Digital Signal Processing Ⅰ

7th Week EXPERIMENT

Report

(5th report of DSP1 course)

Subject	Digital Signal Processing Ⅰ
Professor	Joon-Hyuk Chang
Submission Date	April 14th, 2021
University	Hanyang University
School	College of Engineering
Department	Department of Computer Science & Engineering
Student ID	**Name**
2019009261	최가온(CHOI GA ON)

-Contents-

Ⅰ. Abstraction

1. The Properties of the DTFT

1-1. Linearity

The discrete-time Fourier transform is a linear transformation; that is,

$$\mathcal{F}[\alpha x_1(n) + \beta x_2(n)] = \alpha\mathcal{F}[x_1(n)] + \beta\mathcal{F}[x_2(n)]$$

for every $\alpha, \beta, x_1(n), x_2(n)$.

1-2. Time shifting

A shift in the time domain corresponds to the phase shifting.

$$\mathcal{F}[x(n-k)] = X\left(e^{jw}\right)e^{-jwk}$$

1-3. Frequency shifting

Multiplication by a complex exponential corresponding to a shift in the frequency domain.

$$\mathcal{F}\left[x(n)e^{jw_0 n}\right] = X(e^{j(w-w_0)})$$

1-4. Conjugation

Conjugation in the time domain corresponds to the folding and connjugation in the frequency domain.

$$\mathcal{F}[x^*(n)] = X^*(e^{-jw})$$

1-5. Folding

Folding in the time domain corresponds to the folding in the frequency domain.

$$\mathcal{F}[x(-n)] = X(e^{-jw})$$

1-6. Symmetries in real sequences

We have already studied the conjugate symmetry of real sequences. These real sequences can be decomposed into their even and odd parts, as discussed in Chapter 2.

$$x(n) = x_e(n) + x_o(n)$$

Then,

$$\mathcal{F}[x_e(n)] = Re[X(e^{jw})]$$

$$\mathcal{F}[x_0(n)] = j\ Im[X(e^{jw})]$$

If the sequence x(n) is real and even, then $X(e^{jw})$ is also real and even. Hence only one plot over [0, π] is necessary for its complete representation.

1-7. Convolution

This is one of the most useful properties that makes system analysis convevient in the frequency domain.

$$\mathcal{F}[x_1(n) * x_2(n)] = \mathcal{F}[x_1(n)] * \mathcal{F}[x_2(n)] = \mathrm{X}_1(e^{jw})X_2(e^{jw})$$

1-8. Multiplication

This a dual of the convolution property.

$$\mathcal{F}[x_1(n)\cdot x_2(n)] = \mathcal{F}[x_1(n)] * \mathcal{F}[x_2(n)] \triangleq \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1\left(e^{j\theta}\right)X_2\left(e^{j(w-\theta)}\right)d\theta$$

This convolution-like operation is called a periodic convolution and hence donoted by * . It is discussed in Chapter 5.

1-9. Energy

The energy of the sequence x(n) can be written as

$$\varepsilon_x = \sum_{-\infty}^{\infty}|x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|X\left(e^{jw}\right)\right|^2 dw = \int_0^{\pi}\frac{\left|X\left(e^{jw}\right)\right|^2}{\pi}dw$$

(for real sequences using even symmetry)

This is also known as Parseval's theorem. The energy density spectrum of x(n) is defined as:

$$\Phi_x(w) \triangleq \frac{\left|\mathrm{X}\left(\mathrm{e}^{\mathrm{j}w}\right)\right|^2}{\pi}$$

Then the energy of x(n) in the $[\omega_1, \omega_2]$ band is given by

$$\int_{\omega_1}^{\omega_2}\Phi_x(\omega)\,d\omega \qquad 0 \le \omega_1 < \omega_2 \le \pi$$

Ⅱ. Exercises

In this part, there are four exercise questions. Each exercise consists of code and its result. All documents including Matlab code, result, and this report are uploaded in this website :

https://github.com/Gaon-Choi/ELE3076/tree/main/5_Property_of_DTFT

Example 3.7

In this example we will verify the linearity property using real-valued finite-duration sequences. Let $x_1(n)$ and $x_2(n)$ be two random sequences uniformly distributed between [0, 1] over 0 ≤ n ≤ 10. Then we can use our numerical discrete-time Fourier transform procedure as follows.

(Matlab code)

```
x1 = rand(1, 11); x2 = rand(1, 11); n = 0:10;
alpha = 2; beta = 3; k = 0:500; w = (pi/500)*k;
X1 = x1 * (exp(-1i * pi/500)).^(n'*k);  % DTFT of x1
X2 = x2 * (exp(-1i * pi/500)).^(n'*k);  % DTFT of x2

x = alpha * x1 + beta * x2;             % Linear combination of x1 & x2
X = x * exp((-1i * pi/500)).^(n'*k);     % DTFT of x
% verification
X_check = alpha * X1 + beta * X2;        % Linear combination of X1 & X2
error = max(abs(X-X_check))             % Difference
```

(Result)

```
명령 창
>> example3_7

error =

   7.3241e-15

fx >>
```

Example 3.8

Let x(n) be a random sequence uniformly distributed between [0, 1] over $0 \le n \le 10$ and let $y(n) = x(n - 2)$. Then we can verify the sample shift property (3.6) as follows.

(Matlab code)

```
x = rand(1, 11); n = 0:10;
k = 0:500; w = (pi/500) * k;
X = x * (exp(-1i * pi/500)).^(n'*k);    % DTFT of x
% signal shifted by two samples
y = x; m = n + 2;
Y = y * (exp(-1i * pi/500)).^(m'*k);    % DTFT of y
% verification
Y_check = (exp(-1i * 2).^w).*X;        % multiplication by exp(-j2w)
error = max(abs(Y - Y_check))
```

(Result)

```
명령 창
>> example3_8

error =

   6.9529e-15

fx >>
```

Example 3.9

To verify the frequency shift property (3.7), we will use the graphical approach.

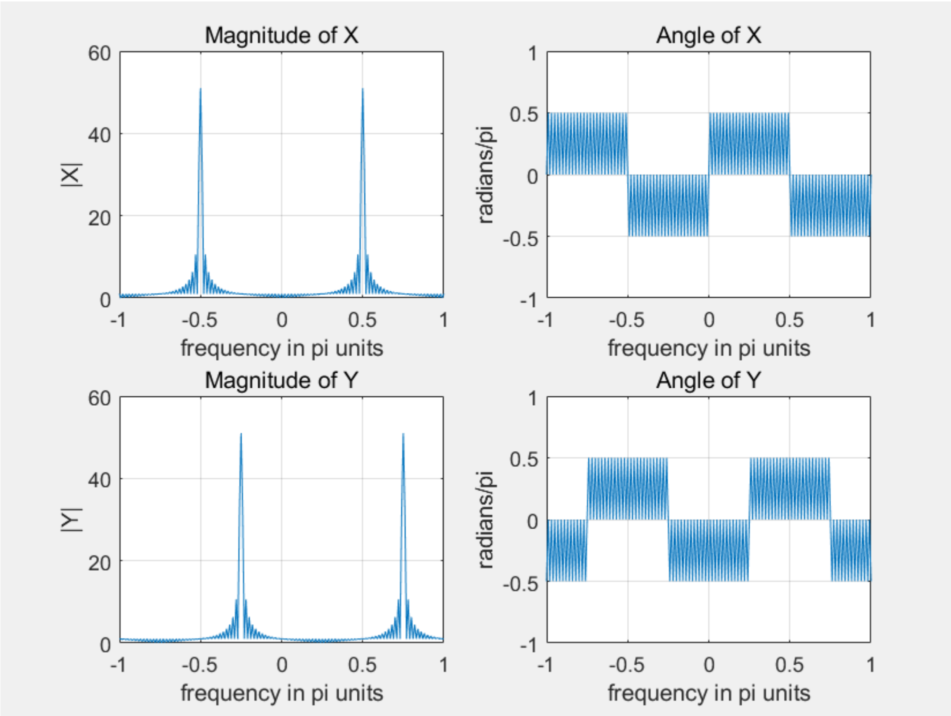
Let

$$x(n) = \cos\left(\frac{\pi n}{2}\right), \quad 0 \le n \le 100 \quad and \quad y(n) = e^{\frac{j\pi n}{4}} x(n)$$

(Matlab code)

```
n = 0:100; x = cos(pi * n/2);
k = -100:100; w = (pi/100) * k;         % frequency between -pi and
+pi
X = x * (exp(-1j * pi/100)).^(n'*k);    % DTFT of x

y = exp(1j * pi * n / 4).*x;           % signal multiplied by
exp(j*pi*n/4)
Y = y * (exp(-1j * pi / 100)).^(n'*k);  % DTFT of y
% Graphical verification
subplot(2, 2, 1); plot(w/pi, abs(X)); grid; axis([-1, 1, 0, 60])
xlabel("frequency in pi units"); ylabel("|X|")
title("Magnitude of X")

subplot(2, 2, 2); plot(w/pi, angle(X)/pi); grid; axis([-1, 1, -1, 1])
xlabel("frequency in pi units"); ylabel("radians/pi")
title("Angle of X")

subplot(2, 2, 3); plot(w/pi, abs(Y)); grid; axis([-1, 1, 0, 60])
xlabel("frequency in pi units"); ylabel("|Y|")
title("Magnitude of Y")

subplot(2, 2, 4); plot(w/pi, angle(Y)/pi); grid; axis([-1, 1, -1, 1])
xlabel("frequency in pi units"); ylabel("radians/pi")
title("Angle of Y")
```

(Result)



Magnitude of X
|X|
frequency in pi units
Angle of X
radians/pi
frequency in pi units
Magnitude of Y
|Y|
frequency in pi units
Angle of Y
radians/pi
frequency in pi units

Example 3.10

To verify the conjugation property (3.8), let x(n) be a complex-valued random sequence over $-5 \le n \le 10$ with real and imaginary parts uniformly distributed between [0, 1]. The MATLAB verification is as follows.

(Matlab code)

```
n = -5:10; x = rand(1, length(n)) + 1j * rand(1, length(n));
k = -100:100; w = (pi/100) * k;        % frequency between -
pi and +pi
X = x * (exp(-1j * pi / 100)).^(n'*k);  % DTFT of x
% conjugation property
y = conj(x);                            % signal conjugation
Y = y * (exp(-1j * pi / 100)).^(n'*k);  % DTFT of y
% verification
Y_check = conj(fliplr(X));              % conj(X(-w))
error = max(abs(Y - Y_check))           % Difference
```

(Result)

명령 창

```
>> example3_10

error =

   1.2219e-13

fx >>
```

Example 3.11

To verify the folding property (3.9), let x(n) be a random sequence over $-5 \le n \le 10$ uniformly distributed between [0, 1]. The MATLAB verification is as follows.

(Matlab code)

```
n = -5:10; x = rand(1, length(n));
k = -100:100; w = (pi/100) * k;          % frequency between -pi and +pi
X = x * (exp(-1j * pi/100)).^(n'*k);      % DTFT of x
% folding property
y = fliplr(x); m = -fliplr(n);           % signal folding
Y = y * (exp(-1j * pi/100)).^(m'*k);      % DTFT of y
% verification
Y_check = fliplr(X);
error = max(abs(Y - Y_check))
```

(Result)

```
명령 창
>> example3_11

error =

   1.0474e-15

fx >>
```

Example 3.12

In this probem we verify the symmetry property (3.10) of real signals.

Let

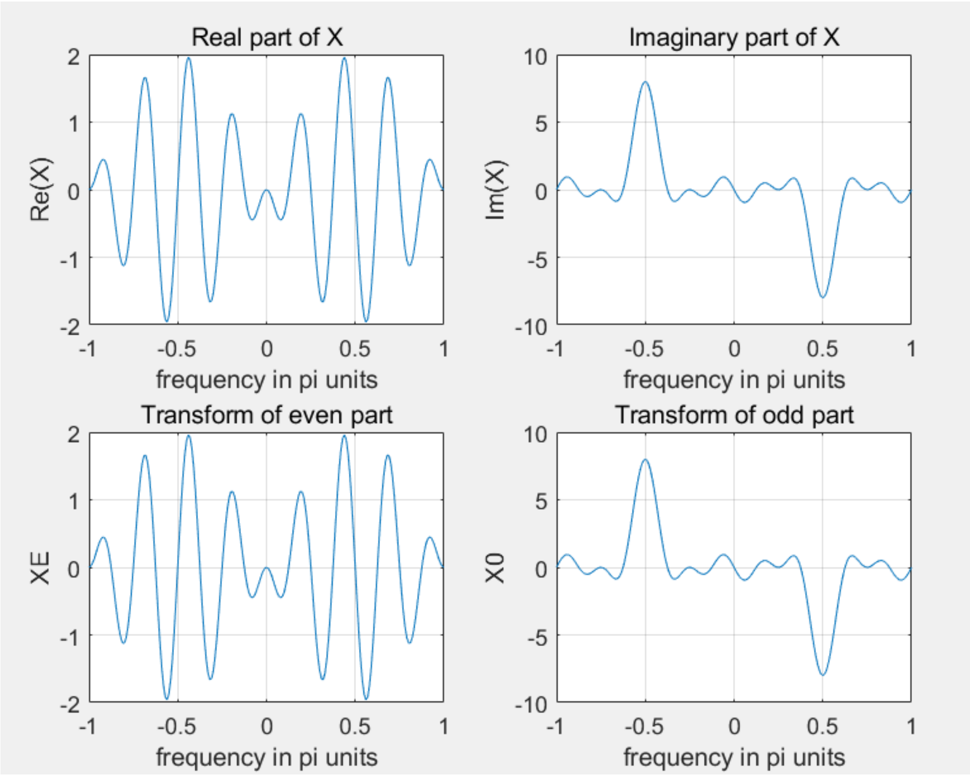
$$x(n) = \sin\left(\frac{\pi n}{2}\right), \qquad -5 \leq n \leq 10$$

Then using the evenodd function developed in Chapter 2, we can compute the even and odd parts of x(n) and then evalutate their discrete-time Fourier transforms. We will provide the numerical as well as graphical verification.

(Matlab code)

```
n = -5:10; x = sin(pi * n / 2);
k = -100:100; w = (pi / 100) * k;          % frequency between -pi and +pi
X = x * (exp(-1j * pi /100)).^(n'*k);       % DTFT of x
% signal decomposition
[xe, xo, m] = evenodd(x, n);               % even and odd parts
XE = xe * (exp(-1j * pi / 100)).^(m'*k);    % DTFT of xe
XO = xo * (exp(-1j * pi / 100)).^(m'*k);    % DTFT of xo

% verification
XR = real(X);                             % real part of X
error1 = max(abs(XE - XR))                  % Difference
XI = imag(X);                             % imag part of X
error2 = max(abs(XO - j*XI))                % Difference

% graphical verification
subplot(2, 2, 1); plot(w/pi, XR); grid; axis([-1, 1, -2, 2])
xlabel("frequency in pi units"); ylabel("Re(X)");
title("Real part of X")
subplot(2, 2, 2); plot(w/pi, XI); grid; axis([-1, 1, -10, 10])
xlabel("frequency in pi units"); ylabel("Im(X)");
title("Imaginary part of X")
subplot(2, 2, 3); plot(w/pi, real(XE)); grid; axis([-1, 1, -2, 2])
xlabel("frequency in pi units"); ylabel("XE");
title("Transform of even part")
subplot(2, 2, 4); plot(w/pi, imag(XO)); grid; axis([-1, 1, -10, 10])
xlabel("frequency in pi units"); ylabel("XO");
title("Transform of odd part")
```

(Result)


Real part of X
Re(X)
frequency in pi units
Imaginary part of X
Im(X)
frequency in pi units
Transform of even part
XE
frequency in pi units
Transform of odd part
XO
frequency in pi units