

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...
1 % WIMALASOORIYA G.H.N.P.D.
2 % 2022E039
3
4 % PART 01
5 % -----
6
7 % Define numerator and denominator
8 num = [1 0 10 3 28];
9 den = [1 0 0.35 -0.35 0.425];
10
11 % Plot Pole-Zero Map
12 figure; % <-- ADD THIS to open a new figure window
13 zplane(num, den);
14 title('Pole-Zero Map of H(z)');
15 xlabel('Real part');
16 ylabel('Imaginary part');
17 grid on;
18
19 % Define coefficients
20 b = [1 0 10 3 28]; % Numerator coefficients (highest to lowest power)
21 a = [1 0 0.35 -0.35 0.425]; % Denominator coefficients
22
23 % Number of samples
24 N = 50; % Length of impulse response
25
26 % Generate impulse input
27 x = zeros(1, N);
28 x(1) = 1; % impulse at n=0
29
30 % Initialize output
31 y = zeros(1, N);
32
33 % Manual filtering using the difference equation
34 for n = 1:N
35     % Feedforward part (numerator)
36     for k = 1:length(b)
37         if n-k+1 > 0
38             y(n) = y(n) + b(k) * x(n-k+1);
39         end
40     end
41     % Feedback part (denominator)
42     for k = 2:length(a)
43         if n-k+1 > 0
44             y(n) = y(n) - a(k) * y(n-k+1);
45         end
46     end
47 end
48
49 % Plot the impulse response
50 figure; % <-- ADD THIS also if you want impulse response in a new figure
51 stem(0:N-1, y, 'filled');
52 title('Impulse Response of H(z)');
53 xlabel('n');
54 ylabel('h[n]');
55 grid on;
56
57 % -----
58 % Plot Magnitude and Phase Response using the Impulse Response
59 % Take FFT of the impulse response
60 Y = fft(y, 512); % 512-point FFT for better resolution
61
62 % Frequency axis
63 f = linspace(0, 1, 512); % Normalized frequency (0 to 1 corresponds to 0 to pi rad/sample)
64
65 % Plot magnitude response
66 figure;
67 plot(f, abs(Y));
68 title('Magnitude Response from Impulse Response');
69 xlabel('Normalized Frequency (\times\pi rad/sample)');
70 ylabel('|H(e^{j\omega})|');
71 grid on;
72
73 % Plot phase response
74 figure;
75 plot(f, angle(Y));
76 title('Phase Response from Impulse Response');
77 xlabel('Normalized Frequency (\times\pi rad/sample)');
78 ylabel('Phase (radians)');
79 grid on;
80
81 % -----
82 % -----
83 % Find Frequency Response Directly using freqz (without impulse response)
84
85 % Directly find the frequency response using freqz
86 [H, w] = freqz(b, a, 512); % 512 points
87
88 % Plot magnitude response
89 figure;
90 plot(w/pi, abs(H));
91 title('Magnitude Response using freqz');
92 xlabel('Normalized Frequency (\times\pi rad/sample)');
93 ylabel('|H(e^{j\omega})|');
94 grid on;
95
96 % Plot phase response
97 figure;
98 plot(w/pi, angle(H));
99 title('Phase Response using freqz');
100 xlabel('Normalized Frequency (\times\pi rad/sample)');
101 ylabel('Phase (radians)');
102 grid on;
103
104 % -----
105
106
107

```

FIGURE 01: CODE

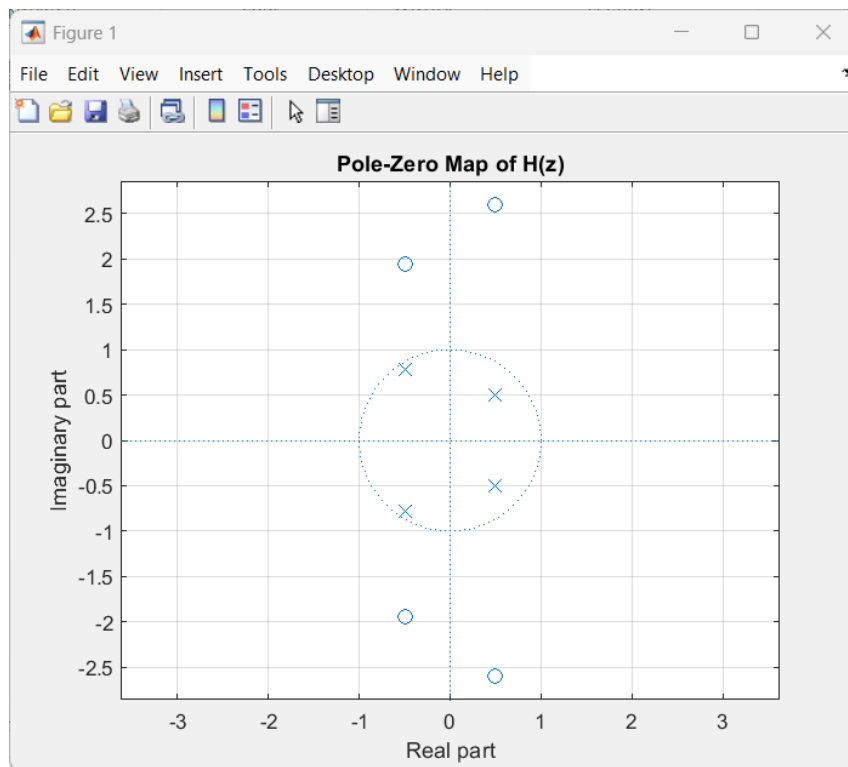


FIGURE 02: ZEROS & POLES

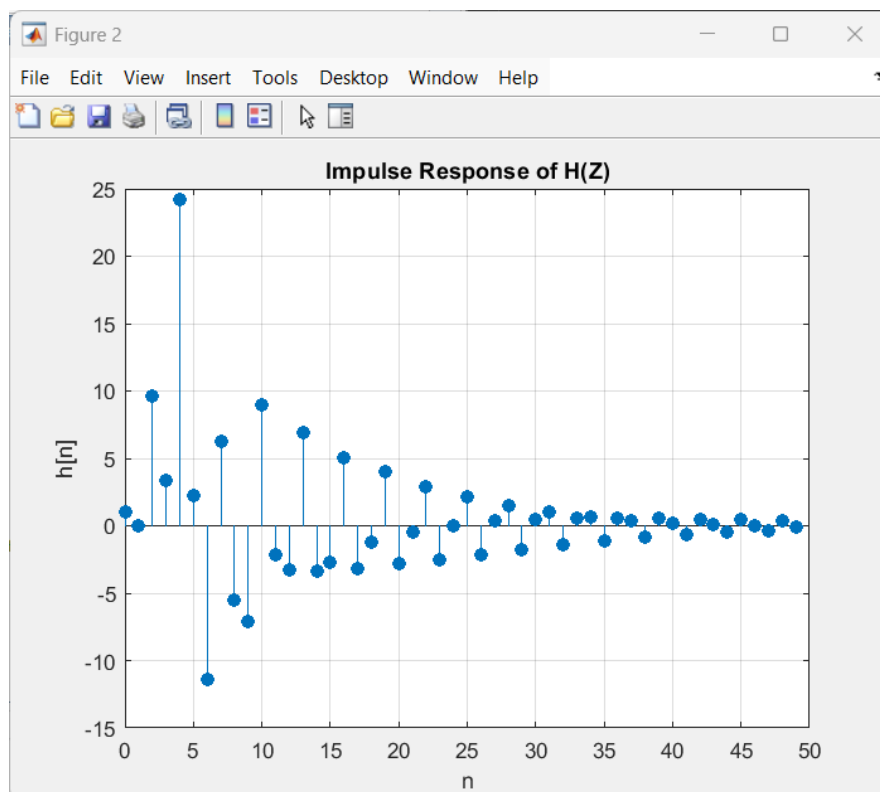


FIGURE 03: IMPULSE RESPONSE

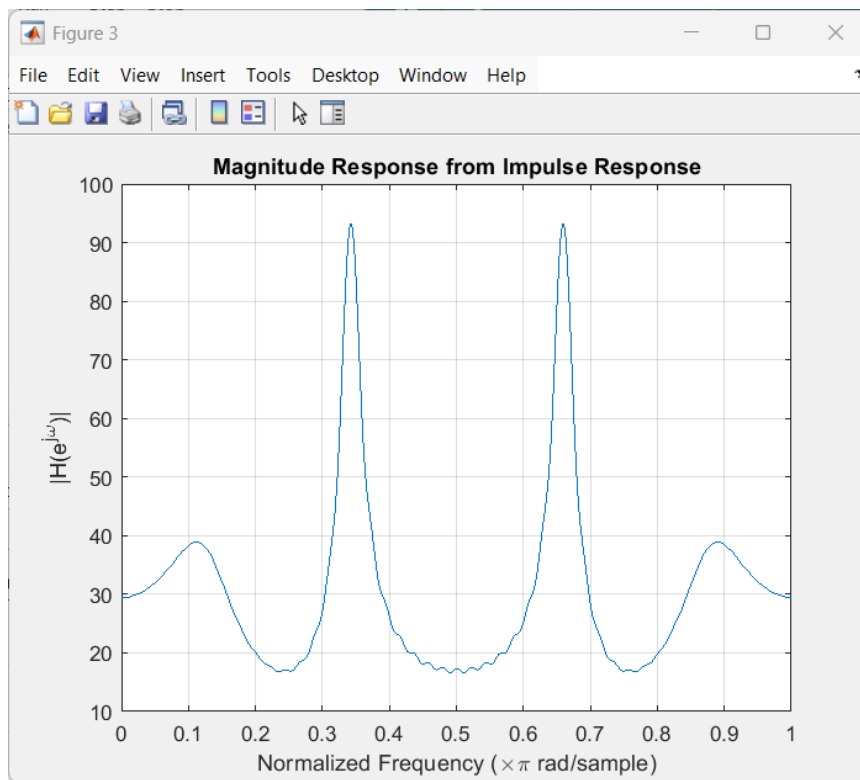


FIGURE 04: MAGNITUDE RESPONSE

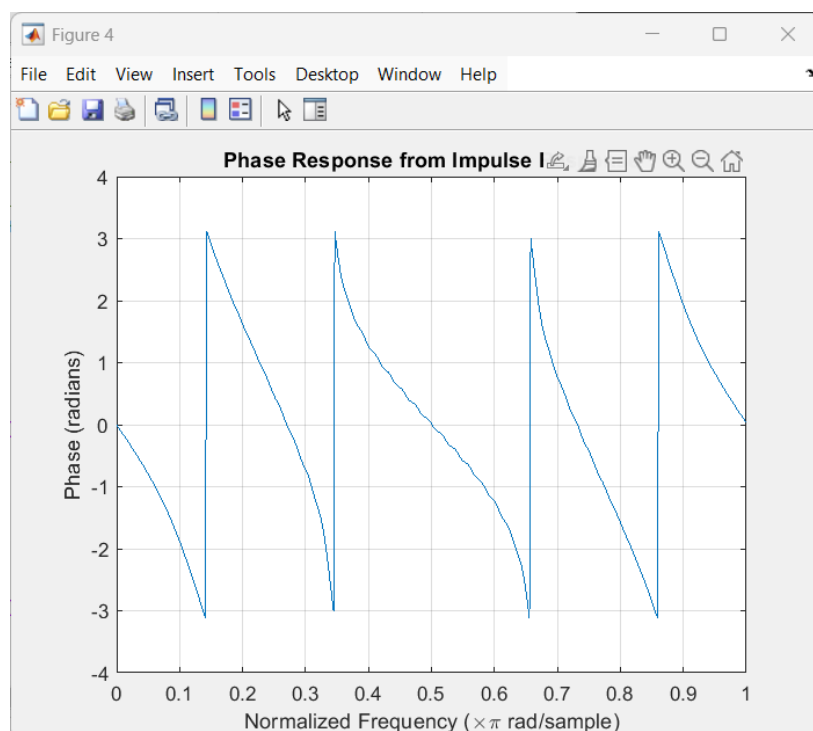


FIGURE 05: PHASE RESPONSE

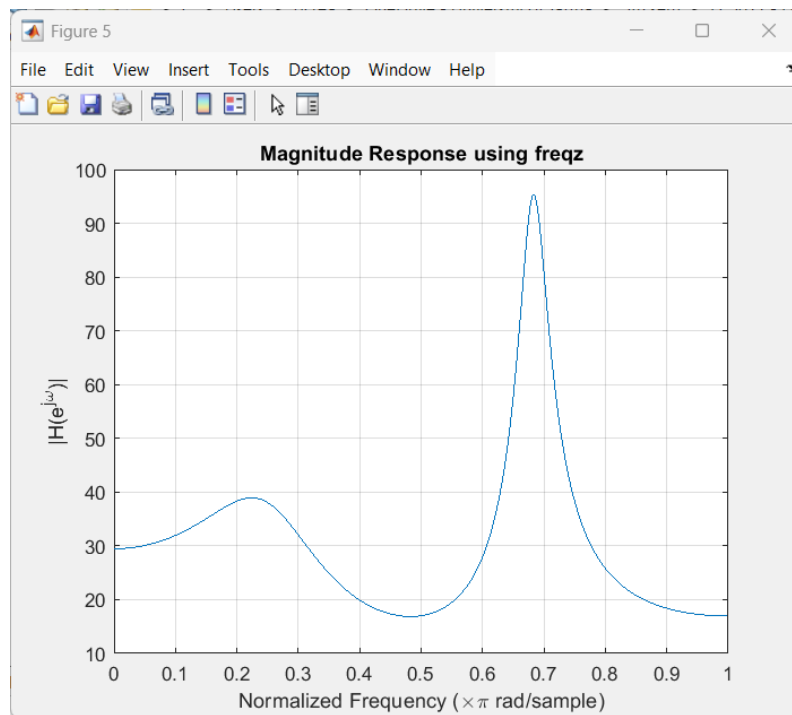


FIGURE 06: MAGNITUDE RESPONSE USING `freqz`

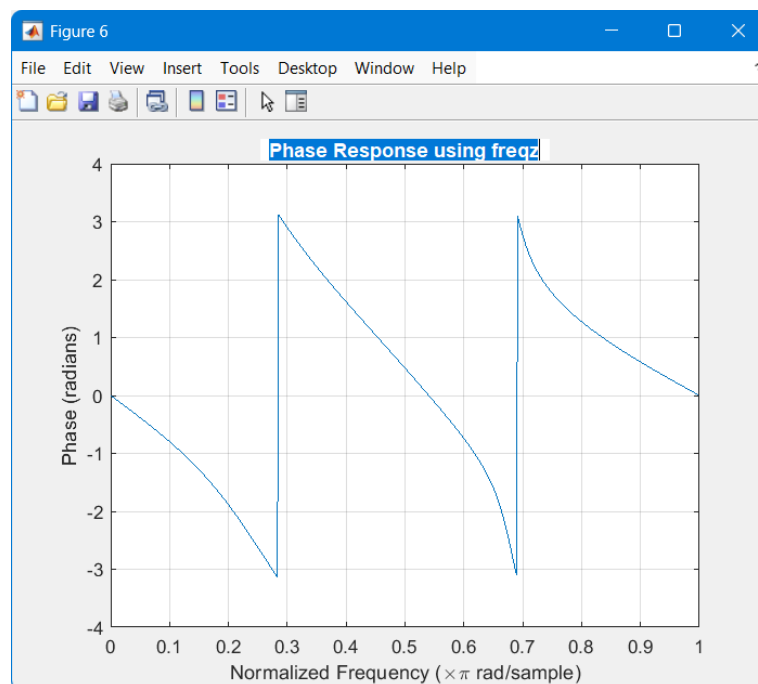
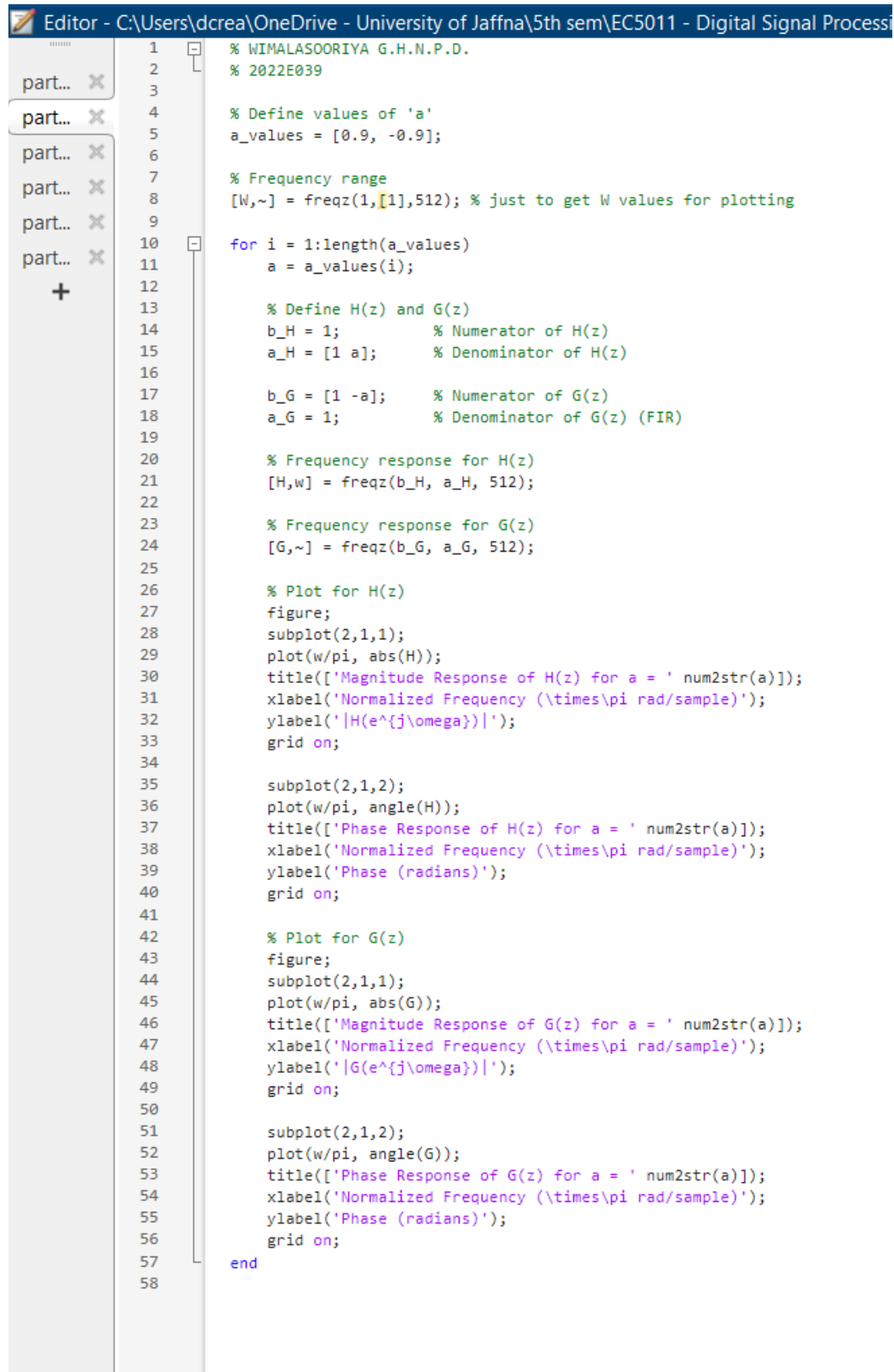


FIGURE 07: PHASE RESPONSE USING `freqz`

PART2: DIGITAL FILTER DESIGN IN MATLAB



```
1 % WIMALASOORIYA G.H.N.P.D.  
2 % 2022E039  
3  
4 % Define values of 'a'  
5 a_values = [0.9, -0.9];  
6  
7 % Frequency range  
8 [W,~] = freqz(1,[1],512); % just to get W values for plotting  
9  
10 for i = 1:length(a_values)  
11     a = a_values(i);  
12  
13     % Define H(z) and G(z)  
14     b_H = 1; % Numerator of H(z)  
15     a_H = [1 a]; % Denominator of H(z)  
16  
17     b_G = [1 -a]; % Numerator of G(z)  
18     a_G = 1; % Denominator of G(z) (FIR)  
19  
20     % Frequency response for H(z)  
21     [H,w] = freqz(b_H, a_H, 512);  
22  
23     % Frequency response for G(z)  
24     [G,~] = freqz(b_G, a_G, 512);  
25  
26     % Plot for H(z)  
27     figure;  
28     subplot(2,1,1);  
29     plot(w/pi, abs(H));  
30     title(['Magnitude Response of H(z) for a = ' num2str(a)]);  
31     xlabel('Normalized Frequency (\times\pi rad/sample)');  
32     ylabel('|H(e^{j\omega})|');  
33     grid on;  
34  
35     subplot(2,1,2);  
36     plot(w/pi, angle(H));  
37     title(['Phase Response of H(z) for a = ' num2str(a)]);  
38     xlabel('Normalized Frequency (\times\pi rad/sample)');  
39     ylabel('Phase (radians)');  
40     grid on;  
41  
42     % Plot for G(z)  
43     figure;  
44     subplot(2,1,1);  
45     plot(w/pi, abs(G));  
46     title(['Magnitude Response of G(z) for a = ' num2str(a)]);  
47     xlabel('Normalized Frequency (\times\pi rad/sample)');  
48     ylabel('|G(e^{j\omega})|');  
49     grid on;  
50  
51     subplot(2,1,2);  
52     plot(w/pi, angle(G));  
53     title(['Phase Response of G(z) for a = ' num2str(a)]);  
54     xlabel('Normalized Frequency (\times\pi rad/sample)');  
55     ylabel('Phase (radians)');  
56     grid on;  
57 end  
58
```

FIGURE 08: CODE

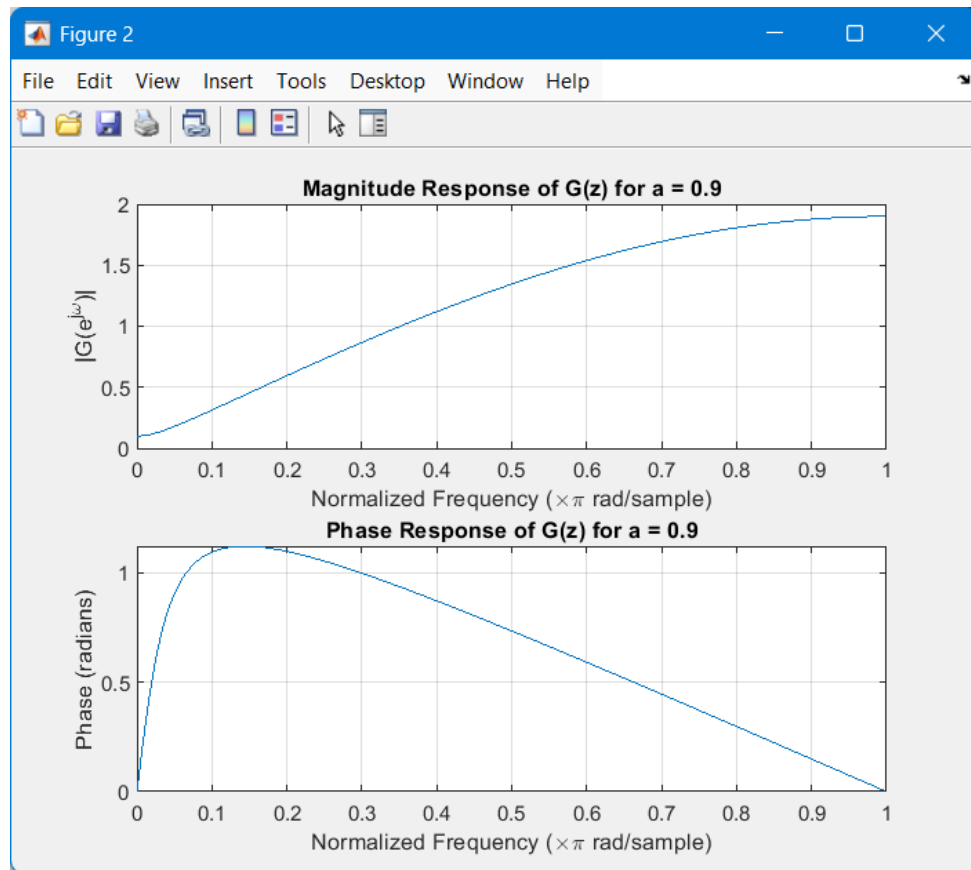


FIGURE 09: MAGNITUDE & PHASE RESPONSE OF $G(Z)$ FOR $A = 0.9$

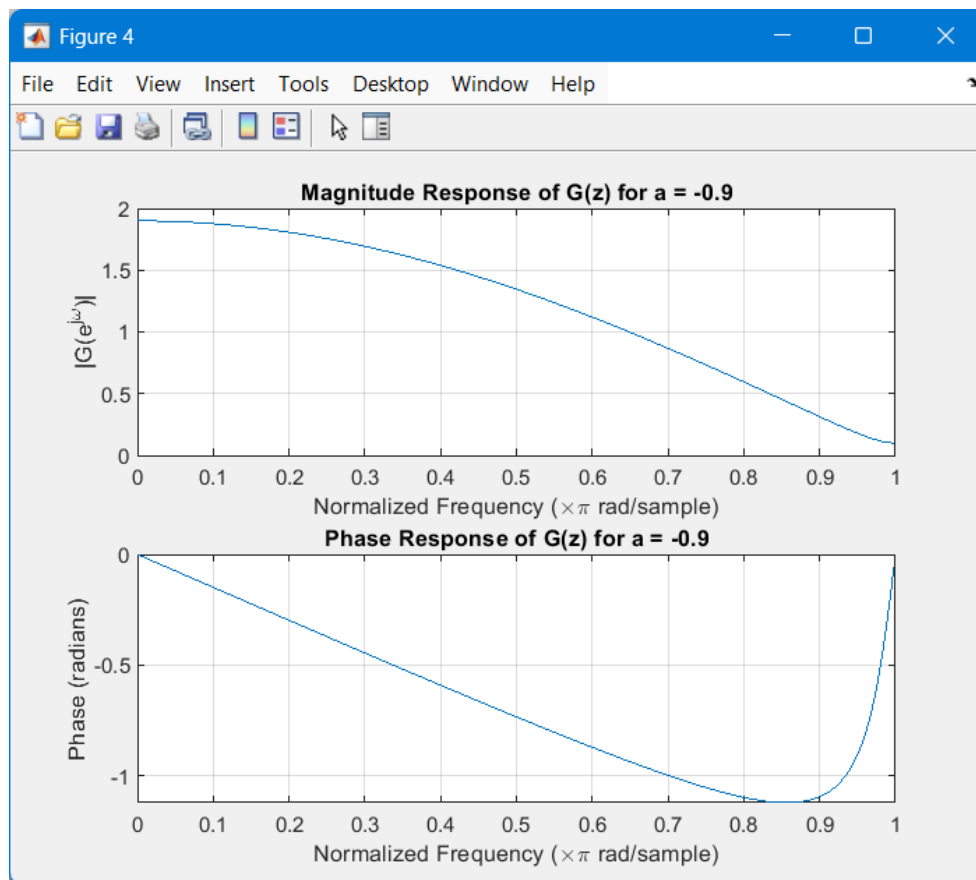


FIGURE 10: MAGNITUDE & PHASE RESPONSE OF $G(Z)$ FOR $A = -0.9$

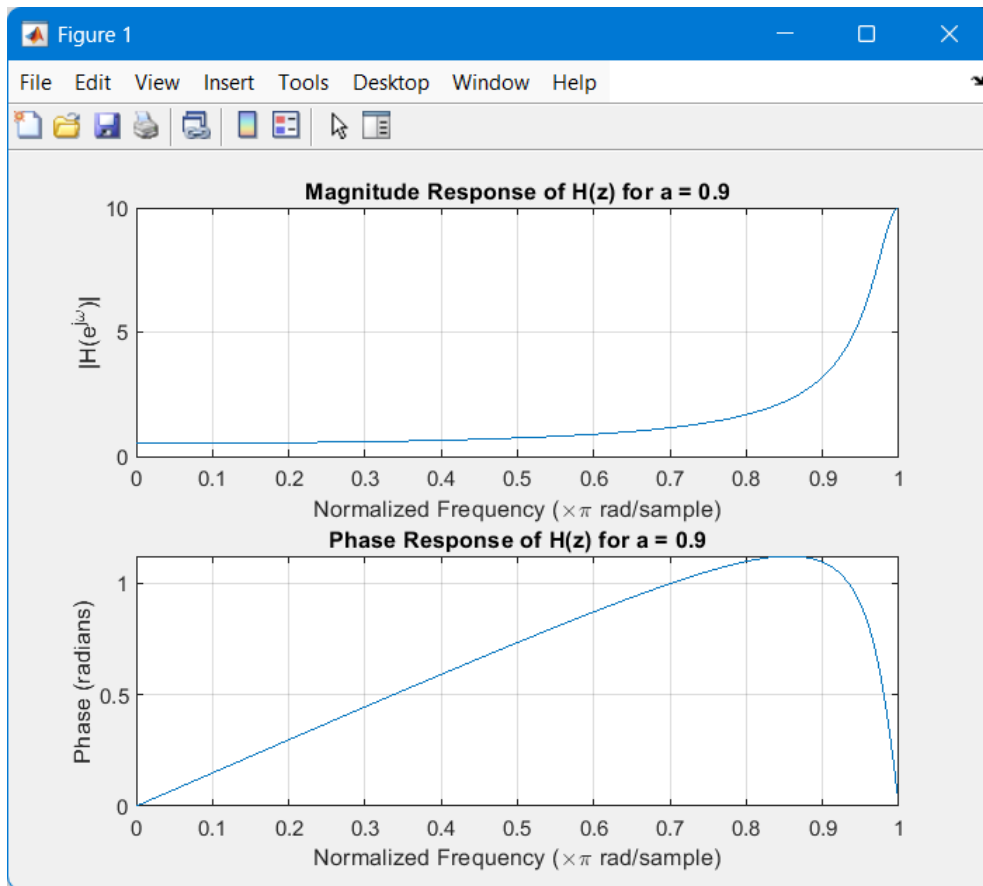


FIGURE 11: MAGNITUDE & PHASE RESPONSE OF $H(Z)$ FOR $A = 0.9$

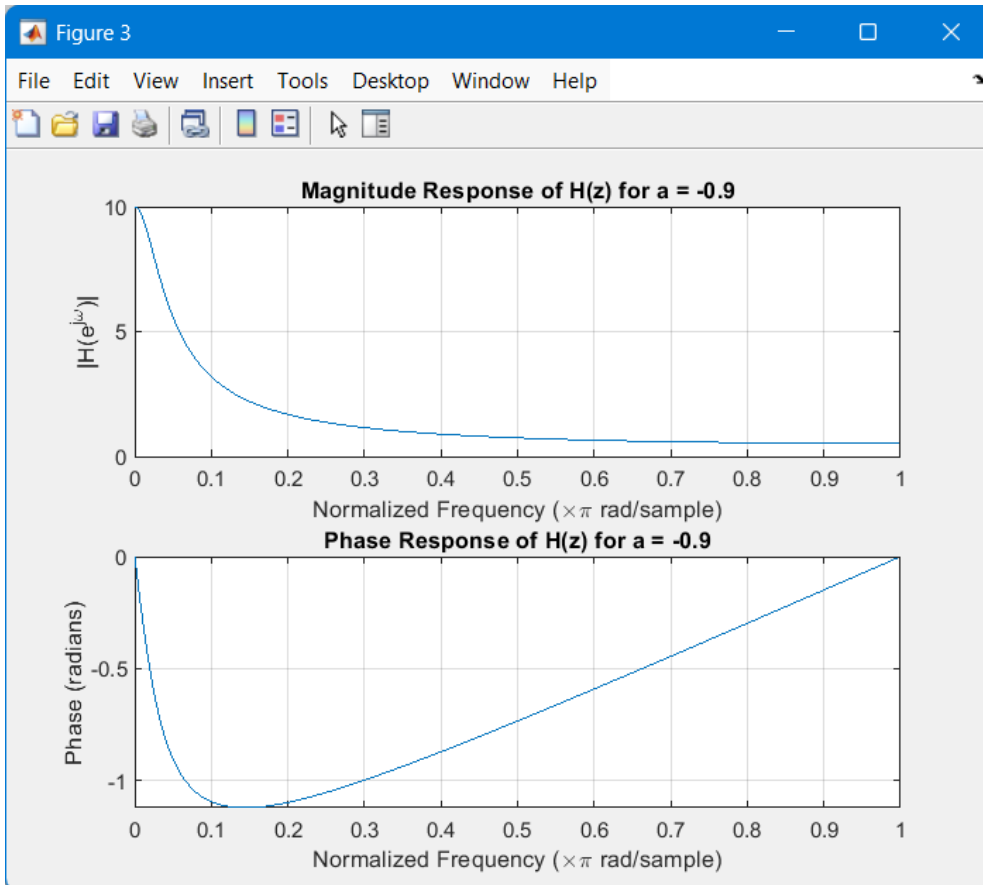


FIGURE 12: MAGNITUDE & PHASE RESPONSE OF $H(Z)$ FOR $A = -0.9$

PART3: ALLPASS FILTER

```
Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Sign
1 % WIMALASOORIYA G.H.N.P.D.
2 % 2022E039
3
4 % Define numerator and denominator
5 b = [1 0 0 0 4]; %  $Z^4 + 4$ 
6 a = [4 0 0 0 1]; %  $4Z^4 + 1$ 
7
8 % Plot pole-zero map
9 figure;
10 zplane(b, a);
11 title('Pole-Zero Map of H(z)');
12 xlabel('Real Part');
13 ylabel('Imaginary Part');
14 grid on;
15
16 % Find zeros (roots of numerator)
17 zeros_H = roots(b);
18
19 % Find poles (roots of denominator)
20 poles_H = roots(a);
21
22 % Frequency response using freqz
23 [H, w] = freqz(b, a, 512);
24
25 % Magnitude Response
26 figure;
27 plot(w/pi, abs(H));
28 title('Magnitude Response of H(z)');
29 xlabel('Normalized Frequency (\times\pi rad/sample)');
30 ylabel('|H(e^{j\omega})|');
31 grid on;
32
33 % Phase Response
34 figure;
35 plot(w/pi, angle(H));
36 title('Phase Response of H(z)');
37 xlabel('Normalized Frequency (\times\pi rad/sample)');
38 ylabel('Phase (radians)');
39 grid on;
40
41
```

FIGURE 13: CODE

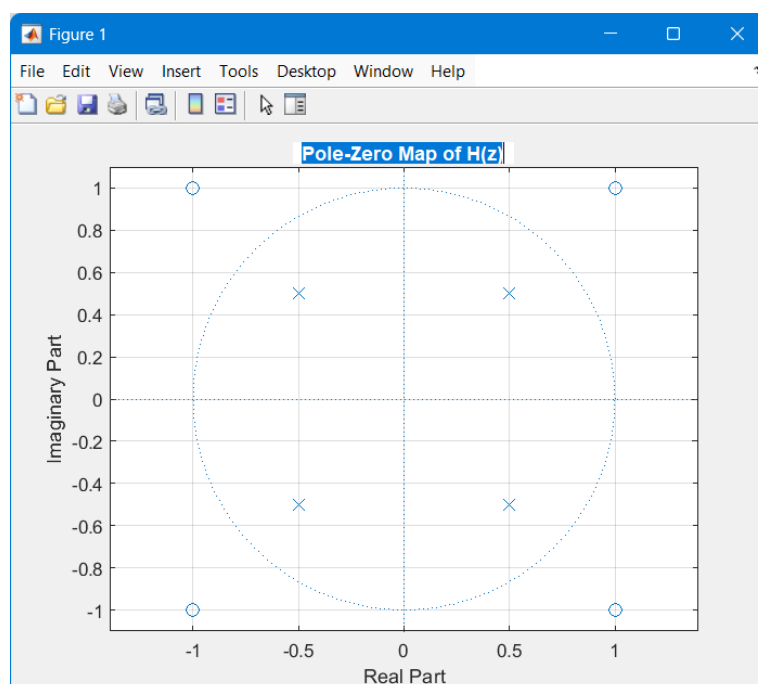


FIGURE 14: POLE-ZERO MAP OF H(Z)

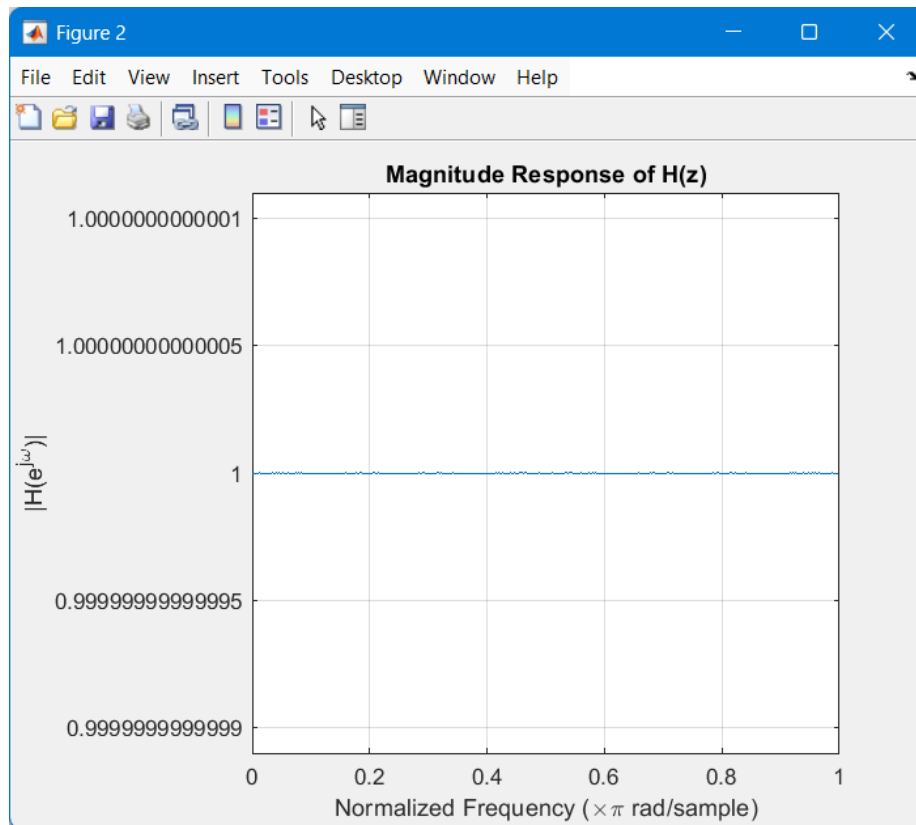


FIGURE 15: MAGNITUDE RESPONSE OF $H(Z)$

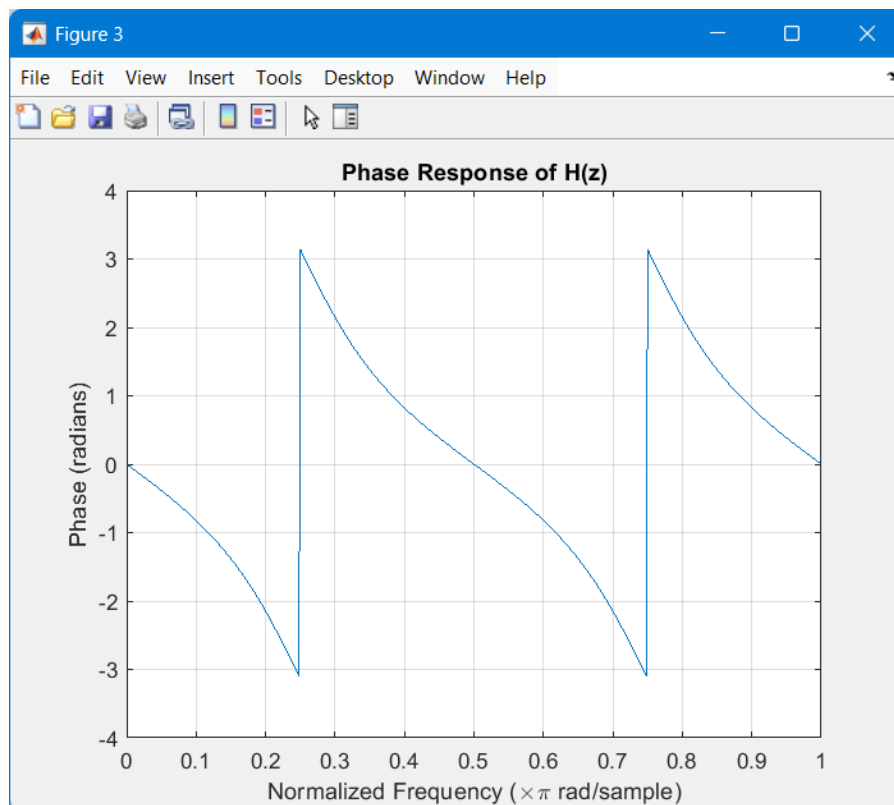
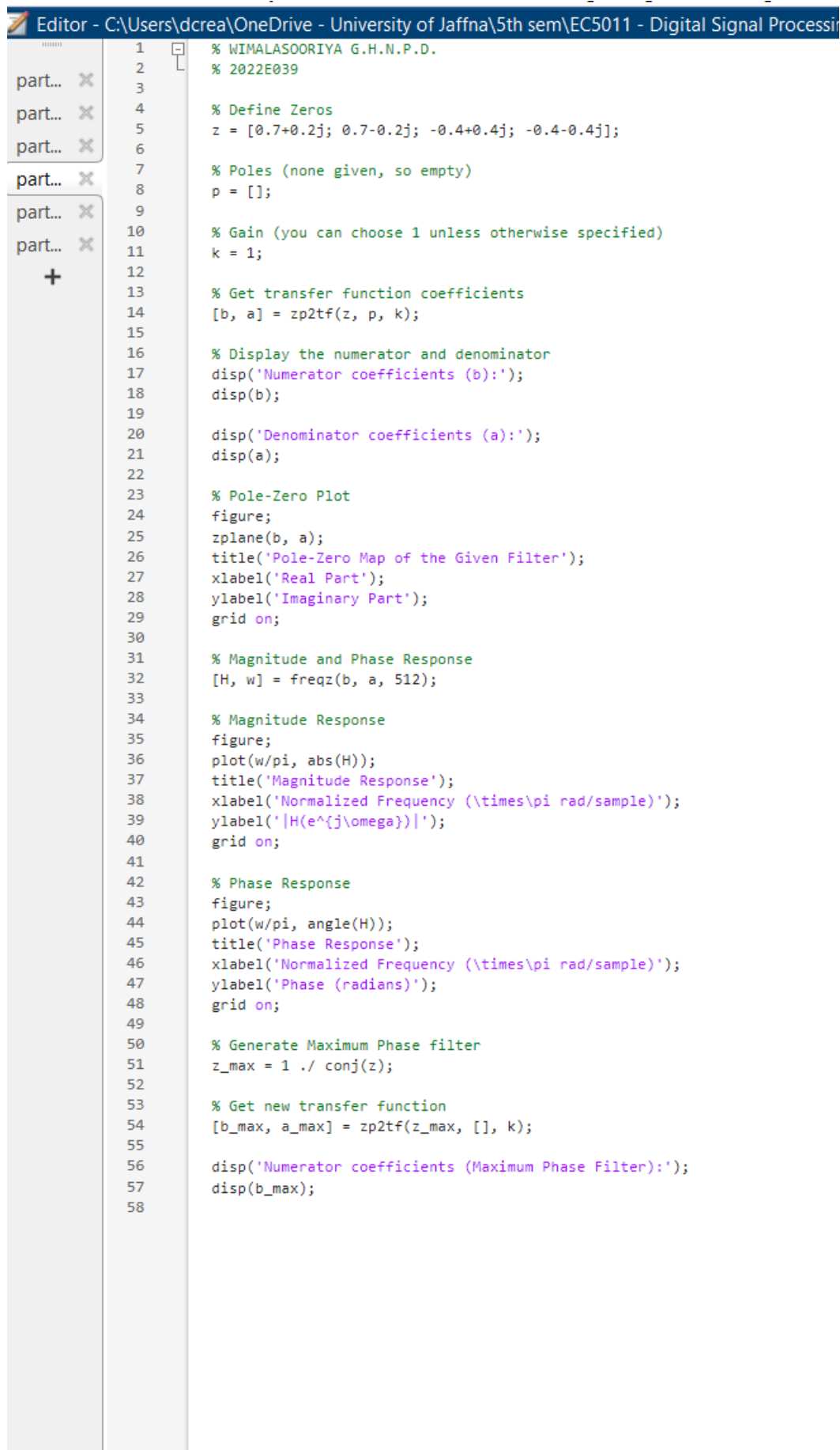


FIGURE 16: PHASE RESPONSE OF $H(Z)$

PART4: MIN PHASE, MAX PHASE & MIXED PHASE



The image shows a MATLAB Editor window with a file named 'Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Signal Processi'. The code is as follows:

```
1 % WIMALASOORIYA G.H.N.P.D.  
2 % 2022E039  
3  
4 % Define Zeros  
5 z = [0.7+0.2j; 0.7-0.2j; -0.4+0.4j; -0.4-0.4j];  
6  
7 % Poles (none given, so empty)  
8 p = [];  
9  
10 % Gain (you can choose 1 unless otherwise specified)  
11 k = 1;  
12  
13 % Get transfer function coefficients  
14 [b, a] = zp2tf(z, p, k);  
15  
16 % Display the numerator and denominator  
17 disp('Numerator coefficients (b):');  
18 disp(b);  
19  
20 disp('Denominator coefficients (a):');  
21 disp(a);  
22  
23 % Pole-Zero Plot  
24 figure;  
25 zplane(b, a);  
26 title('Pole-Zero Map of the Given Filter');  
27 xlabel('Real Part');  
28 ylabel('Imaginary Part');  
29 grid on;  
30  
31 % Magnitude and Phase Response  
32 [H, w] = freqz(b, a, 512);  
33  
34 % Magnitude Response  
35 figure;  
36 plot(w/pi, abs(H));  
37 title('Magnitude Response');  
38 xlabel('Normalized Frequency (\times\pi rad/sample)');  
39 ylabel('|H(e^{j\omega})|');  
40 grid on;  
41  
42 % Phase Response  
43 figure;  
44 plot(w/pi, angle(H));  
45 title('Phase Response');  
46 xlabel('Normalized Frequency (\times\pi rad/sample)');  
47 ylabel('Phase (radians)');  
48 grid on;  
49  
50 % Generate Maximum Phase filter  
51 z_max = 1 ./ conj(z);  
52  
53 % Get new transfer function  
54 [b_max, a_max] = zp2tf(z_max, [], k);  
55  
56 disp('Numerator coefficients (Maximum Phase Filter):');  
57 disp(b_max);  
58
```

FIGURE 17: CODE

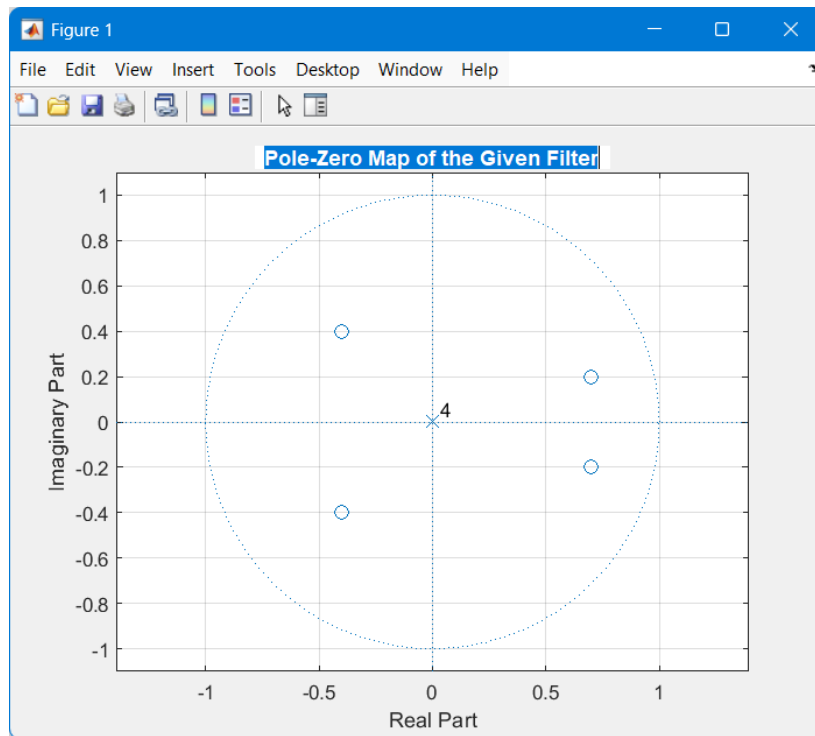


FIGURE 18: POLE-ZERO MAP

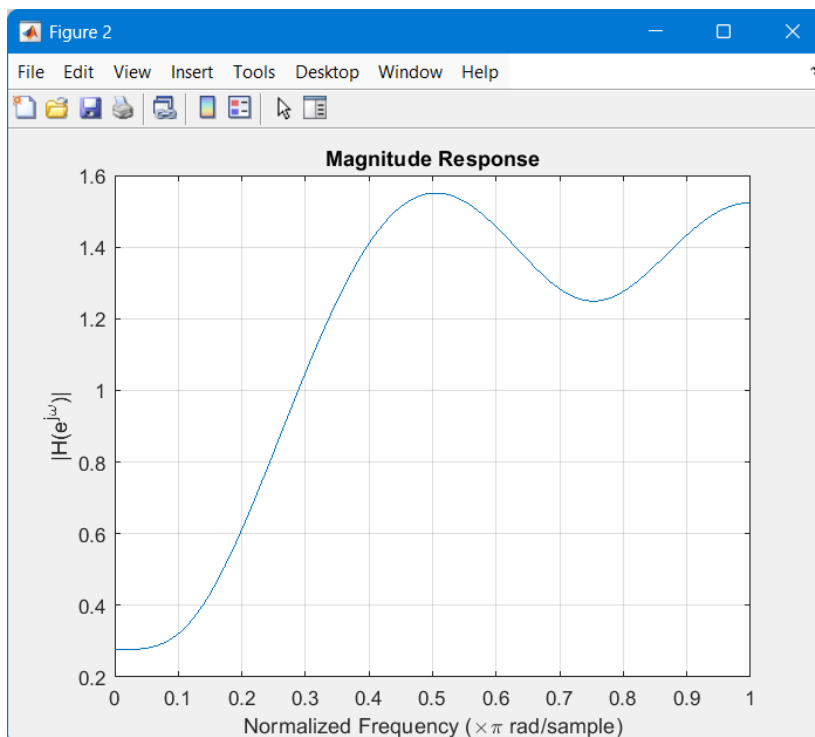


FIGURE 19: MAGNITUDE RESPONSE

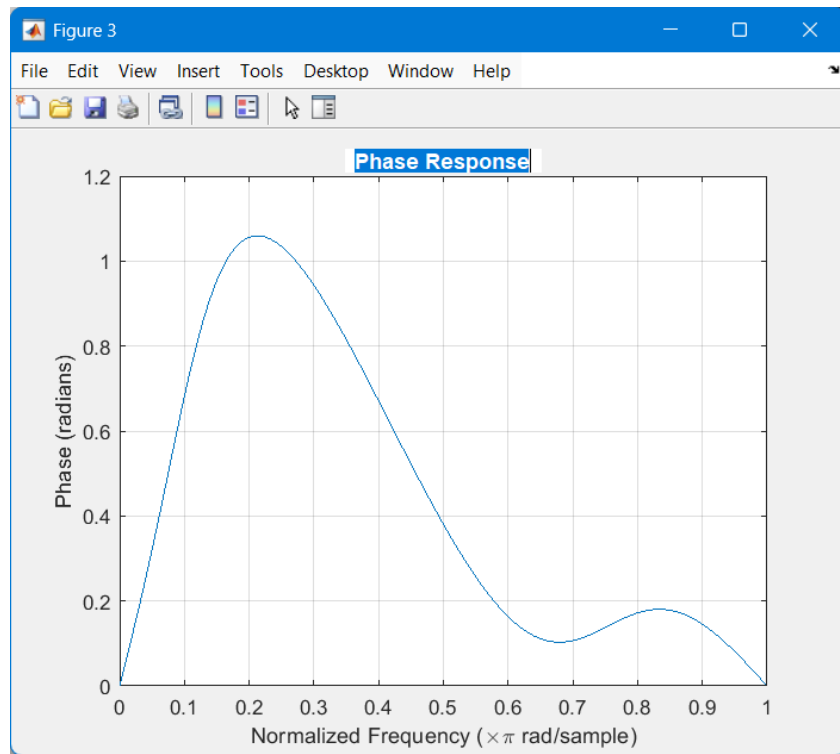


FIGURE 20: PHASE RESPONSE

```
>> 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

```
Editor - C:\Users\dcrea\OneDrive - University of Jaffna\5th sem\EC5011 - Digital Signal Processing\Labs\Lab
part... x 1 % WIMALASOORIYA G.H.N.P.D.
part... x 2 % 2022E039
part... x 3
part... x 4 % Define the coefficients of the numerator
part... x 5 b = [5 0 0 0 26 0 0 5]; % 5Z^8 + 26Z^4 + 5
part... x 6
part... x 7 % Define the denominator (just 1)
+ 8 a = 1;
9
10 % Find the zeros
11 zeros_G = roots(b);
12
13 % Plot the pole-zero map (no poles)
14 figure;
15 zplane(b, a);
16 title('Pole-Zero Map of G(Z)');
17 xlabel('Real Part');
18 ylabel('Imaginary Part');
19 grid on;
20
21 % Display the zeros
22 disp('Zeros of G(Z):');
23 disp(zeros_G);
24
25 % Use fvtool to analyze the filter
26 fvtool(b, a);
27
```

FIGURE 22: CODE

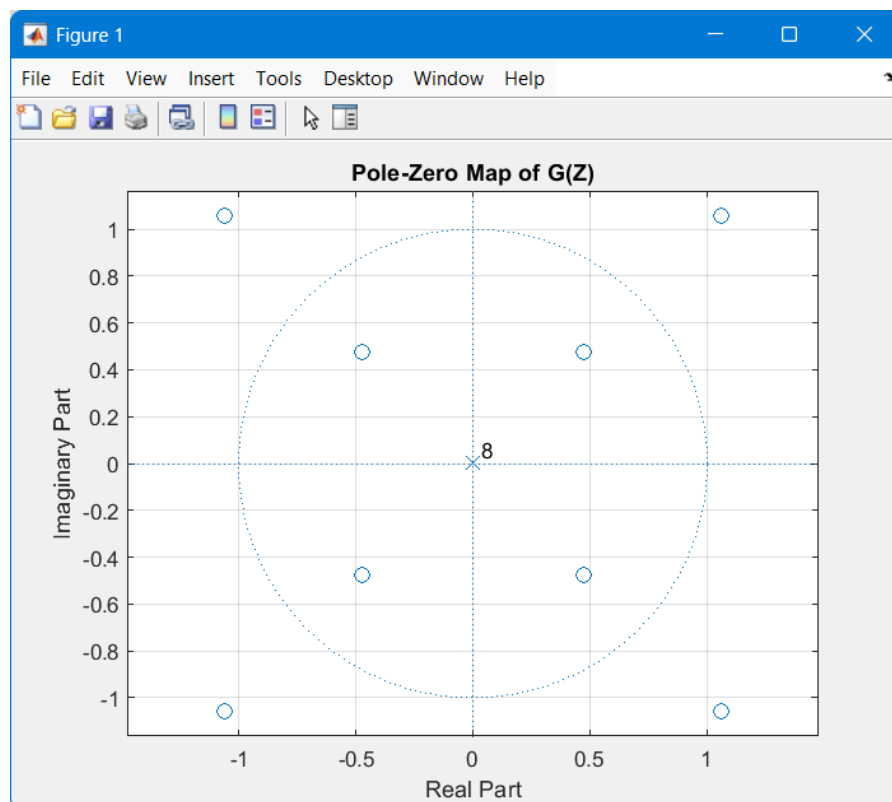


FIGURE 23: POLE-ZERO MAP

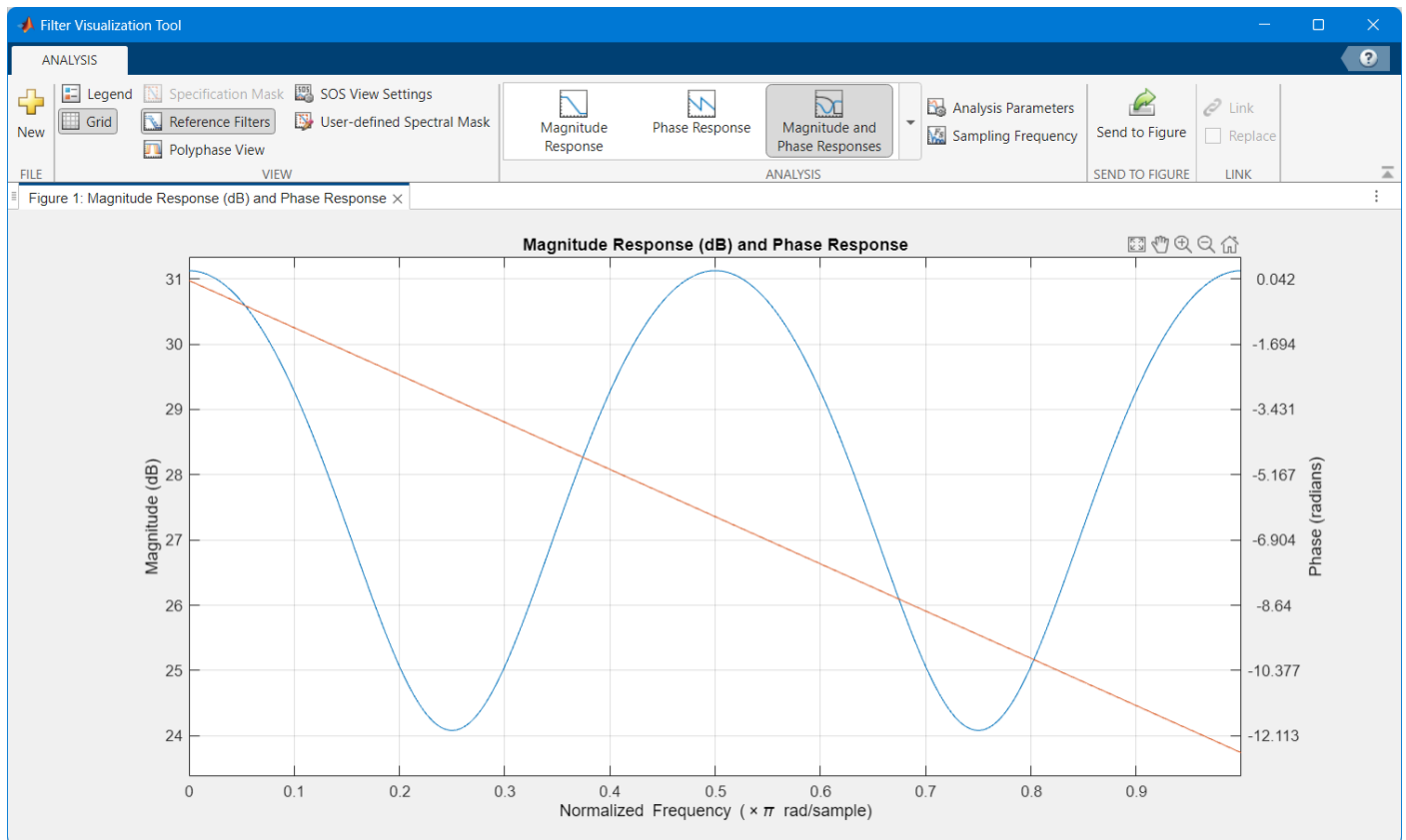
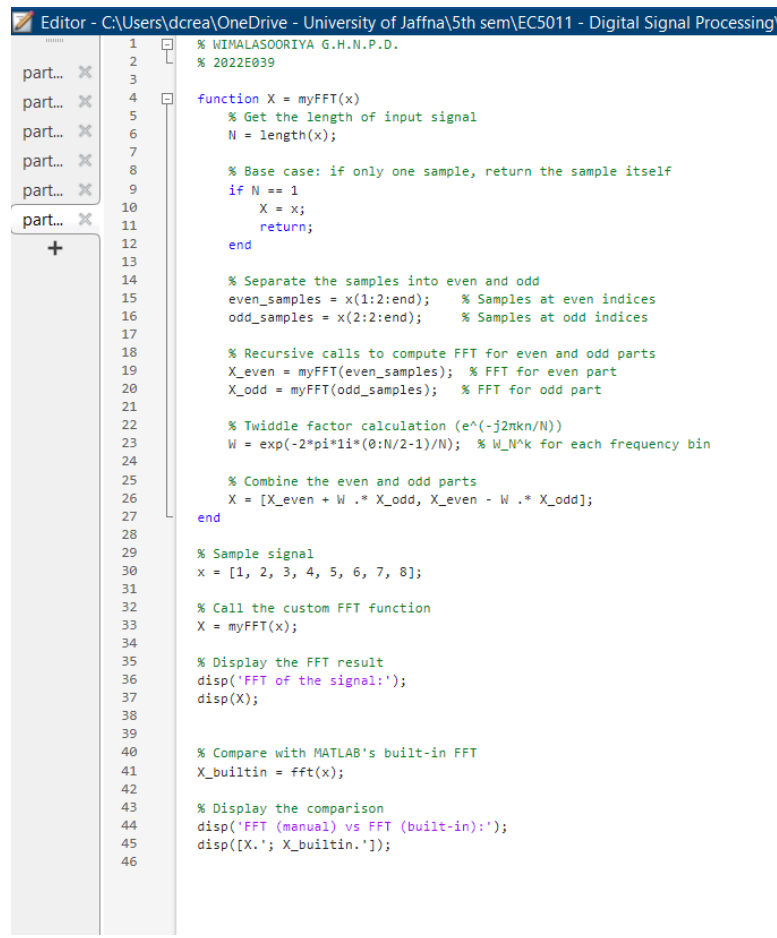


FIGURE 24: MAGNITUDE RESPONSE (dB) & PHASE RESPONSE

```
>> 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
```

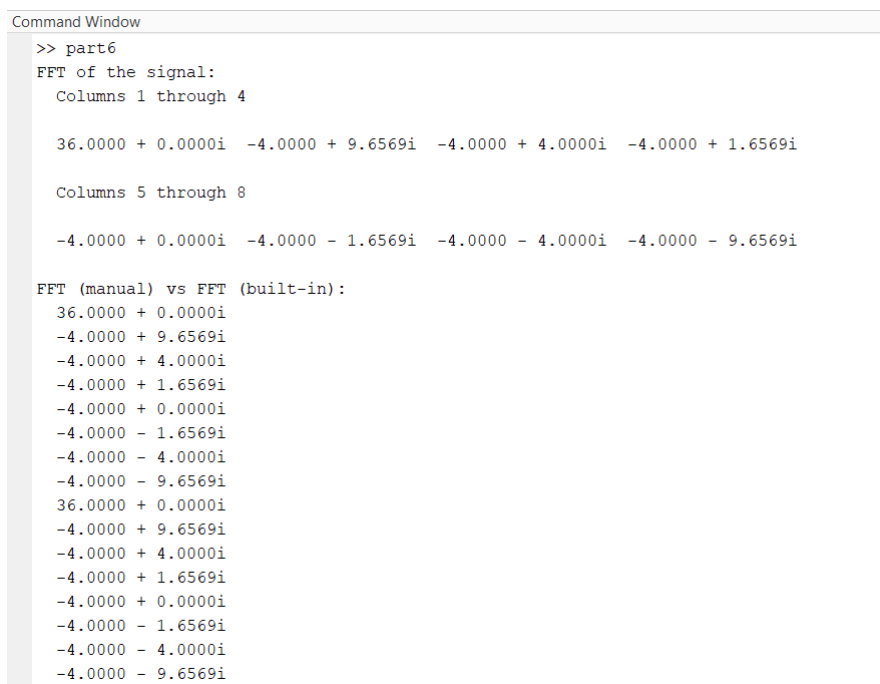
FIGURE 25: TERMINAL OUTPUT

PART6 (OPTIONAL): FAST FOURIER TRANSFORM IMPLEMENTATION



```
1 % WIMALASOORIYA G.H.N.P.D.  
2 % 2022E039  
3  
4 function X = myFFT(x)  
5 % Get the length of input signal  
6 N = length(x);  
7  
8 % Base case: if only one sample, return the sample itself  
9 if N == 1  
10     X = x;  
11     return;  
12 end  
13  
14 % Separate the samples into even and odd  
15 even_samples = x(1:2:end); % Samples at even indices  
16 odd_samples = x(2:2:end); % Samples at odd indices  
17  
18 % Recursive calls to compute FFT for even and odd parts  
19 X_even = myFFT(even_samples); % FFT for even part  
20 X_odd = myFFT(odd_samples); % FFT for odd part  
21  
22 % Twiddle factor calculation ( $e^{-j2\pi kn/N}$ )  
23 W = exp(-2*pi*1i*(0:N/2-1)/N); %  $W_N^k$  for each frequency bin  
24  
25 % Combine the even and odd parts  
26 X = [X_even + W .* X_odd, X_even - W .* X_odd];  
27 end  
28  
29 % Sample signal  
30 x = [1, 2, 3, 4, 5, 6, 7, 8];  
31  
32 % Call the custom FFT function  
33 X = myFFT(x);  
34  
35 % Display the FFT result  
36 disp('FFT of the signal:');  
37 disp(X);  
38  
39  
40 % Compare with MATLAB's built-in FFT  
41 X_builtin = fft(x);  
42  
43 % Display the comparison  
44 disp('FFT (manual) vs FFT (built-in):');  
45 disp([X.'; X_builtin.']);  
46
```

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