DSP2 Week 6 experiment Report

2019023436 Kim, Hyunsoo

EXERCISE 1

a) (Source Code)

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

N = length(xn);
n = [0:1:N-1];
k = [0:1:N-1];
WN = exp(-j*2*pi/N);
nk = n'*k;
WNnk = WN.^nk;
Xk = xn*WNnk;
end
```

Above code is a function of DFT which implemented $X[k] = \sum_{n=0}^{N-1} x[n]\exp\left(-j\frac{2\pi k}{N}n\right)$. It receives a row vector xn and returns a row vector Xk.

b) (Source Code)

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

N = length(Xk);
WN = exp(j*2*pi/N);
n = [0:1:N-1];
k = [0:1:N-1];
nk = n'*k;
WNnk = WN.^nk;
xn = Xk*WNnk/N;
end
```

Above code is a function of IDFT which implemented $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\exp\left(j\frac{2\pi k}{N}n\right)$. It receives a row vector Xk and returns a row vector xn.

c) (Source Code)

```
x = zeros(8,1);
x(1:4) = 1;
x = x'

y = idft(dft(x))
abs(x-y)
```

zeros returns a column vector so I transposed it and inserted to the function dft.

I calculated the difference of the 'IDFT result of DFT of *x* (=y)' and *x*. Below is the result.

(Result)

```
x = 1×8
     1     1     1     1     0     0     0     0

y = 1×8 complex
   1.0000 - 0.0000i   1.0000 - 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i  -0.0000 + 0.0000i  -0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i

ans = 1×8
10^-14 ×
    0.0355    0.0449    0.0242    0.0434    0.0600    0.0795    0.1072    0.1927
```

y looks exactly the same as x but the difference between x and y are not 0s. It seems like an error of computer. But they are very small numbers (around e-14) so we can ignore them.

Exercise 2

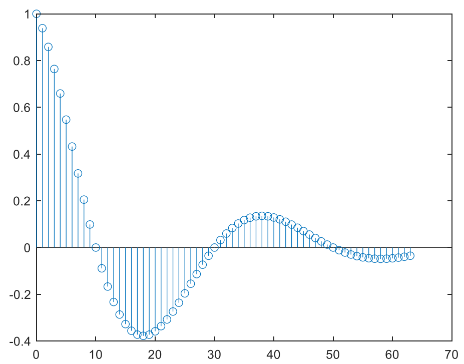
a) (Source Code)

```
n = 0:63
x = (0.95 .^ n) .* cos(pi/20*n)

stem(n, x)
```

x is the implementation of $x[n] = (0.95)^n \cos\left(\frac{\pi}{20}n\right)$ and is a row vector.

(Result)



b) (Source Code)

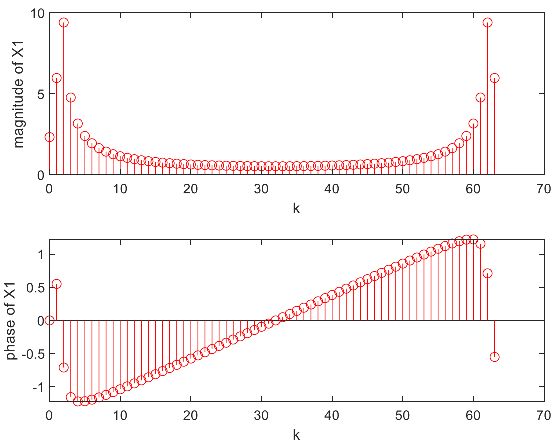
```
N = length(n);
k = 0:N-1;
X1 = dft(x);
magX1 = abs(X1);
angX1 = angle(X1);

subplot(2,1,1);
stem(k, magX1, 'r')
xlabel('k');
ylabel('magnitude of X1')

subplot(2,1,2);
stem(k, angX1, 'r')
xlabel('k');
ylabel('phase of X1')
```

X1 is the dft of *x*. I plotted the magnitude of *X1* and phase of *X1* on each subplot.

(Result)



c) (Source Code)

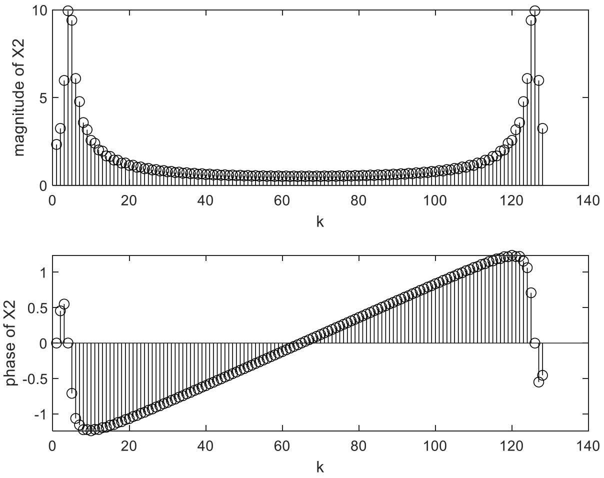
```
n2 = 0:127
x2 = [x, zeros(1,64)]
X2 = dft(x2);
magX2 = abs(X2);
angX2 = angle(X2);

subplot(2,1,1);
stem(magX2, 'k');
xlabel('k');
ylabel('magnitude of X2')

subplot(2,1,2);
stem(angX2, 'k');
xlabel('k');
ylabel('phase of X2');
```

x2 is a row vector whose 64 elements are *x* and 64 elements are 0s. *X2* is dft of *x2*. I plotted the magnitude of *X2* and phase of *X2* on each subplot.

(Result)



DFT maps N-point time-domain signal x[0]~x[N] into a discrete periodic sequence X[k], which has periodicity of N and N times sampled frequency. The length of x[n] of b) is 64 (=N) and the length of x[n] of this c) is 128 (=N). Thus, the result graphs look very similar to b) but are more sampled (128).

d) (Source Code)

```
subplot(2,1,1)
y = fft(x)
stem(abs(y))
xlabel('k')
ylabel('magnitude of Y')
```

I don't know what the fft is but I can see the result graph is very similar to that of b).

(Result)