Digital Signal Processing Ⅱ

9^{th} EXPERIMENT

Report

(WEEK10 report of DSP2 course)

Subject	Digital Signal Processing Ⅱ
Professor	Je Hyeong Hong
Submission Date	November 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_experiment09

Exercise 1

exercise1-a)

Generate a signal x that is $\sin(2\pi t)$.

Generate a signal x_1 that is $\sin(2\pi t)$ with standard Gaussian noise multiplied by 0.15.

Generate Low-pass filter h that is $\frac{1}{4}\delta[n] + \frac{1}{2}\delta[n-1] + \frac{1}{4}\delta[n-2]$.

(MATLAB Code) * Refer to lab10_exercise1.m

```
time_interval = 0.01;
t = 0:time_interval:1;

% Exercise1-(a)
subplot(2, 3, 1);
x = sin(4*pi*t);
plot(t, x); title("original signal");

subplot(2, 3, 2);
x1 = x + 0.15 * randn(1, length(t));
plot(t, x1); title("corrupted signal");

subplot(2, 3, 3);
h = [1/4 1/2 1/4];
```

(Results)

(Skip)

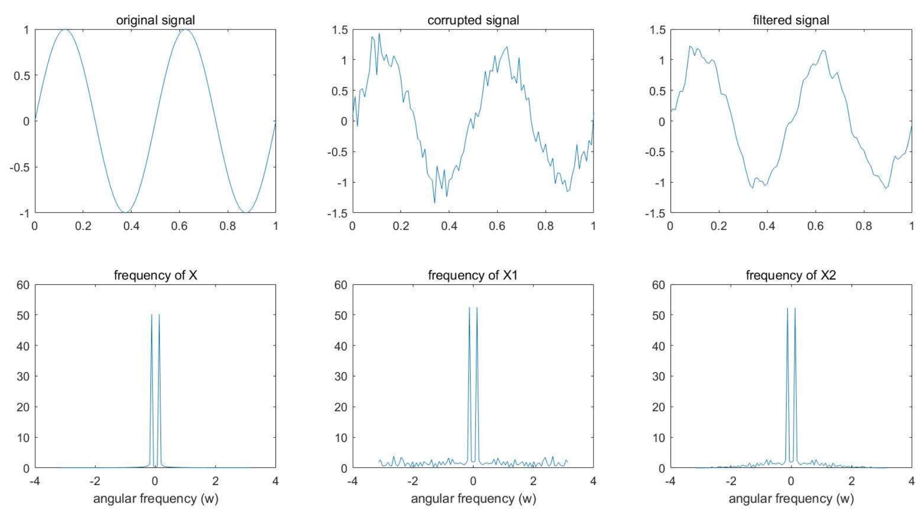
exercise1-b)

Calculate $x_2 = x_1 * h$, its magnitude of frequency (use ‘fft’ and ‘fftshift’ function) and plot the signal and magnitude of frequency of x, x_1, x_2.

(MATLAB Code) * Refer to lab10_exercise1.m

```
% Exercise1-(b)
x2 = conv(x1, h);
x2(1) = []; x2(end)=[];
plot(t, x2); title("filtered signal");

interval_ = -pi:0.01*2*pi:+pi;
subplot(2, 3, 4);
X = fft(x); X = fftshift(X); magX = abs(X); plot(interval_,
magX);
title("frequency of X"); xlabel("angular frequency (w)");

subplot(2, 3, 5);
X1 = fft(x1); X1 = fftshift(X1); magX1 = abs(X1);
plot(interval_, magX1);
title("frequency of X1"); xlabel("angular frequency (w)");

subplot(2, 3, 6);

X2 = fft(x2); X2 = fftshift(X2); magX2 = abs(X2);
plot(interval_, magX2);
title("frequency of X2"); xlabel("angular frequency (w)");
```

(Results)



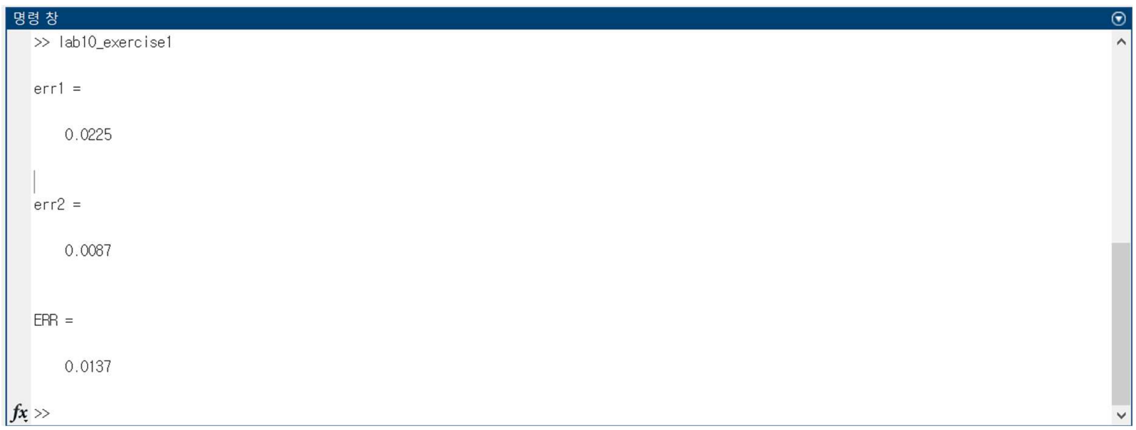
exercise1-c)

Calculate the mse of x with x_1, x_2 respectively, and calculate the difference.

(MATLAB Code) * Refer to lab10_exercise1.m

```
% Exercise1-(c)
err1 = mean((x - x1).^2)
err2 = mean((x - x2).^2)
ERR = abs(err1 - err2)
```

(Results)



exercise1-d)

Explain briefly the result of Q1-b, c.

After adding some noise with the gaussian distribution, the signal became corrupted, having small frequencies in overall frequency domain, including higher frequency region. To the listener, it will be an unwanted component. The error between the original signal and the corrupted signal is represented by 'err1'.

We pass the corrupted signal into a low-pass filter. This is implemented in 'h', and the corrupted signal is passed into the filter by applying convolution, $x1 * h$. Now, the new signal generated by convolution operation have little high-frequency components, in other words, we eliminated the high-frequency noise. Thus, the new signal lost its noise that we made, as a result, it will be a more precise reconstruction version of x, compared to the corrupted signal. Its error is represented in 'err2'.

(ALL code) lab10_exercise1.m

```
time_interval = 0.01;
t = 0:time_interval:1;

% Exercise1-(a)
subplot(2, 3, 1);
x = sin(4*pi*t);
plot(t, x); title("original signal");

subplot(2, 3, 2);
x1 = x + 0.15 * randn(1, length(t));
plot(t, x1); title("corrupted signal");

subplot(2, 3, 3);
h = [1/4 1/2 1/4];

% Exercise1-(b)
x2 = conv(x1, h);
x2(1) = []; x2(end)=[];
plot(t, x2); title("filtered signal");

interval_ = -pi:0.01*2*pi:+pi;
subplot(2, 3, 4);
X = fft(x); X = fftshift(X); magX = abs(X); plot(interval_,
magX);
title("frequency of X"); xlabel("angular frequency (w)");

subplot(2, 3, 5);
X1 = fft(x1); X1 = fftshift(X1); magX1 = abs(X1);
plot(interval_, magX1);
title("frequency of X1"); xlabel("angular frequency (w)");

subplot(2, 3, 6);

X2 = fft(x2); X2 = fftshift(X2); magX2 = abs(X2);
plot(interval_, magX2);
title("frequency of X2"); xlabel("angular frequency (w)");

% Exercise1-(c)
err1 = mean((x-x1).^2)
err2 = mean((x-x2).^2)
ERR = abs(err1-err2)
```

Exercise 2

exercise2-a)

Load the audio file 'dsp2_experiment10.wav' and convert this signal by low pass filtering and high pass filtering.

(Hint: use 'audioread' function)

(MATLAB Code) * Refer to lab10_exercise2.m

```
[y, Fs] = audioread(".\dsp2_experiment10.wav");

% N = length(y);

%for i = 1:N-2
%    %yh(i) = (y(i+1)-y(i))/2;
%    yl(i) = (y(i+2) + 2*y(i+1) + y(i))/4;
%end

% sound(y, Fs);

h = [-1/2 1/2];
pad = zeros(1, 30);
h = [h,pad]; H = fft(h); H = fftshift(H);
magH = abs(H); plot(magH);
lpf = [1/2 1/2]; hpf = [-1/2 1/2];
audio_low = conv(y, lpf);
audio_high = conv(y, hpf);
```

(Results)

(Skip)

exercise2-b)

By use 'sound' function and compare two of these by listening these sound.

(MATLAB Code) * Refer to lab10_exercise2.m

```
% original sound
sound(y, Fs)

% low-pass filtered sound
% sound(audio_low, Fs);

% high-pass filtered sound
% sound(audio_high, Fs);
```

(Results)

Except for one of the three sound sections, the other two lines are treated with comment and then listen to the sound one by one. The original sound included a boy's voice and beep sound.

In the low-pass filtered sound, the beep sounded weaker than before, and the male voice became stronger compared to the original version. In contrast, hearing the high-pass filtered sound, the beep sound became stronger than before, and the male voice became weaker.

The beep sound is located in high-frequency region, while the boy's voice is in the low-pass region. They can be distinguished by low-pass filter and high-pass filter.

exercise2-c)

Calculate and plot the frequency response of original signal, low pass filtered signal and high pass filtered signal respectively.

(MATLAB Code) * Refer to lab10_exercise2.m

```
index = -pi:2*pi/(length(y)-1):pi;
subplot(3, 1, 1); plot(-pi:2*pi/(length(y)-1):pi,
abs(fftshift(fft(y))));
title("original sound"); xlabel("angular frequency (w)");
subplot(3, 1, 2); plot(-pi:2*pi/(length(y)):pi,
abs(fftshift(fft(audio_low))));
title("low-pass filtered sound"); xlabel("angular frequency
(w)");
subplot(3, 1, 3); plot(-pi:2*pi/(length(y)):pi,
abs(fftshift(fft(audio_high))));
title("high-pass filtered sound"); xlabel("angular frequency
(w)");
```

(Results)