Digital Signal Processing Ⅱ

6^{th} EXPERIMENT

Report

(6th report of DSP2 course)

Subject	Digital Signal Processing Ⅱ
Professor	Je Hyeong Hong
Submission Date	October 9th, 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_experiment06

Exercise 1

Question A. Create a function that compute 'discrete Fourier transform' and save it as 'dft.m'.

(MATLAB Code) dft.m

```
function [Xk] = dft(xn)
% Compute DFT
% Xk = DFT coeff. array over 0 <= k <= N-1
% xn = N-point finite-duration sequence
% N = Length of DFT

N = length(xn);
n = [0:1:N-1];          % row vector for n
k = [0:1:N-1];          % row vector for k
WN = exp(-j*2*pi/N);     % Wn factor
nk = n'*k;              % creates a N by N matrix of nk values
WNnk = WN.^nk;          % DFT matrix
Xk = xn * WNnk;         % row vector for DFT coefficients
```

Question B. Create a function that compute 'inverse discrete Fourier transform' and save it as 'idft.m'.

(MATLAB Code) idft.m

```
function [xn] = idft(Xk)
% Comupte iDFT
% xn = N-point sequence over 0 <= k <= N-1
% Xk = DFT coeff. array over 0 <= k <= N-1
% N = Length of DFT

N = length(Xk);
n = [0:1:N-1];          % row vector for n
k = [0:1:N-1];          % row vector for k
```

```
WN = exp(-j*2*pi/N);     % Wn factor
nk = n'*k;
WNnk = WN.^(-nk);        % DFT matrix
xn = abs(Xk*WNnk)/N;
```

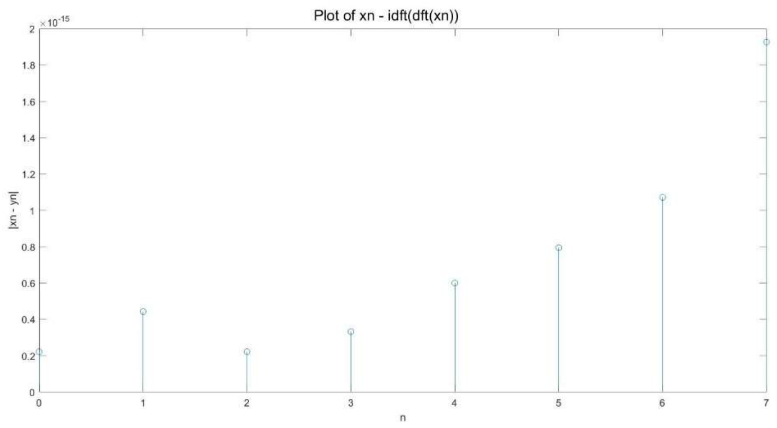
Question C. Generate the sequence that '$x[n] = \{1,1,1,1,0,0,0,0\}$'.

Calculate $y = \mathrm{idft}(\mathrm{dft}(x))$, and check x and y are equal by using $\mathrm{abs}(x - y)$.

(MATLAB Code) lab6_exercise1_c.m

```
xn = [1 1 1 1 0 0 0 0];
yn = idft(dft(xn));

xn_minus_yn = xn - yn;

stem(0:length(xn)-1, abs(xn-yn))
xlabel("n"); ylabel("|xn - yn|");
hold on
sgtitle("Plot of xn - idft(dft(xn))")
```

(Result)



The unit of the y-axis is "$\times 10^{-15}$", which represents that |xn – yn| is substantially zero. The error is presumed to have occurred because N is finite and there is tiny error when computer computes with data associated with float type, which can be ignored in reality.

Exercise 2

Question A. Generate and plot (using stem function) the sequence

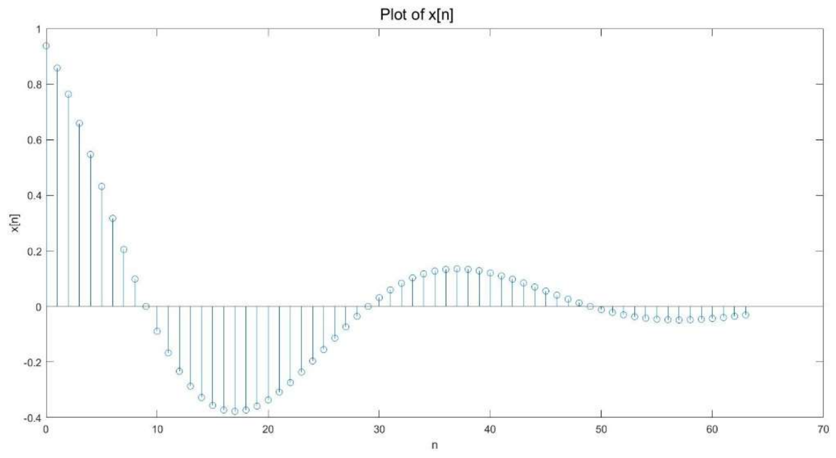
$$x[n] = (0.95)^n \cos\left(\frac{\pi}{20}n\right)$$

for $0 \leq n \leq 63$. Note that MATLAB's indexing for the first element starts with 1 and not 0, so you will have to adjust for this in your plot.

(MATLAB Code) lab6_exercise2_a.m

```
N = 64;

for n = 1:N
    x(n) = ((0.95)^n) * cos(n * pi/20);
end

stem(0:N-1, x);
xlabel("n"); ylabel("x[n]");
sgtitle("Plot of x[n]")
```

(Result)

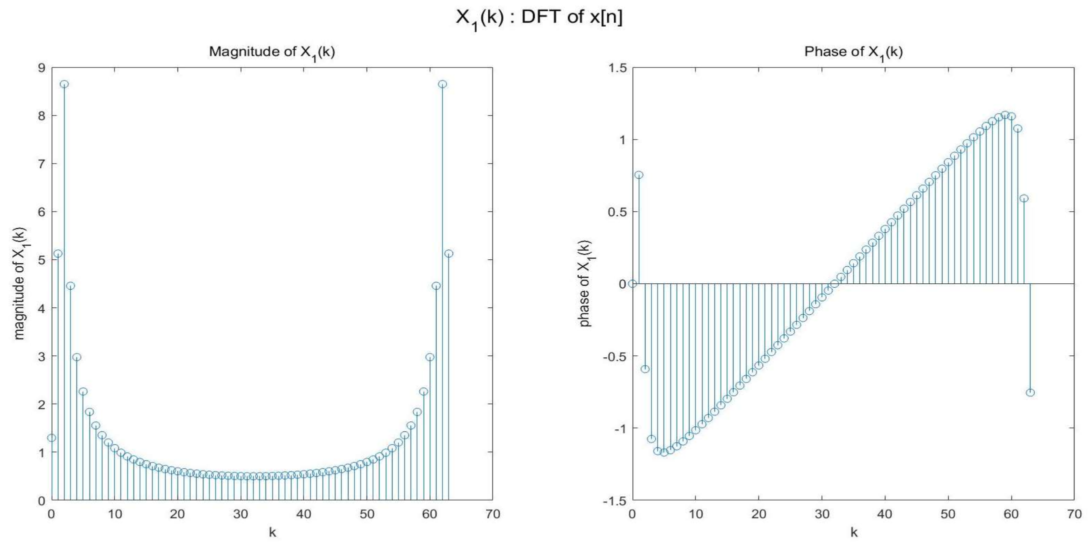


Question B. Calculate $X_1[k]$ which is the DFT for the sequence $x[n]$ and plot (using stem function) its magnitude and its angle.

(MATLAB Code) lab6_exercise2_b.m

```
N = 64;

for n = 1:N
    x(n) = ((0.95)^n) * cos(n * pi/20);
end

x_dft = dft(x);
subplot(1, 2, 1);
stem(0:N-1, abs(x_dft));
xlabel("k"); ylabel("magnitude of X_1(k)");
title("Magnitude of X_1(k)");

subplot(1, 2, 2);
stem(0:N-1, angle(x_dft));
xlabel("k"); ylabel("phase of X_1(k)");
title("Phase of X_1(k)");

sgtitle("X_1(k) : DFT of x[n]");
```

(Result)



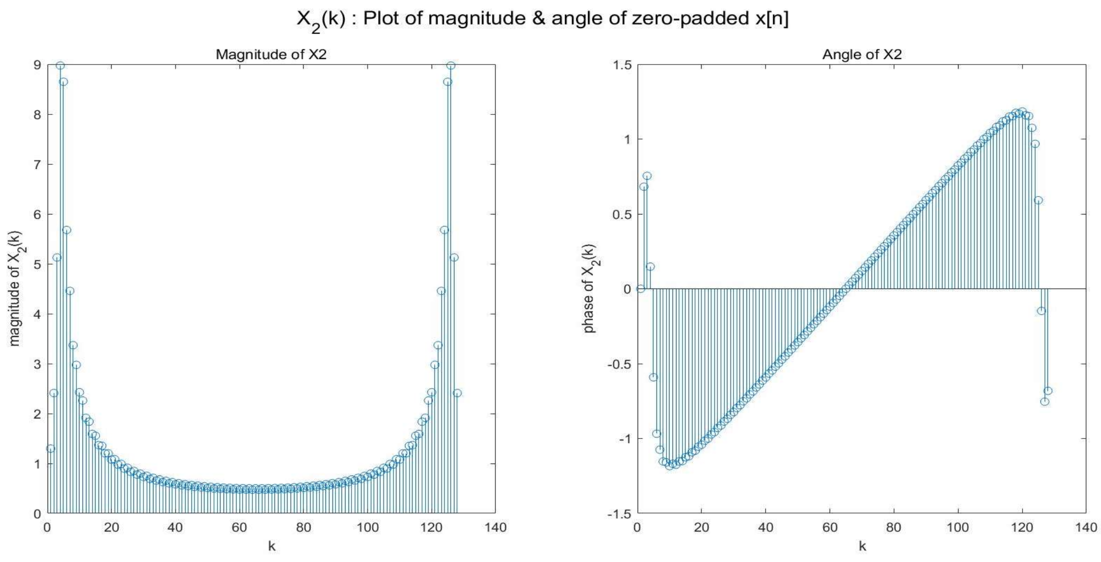
Question C. Calculate $X_2[k]$ which is the 128-point DFT of the 128-point sequence obtained by appending $x[n]$ in 'problem Q2-a' with 64 zeros and plot (using stem function) its magnitude and its angle.

And, compare it to the result of problem Q2-b, and explain the difference theoretically by using the definition of DFT.

(MATLAB Code) lab6_exercise2_c.m

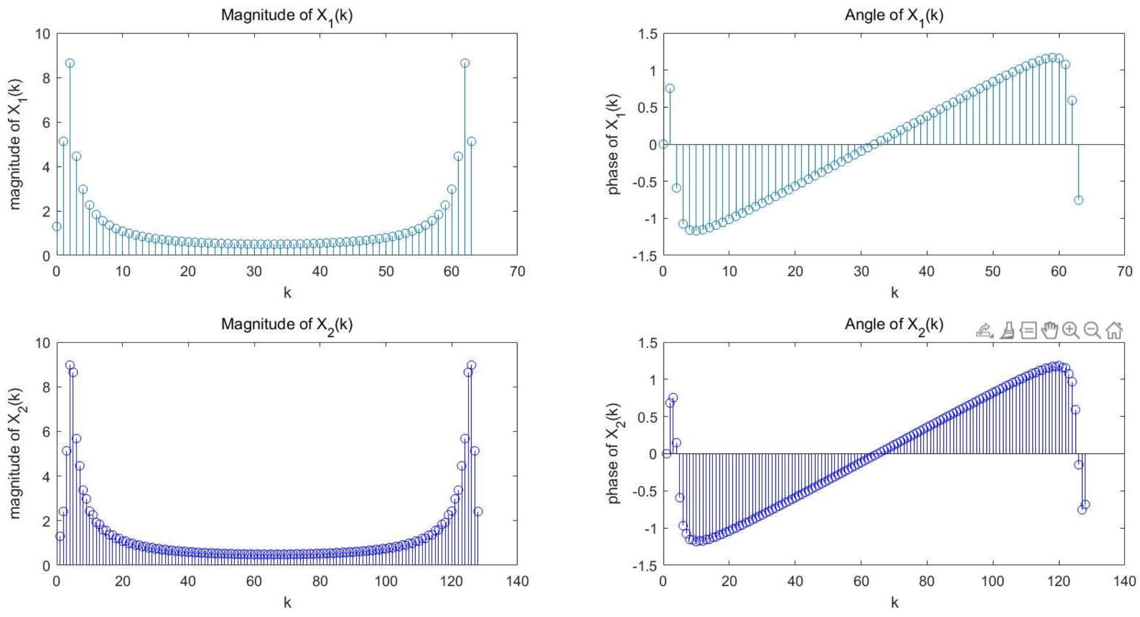
```
N = 64;

for n = 1:N
    x(n) = ((0.95)^n) * cos(n * pi/20);
end

for n = 1:(2*N)
    if n <= N
        x2(n) = x(n);
    else
        x2(n) = 0;  % zero-padding
    end
end
stem(0:2*N-1, x2);

x2_dft = dft(x2);
mag = abs(x2_dft); ang = angle(x2_dft);

subplot(1, 2, 1);
stem(1:2*N, mag);
xlabel("k"); ylabel("magnitude of X_2(k)"); title("Magnitude
of X_2(k)");

subplot(1, 2, 2);
stem(1:2*N, ang);
xlabel("k"); ylabel("phase of X_2(k)"); title("Angle of
X_2(k)");

sgtitle("X_2(k) : Plot of magnitude & angle of zero-padded
x[n]");
```

(Result)



The following plot represents the comparation with Q2-b.

(The MATLAB code is in lab6_exercise2_c2.m file.)



The difference between Q2-c and Q2-b is that the k of Q2-c has 2-times longer range compared to that of Q2-b. However, the total shape of magnitude and phase is drawn equally.

The definition of DFT(Discrete Fourier Transform) is:

$$X[k] = \frac{1}{N}\sum_{k=0}^{N-1} x[n]e^{-j\frac{2\pi k}{N}n}$$

The upper mathematical definition assumes the case of N-point DFT on N-length finite signal. Now let us consider adding zeros of length M at the right side of $X[k]$.

Then, the terms of $x[n]$ become zero if $N \leq n < N + M$.

$$x[N] = x[N+1] = \cdots = X[N+M-1] = 0$$

※ Note that the index in MATLAB starts from 1, not 0.

After inserting the M zeros,

$$X[k] = \frac{1}{N+M}\sum_{k=0}^{N-1} x[n]e^{-j\frac{2\pi k}{N+M}n}$$

The magnitude and phase make no change even if some zeros are padded, and it can be clarified when considering the definition. It means zero padding creates more samples between finite frequency range, for example, [0, 2π]. In other words, the distance between Ω_k and Ω_{k+1} decreases.

However, it can't contribute to more high resolution. Further, L-point DFT of a N-length sequence is said to be a Lagrange Polynomial Interpolation of the N-point DFT samples. Hence, there is not any spectral information increment. It is widely used to make the DFT smoother.

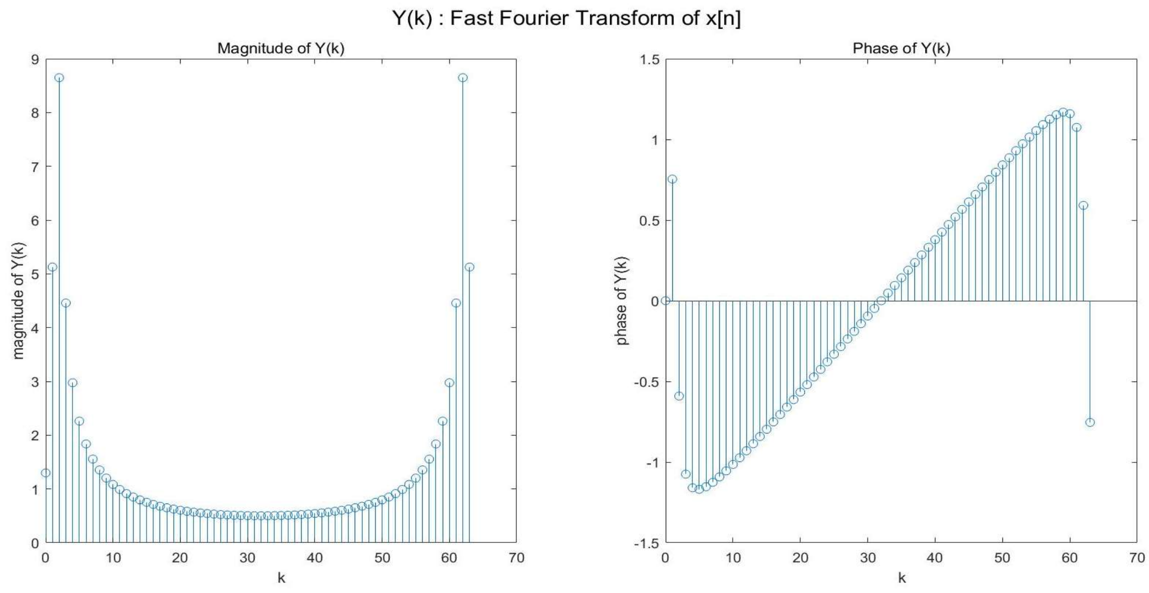
Question D. Calculate $Y[k]$ which is the FFT for the sequence $x[n]$ by using 'fft' function and plot (using stem function) its magnitude.

And compare it to the result of 'problem Q2-b', check that the two are very similar.

(MATLAB Code) lab6_exercise2_d.m

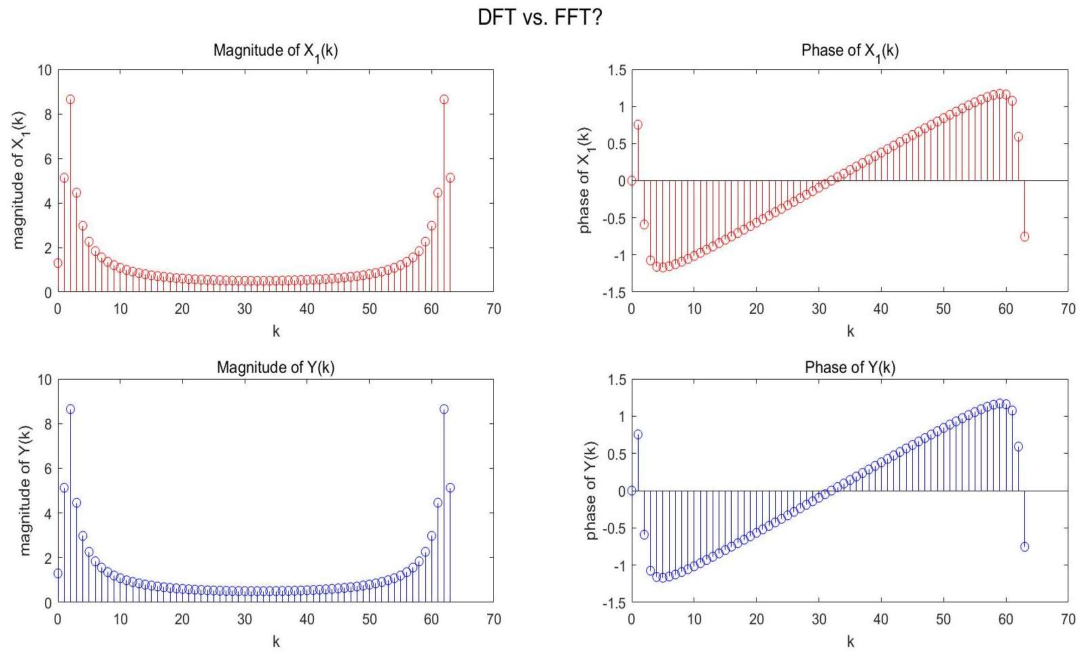
```
N = 64;

for n = 1:N
    x(n) = ((0.95)^n) * cos(n * pi/20);
end

y = fft(x); % Fast Fourier Transform

subplot(1, 2, 1);
stem(0:N-1, abs(y));
xlabel("k"); ylabel("magnitude of Y(k)");
title("Magnitude of Y(k)");

subplot(1, 2, 2);
stem(0:N-1, angle(y));
xlabel("k"); ylabel("phase of Y(k)");
title("Phase of Y(k)");

sgtitle("Y(k) : Fast Fourier Transform of x[n]");
```

(Result)



The following plot represents the comparation with Q2-b.

(The MATLAB code is in lab6_exercise2_d2.m file.)



References

[1] Fredrik Lindsten. *A remark on zero-padding for increased frequency resolution*. Lund, Sweden, 2010. https://www.control.isy.liu.se/en/student/tsrt78/zeropadding.pdf