Discrete-Time Signals in the Frequency Domain 3

3.1 Introduction

In the previous two exercises you dealt with the time-domain representation of discrete-time signals and systems, and investigated their properties. Further insight into the properties of such signals and systems is obtained by their representation in the frequency-domain. To this end three commonly used representations are the discrete-time Fourier transform (DTFT), the discrete Fourier transform (DFT), and the z-transform. In this exercise you will study all three representations of a discrete-time sequence.

3.2 Background Review

R3.1 The *discrete-time Fourier transform* (DTFT) $X(e^{j\omega})$ of a sequence $x[n]$ is defined by

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}. \tag{3.1}$$

In general $X(e^{j\omega})$ is a complex function of the real variable ω and can be written as

$$X(e^{j\omega}) = X_{re}(e^{j\omega}) + jX_{im}(e^{j\omega}), \tag{3.2}$$

where $X_{re}(e^{j\omega})$ and $X_{im}(e^{j\omega})$ are, respectively, the real and imaginary parts of $X(e^{j\omega})$, and are real functions of ω. $X(e^{j\omega})$ can alternately be expressed in the form

$$X(e^{j\omega}) = |X(e^{j\omega})|e^{j\theta(\omega)}, \tag{3.3}$$

where

$$\theta(\omega) = \arg\{X(e^{j\omega})\}. \tag{3.4}$$

The quantity $|X(e^{j\omega})|$ is called the *magnitude function* and the quantity $\theta(\omega)$ is called the *phase function* , with both functions again being real functions of ω. In many applications, the Fourier transform is called the *Fourier spectrum* and, likewise, $|X(e^{j\omega})|$ and $\theta(\omega)$ are referred to as the *magnitude spectrum* and *phase spectrum*, respectively.

R3.2 The DTFT $X(e^{j\omega})$ is a periodic continuous function in ω with a period 2π.

R3.3 For a real sequence $x[n]$, the real part $X_{re}(e^{j\omega})$ of its DTFT and the magnitude function $|X(e^{j\omega})|$ are even functions of ω, whereas the imaginary part $X_{im}(e^{j\omega})$ and the phase function $\theta(\omega)$ are odd functions of ω.

R3.4 The *inverse discrete-time Fourier transform* $x[n]$ of $X(e^{j\omega})$ is given by

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega. \tag{3.5}$$

R3.5 The Fourier transform $X(e^{j\omega})$ of a sequence $x[n]$ exists if $x[n]$ is *absolutely summable*, that is,

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty, \tag{3.6}$$

R3.6 The DTFT satisfies a number of useful properties that are often uitilized in a number of applications. A detailed listing of these properties and their analytical proofs can be found in any text on digital signal processing. These properties can also be verified using MATLAB. We list below a few selected properties that will be encountered later in this exercise.

Time-Shifting Property – If $G(e^{j\omega})$ denotes the DTFT of a sequence $g[n]$, then the DTFT of the time-shifted sequence $g[n - n_o]$ is given by $e^{-j\omega n_o} G(e^{j\omega})$.

Frequency-Shifting Property – If $G(e^{j\omega})$ denotes the DTFT of a sequence $g[n]$, then the DTFT of the sequence $e^{j\omega_o n} g[n]$ is given by $G(e^{j(\omega-\omega_o)})$.

Convolution Property – If $G(e^{j\omega})$ and $H(e^{j\omega})$ denote the DTFTs of the sequences $g[n]$ and $h[n]$, respectively, then the DTFT of the sequence $g[n] \star h[n]$ is given by $G(e^{j\omega})H(e^{j\omega})$.

Modulation Property – If $G(e^{j\omega})$ and $H(e^{j\omega})$ denote the DTFTs of the sequences $g[n]$ and $h[n]$, respectively, then the DTFT of the sequence $g[n]h[n]$ is given by

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} G(e^{j\theta}) H(e^{j(\omega-\theta)}) d\theta.$$

Time-Reversal Property – If $G(e^{j\omega})$ denotes the DTFT of a sequence $g[n]$, then the DTFT of the time-reversed sequence $g[-n]$ is given by $G(e^{-j\omega})$.

R3.7 The N-point *discrete Fourier transform* (DFT) of a finite-length sequence $x[n]$, defined for $0 \le n \le N-1$, is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0, 1, \ldots, N-1, \tag{3.7}$$

where

$$W_N = e^{-j2\pi/N}. \tag{3.8}$$

R3.8 The N-point DFT $X[k]$ of a length-N sequence $x[n]$, $n = 0, 1, \ldots, N-1$, is simply the frequency samples of its DTFT $X(e^{j\omega})$ evaluated at N uniformly spaced frequency points, $\omega = \omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, that is,

$$X[k] = X(e^{j\omega})|_{\omega=2\pi k/N}, \qquad k = 0, 1, \ldots, N-1. \tag{3.9}$$

R3.9 The N-point *circular convolution* of two length-N sequences $g[n]$ and $h[n]$, $0 \leq n \leq N-1$, is defined by

$$y_C[n] = \sum_{m=0}^{N-1} g[m]h[\langle n-m\rangle_N], \tag{3.10}$$

where $\langle n\rangle_N = n$ modulo N. The N-point circular convolution operation is usually denoted as

$$y_C[n] = g[n] \,Ⓝ\, h[n]. \tag{3.11}$$

R3.10 The linear convolution of a length-N sequence $g[n]$, $0 \leq n \leq N-1$, with a length-M sequence $h[n]$, $0 \leq n \leq M-1$, can be obtained by a $(N+M-1)$-point circular convolution of two length-$(N+M-1)$ sequences, $g_e[n]$ and $h_e[n]$,

$$y_L[n] = g[n] \circledast h[n] = g_e[n] \,Ⓝ\, h_e[n], \tag{3.12}$$

where $g_e[n]$ and $h_e[n]$ are obtained by *appending* $g[n]$ and $h[n]$ with zero-valued samples:

$$g_e[n] = \begin{cases} g[n], & 0 \leq n \leq N-1, \\ 0, & N \leq n \leq N+M-1, \end{cases} \tag{3.13}$$

$$h_e[n] = \begin{cases} h[n], & 0 \leq n \leq M-1, \\ 0, & M \leq n \leq N+M-1. \end{cases} \tag{3.14}$$

R3.11 The DFT satisfies a number of useful properties that are often utilized in a number of applications. A detailed listing of these properties and their analytical proofs can be found in any text on digital signal processing. These properties can also be verified using MATLAB. We list below a few selected properties that will be encountered later in this exercise.

Circular Time-Shifting Property – If $G[k]$ denotes the N-point DFT of a length-N sequence $g[n]$, then the N-point DFT of the circularly time-shifted sequence $g[\langle n-n_o\rangle_N]$ is given by $W_N^{kn_o}G[k]$ where $W_N = e^{-j2\pi/N}$.

Circular Frequency-Shifting Property – If $G[k]$ denotes the N-point DFT of a length-N sequence $g[n]$, then the N-point DFT of the sequence $W_N^{-k_o n}g[n]$ is given by $G[\langle k-k_o\rangle_N]$.

Circular Convolution Property – If $G[k]$ and $H[k]$ denote the N-point DFTs of the length-N sequences $g[n]$ and $h[n]$, respectively, then the N-point DFT of the circularly convolved sequence $g[n] \,Ⓝ\, h[n]$ is given by $G[k]H[k]$.

Parseval's Relation – If $G[k]$ denotes the N-point DFT of a length-N sequence $g[n]$, then

$$\sum_{n=0}^{N-1} |g[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |G[k]|^2. \tag{3.15}$$

R3.12 The periodic even part $g_{pe}[n]$ and the periodic odd part $g_{po}[n]$ of a length-N real sequence $g[n]$ are given by

$$g_{pe}[n] = \tfrac{1}{2}\left(g[n] + g[\langle -n\rangle_N]\right), \tag{3.16}$$

$$g_{po}[n] = \tfrac{1}{2}\left(g[n] - g[\langle -n\rangle_N]\right). \tag{3.17}$$

If $G[k]$ denotes the N-point DFT of $g[n]$, then the N-point DFTs of $g_{pe}[n]$ and $g_{po}[n]$ are given by Re$\{G[k]\}$ and Im$j\ \{G[k]\}$, respectively.

R3.13 Let $g[n]$ and $h[n]$ be two length-N real sequences, with $G[k]$ and $H[k]$ denoting their respective N-point DFTs. These two N-point DFTs can be computed efficiently using a single N-point DFT $X[k]$ of a complex length-N sequence $x[n]$ defined by $x[n] = g[n] + jh[n]$ using

$$G[k] = \tfrac{1}{2}\left(X[k] + X^*[\langle -k\rangle_N]\right), \tag{3.18}$$

$$H[k] = \tfrac{1}{2j}\left(X[k] - X^*[\langle -k\rangle_N]\right). \tag{3.19}$$

R3.14 Let $v[n]$ be a real sequence of length $2N$ with $V[k]$ denoting its $2N$-point DFT. Define two real sequences $g[n]$ and $h[n]$ of length N each as

$$g[n] = v[2n] \ \ and \ \ h[n] = v[2n+1], \qquad 0 \le n < N, \tag{3.20}$$

with $G[k]$ and $H[k]$ denoting their N-point DFTs. Then the $2N$-point DFT $V[k]$ of $v[n]$ can be computed from the two N-point DFTs, $G[k]$ and $H[k]$, using

$$V[k] = G[\langle k\rangle_N] + W_{2N}^{k} H[\langle k\rangle_N], \qquad 0 \le k \le 2N-1. \tag{3.21}$$

R3.15 The *z-transform* $G(z)$ of a sequence $g[n]$ is defined as

$$G(z) = \mathcal{Z}\{g[n]\} \sum_{n=-\infty}^{\infty} g[n] z^{-n}, \tag{3.22}$$

where z is a complex variable. The set $\Re$ of values of z for which the z-transform $G(z)$ converges is called its *region of convergence* (ROC). In general, the region of convergence $\Re$ of a z-transform of a sequence $g[n]$ is an annular region of the z-plane:

$$R_{g-} < |z| < R_{g+}, \tag{3.23}$$

where $0 \le R_{g-} < R_{g+} \le \infty$.

R3.16 In the case of LTI discrete-time systems , all pertinent z-transforms are rational functions of z^{-1}, that is, they are ratios of two polynomials in z^{-1}:

$$G(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1 z^{-1} + \ldots + p_{M-1} z^{-(M-1)} + p_M z^{-M}}{d_0 + d_1 z^{-1} + \ldots + d_{N-1} z^{-(N-1)} + d_N z^{-N}}, \tag{3.24}$$

which can be alternately written in factored form as

$$G(z) = \frac{p_0}{d_0} \frac{\prod_{r=1}^{M}(1 - \xi_r z^{-1})}{\prod_{s=1}^{N}(1 - \lambda_s z^{-1})} = \frac{p_0}{d_0} z^{N-M} \frac{\prod_{r=1}^{M}(z - \xi_r)}{\prod_{s=1}^{N}(z - \lambda_s)}. \tag{3.25}$$

The *zeros* of $G(z)$ are given by $z = \xi_r$ while the *poles* are given by $z = \lambda_s$. There are additional $(N-M)$ zeros at $z = 0$ (the origin in the z-plane) if $N > M$ or additional $(M-N)$ poles at $z = 0$ if $N < M$.

R3.17 For a sequence with a rational z-transform, the ROC of the z-transform cannot contain any poles and is bounded by the poles.

R3.18 The *inverse z-transform* $g[n]$ of a z-transform $G(z)$ is given by

$$g[n] = \frac{1}{2\pi j} \oint_C G(z)\, z^{n-1}\, dz, \tag{3.26}$$

where C is a counterclockwise contour encircling the point $z0$ in the ROC of $G(z)$.

R3.19 A rational z-transform $G(z) = P(z)/D(z)$, where the degree of the polynomial $P(z)$ is M and the degree of the polynomial $D(z)$ is N, and with distinct poles at $z = \lambda_s, s = 1, 2, \ldots, N$, can be expressed in a partial-fraction expansion form given by

$$G(z) = \sum_{\ell=0}^{M-N} \eta_\ell z^{-\ell} + \sum_{s=0}^{N} \frac{\rho_s}{1 - \lambda_s z^{-1}}, \tag{3.27}$$

assuming $M \geq N$. The constants ρ_s in the above expression, called the *residues* , are given by

$$\rho_s = (1 - \lambda_s z^{-1}) G(z)|_{z=\lambda_s}. \tag{3.28}$$

If $G(z)$ has multiple poles , the partial-fraction expansion is of slightly different form. For example, if the pole at $z = \nu$ is of multiplicity L and the remaining $N - L$ poles are simple and at $z = \lambda_s, s1, 2, \ldots, N - L$, then the general partial fraction expansion of $G(z)$ takes the form

$$G(z) = \sum_{\ell=0}^{M-N} \eta_\ell z^{-\ell} + \sum_{s=0}^{N-L} \frac{\rho_s}{1 - \lambda_s z^{-1}} + \sum_{r=1}^{L} \frac{\gamma_r}{(1 - \nu\, z^{-1})^r}, \tag{3.29}$$

where the constants γ_r (no longer called the residues for $r \neq 1$) are computed using the formula

$$\gamma_r = \frac{1}{(L-r)!(-\nu)^{L-r}} \frac{d^{L-r}}{d(z^{-1})^{L-r}} \left[(1 - \nu z^{-1})^L G(z)\right]_{z=\nu}, \qquad r = 1, \ldots, L, \tag{3.30}$$

and the residues ρ_s are calculated using Eq. (3.28).

3.3 MATLAB Commands Used

The MATLAB commands you will encounter in this exercise are as follows:

General Purpose Commands

```
disp
```

Operators and Special Characters

```
:     .     +     -     *     /     ;
%     <     >     .*    ^     .^    ~=
```

Language Constructs and Debugging

```
break    end      error    for      function
if       input    pause
```

Elementary Matrices and Matrix Manipulation

```
fliplr   i   pi   zeros   :
```

Elementary Functions

```
abs    angle    conj    exp    imag    real
rem
```

Polynomial and Interpolation Functions

```
conv
```

Two-Dimensional Graphics

```
axis     grid     plot     stem     title
xlabel   ylabel
```

General Purpose Graphics Functions

```
clf    subplot
```

Character String Functions

```
num2str
```

Data Analysis and Fourier Transform Functions

```
fft    ifft    max    min
```

Signal Processing Toolbox

freqz	impz	residuez	tf2zp	zp2sos
zp2tf	zplane			

For additional information on these commands, see the *MathWorks Online Documentation* [Mat05] or type `help commandname` in the Command window. A brief explanation of the MATLAB functions used here can be found in Appendix B.

3.4 Discrete-Time Fourier Transform

The discrete-time Fourier transform (DTFT) $X(e^{j\omega})$ of a sequence $x[n]$ is a continuous function of ω. Since the data in MATLAB is in vector form, $X(e^{j\omega})$ can only be evaluated at a prescribed set of discrete frequencies. Moreover, only a class of the DTFT that is expressed as a rational function in $e^{-j\omega}$ in the form

$$X(e^{j\omega}) = \frac{p_0 + p_1 e^{-j\omega} + \ldots + p_M e^{-j\omega M}}{d_0 + d_1 e^{-j\omega} + \ldots + d_N e^{-j\omega N}}, \tag{3.31}$$

can be evaluated. In the following two projects you will learn how to evaluate and plot the DTFT and study certain properties of the DTFT using MATLAB.

Project 3.1 DTFT Computation

The DTFT $X(e^{j\omega})$ of a sequence $x[n]$ of the form of Eq. (3.31) can be computed easily at a prescribed set of L discrete frequency points $\omega = \omega_\ell$ using the MATLAB function `freqz`. Since $X(e^{j\omega})$ is a continuous function of ω, it is necessary to make L as large as possible so that the plot generated using the command `plot` provides a resonable replica of the actual plot of the DTFT. In MATLAB, `freqz` computes the L-point DFT of the sequences $\{p_0 \quad p_1 \ldots p_M\}$ and $\{d_0 \quad d_1 \ldots d_M\}$, and then forms their ratio to arrive at $X(e^{j\omega_\ell}), \ell = 1, 2, \ldots, L$. For faster computation, L should be chosen as a power of 2, such as 256 or 512.

Program P3_1 can be used to evaluate and plot the DTFT of the form of Eq. (3.31).

```
% Program P3_1
% Evaluation of the DTFT
clf;
% Compute the frequency samples of the DTFT
w = -4*pi:8*pi/511:4*pi;
num = [2 1];den = [1 -0.6];
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
```

```
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,angle(h));grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase, radians');
```

Questions:

Q3.1 What is the expression of the DTFT being evaluated in Program P3_1? What is the function of the MATLAB command `pause`?

Q3.2 Run Program P3_1 and compute the real and imaginary parts of the DTFT, and the magnitude and phase spectra . Is the DTFT a periodic function of ω? If it is, what is the period? Explain the type of symmetries exhibited by the four plots.

Q3.3 Modify Program P3_1 to evaluate in the range $0 \le \omega \le \pi$ the following DTFT:

$$U(e^{j\omega}) = \frac{0.7 - 0.5e^{-j\omega} + 0.3e^{-j2\omega} + e^{-j3\omega}}{1 + 0.3e^{-j\omega} - 0.5e^{-j2\omega} + 0.7e^{-j3\omega}},$$

and repeat Question Q3.2. Comment on your results. Can you explain the jump in the phase spectrum ? The jump can be removed using the MATLAB command `unwrap`. Evaluate the phase spectrum with the jump removed.

Q3.4 Modify Program P3_1 to evaluate the DTFT of the following finite-length sequence:

$$g[n] = [1 \quad 3 \quad 5 \quad 7 \quad 9 \quad 11 \quad 13 \quad 15 \quad 17],$$

and repeat Question Q3.2. Comment on your results. Can you explain the jumps in the phase spectrum?

Q3.5 How would you modify Program P3_1 to plot the phase in degrees?

Project 3.2 DTFT Properties

Most of the properties of the DTFT can be verified using MATLAB. In this project you shall verify the properties listed in R3.6. Since all data in MATLAB have to be finite-length vectors, the sequences being used to verify the properties are thus restricted to be of finite length.

Program P3_2 can be used to verify the time-shifting property of the DTFT.

```
% Program P3_2
% Time-Shifting Properties of DTFT
clf;
w = -pi:2*pi/255:pi; wo = 0.4*pi; D = 10;
num = [1 2 3 4 5 6 7 8 9];
h1 = freqz(num, 1, w);
h2 = freqz([zeros(1,D) num], 1, w);
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('Magnitude Spectrum of Original Sequence')
subplot(2,2,2)
plot(w/pi,abs(h2));grid
title('Magnitude Spectrum of Time-Shifted Sequence')
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('Phase Spectrum of Original Sequence')
subplot(2,2,4)
plot(w/pi,angle(h2));grid
title('Phase Spectrum of Time-Shifted Sequence')
```

Questions:

Q3.6 Modify Program P3_2 by adding appropriate comment statements and program statements for labeling the two axes of each plot being generated by the program. Which parameter controls the amount of time-shift?

Q3.7 Run the modified program and comment on your results.

Q3.8 Repeat Question Q3.7 for a different value of the time-shift.

Q3.9 Repeat Question Q3.7 for two different sequences of varying lengths and two different time-shifts.

Program P3_3 can be used to verify the frequency-shifting property of the DTFT.

```
% Program P3_3
% Frequency-Shifting Properties of DTFT
```

```
clf;
w = -pi:2*pi/255:pi; wo = 0.4*pi;
num1 = [1 3 5 7 9 11 13 15 17];
L = length(num1);
h1 = freqz(num1, 1, w);
n = 0:L-1;
num2 = exp(wo*i*n).*num1;
h2 = freqz(num2, 1, w);
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('Magnitude Spectrum of Original Sequence')
subplot(2,2,2)
plot(w/pi,abs(h2));grid
title('Magnitude Spectrum of Frequency-Shifted Sequence')
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('Phase Spectrum of Original Sequence')
subplot(2,2,4)
plot(w/pi,angle(h2));grid
title('Phase Spectrum of Frequency-Shifted Sequence')
```

Questions:

Q3.10 Modify Program P3_3 by adding appropriate comment statements and program statements for labeling the two axes of each plot being generated by the program. Which parameter controls the amount of frequency-shift?

Q3.11 Run the modified program and comment on your results.

Q3.12 Repeat Question Q3.11 for a different value of the frequency-shift.

Q3.13 Repeat Question Q3.11 for two different sequences of varying lengths and two different frequency-shifts.

Program P3_4 can be used to verify the convolution property of the DTFT.

```
% Program P3_4
% Convolution Property of DTFT
clf;
w = -pi:2*pi/255:pi;
x1 = [1 3 5 7 9 11 13 15 17];
x2 = [1 -2 3 -2 1];
y = conv(x1,x2);
h1 = freqz(x1, 1, w);
h2 = freqz(x2, 1, w);
hp = h1.*h2;
```

```
h3 = freqz(y,1,w);
subplot(2,2,1)
plot(w/pi,abs(hp));grid
title('Product of Magnitude Spectra')
subplot(2,2,2)
plot(w/pi,abs(h3));grid
title('Magnitude Spectrum of Convolved Sequence')
subplot(2,2,3)
plot(w/pi,angle(hp));grid
title('Sum of Phase Spectra')
subplot(2,2,4)
plot(w/pi,angle(h3));grid
title('Phase Spectrum of Convolved Sequence')
```

Questions:

Q3.14 Modify Program P3_4 by adding appropriate comment statements and program statements for labeling the two axes of each plot being generated by the program.

Q3.15 Run the modified program and comment on your results.

Q3.16 Repeat Question Q3.15 for two different sets of sequences of varying lengths.

Program P3_5 can be used to verify the modulation property of the DTFT.

```
% Program P3_5
% Modulation Property of DTFT
clf;
w = -pi:2*pi/255:pi;
x1 = [1 3 5 7 9 11 13 15 17];
x2 = [1 -1 1 -1 1 -1 1 -1 1];
y = x1.*x2;
h1 = freqz(x1, 1, w);
h2 = freqz(x2, 1, w);
h3 = freqz(y,1,w);
subplot(3,1,1)
plot(w/pi,abs(h1));grid
title('Magnitude Spectrum of First Sequence')
subplot(3,1,2)
plot(w/pi,abs(h2));grid
title('Magnitude Spectrum of Second Sequence')
subplot(3,1,3)
plot(w/pi,abs(h3));grid
title('Magnitude Spectrum of Product Sequence')
```

Questions:

Q3.17 Modify Program P3_5 by adding appropriate comment statements and program statements for labeling the two axes of each plot being generated by the program.

Q3.18 Run the modified program and comment on your results.

Q3.19 Repeat Question Q3.18 for two different sets of sequences of varying lengths.

Program P3_6 can be used to verify the time-reversal property of the DTFT.

```
% Program P3_6
% Time-Reversal Property of DTFT
clf;
w = -pi:2*pi/255:pi;
num = [1 2 3 4];
L = length(num)-1;
h1 = freqz(num, 1, w);
h2 = freqz(fliplr(num), 1, w);
h3 = exp(w*L*i).*h2;
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('Magnitude Spectrum of Original Sequence')
subplot(2,2,2)
plot(w/pi,abs(h3));grid
title('Magnitude Spectrum of Time-Reversed Sequence')
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('Phase Spectrum of Original Sequence')
subplot(2,2,4)
plot(w/pi,angle(h3));grid
title('Phase Spectrum of Time-Reversed Sequence')
```

Questions:

Q3.20 Modify Program P3_6 by adding appropriate comment statements and program statements for labeling the two axes of each plot being generated by the program. Explain how the program implements the time-reversal operation.

Q3.21 Run the modified program and comment on your results.

Q3.22 Repeat Question Q3.21 for two different sequences of varying lengths.

3.5 Discrete Fourier Transform

The discrete Fourier transform (DFT) $X[k]$ of a finite-length sequence $x[n]$ can be easily computed in MATLAB using the function `fft`. There are two versions of this function. `fft(x)` computes the DFT $X[k]$ of the sequence $x[n]$ where the length of $X[k]$ is the same as that of $x[n]$. `fft(x,L)` computes the L-point DFT of a sequence $x[n]$ of length N where $L \geq N$. If $L > N$, $x[n]$ is zero-padded with $L - N$ trailing zero-valued samples before the DFT is computed. The inverse discrete Fourier transform (IDFT) $x[n]$ of a DFT sequence $X[k]$ can likewise be computed using the function `ifft`, which also has two versions.

Project 3.3 DFT and IDFT Computations

Questions:

Q3.23 Write a MATLAB program to compute and plot the L-point DFT $X[k]$ of a sequence $x[n]$ of length N with $L \geq N$ and then to compute and plot the L-point IDFT of $X[k]$. Run the program for sequences of different lengths N and for different values of the DFT length L. Comment on your results.

Q3.24 Write a MATLAB program to compute the N-point DFT of two length-N real sequences using a single N-point DFT and compare the result by computing directly the two N-point DFTs (see R3.13).

Q3.25 Write a MATLAB program to compute the $2N$-point DFT of a length-$2N$ real sequence using a single N-point DFT and compare the result by computing directly the $2N$-point DFT (see R3.14).

Project 3.4 DFT Properties

Two important concepts used in the application of the DFT are the *circular-shift* of a sequence and the *circular convolution* of two sequences of the same length. As these operations are needed in verifying certain properties of the DFT, we implement them as MATLAB functions `circshift1` and `circonv` as indicated below:

```
function y = circshift1(x,M)
% Develops a sequence y obtained by
% circularly shifting a finite-length
% sequence x by M samples
if abs(M) > length(x)
    M = rem(M,length(x));
end
if M < 0
    M = M + length(x);
end
```

```
y = [x(M+1:length(x)) x(1:M)];}

function y = circonv(x1,x2)
L1 = length(x1); L2 = length(x2);
if L1 ~= L2, error('Sequences of unequal lengths'), end
y = zeros(1,L1);
x2tr = [x2(1) x2(L2:-1:2)];
for k = 1:L1
    sh = circshift1(x2tr,1-k);
    h = x1.*sh;
    y(k) = sum(h);
end
```

Questions:

Q3.26 What is the purpose of the command `rem` in the function `circshift1`?

Q3.27 Explain how the function `circshift1` implements the circular time-shifting operation.

Q3.28 What is the purpose of the operator `~=` in the function `circonv`?

Q3.29 Explain how the operation of the function `circonv` implements the circular convolution operation.

Program P3_7 can be used to illustrate the concept of circular shift of a finite-length sequence. It employs the function `circshift1`.

```
% Program P3_7
%  Illustration of Circular Shift of a Sequence
clf;
M = 6;
a = [0 1 2 3 4 5 6 7 8 9];
b = circshift1(a,M);
L = length(a)-1;
n = 0:L;
subplot(2,1,1);
stem(n,a);axis([0,L,min(a),max(a)]);
title('Original Sequence');
subplot(2,1,2);
stem(n,b);axis([0,L,min(a),max(a)]);
title(['Sequence Obtained by Circularly Shifting by ',num2str(M),'
Samples']);
```

Questions:

Q3.30 Modify Program P3_7 by adding appropriate comment statements and program statements for labeling each plot being generated by the program. Which parameter determines the amount of time-shifting? What happens if the amount of time-shift is greater than the sequence length?

Q3.31 Run the modified program and verify the circular time-shifting operation.

Program P3_8 can be used to illustrate the circular time-shifting property of the DFT. It employs the function `circshift1`.

```
% Program P3_8
% Circular Time-Shifting Property of DFT
clf;
x = [0 2 4 6 8 10 12 14 16];
N = length(x)-1; n = 0:N;
y = circshift1(x,5);
XF = fft(x);
YF = fft(y);
subplot(2,2,1)
stem(n,abs(XF)); grid
title('Magnitude of DFT of Original Sequence');
subplot(2,2,2)
stem(n,abs(YF)); grid
title('Magnitude of DFT of Circularly Shifted Sequence');
subplot(2,2,3)
stem(n,angle(XF)); grid
title('Phase of DFT of Original Sequence');
subplot(2,2,4)
stem(n,angle(YF)); grid
title('Phase of DFT of Circularly Shifted Sequence');
```

Questions:

Q3.32 Modify Program P3_8 by adding appropriate comment statements and program statements for labeling each plot being generated by the program. What is the the amount of time-shift?

Q3.33 Run the modified program and verify the circular time-shifting property of the DFT.

Q3.34 Repeat Question Q3.33 for two different amounts of time-shift.

Q3.35 Repeat Question Q3.33 for two different sequences of different lengths.

Program P3_9 can be used to illustrate the circular convolution property of the DFT. It employs the function `circonv`.

```
% Program P3_9
% Circular Convolution Property of DFT
g1 = [1 2 3 4 5 6]; g2 = [1 -2 3 3 -2 1];
ycir = circonv(g1,g2);
disp('Result of circular convolution = ');disp(ycir)
G1 = fft(g1); G2 = fft(g2);
yc = real(ifft(G1.*G2));
disp('Result of IDFT of the DFT products = ');disp(yc)
```

Questions:

Q3.36 Run Program P3_9 and verify the circular convolution property of the DFT.

Q3.37 Repeat Question Q3.36 for two other different sets of equal-length sequences.

Program P3_10 can be used to illustrate the relation between circular and linear convolutions (see R3.10).

```
% Program P3_10
% Linear Convolution via Circular Convolution
g1 = [1 2 3 4 5];g2 = [2 2 0 1 1];
g1e = [g1 zeros(1,length(g2)-1)];
g2e = [g2 zeros(1,length(g1)-1)];
ylin = circonv(g1e, g2e);
disp('Linear convolution via circular convolution = ');disp(ylin);
y = conv(g1, g2);
disp('Direct linear convolution = ');disp(y)
```

Questions:

Q3.38 Run Program P3_10 and verify that linear convolution can be obtained via circular convolution.

Q3.39 Repeat Question Q3.38 for two other different sets of sequences of unequal lengths.

Q3.40 Write a MATLAB program to develop the linear convolution of two sequences via the DFT of each. Using this program verify the results of Questions Q3.38 and Q3.39.

Program P3_11 can be used to verify the relation between the DFT of a real sequence, and the DFTs of its periodic even and the periodic odd parts (see R3.12).

```
% Program P3_11
% Relations between the DFTs of the Periodic Even
% and Odd Parts of a Real Sequence
x = [1 2 4 2 6 32 6 4 2 zeros(1,247)];
x1 = [x(1) x(256:-1:2)];
xe = 0.5 *(x + x1);
XF = fft(x);
XEF = fft(xe);
clf;
k = 0:255;
subplot(2,2,1);
plot(k/128,real(XF)); grid
ylabel('Amplitude');
title('Re(DFT\{x[n]\})');
subplot(2,2,2);
plot(k/128,imag(XF)); grid ylabel('Amplitude');
title('Im(DFT\{x[n]\})');
subplot(2,2,3);
plot(k/128,real(XEF)); grid
xlabel('Time index n'); ylabel('Amplitude');
title('Re(DFT\{x_{e}[n]\} )');
subplot(2,2,4);
plot(k/128,imag(XEF)); grid
xlabel('Time index n');ylabel('Amplitude');
title('Im(DFT\{x_{e}[n]\})');
```

Questions:

Q3.41 What is the relation between the sequences `x1[n]` and `x[n]`?

Q3.42 Run Program P3_11. The imaginary part of `XEF` should be zero as the DFT of the periodic even part is simply the real part of `XEF` of the original sequence. Can you verify that? How can you explain the simulation result?

Q3.43 Modify the program to verify the relation between the DFT of the periodic odd part and the imaginary part of `XEF`.

Parseval's relation (Eq. (3.15)) can be verified using the following program.

```
% Program P3_12
% Parseval's Relation
x = [(1:128) (128:-1:1)];
XF = fft(x);
a = sum(x.*x)
b = round(sum(abs(XF).^2)/256})
```

Questions:

Q3.44 Run Program P3_12. Do you get the same values for `a` and `b`?

Q3.45 Modify the program in such a way that you do not have to use the command `abs(XF)`. Use the MATLAB command `conj(x)` to compute the complex conjugate of `x`.

3.6 z-Transform

As in the case of the discrete-time Fourier transform, we restrict our attention here to a z-transform $G(z)$ of a sequence $g[n]$ that is a rational function of the complex variable z^{-1} and expressed in the form of a ratio of polynomials in z^{-1} as in Eq. (3.24) or in factored form as in Eq. (3.25). Some of the operations that are of interest in practice are as follows. (1) Evaluate the z-transform $G(z)$ on the unit circle, that is, evaluate $G(e^{j\omega})$; (2) develop the pole-zero plot of $G(z)$; (3) develop the factored form of $G(z)$; (4) determine the inverse z-transform $g[n]$ of $G(z)$; and (5) make a partial-fraction expansion of $G(z)$. In the next two projects you will learn how to perform the above operations using MATLAB.

Project 3.5 Analysis of z-Transforms

The function `freqz` can be used to evaluate the values of a rational z-transform on the unit circle . To this end, Program P3_1 can be used without any modifications.

Question:

Q3.46 Using Program P3_1 evaluate the following z-transform on the unit circle:

$$G(z) = \frac{2 + 5z^{-1} + 9z^{-2} + 5z^{-3} + 3z^{-4}}{5 + 45z^{-1} + 2z^{-2} + z^{-3} + z^{-4}}. \tag{3.32}$$

The pole-zero plot of a rational z-transform $G(z)$ can be readily obtained using the function `zplane`. There are two versions of this function. If the z-transform is given in the form of a rational function as in Eq. (3.32), the command to use is `zplane(num, den)` where `num` and `den` are row vectors containing the coefficients of the numerator and denominator polynomials of $G(z)$ in ascending powers of z^{-1}. On the other hand, if the zeros and poles of $G(z)$ are given, the command to use is `zplane(zeros, poles)` where `zeros` and `poles` are column vectors. In the pole-zero plot generated by MATLAB, the location of a pole is indicated by the symbol $\times$ and the location of a zero is indicated by the symbol $\circ$.

The function `tf2zp` can be used to determine the zeros and poles of a rational z-transform $G(z)$. The program statement to use is `[z, p, k] = tf2zp(num,den)` where `num` and `den` are row vectors containing the coefficients of the numerator and denominator polynomials of $G(z)$ in ascending powers of z^{-1} and the output file contains the gain constant `k` and the computed zeros and poles given as column vectors `z` and `p`, respectively. The factored form of the z-transform can be obtained from the zero-pole description using the function `sos = zp2sos(z,p,k)`. The function computes the coefficients of each second-order factor given as an $L \times 6$ matrix `sos` where

$$\texttt{sos} = \begin{bmatrix} b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & a_{0L} & a_{1L} & a_{2L} \end{bmatrix},$$

where the ℓth row contains the coefficients of the numerator and the denominator of the ℓth second-order factor of the z-transform $G(z)$:

$$G(z) = \prod_{\ell=1}^{L} \frac{b_{0\ell} + b_{1\ell}\, z^{-1} + b_{2\ell}\, z^{-2}}{a_{0\ell} + a_{1\ell}\, z^{-1} + a_{2\ell}\, z^{-2}}.$$

Questions:

Q3.47 Write a MATLAB program to compute and display the poles and zeros, to compute and display the factored form, and to generate the pole-zero plot of a z-transform that is a ratio of two polynomials in z^{-1}. Using this program, analyze the z-transform $G(z)$ of Eq. (3.32).

Q3.48 From the pole-zero plot generated in Question Q3.47, determine the number of regions of convergence (ROC) of $G(z)$. Show explicitly all possible ROCs . Can you tell from the pole-zero plot whether or not the DTFT exists?

The reverse process of converting a z-transform given in the form of zeros, poles, and the gain constant to a rational form can be implemented using the function `zp2tf`. The program statement to use is `[num,den] = zp2tf(z,p,k)`.

Question:

Q3.49 Write a MATLAB program to compute and display the rational z-transform from its zeros, poles and gain constant. Using this program, determine the rational form of a z-transform whose zeros are at $\xi_1 = 0.3, \xi_2 = 2.5, \xi_3 = -0.2 + j\,0.4$, and $\xi_4 = -0.2 - j\,0.4$; the poles are at $\lambda_1 = 0.5, \lambda_2 = -0.75, \lambda_3 0.6 + j\,0.7$, and $\lambda_4 = 0.6 - j\,0.7$; and the gain constant k is 3.9.

Project 3.6 Inverse z-Transform

The inverse $g[n]$ of a rational z-transform $G(z)$ can be computed using MATLAB in basically two different ways . To this end, it is necessary to know a priori the ROC of $G(z)$.

The function `impz` provides the samples of the time-domain sequence, which is assumed to be causal. There are three versions of this function: `[g,t] = impz(num,den)`, `[g,t] impz(num,den, L)`, and `[g,t] = impz(num,den, L, FT)`, where `num` and `den` are row vectors containing the coefficients of the numerator and denominator polynomials of $G(z)$ in ascending powers of z^{-1}, `L` is the desired number of the samples of the inverse transform, `g` is the vector containing the samples of the inverse transform starting with the sample at $n = 0$, `t` is the length of `g`, and `FT` is the specified sampling frequency in Hz with default value of unity.

A closed-form expression for the inverse of a rational z-transform can be obtained by first performing a partial-fraction expansion using the function `residuez` and then determining the inverse of each term in the expansion by looking up a table of z-transforms. The function `residuez` can also be used to convert a z-transform given in the form of a partial-fraction expansion to a ratio of polynomials in z^{-1}.

Questions:

Q3.50 Write a MATLAB program to compute the first L samples of the inverse of a rational z-transform where the value of L is provided by the user through the command `input`. Using this program compute and plot the first 50 samples of the inverse of $G(z)$ of Eq. (3.32). Use the command `stem` for plotting the sequence generated by the inverse transform.

Q3.51 Write a MATLAB program to determine the partial-fraction expansion of a rational z-transform. Using this program determine the partial-fraction expansion of $G(z)$ of Eq. (3.32) and then its inverse z-transform $g[n]$ in closed form. Assume $g[n]$ to be a causal sequence.

3.7 Background Reading

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications*. McGraw-Hill, New York NY, second edition, 1993. Chs. 2, 13.

[2] E. Cunningham. *Digital Filtering: An Introduction*. Houghton-Mifflin, Boston MA, 1992. Ch. 3.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York, NY, 1988. Secs. 2.1, 3.1–3.3, 6.1–6.4.

[4] L.B. Jackson. *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996. Ch. 3 and Secs. 6.1, 6.2, 7.1, 7.2.

[5] R. Kuc. *Introduction to Digital Signal Processing*. McGraw-Hill, New York NY, 1988. Chs. 3–5.

[6] L.C. Ludeman. *Fundamentals of Digital Signal Processing*. Harper & Row, New York NY, 1986. Secs. 1.4, 2.1, 2.2, 6.3.