Frequency response using freqz
[H, w] = freqz(b, a, 512);

% Magnitude Response
figure;
plot(w/pi, abs(H));
title('Magnitude Response of H(z)');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('|H(e^{j\omega})|');
grid on;

% Phase Response
figure;
plot(w/pi, angle(H));
title('Phase Response of H(z)');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
grid on;
```

FIGURE 13: CODE

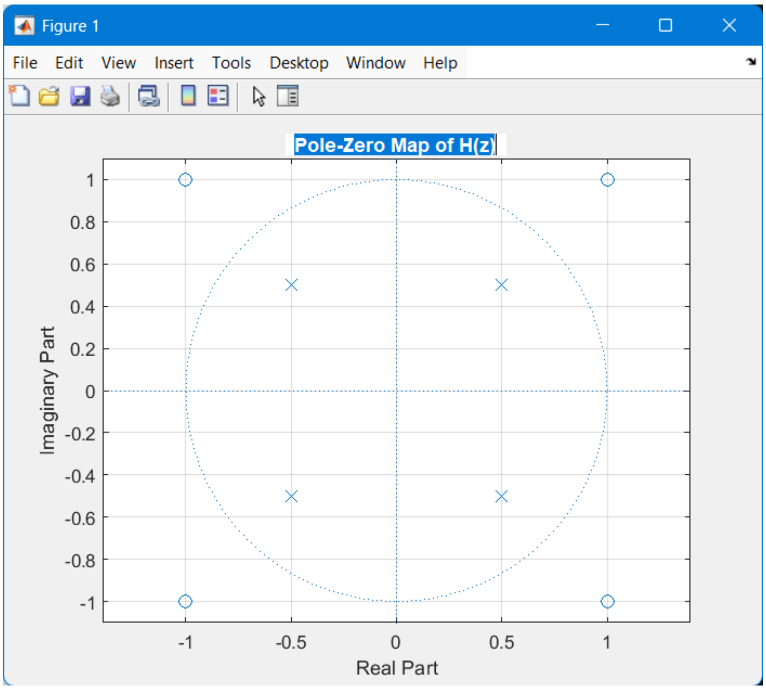

FIGURE 14: POLE-ZERO MAP OF H(Z)

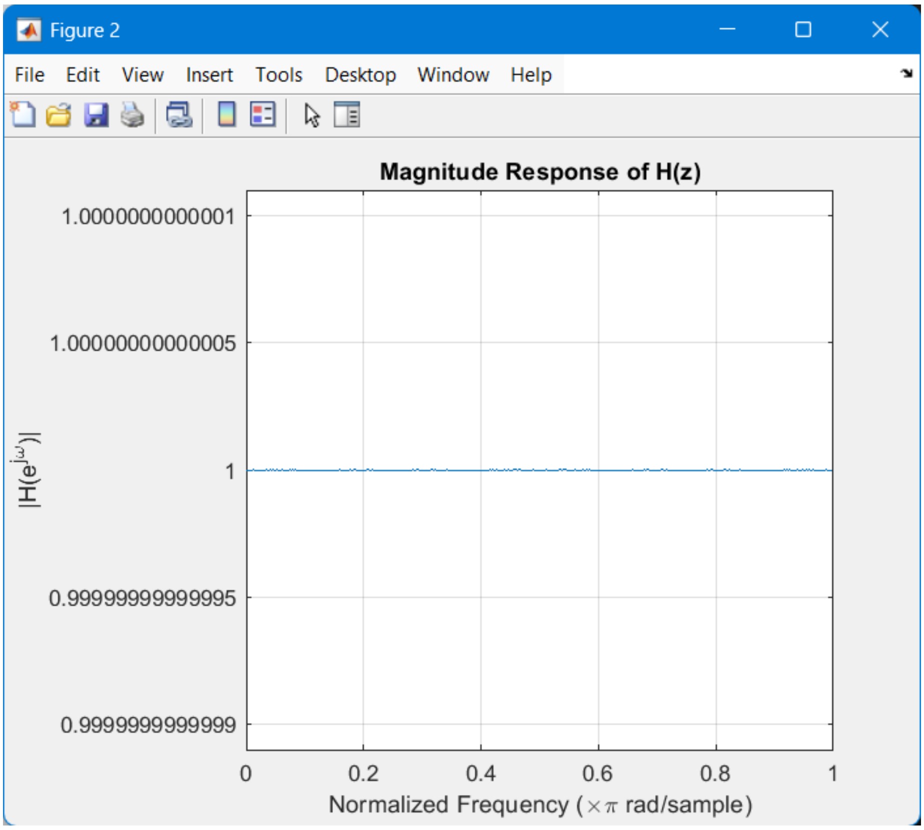

FIGURE 15: MAGNITUDE RESPONSE OF H(Z)

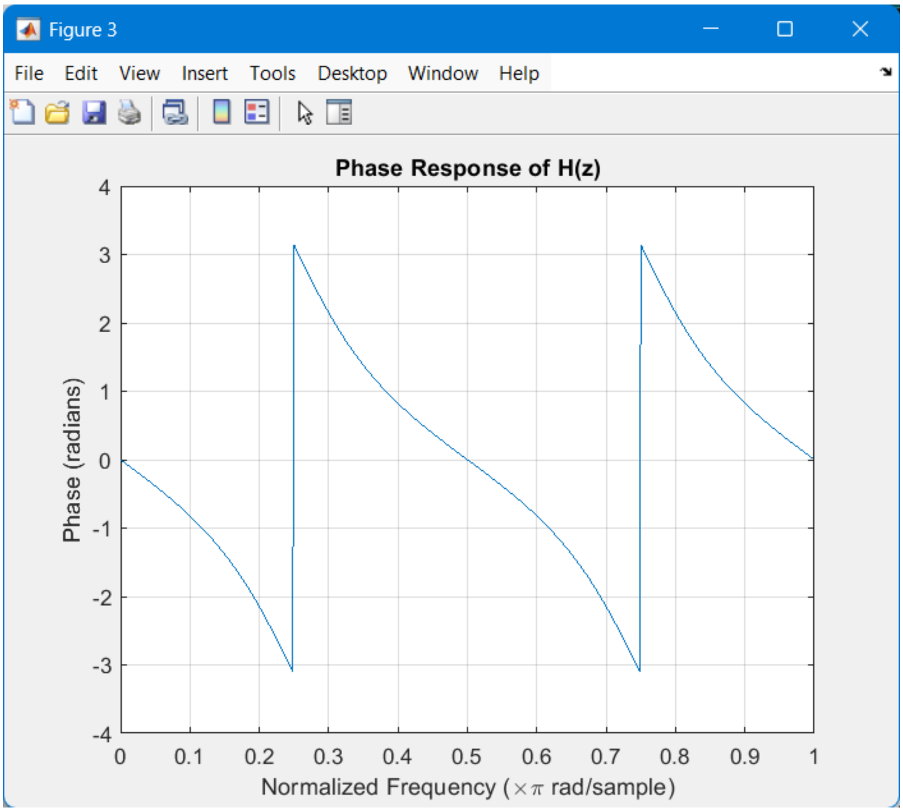

FIGURE 16: PHASE RESPONSE OF H(Z)

PART4: MIN PHASE, MAX PHASE & MIXED PHASE

Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Signal Processi

```matlab
% WIMALASOORIYA G.H.N.P.D.
% 2022E039

% Define Zeros
z = [0.7+0.2j; 0.7-0.2j; -0.4+0.4j; -0.4-0.4j];

% Poles (none given, so empty)
p = [];

% Gain (you can choose 1 unless otherwise specified)
k = 1;

% Get transfer function coefficients
[b, a] = zp2tf(z, p, k);

% Display the numerator and denominator
disp('Numerator coefficients (b):');
disp(b);

disp('Denominator coefficients (a):');
disp(a);

% Pole-Zero Plot
figure;
zplane(b, a);
title('Pole-Zero Map of the Given Filter');
xlabel('Real Part');
ylabel('Imaginary Part');
grid on;

% Magnitude and Phase Response
[H, w] = freqz(b, a, 512);

% Magnitude Response
figure;
plot(w/pi, abs(H));
title('Magnitude Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('|H(e^{j\omega})|');
grid on;

% Phase Response
figure;
plot(w/pi, angle(H));
title('Phase Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
grid on;

% Generate Maximum Phase filter
z_max = 1 ./ conj(z);

% Get new transfer function
[b_max, a_max] = zp2tf(z_max, [], k);

disp('Numerator coefficients (Maximum Phase Filter):');
disp(b_max);

```

FIGURE 17: CODE

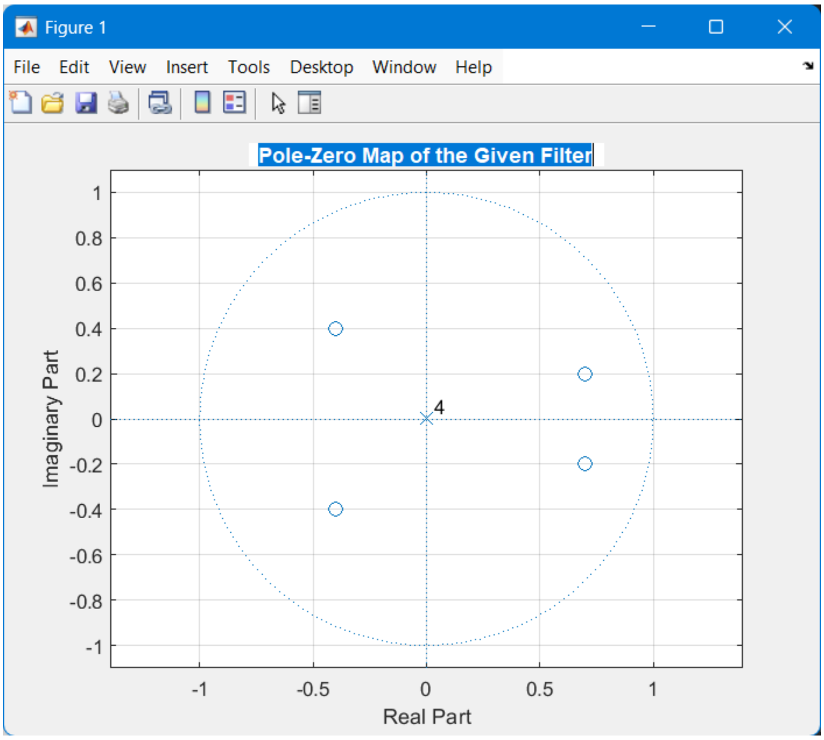

FIGURE 18: POLE-ZERO MAP

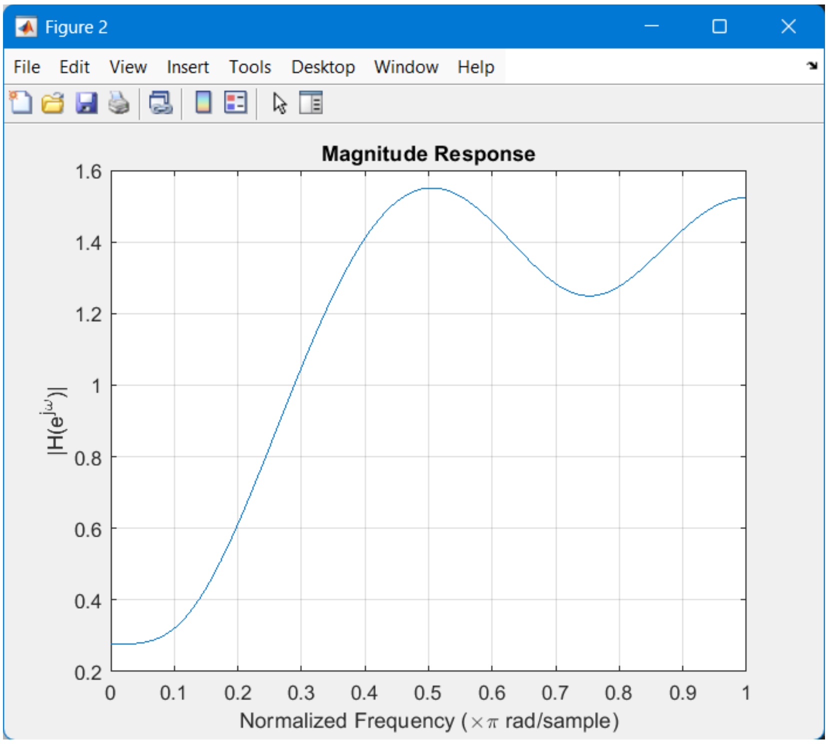

FIGURE 19: MAGNITUDE RESPONSE

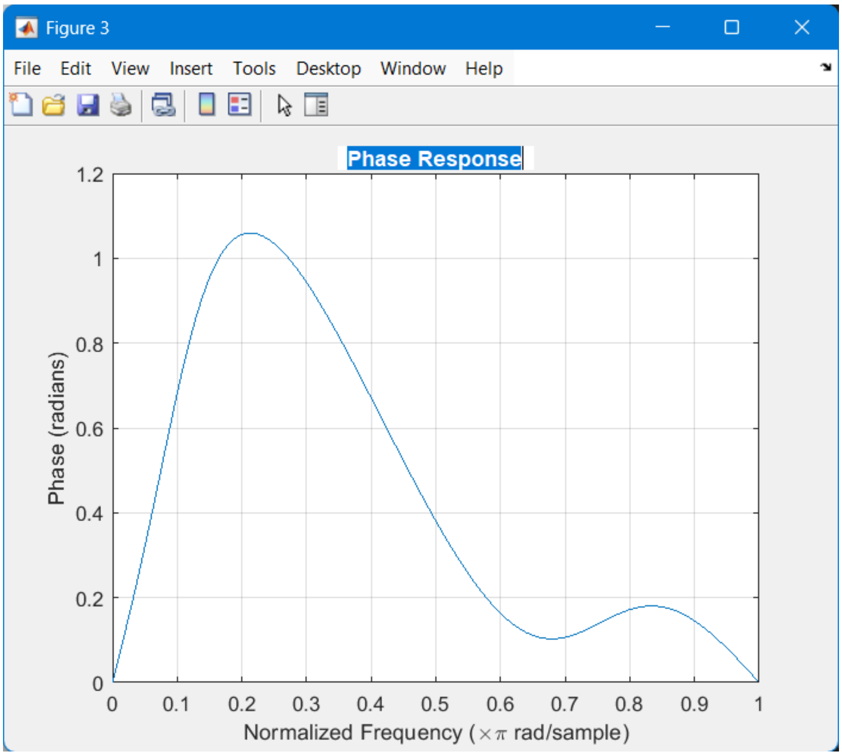

FIGURE 20: PHASE RESPONSE

```
>> part4
Numerator coefficients (b):
    1.0000   -0.6000   -0.2700   -0.0240    0.1696

Denominator coefficients (a):
     1

Numerator coefficients (Maximum Phase Filter):
    1.0000   -0.1415   -1.5920   -3.5377    5.8962
```

FIGURE 21: TERMINAL OUTPUT

PART5: LINEAR PHASE FILTER

```
% WIMALASOORIYA G.H.N.P.D.
% 2022E039

% Define the coefficients of the numerator
b = [5 0 0 0 26 0 0 0 5];  % 5Z^8 + 26Z^4 + 5

% Define the denominator (just 1)
a = 1;

% Find the zeros
zeros_G = roots(b);

% Plot the pole-zero map (no poles)
figure;
zplane(b, a);
title('Pole-Zero Map of G(Z)');
xlabel('Real Part');
ylabel('Imaginary Part');
grid on;

% Display the zeros
disp('Zeros of G(Z):');
disp(zeros_G);

% Use fvtool to analyze the filter
fvtool(b, a);
```

FIGURE 22: CODE

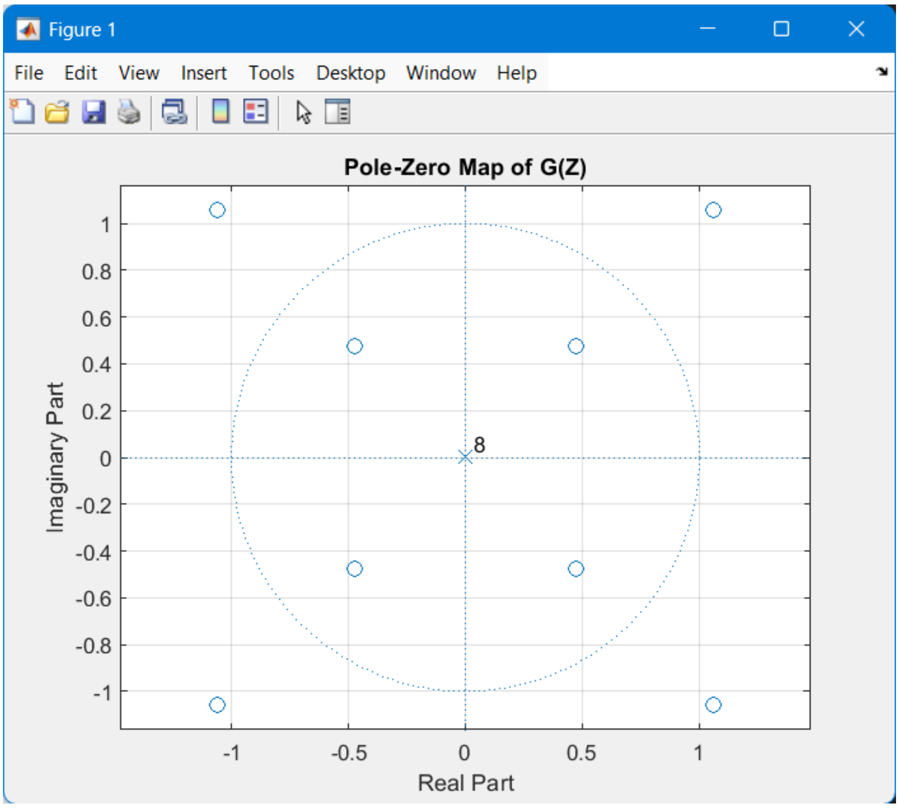

FIGURE 23: POLE-ZERO MAP

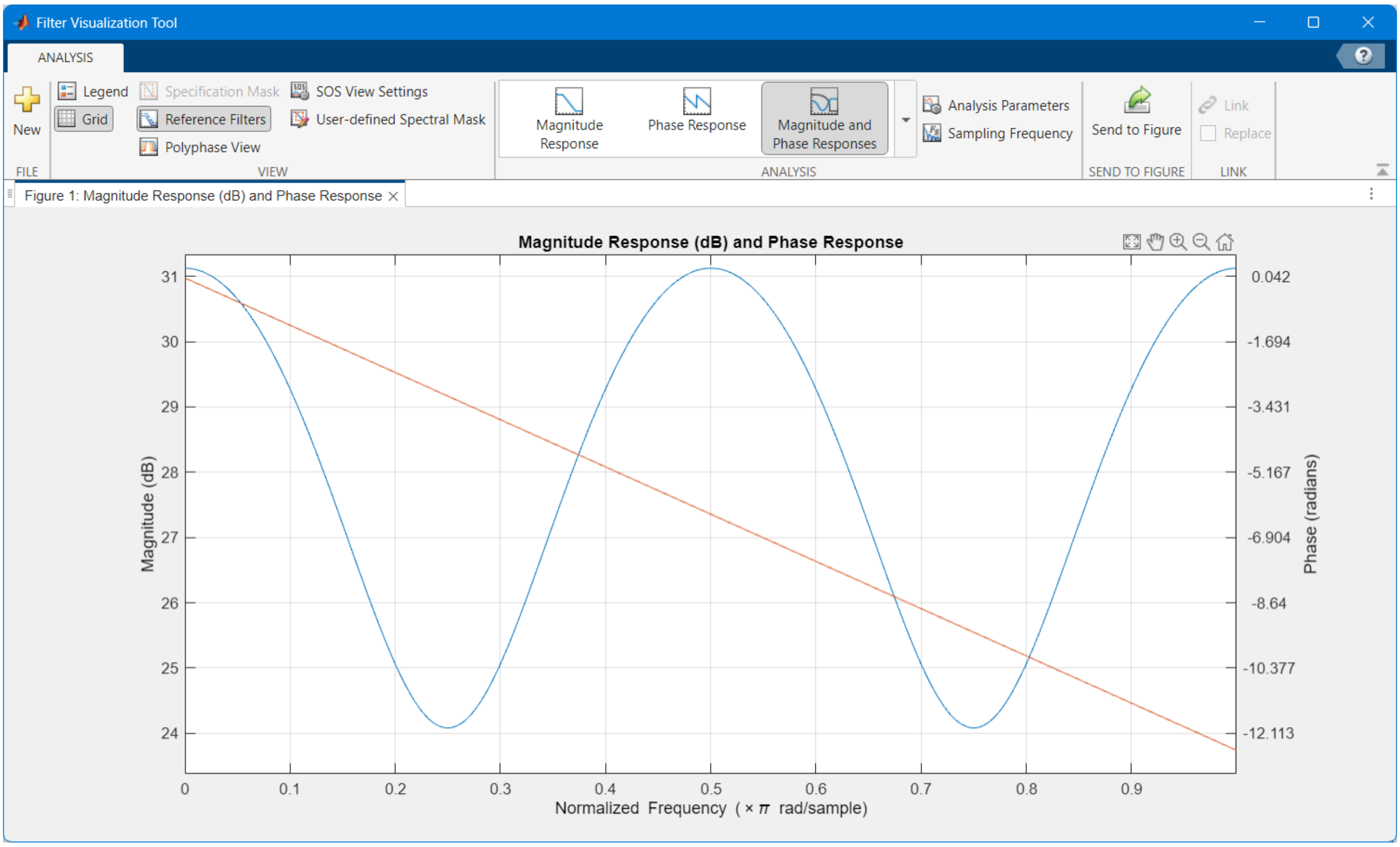

FIGURE 24: MAGNITUDE RESPONSE (dB) & PHASE RESPONSE

```
>> part5
Zeros of G(Z):
  -1.0574 + 1.0574i
  -1.0574 - 1.0574i
   1.0574 + 1.0574i
   1.0574 - 1.0574i
  -0.4729 + 0.4729i
  -0.4729 - 0.4729i
   0.4729 + 0.4729i
   0.4729 - 0.4729i
```

FIGURE 25: TERMINAL OUTPUT

PART6 (OPTIONAL): FAST FOURIER TRANSFORM IMPLEMENTATION

Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Signal Processing\

```matlab
% WIMALASOORIYA G.H.N.P.D.
% 2022E039

function X = myFFT(x)
    % Get the length of input signal
    N = length(x);

    % Base case: if only one sample, return the sample itself
    if N == 1
        X = x;
        return;
    end

    % Separate the samples into even and odd
    even_samples = x(1:2:end);    % Samples at even indices
    odd_samples = x(2:2:end);     % Samples at odd indices

    % Recursive calls to compute FFT for even and odd parts
    X_even = myFFT(even_samples);  % FFT for even part
    X_odd = myFFT(odd_samples);   % FFT for odd part

    % Twiddle factor calculation (e^(-j2πkn/N))
    W = exp(-2*pi*1i*(0:N/2-1)/N);  % W_N^k for each frequency bin

    % Combine the even and odd parts
    X = [X_even + W .* X_odd, X_even - W .* X_odd];
end

% Sample signal
x = [1, 2, 3, 4, 5, 6, 7, 8];

% Call the custom FFT function
X = myFFT(x);

% Display the FFT result
disp('FFT of the signal:');
disp(X);


% Compare with MATLAB's built-in FFT
X_builtin = fft(x);

% Display the comparison
disp('FFT (manual) vs FFT (built-in):');
disp([X.'; X_builtin.']);
```

FIGURE 26: CODE

Command Window

```
>> part6
FFT of the signal:
  Columns 1 through 4

  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i  -4.0000 + 1.6569i

  Columns 5 through 8

  -4.0000 + 0.0000i  -4.0000 - 1.6569i  -4.0000 - 4.0000i  -4.0000 - 9.6569i

FFT (manual) vs FFT (built-in):
  36.0000 + 0.0000i
  -4.0000 + 9.6569i
  -4.0000 + 4.0000i
  -4.0000 + 1.6569i
  -4.0000 + 0.0000i
  -4.0000 - 1.6569i
  -4.0000 - 4.0000i
  -4.0000 - 9.6569i
  36.0000 + 0.0000i
  -4.0000 + 9.6569i
  -4.0000 + 4.0000i
  -4.0000 + 1.6569i
  -4.0000 + 0.0000i
  -4.0000 - 1.6569i
  -4.0000 - 4.0000i
  -4.0000 - 9.6569i
```

FIGURE 27: TERMINAL OUTPUT