# DSP2 Week 7 experiment Report

2019023436 Kim, Hyunsoo

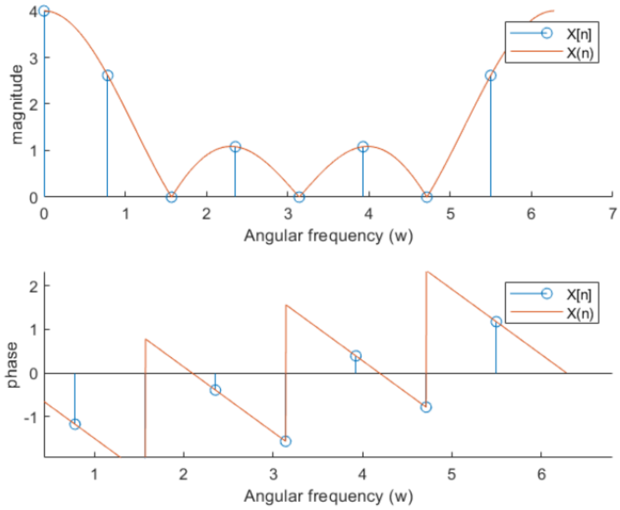
EXERCISE 1

(Source Code)

```
n = 0:7;
N = length(n);
k = 0:N-1;
x = [1 1 1 1 0 0 0 0];
X=dft(x);
magX = abs(X);
angX = angle(X);
xw = ((2*pi)/N)*k;
yw = ((2*pi)/2000)*(0:1999);
Y = 1 + exp(-j.*yw) + exp(-j*2.*yw) + exp(-j*3.*yw);
magY = abs(Y);
angY = angle(Y);

subplot(2,1,1);
hold on;
stem(xw, magX);
plot(yw, magY);
legend('X[n]', 'X(n)');
ylabel('magnitude');
xlabel('Angular frequency (w)');
hold off;

subplot(2,1,2);
hold on;
stem(xw, angX);
plot(yw, angY);
legend('X[n]', 'X(n)');
ylabel('phase');
xlabel('Angular frequency (w)');
hold off;
```

X implements the 8-point DFT of x[n]. Y implements the DTFT of x[n]. xw and yw looks very similar, but xw is N-point sampled vector whereas yw is 2000-point which makes the yw looks like a continuous vector.

(Result)




As we can see in the above graph, DFT of x[n] is included in the DTFT of x[n].

Exercise 2

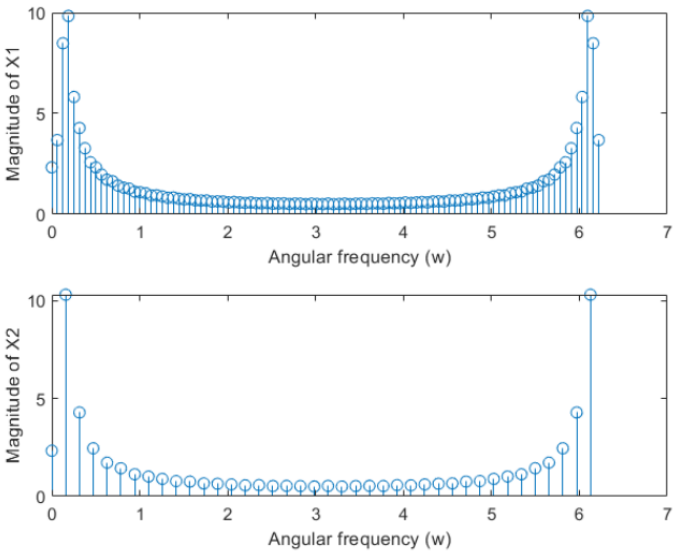
a) (Source Code)

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

M1 = 100;
M2 = 40;

k1 = [0:M1-1];
k2 = [0:M2-1];

DT_samp = @(x, k, n) x*exp(-j*2*pi/length(k)).^(n'*k);

X1 = DT_samp(x, k1, n);
X2 = DT_samp(x, k2, n);

magX1 = abs(X1);
magX2 = abs(X2);

w1 = ((2*pi)/M1)*k1;
w2 = ((2*pi)/M2)*k2;

subplot(2,1,1);
stem(w1, magX1)
xlabel("Angular frequency (w)");
ylabel("Magnitude of X1");

subplot(2,1,2);
stem(w2, magX2);
xlabel("Angular frequency (w)");
ylabel("Magnitude of X2");
```

$x$ is the implementation of $x[n] = (0.95)^n \cos\left(\frac{\pi}{20}n\right)$ for 0<= n <= 63. We plotted

2 kinds of DFT of x[n]; One is 100-point DFT of x[n], the other is 40-point DFT of x[n] where the length of x[n] is 64.

(Result)

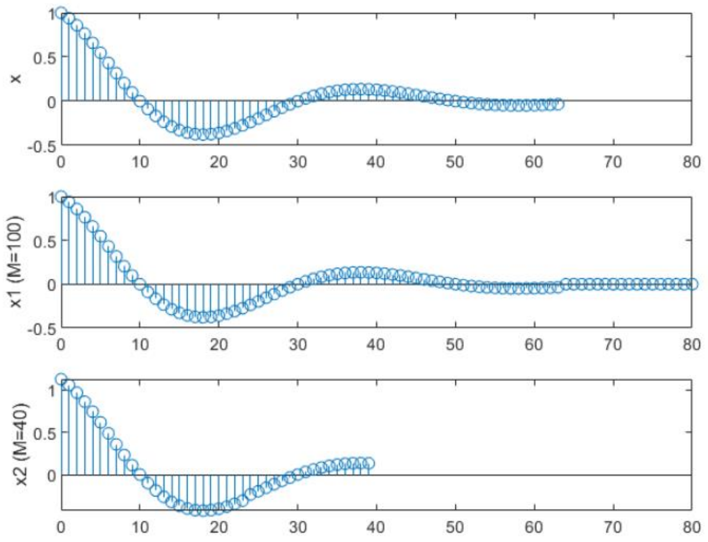



b) (Source Code)

```
x1 = idft(X1)
x2 = idft(X2)

subplot(3,1,1);
stem(n, real(x));
ylabel("x");
xlim([0 80]);
subplot(3,1,2);
stem(k1, real(x1));
ylabel("x1 (M=100)");
xlim([0 80]);
subplot(3,1,3);
stem(k2, real(x2));
ylabel("x2 (M=40)");
xlim([0 80]);
```

I calculated the inverse dft of X1 and X2 using the idft function which I made on the last lecture.

(Result)



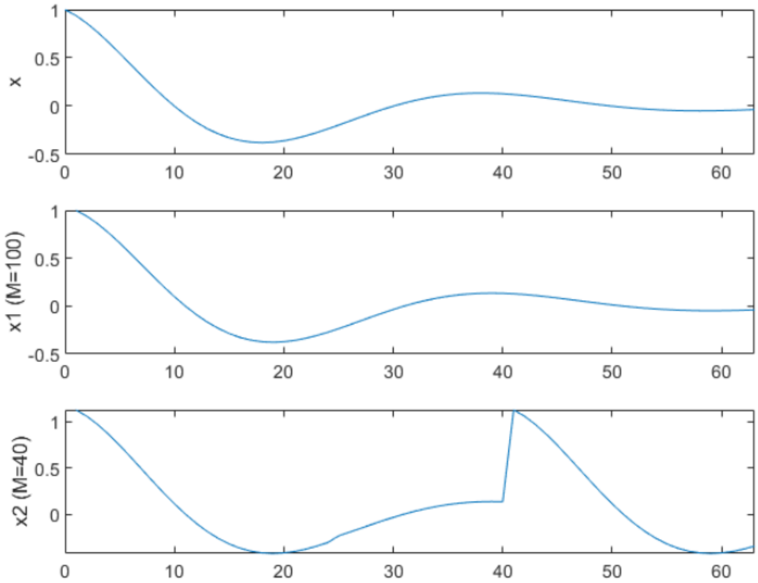

First graph is the original signal x[n]. Second graph is a recovered signal where M is 100 and it shows zero padding on 64<=n<=80. May be there is more zero padding until n=100. Third graph is a recovered signal where M is 40, which is way less than the length of the original signal x. It perfectly recovered till the n is 40.

c) (Source Code)

```
x1_ = idfs(X1)
x2_ = idfs(X2)

subplot(3,1,1);
plot(n, x);
ylabel("x");
xlim([0 63]);
subplot(3,1,2);
plot(real(x1_));
ylabel("x1 (M=100)");
xlim([0 63]);
subplot(3,1,3);
plot(real(x2_));
ylabel("x2 (M=40)");
xlim([0 63]);
```

x1_ and x2_ is a inverse discrete Fourier series of X1 and X2 respectively.

(Result)




x1 is exactly recovered from X1. x2 is exactly recovered until 40 but it went wrong after 40. The temporal aliasing has occurred.