Digital Signal Processing Ⅱ

3^{rd} EXPERIMENT

Report

(3rd report of DSP2 course)

Subject	Digital Signal Processing Ⅱ
Professor	Je Hyeong Hong
Submission Date	September 19th, 2021
University	Hanyang University
School	College of Engineering
Department	Department of Computer Science & Engineering
Student ID	**Name**
2019009261	최가온(CHOI GA ON)

Exercises

In this part, there are several 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/ELE3077/tree/main/lab_experiment03

Exercise 1

Calculate the DTFT of $x[n] = (0.5)^n u(n)$ with your own hands. And plot magnitude, angle, real part, imaginary part at 501 equispaced points between [0, π].

By the definition of Discrete-time Fourier Transform(DTFT),

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} = \sum_{n=-\infty}^{\infty} (0.5)^n u[n] e^{-j\Omega n} = \sum_{n=0}^{\infty} (0.5)^n e^{-j\Omega n}$$

Nesting two terms $(0.5)^n$ and $e^{-j\Omega n}$ together with common part, and calculating with geometric sum formula,

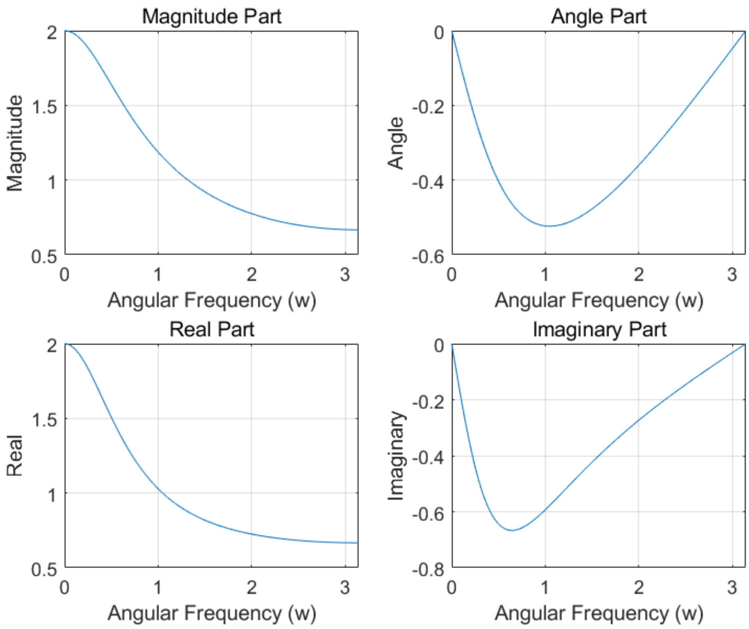
$$X(e^{j\Omega}) = \sum_{n=0}^{\infty} \left(0.5e^{-j\Omega}\right)^n = \frac{1}{1 - 0.5e^{-j\Omega}} = \frac{e^{j\Omega}}{e^{j\Omega} - 0.5}$$

(MATLAB code) lab3_exercise1.m

```
w = [0:1:500] * pi / 500;   % [0, pi] axis divided into 501 points
X = exp(j*w) ./ (exp(j*w) - 0.5 * ones(1, 501));

magX = abs(X);      % magnitude
angX = angle(X);    % angle
realX = real(X);    % real part
imagX = imag(X);    % imaginary part

subplot(2, 2, 1);   plot(w, magX); grid
xlabel("Angular Frequency (w)"); ylabel("Magnitude");
title("Magnitude Part");

subplot(2, 2, 2);   plot(w, angX); grid
```

```
xlabel("Angular Frequency (w)"); ylabel("Angle");
title("Angle Part");

subplot(2, 2, 3);   plot(w, realX); grid
xlabel("Angular Frequency (w)"); ylabel("Real");
title("Real Part");

subplot(2, 2, 4);   plot(w, imagX); grid
xlabel("Angular Frequency (w)"); ylabel("Imaginary");
title("Imaginary Part");
```

(Result)



Exercise 2

(a) Let $x[n]$ be a random sequence uniformly distributed between [0, 1] over $0 \leq n \leq 10$. Calculate numerically $X(e^{j\Omega})$ and plot its magnitude and its angle.

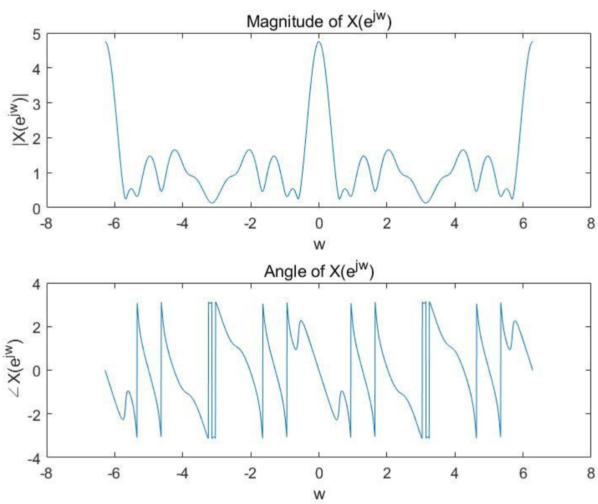
And, check its periodicity.

Hint: Equispaced frequencies between $[-2\pi, 2\pi]$

(MATLAB code) lab3_exercise2_a.m

```
x = rand(1, 11);
n = 0:10;
k = -1000:1000;
w = (pi/500) * k;

X = x * (exp(-j * pi / 500)).^(n'*k);

subplot(2, 1, 1);
plot(w, abs(X));
title("Magnitude of X(e^j^w)");
xlabel("w"); ylabel("|X(e^j^w)|");

subplot(2, 1, 2);
plot(w, angle(X));
title("Angle of X(e^j^w)");
xlabel("w"); ylabel("¡ÐX(e^j^w)");
```

(Results)

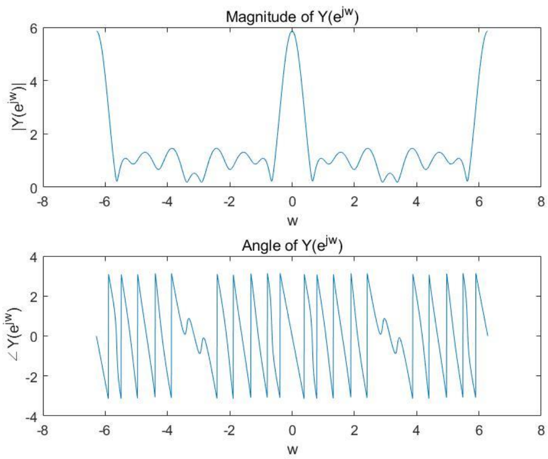


(b) Let $y[n] = x(n-2)$. Calculate numerically $Y(e^{j\Omega})$ and plot its magnitude and its angle.

(MATLAB code) lab3_exercise2_b.m

```
x = rand(1, 11);
n = 0:10;
k = -1000:1000;
w = (pi/500) * k;

y = x;  m = n + 2;
Y = y * (exp(-j * pi / 500)).^(m'*k);

subplot(2, 1, 1);
plot(w, abs(Y));
title("Magnitude of Y(e^j^w)");
xlabel("w"); ylabel("|Y(e^j^w)|");

subplot(2, 1, 2);
plot(w, angle(Y));
title("Angle of Y(e^j^w)");
xlabel("w"); ylabel("¡ÐY(e^j^w)");
```

(Results)



(c) Create $Y_{check} = X\left(e^{j\Omega}\right)e^{-j\Omega 2}$. And calculate the maximum value of difference between Y and Y_{check} to verify the sample shift property.

Hint: calculate abs(Y-Y_check), No (abs(Y)-abs(Y_check))!

(MATLAB code) lab3_exercise2_c.m

```
x = rand(1, 11);
n = 0:10;
k = -1000:1000;
w = (pi/500) * k;

X = x * (exp(-j * pi / 500)).^(n'*k);

y = x;  m = n + 2;
Y = y * (exp(-j * pi / 500)).^(m'*k);

Y_check = (exp(-j * 2).^w) .* X;

max_error = max(abs(Y - Y_check))
```

(Results)

Exercise 3

Sample signal and reconstruct from the samples.

- Generate 5Hz sinewave in the domain [-0.5 : 0.001 : 0.5]
- Make discrete signals by sampling the above signals at 8Hz, 12Hz
- Make the reconstructed signals by using the sinc function
- Plot above signals and compare between the original and reconstructed signals

(MATLAB code) lab3_exercise3.m

```
f0 = 5;

t = -0.5:0.001:0.5;
y = cos(2 * pi * f0 * t);

subplot(3, 2, 1:2);
set(gca,'XTick', (-0.5:0.1:0.5)); set(gca, "YTick", (-
1:1.0:1));
plot(t, y);
title("Original Signal");

subplot(3, 2, 3);
f1 = 8; t1 = -0.5:(1 / f1):0.5; Ts1 = 1 / f1;
yn_1 = cos(2 * pi * f0 * t1);
set(gca,'XTick', (-0.5:0.5:0.5)); set(gca, "YTick", (-
1:1.0:1));
stem(t1, yn_1);
title("8Hz Sampling");

subplot(3, 2, 4);
yn_r1 = zeros(1, length(t));
for i = 1:length(t1)
    for k = 1:length(t)
        yn_r1(k) = yn_r1(k) + yn_1(i) * sinc((t(k) - t1(i)) /
Ts1);
    end
end
plot(t, yn_r1);
hold on
stem(t1, yn_1)
```

```
title("8Hz Reconstruction");

subplot(3, 2, 5);
f2 = 12; t2 = -0.5:1 / f2:0.5;
yn_2 = cos(2*pi*f0*t2);
set(gca,'XTick', (-0.5:0.5:0.5)); set(gca, "YTick", (-
1:1.0:1));
stem(t2, yn_2);
title("12Hz Sampling");

subplot(3, 2, 6);
yn_r2 = zeros(1, length(t));
Ts2 = 1 / f2;
for i = 1:length(t2)
    for k = 1:length(t)
        yn_r2(k) = yn_r2(k) + yn_2(i) * sinc((t(k) - t2(i)) /
Ts2);
    end
end

plot(t, yn_r2);
hold on
stem(t2, yn_2)
title("12Hz Reconstruction");
```

(Results)