Digital Signal Processing Ⅱ

11^{th} EXPERIMENT

Report

(WEEK12 report of DSP2 course)

Subject	Digital Signal Processing Ⅱ
Professor	Je Hyeong Hong
Submission Date	November 25th, 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_experiment11

Exercise 1

exercise1-a)

A filter is described by the following difference equation:

$$y[n] = 2x[n] + 2.76x[n-1] + 2.622x[n-2] + 2.6740x[n-3] + 1.8x[n-4]$$

determine its lattice form.

(MATLAB Code) lab12_exercise1_a.m

```
b = [2 2.76 2.622 2.6740, 1.8];
k = tf2latc(b)
```

(Results)

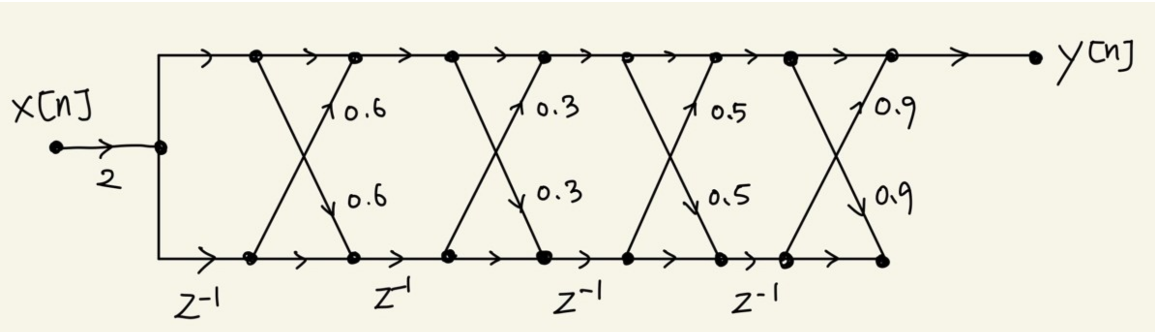
exercise1-b)

Draw the block diagram of direct form and lattice form of Q2-a's filter by using the signal-flow graph (SGF) with your own hands.

(MATLAB Code)

(SKIP)

(Results)



exercise1-c)

Generate an impulse response (the length of sequence = 4) and filter it by the direct form ('filter') and the direct form ('latcfilt') respectively. And check the result is same.

(MATLAB Code) lab12_exercise1_c.m

```
b = [2 2.76 2.622 2.6740, 1.8];
k = tf2latc(b);

delta = [1 0 0 0];

output_dir = filter(b, 1, delta)
output_lat = 2 * latcfilt(k, delta)
```

(Results)

```
명령 창
>> lab12_exercise1_c

output_dir =

    2.0000    2.7600    2.6220    2.6740


output_lat =

    2.0000    2.7600    2.6220    2.6740

fx >>
```

Exercise 2

exercise2-a)

Generate a filter with linear property and plot its 'angular frequency' versus 'phase' plot.

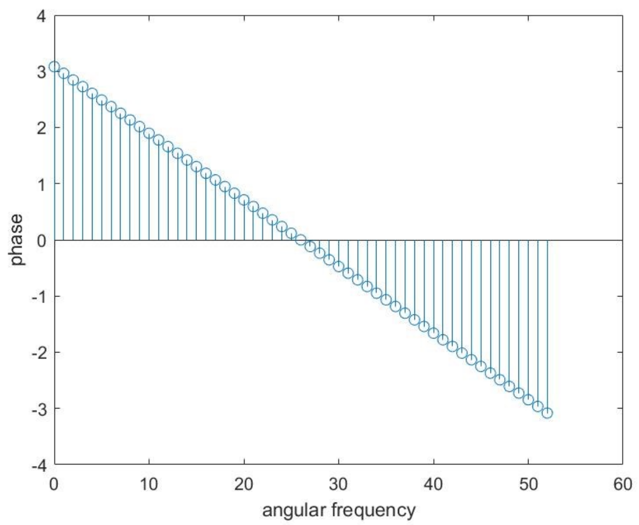
(for example, $y[n] = \frac{1}{N+M+1}\sum_{k=-N}^{M} x[n-k]$)

(MATLAB Code) lab12_exercise2_a.m

```
h = [1/4 1/2 1/4];
h = [h, zeros(1, 50)];
k = [0:length(h)-1];

H = fft(h); H = fftshift(H);
angH = angle(H);
stem(k, angH)   % Linear Phase
xlabel("angular frequency"); ylabel("phase");
```

(Results)



exercise2-b)

Generate a filter with non-linear property and plot its 'angular frequency' versus 'phase' plot.

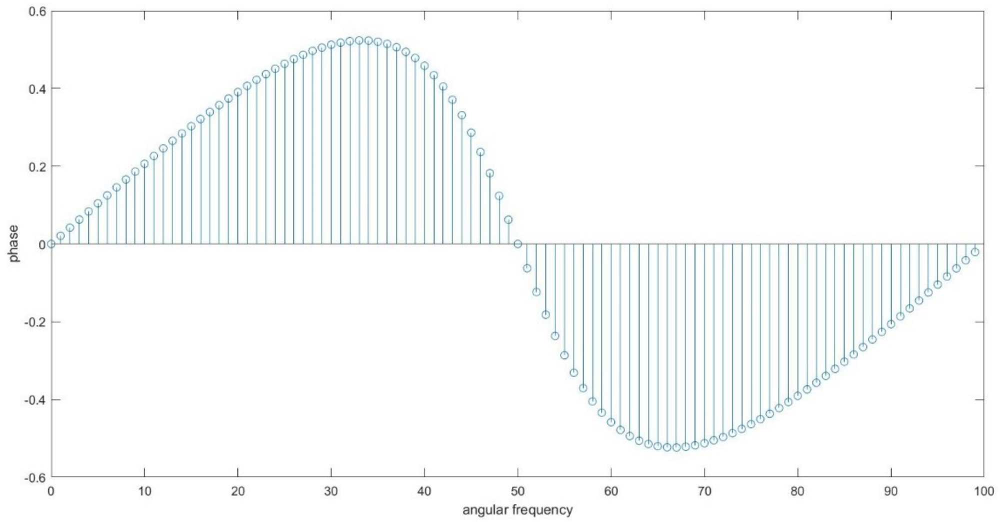
(for example, $y[n] - ay[n-1] = x[n]$)

(MATLAB Code) lab12_exercise2_b.m

```
h = (1/2).^(1:50);
h = [h, zeros(1, 50)];
k = [0:length(h)-1];

H = fft(h); H = fftshift(H);
angH = angle(H);
stem(k, angH)   % NON-Linear Phase
xlabel("angular frequency"); ylabel("phase");
```

(Results)



exercise2-c)

Explain briefly the linear-phase properties based on the results of Q2-a and Q2-b.

(Explanation)[1]

A filter is called a linear phase filter if the phase component of the frequency response is a linear function of frequency. For a continuous-time application, the frequency response of the filter is the Fourier transform of the filter's impulse response, and a linear phase version has the form:

$$H(\omega) = A(\omega)e^{-j\omega\tau}$$

where:

$A(\omega)$ is a real-valued function.

τ is the group delay.

For a discrete-time application, the discrete-time Fourier transform of the linear phase impulse response has the form:

$$H_{2\pi}(\omega) = A(\omega)e^{-\frac{j\omega k}{2}}$$

where:

$A(\omega)$ is a real-valued function with 2π periodicity.

k is an integer, and $\frac{k}{2}$ is the group delay in units of samples.

L.P. is a property of a filter where the phase response of the filter is a linear function of frequency. The result is that all frequency components of the input signal are shifted in time (usually delayed) by the same constant amount (the slope

---

[1] https://en.wikipedia.org/wiki/Linear_phase

of the linear function), which is referred to as the group delay. Consequently, there is no phase distortion due to the time delay of frequencies relative to one another.

For discrete-time signals, perfect linear phase is easily achieved with a finite impulse response (FIR) filter by having coefficients which are symmetric or anti-symmetric.