Digital signal generation and time domain processing case3

Digital Signal Processing Experiment

Student Name: Liu Xingyan
Student Number: 202264690069
Professional Class: 22 Artificial Intelligence Class 1

Instructor: Ning Gengxin

Starting Semester: 2023-2024 second semester of the academic year

School of Future Technology

2024-05-08

1 Discrete System Frequency Domain Analysis

1.1 Experimental Purpose

The purpose of this experiment is to analyze discrete-time systems in the frequency domain and understand their behavior using frequency response analysis and pole-zero plots. By exploring the system's frequency response, the experiment aims to provide insights into how the system affects different frequency components of input signals. Additionally, the experiment will examine the significance of zero and pole locations in defining the characteristics of discrete-time systems.

1.2 Experimental Principle

1.2.1 Frequency Response Analysis

Discrete-time systems can be represented using difference equations of the form:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{j=1}^{N} a_j y[n-j], \tag{1}$$

where $x[n]$ and $y[n]$ are the input and output sequences, respectively, and b_k and a_j are system coefficients. The frequency response of such a system is derived from its system function $H(z)$, defined as:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}}. \tag{2}$$

For a stable system, the poles should lie inside the unit circle. The frequency response $H(e^{j\omega})$ is obtained by evaluating $H(z)$ on the unit circle $z = e^{j\omega}$:

$$H(e^{j\omega}) = \frac{B(e^{j\omega})}{A(e^{j\omega})}. \tag{3}$$

The magnitude and phase responses are then calculated as follows:

$$|H(e^{j\omega})| = \sqrt{\mathrm{Re}\{H(e^{j\omega})\}^2 + \mathrm{Im}\{H(e^{j\omega})\}^2} \tag{4}$$

$$\arg\{H(e^{j\omega})\} = \arctan\left(\frac{\mathrm{Im}\{H(e^{j\omega})\}}{\mathrm{Re}\{H(e^{j\omega})\}}\right). \tag{5}$$

1.2.2 Zero-Pole Analysis

The system function $H(z)$ can also be expressed in terms of its zeros and poles:

$$H(z) = K\frac{(z-z_1)(z-z_2)\ldots(z-z_M)}{(z-p_1)(z-p_2)\ldots(z-p_N)}, \tag{6}$$

where z_k are the zeros, p_k are the poles, and K is a gain factor. The poles and zeros provide valuable insights into the system's frequency response. The pole-zero plot visually represents these characteristics.

Given the digital system with the following transfer function:

$$H(z) = \frac{0.0528 + 0.0797z^{-1} + 0.1295z^{-2} + 0.1295z^{-3} + 0.0797z^{-4} + 0.0528z^{-5}}{1 - 1.8107z^{-1} + 2.4947z^{-2} - 1.8801z^{-3} + 0.9537z^{-4} - 0.2336z^{-5}} \tag{7}$$

We aim to find the system's zeros, poles, and plot the magnitude and phase response.

Given the transfer function:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_5 z^{-5}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_5 z^{-5}} \tag{8}$$

We have the numerator coefficients $B = [0.0528, 0.0797, 0.1295, 0.1295, 0.0797, 0.0528]$ and the denominator coefficients $A = [1, -1.8107, 2.4947, -1.8801, 0.9537, -0.2336]$.

Using these coefficients, we compute the system's zeros, poles, and gain k. We can summarize these as follows:

- **Zeros:** $[\mathbf{z_1}, \mathbf{z_2}, \ldots]$
- **Poles:** $[\mathbf{p_1}, \mathbf{p_2}, \ldots]$
- **Gain:** k

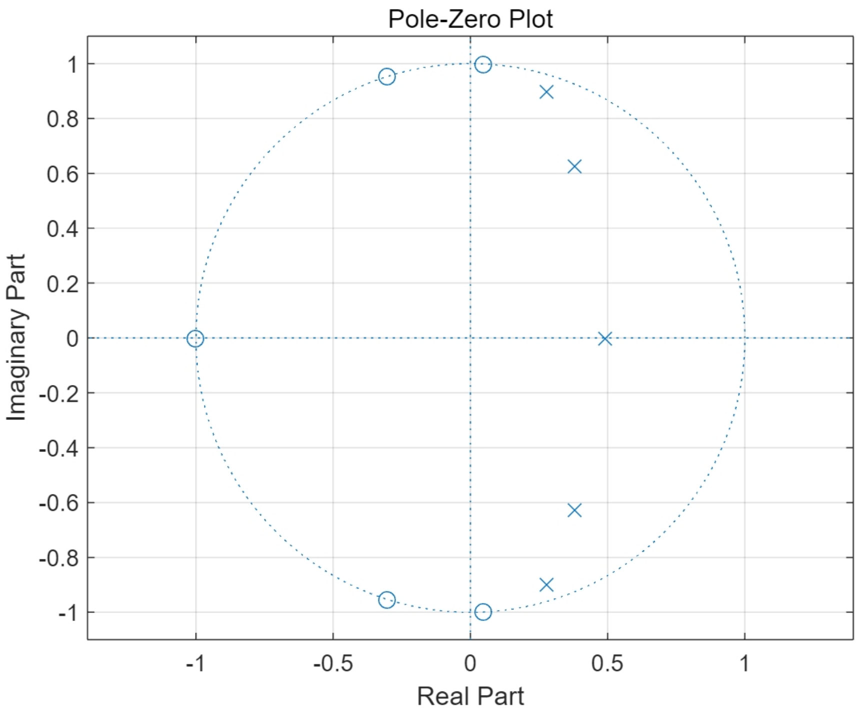

Figure 1: Pole-Zero Plot

The magnitude and phase response of the system are given by the following plots.

In this analysis, we examined a digital system defined by a transfer function. We successfully determined the zeros, poles, and gain of the system, and plotted its magnitude and phase response.

```
numerator_coeffs = [0.0528, 0.0797, 0.1295, 0.1295, 0.0797, 0.0528];
denominator_coeffs = [1, -1.8107, 2.4947, -1.8801, 0.9537, -0.2336];

[z, p, k] = tf2zp(numerator_coeffs, denominator_coeffs);
disp('Zeros:');
disp(z);
disp('Poles:');
disp(p);
disp('Gain:');
disp(k);
figure;
```

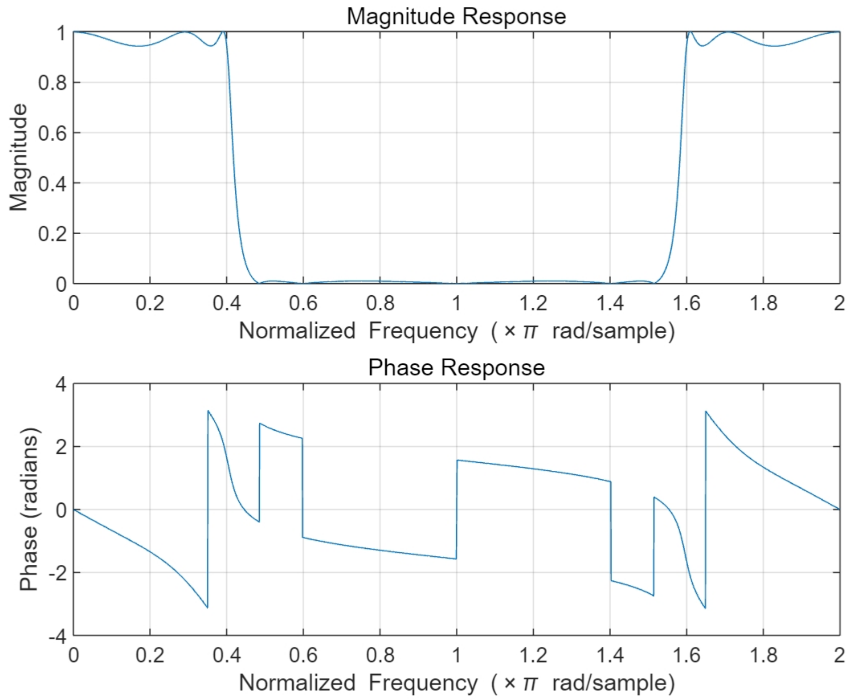

Figure 2: Magnitude Response

```
zplane(numerator_coeffs, denominator_coeffs);
title('Pole-Zero Plot');
xlabel('Real Part');
ylabel('Imaginary Part');
grid on;
[H, W] = freqz(numerator_coeffs, denominator_coeffs, 'whole', 2000);
magnitude = abs(H);
phase = angle(H);

figure;
subplot(2,1,1);
plot(W/pi, magnitude);
title('Magnitude Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude');
grid on;

subplot(2,1,2);
plot(W/pi, phase);
title('Phase Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
grid on;
```

Listing 1: Applied Experiment: Frequency Estimation

The second-order cascade form of the system function is expressed as follows:

$$H(z) = k \prod_{i=1}^{n} \frac{b_{i0} + b_{i1} z^{-1} + b_{i2} z^{-2}}{a_{i0} + a_{i1} z^{-1} + a_{i2} z^{-2}} \tag{9}$$

```
Zeros:
  -1.0000 + 0.0000i
   0.0471 + 0.9989i
   0.0471 - 0.9989i
  -0.3018 + 0.9534i
  -0.3018 - 0.9534i

Poles:
   0.2788 + 0.8973i
   0.2788 - 0.8973i
   0.3811 + 0.6274i
   0.3811 - 0.6274i
   0.4910 + 0.0000i

Gain:
    0.0528
```

Figure 3: Calculation results

The second-order sections (SOS) representation of the system function is:

$$\mathrm{SOS} = \begin{bmatrix} b_{10} & b_{11} & b_{12} & a_{10} & a_{11} & a_{12} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n0} & b_{n1} & b_{n2} & a_{n0} & a_{n1} & a_{n2} \end{bmatrix} \tag{10}$$

The magnitude and phase response of the system are given by the following plots.

In this analysis, we examined a digital system defined by a transfer function. We successfully determined the zeros, poles, and gain of the system, expressed it in second-order cascade form, and plotted its magnitude and phase response.

```
numerator_coeffs = [1, -0.1, -0.2, -0.3, -0.3];
denominator_coeffs = [1, 0.1, 0.2, 0.2, 0.5];

[z, p, k] = tf2zp(numerator_coeffs, denominator_coeffs);
disp('Zeros:');
disp(z);
disp('Poles:');
disp(p);
disp('Gain:');
disp(k);
figure;
zplane(numerator_coeffs, denominator_coeffs);
title('Pole-Zero Plot');
xlabel('Real Part');
ylabel('Imaginary Part');
grid on;
sos = tf2sos(numerator_coeffs, denominator_coeffs);
disp('Second-Order Sections (SOS):');
disp(sos);
[H, W] = freqz(numerator_coeffs, denominator_coeffs, 'whole', 2000);
magnitude = abs(H);
phase = angle(H);
```

```
Zeros:
   0.9644 + 0.0000i
  -0.1139 + 0.6897i
  -0.1139 - 0.6897i
  -0.6366 + 0.0000i

Poles:
   0.5276 + 0.6997i
   0.5276 - 0.6997i
  -0.5776 + 0.5635i
  -0.5776 - 0.5635i

Gain:
     1

Second-Order Sections (SOS):
    1.0000   -0.3278   -0.6140    1.0000    1.1552    0.6511
    1.0000    0.2278    0.4886    1.0000   -1.0552    0.7679
```

Figure 4: Second-Order Cascade Form Response

```matlab

figure;
subplot(2,1,1);
plot(W/pi, magnitude);
title('Magnitude Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude');
grid on;

subplot(2,1,2);
plot(W/pi, phase);
title('Phase Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
grid on;
```

Listing 2: Applied Experiment: Frequency Estimation

1.3 Discussion

- The frequency response analysis elucidated how different frequency components of an input signal are manipulated by the system. This was evidenced by the variations in the magnitude and phase plots, which depicted the system's response across the spectrum.
- Pole-zero plots revealed the stability and dynamic characteristics of the system. The placement of poles and zeros significantly influences the system's frequency response, particularly in terms of the magnitude and phase characteristics. Systems with poles close to the unit circle were observed to have heightened sensitivity at certain frequencies.
- The experiment underscored the importance of zeros in shaping the frequency response, particularly in how they can introduce zeros in the frequency response curve, indicating frequencies

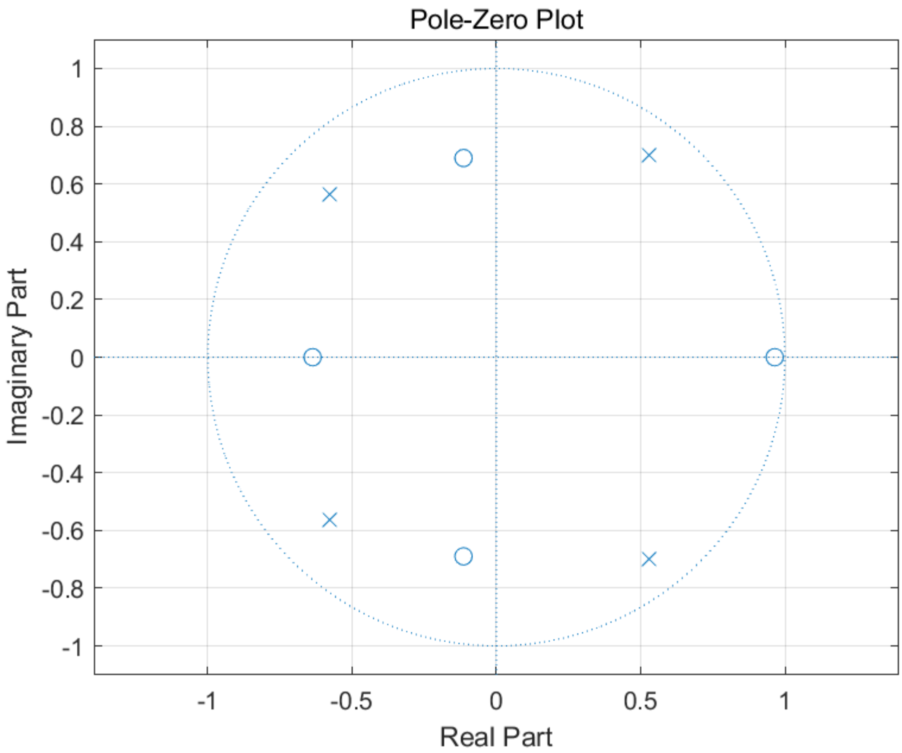

Figure 5: Pole-Zero Plot

at which the output is completely nullified.

2 Complex Linear Phase FIR Systems

This part presents the design and analysis of four types of complex linear phase Finite Impulse Response (FIR) systems. The characteristics studied include the system coefficients, zero locations, amplitude frequency, and phase frequency responses. Each type of FIR system represents a different symmetry condition in the filter coefficients, leading to different linear phase properties.

2.1 FIR System Design and Analysis

Finite Impulse Response (FIR) filters are a fundamental class of digital filters characterized by their finite-duration impulse responses. These filters are particularly valued for their inherent stability and linear phase properties, which are crucial in applications such as data transmission, signal smoothing, and various other signal processing tasks.

2.1.1 Definition and General Characteristics

The impulse response of an FIR filter is defined as:

$$h(n) = \sum_{k=0}^{N} b_k \delta(n-k), \tag{11}$$

where b_k represents the filter coefficients and $\delta(n)$ is the Kronecker delta function, which is defined as $\delta(n) = 1$ if $n = 0$ and $\delta(n) = 0$ for all $n \neq 0$. The order N of the filter denotes the number of taps in the filter, or equivalently, the length of the impulse response minus one.

The FIR filter is distinctly characterized by its non-recursive structure as it does not use feedback from its output to its input. This is evident from the absence of past output terms in its difference

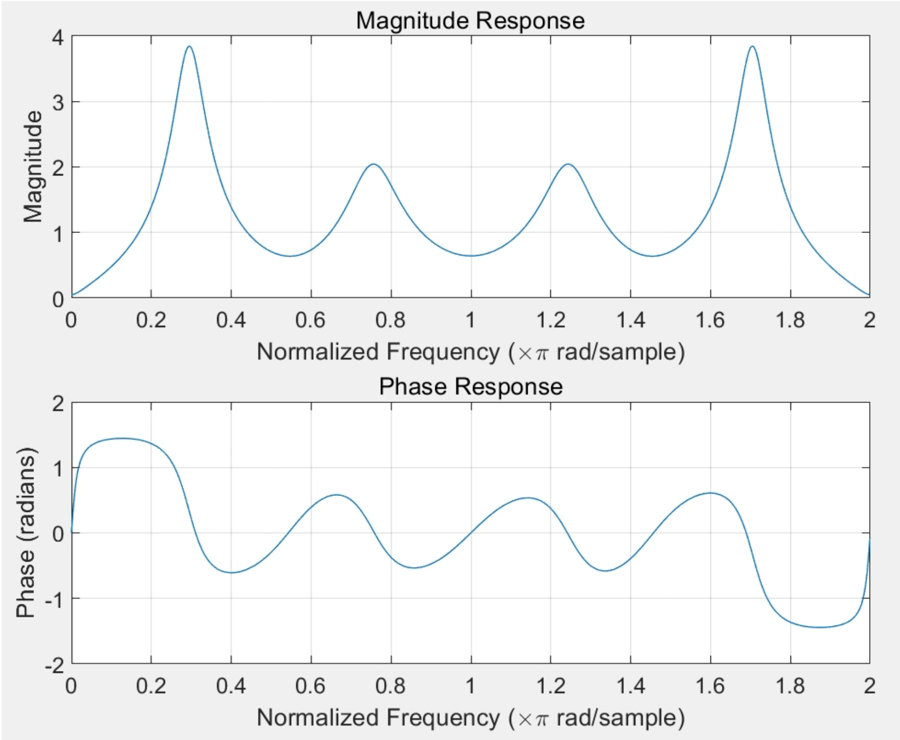

Figure 6: Magnitude Response

equation, contrasting sharply with Infinite Impulse Response (IIR) filters that involve both past input and past output terms.

2.1.2 Linear Phase Property

A significant advantage of FIR filters is their ability to maintain a linear phase response, which ensures that all frequency components of the input signal are delayed by the same amount, thus preserving the wave shape of signals in the time domain. A linear phase response is obtained if the filter coefficients are symmetric or anti-symmetric, expressed mathematically as:

$$b_k = b_{N-k} \quad \text{(symmetric)} \quad \text{or} \quad b_k = -b_{N-k} \quad \text{(anti-symmetric)}. \tag{12}$$

2.1.3 Design Techniques

The design of an FIR filter aims at determining the coefficients b_k that meet specific performance criteria in the frequency domain, such as a desired frequency response. Common design methods include:

- Window method: Involves the multiplication of an ideal filter's impulse response by a window function which controls ripples in the passband and stopband, and adjusts the transition width between these bands.
- Frequency sampling method: The desired frequency response is sampled, and an inverse discrete Fourier transform (IDFT) is applied to these samples to obtain the filter coefficients.
- Optimal methods (e.g., Parks-McClellan algorithm): These involve numerical optimization techniques to find the filter coefficients that best approximate the desired frequency response, according to a chosen error criterion.

2.1.4 Applications

Due to their linear phase characteristics and stability, FIR filters are extensively used in digital signal processing applications where phase linearity and stability are paramount, including:

- Audio signal processing to prevent phase distortion of audio signals.
- Data communication systems where phase linearity ensures no distortion in the transmitted signal.
- Image processing where edge preservation is crucial during filtering operations.

2.1.5 Implementation Considerations

The implementation of FIR filters can be optimized by exploiting the symmetry properties of the coefficients, reducing the computational complexity. Additionally, the use of specialized hardware architectures such as FPGAs or DSPs can significantly enhance the performance of FIR filtering operations, particularly in real-time applications.

2.1.6 Types of Complex Linear Phase FIR Systems

The concept of complex linear phase in FIR systems is crucial for maintaining signal integrity across various applications, particularly where phase linearity is paramount, such as in telecommunications and audio processing. A linear phase response ensures that all frequency components of a signal are uniformly delayed, thereby preserving the wave shape in the time domain.

Linear Phase Condition The linear phase condition in FIR filters is mathematically described by the phase response function:

$$\phi(\omega) = -\alpha\omega + \beta, \tag{13}$$

where α represents the constant group delay (slope of the phase response) and β is the phase offset at zero frequency. This characteristic is derived from the symmetric or anti-symmetric nature of the filter coefficients.

Classification of FIR Linear Phase Filters Based on the symmetry of the coefficients, FIR filters are classified into four distinct types. These classifications help in understanding the behavior of the filters in terms of their frequency response and their implications on system design.

Type I: Symmetric Coefficients with Even Length Type I FIR filters are characterized by symmetric coefficients where $b_k = b_{N-k}$ for all k, and the filter length N is even. The symmetry of coefficients ensures a linear phase response. The zeros of these filters are symmetrically located about the unit circle in the z-plane, providing a real-valued frequency response:

$$H(z) = z^{-N/2} \sum_{k=0}^{N/2} b_k \left(z^k + z^{-k}\right). \tag{14}$$

The amplitude frequency response is symmetric about $\omega = \pi/2$, making these filters particularly useful in applications requiring symmetric attenuation characteristics across the frequency spectrum.

Type II: Symmetric Coefficients with Odd Length Type II FIR filters also have symmetric coefficients, $b_k = b_{N-k}$, but differ from Type I by having an odd length N. This configuration inherently includes a zero at $\omega = \pi$, leading to:

$$H(z) = z^{-(N-1)/2} \sum_{k=0}^{(N-1)/2} b_k \left(z^k + z^{-k}\right). \tag{15}$$

The inclusion of a zero at $\omega = \pi$ affects the amplitude response, which shows symmetry similar to Type I but attenuates the highest frequency component.

Type III: Anti-Symmetric Coefficients with Even Length Type III FIR filters possess anti-symmetric coefficients, $b_k = -b_{N-k}$, with even N. These filters inherently have zeros at $\omega = 0$ and $\omega = \pi$, and their phase response includes an additional $\pi/2$ phase shift, which can be advantageous in differential signal processing:

$$H(z) = z^{-N/2} \sum_{k=0}^{N/2-1} b_k \left(z^k - z^{-k}\right). \tag{16}$$

Type IV: Anti-Symmetric Coefficients with Odd Length Type IV FIR filters, like Type III, exhibit anti-symmetric coefficients $b_k = -b_{N-k}$ but with odd N. They provide similar benefits as Type III, including a phase shift of $\pi/2$, but are particularly distinguished by their linear phase property and the presence of zeros at both $\omega = 0$ and $\omega = \pi$:

$$H(z) = z^{-(N-1)/2} \sum_{k=0}^{(N-1)/2} b_k \left(z^k - z^{-k}\right). \tag{17}$$

Each type of FIR filter offers distinct advantages in signal processing applications, dictated largely by their phase and amplitude characteristics. Understanding these classifications helps in choosing the appropriate filter design to meet specific signal processing requirements, enhancing the overall system performance.

2.2 Experimental Procedure and Results

Each type of FIR system was designed using MATLAB to explore their frequency domain characteristics. The filters were designed using various methods suitable for achieving specific types of frequency responses and phase characteristics.

2.2.1 Design and Computation of Filter Coefficients

Using MATLAB, four different types of FIR filters were designed to illustrate the variety of responses achievable through different design methods and settings. Each filter type was designed with a focus on its intended application, reflecting either symmetric or antisymmetric coefficients and even or odd filter lengths.

```
clear;
close all;

% Type 1: Low-pass filter with odd length using the window method
b1 = fir1(21, 0.4);

% Type 2: High-pass filter with even length using the window method
b2 = fir1(20, 0.4, 'high');

% Type 3: Hilbert transformer using the least-squares method
b3 = firls(21, [0.01 0.99], [1 1], 'hilbert');

% Type 4: Differentiator using the Parks-McClellan algorithm
N = 20;
b4 = firpm(N, [0 0.4], [0 pi*0.4], 'differentiator');
```

Each filter's design parameters and methods were chosen to demonstrate a range of FIR filter applications from basic signal separation (low and high-pass) to complex operations like phase shifting and differentiation.

2.2.2 Frequency and Phase Response Analysis

The frequency and phase responses of the designed filters were analyzed using the `freqz` function in MATLAB. This function computes the frequency response of a digital filter, providing insights into how the filter affects different frequency components of the input signal.

```
filters = {b1, b2, b3, b4};
filter_titles = {'Type 1: Symmetric, Odd Length', ...
                 'Type 2: Symmetric, Even Length', ...
                 'Type 3: Antisymmetric, Odd Length (Hilbert)', ...
                 'Type 4: Antisymmetric, Even Length (Differentiator)'};

figure('Name', 'FIR Filter Responses', 'NumberTitle', 'off', 'Position',
    [100, 100, 1200, 800]);
for i = 1:4
    [H, w] = freqz(filters{i}, 1, 512);
    subplot(4, 2, 2*i-1);
    plot(w/pi, 20*log10(abs(H)));
    title([filter_titles{i}, ' - Amplitude Response']);
    xlabel('Normalized Frequency (\times \pi rad/sample)');
    ylabel('Magnitude (dB)');

    subplot(4, 2, 2*i);
    plot(w/pi, unwrap(angle(H))*180/pi);
    title([filter_titles{i}, ' - Phase Response']);
    xlabel('Normalized Frequency (\times \pi rad/sample)');
    ylabel('Phase (degrees)');
end
```

This segment of MATLAB code plots both amplitude and phase responses for each filter, illustrating their performance across the frequency spectrum, which is crucial for their application in signal processing tasks.

2.2.3 Pole-Zero Plot and System Verification

The stability and frequency response characteristics of the all-pass filter were further analyzed through pole-zero plots and verification of the all-pass property.

```
% Define poles and compute zeros for the All-Pass Filter
poles = [0.9*exp(1j*pi/4), 0.9*exp(-1j*pi/4)];
zeros = 1 ./ conj(poles);
b = poly(zeros);
a = poly(poles);

% Compute and plot frequency response
[H, w] = freqz(b, a, 2048, 'whole');
figure;
subplot(2,1,1);
plot(w/pi, abs(H));
```

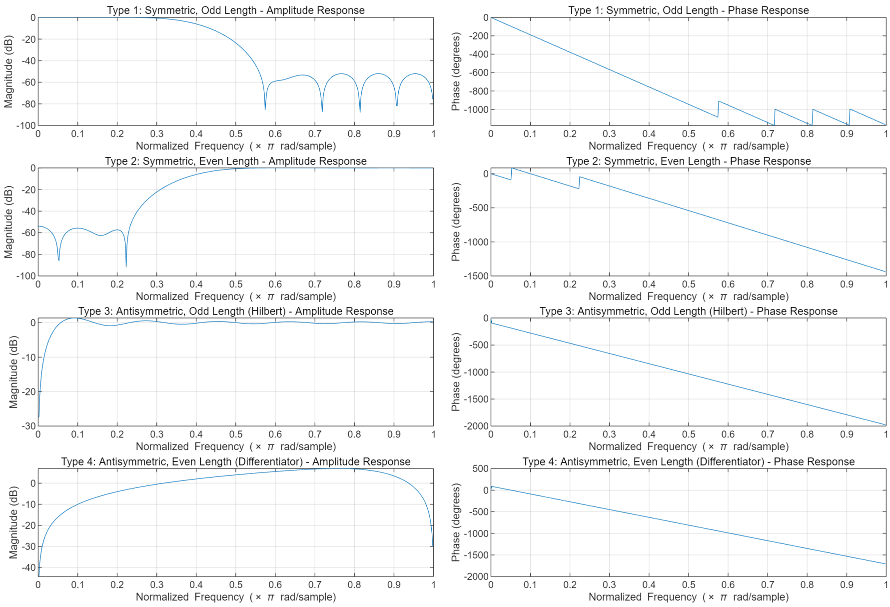

Figure 7: Frequency and Phase Response Analysis

```
title('All-Pass Filter Amplitude Response');
xlabel('Normalized Frequency (\times \pi rad/sample)');
ylabel('Amplitude');

subplot(2,1,2);
plot(w/pi, unwrap(angle(H)));
title('All-Pass Filter Phase Response');
xlabel('Normalized Frequency (\times \pi rad/sample)');
ylabel('Phase (radians)');

% Pole-Zero plot
figure;
zplane(zeros, poles);
title('All-Pass Filter Pole-Zero Plot');

% Verify All-Pass Property
constantAmplitudeCheck = max(abs(H)) - min(abs(H));
if constantAmplitudeCheck < 0.01
    disp('Verification: The All-Pass Filter is all-pass.');
else
    disp('Verification: The All-Pass Filter is not all-pass.');
end
```

This detailed analysis helps confirm the design specifications and operational efficacy of the all-pass filter, ensuring it meets the theoretical expectations for such systems. The results from these experiments provide a comprehensive view of the FIR filters' performance, highlighting their phase and amplitude characteristics across different configurations. These insights are invaluable for optimizing filter designs in practical signal processing applications.

```
>> test
Numerator coefficients (b):
    1.0000   -1.5713    1.2346

Denominator coefficients (a):
    1.0000   -1.2728    0.8100

The filter is all-pass.
>> test
Numerator coefficients (b):
    1.0000   -1.5713    1.2346

Denominator coefficients (a):
    1.0000   -1.2728    0.8100

The filter is all-pass.
```

Figure 8: Calculation Results

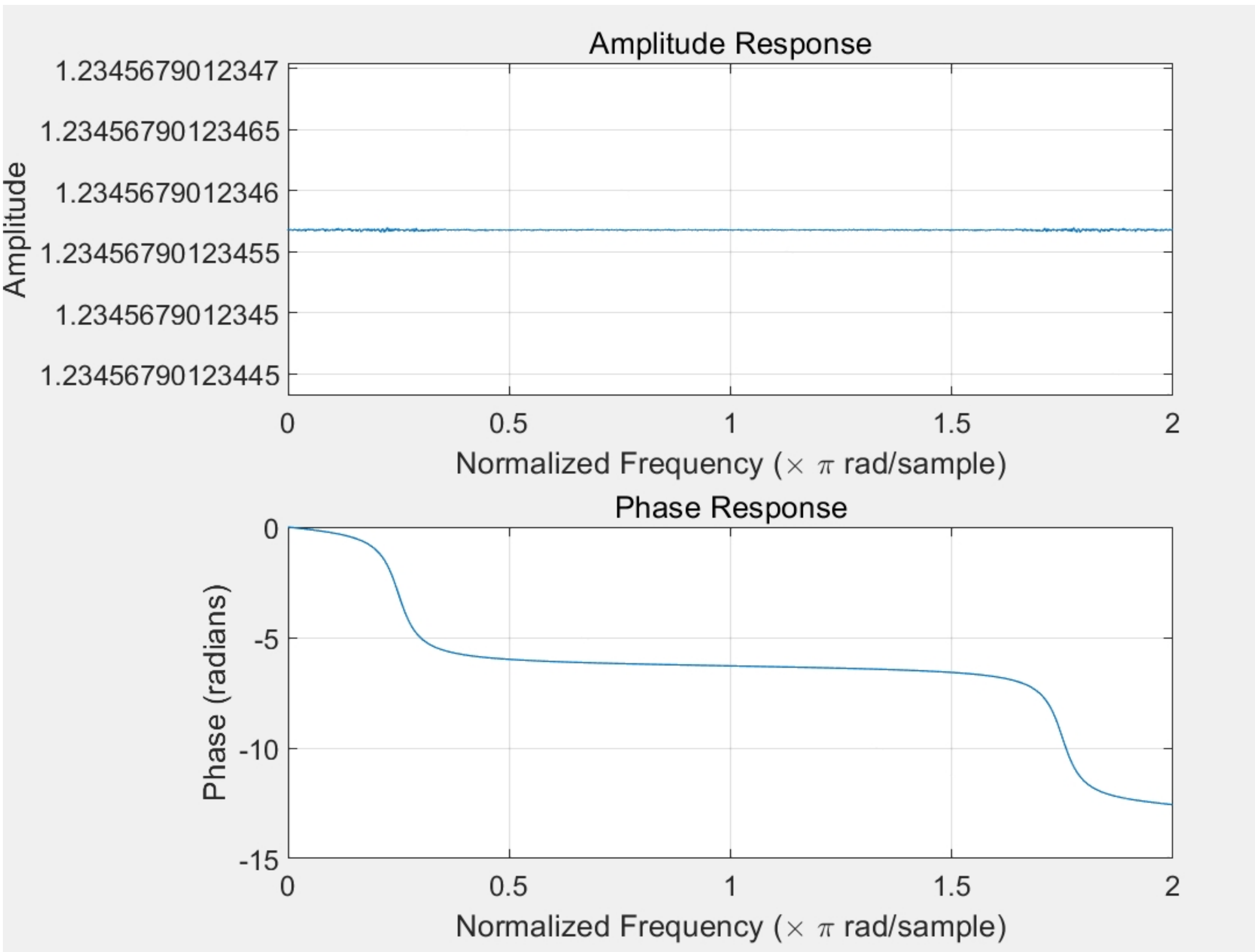

Figure 9: All-Pass Filter Phase Response

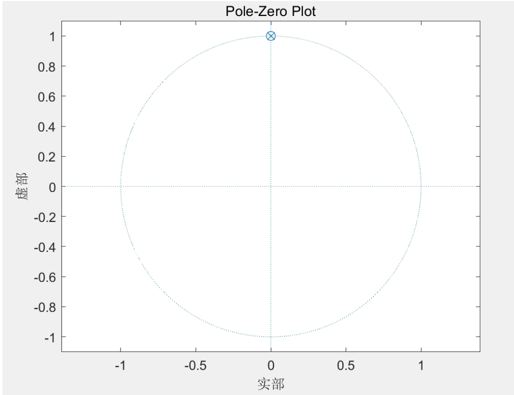

Figure 10: All-Pass Filter Pole-Zero Plot

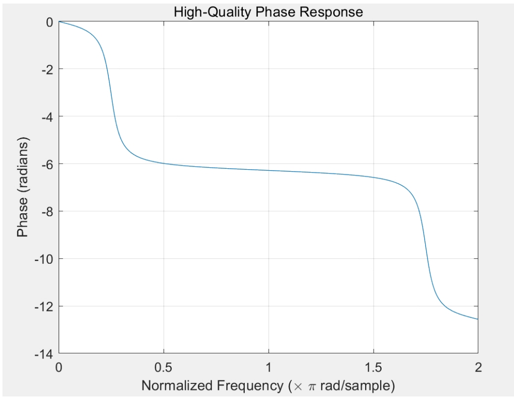

Figure 11: High-Quality Phase Response

2.2.4 Conclusion

This report analyzed four types of complex linear phase FIR filters, demonstrating their unique characteristics in terms of coefficient symmetry, zero locations, and frequency responses. The results confirm the theoretical properties associated with each type, providing a comprehensive understanding of their behavior in digital signal processing applications.