

# DSP Simulation Project #1

2018142023 조성민

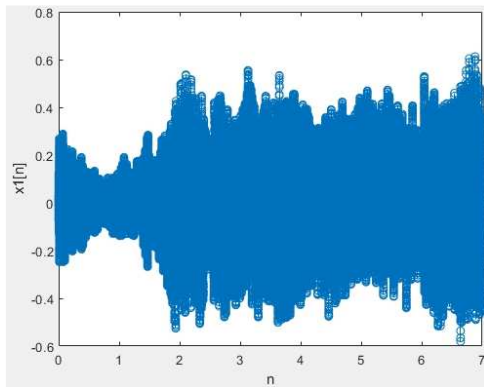
## < Up-sampling, Down-sampling & Filtering process >

주어진 2개의 음성신호는 7 초 길이의 'wav' 파일을 아래의 조건으로 각각 sampling한 신호입니다.

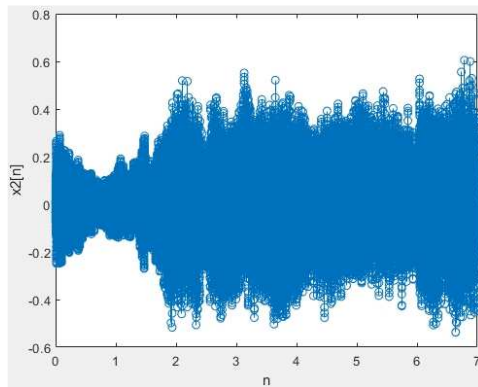
(336,000 point, Sampling rate  $f_{s1} = 48,000\text{Hz}$ ,  $T_{s1} = \frac{1}{48,000}\text{sec}$ ,  $\omega_{s1} = 96,000\pi$ ),

(67,200 point, Sampling rate  $f_{s2} = 9,600\text{Hz}$ ,  $T_{s2} = \frac{1}{9,600}\text{sec}$ ,  $\omega_{s2} = 19,200\pi$ ).

1-1. 각 음성신호 파일 'Sample1.wav'와 'Sample2.wav'의 음성 sample들을  $x_1[n]$ ,  $x_2[n]$ 라고 했을 때,  $x_1[n]$ ,  $x_2[n]$ 의 파형을 그리세요.

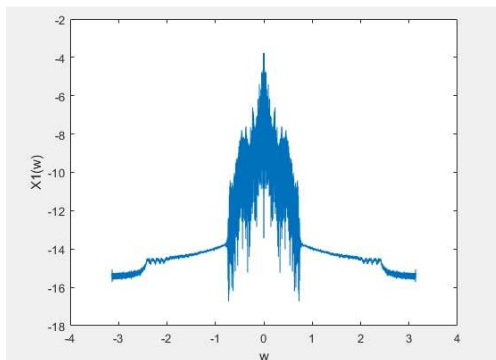


$x_1[n]$

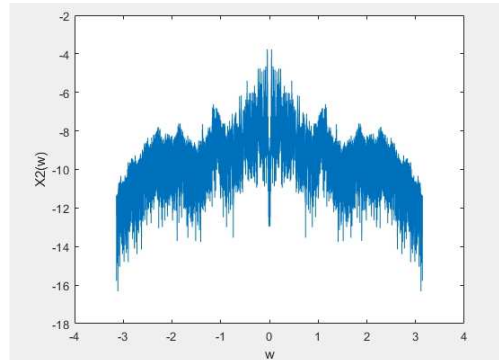


$x_2[n]$

1-2. 문제 1-1에서 다룬  $x_1[n]$ ,  $x_2[n]$ 의 Spectrum을  $X_1(\omega)$ ,  $X_2(\omega)$ 라고 했을 때,  $X_1(\omega)$ ,  $X_2(\omega)$ 의 Magnitude 그래프를 Log scale로 그리세요.



$X_1(w)$



$X_2(w)$

1-3. 음성 sample  $x_1[n], x_2[n]$ 들을 청취해보고 어떤 차이가 있는지 기술하세요.

- $x_1[n]$ 의 경우는 노래의 음이 올라가면서 음량이 커지는 듯한 느낌이 나고,  $x_2[n]$ 의 경우는 전반적으로 음량의 차이가 없는 것으로 느껴진다.

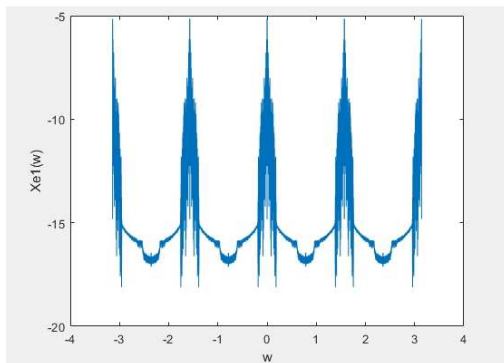
2. Up-sampling 하기 전에, 원래 신호 ( $x_1[n], x_2[n]$ ) 사이에 0을 삽입하는 방법 (Zero-padding)이 있습니다. Zero-padding 한 신호를  $x_e[n]$ 라 하면, 아래와 같은 관계를 갖습니다. ( $L = 4$ )

$$x_e[n] = \begin{cases} x[n/L] & \text{for } n = Lk \\ 0 & \text{otherwise} \end{cases}$$

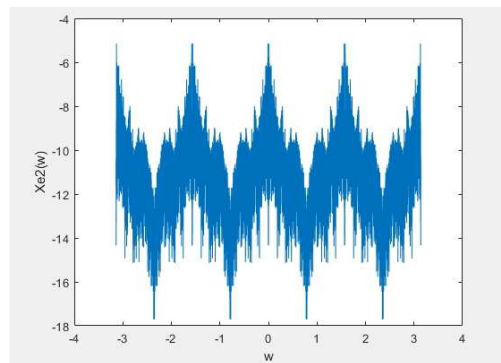
2-1.  $x_1[n], x_2[n]$ 에 대한  $x_{e1}[n], x_{e2}[n]$ 을 청취해보고 원래 신호와 어떤 차이가 있는지 기술하세요.

- $x_{e1}[n]$ 은 원래 신호에 비해 잡음이 더 많아졌고  $x_{e2}[n]$ 은 소리가 깨진다.

2-2.  $x_{e1}[n], x_{e2}[n]$ 의 Spectrum인  $X_{e1}(\omega), X_{e2}(\omega)$ 의 Magnitude 그래프를 Log scale로 그리세요.



$X_{e1}(\omega)$



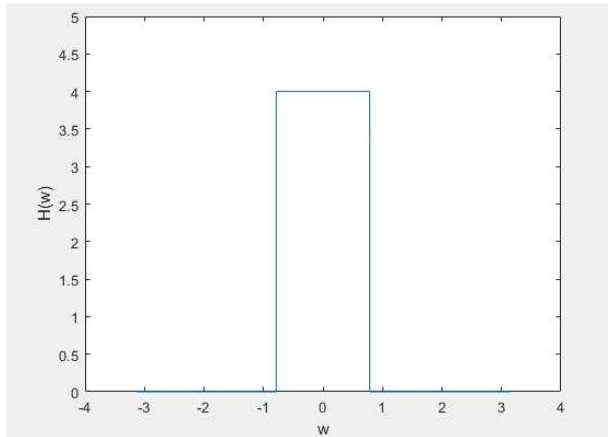
$X_{e2}(\omega)$

2-3. 문제 1-2의  $X_1(\omega), X_2(\omega)$ 의 Magnitude 그래프와 문제 2-2의  $X_{e1}(\omega), X_{e2}(\omega)$ 의 Magnitude 그래프를 비교해보세요.

-각각의 파형이 한 주기( $-\pi \sim \pi$ ) 내에서 4배로 증가한다.

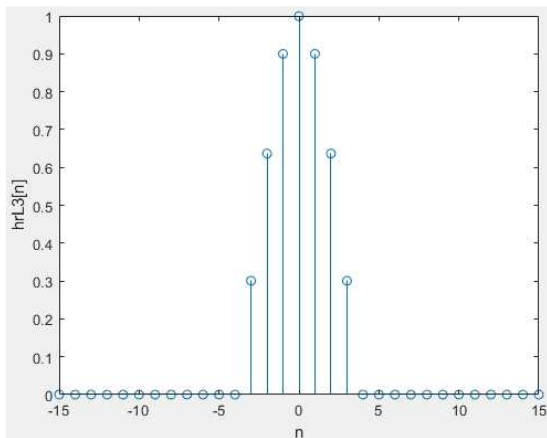
3. Up-sampling을 하기 위해 적절한 Reconstruction Filter  $h_r[n]$ 의 Frequency Response  $H_r(\omega)$ 를 gain과 cutoff frequency에 주의하여 설계합니다. 설계한  $H_r(\omega)$ 를 Inverse Fourier transform을 이용하여 시간축에서 필터를 구현합니다. 그런데 Ideal low-pass filter는  $(-\infty, \infty)$ 의 길이를 필요로 하기에 시간축에서의 완전한 구현은 불가능하므로 Ideal low-pass filter를 각각 **3points, 11points로 rectangular windowing**합니다.

3-1.  $H_r(\omega)$ 의 Magnitude 그래프를 그리세요.

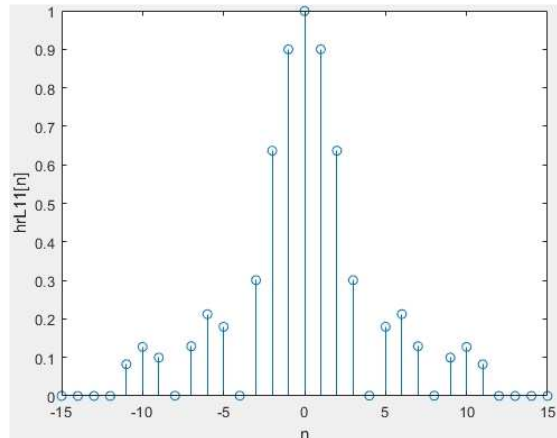


$H_r(\omega)$

3-2. 구현한  $h_r[n]$ 를 각각 rectangular windowing한  $h_{rL3}[n]$ ,  $h_{rL11}[n]$ 의 파형을 그리세요.

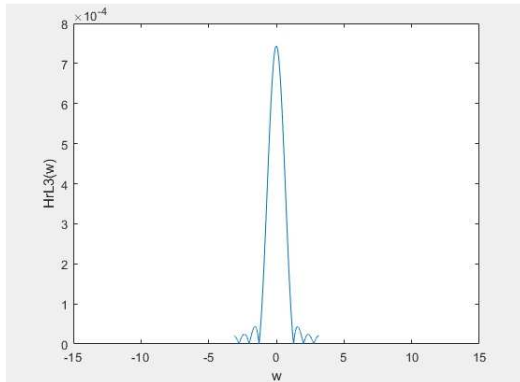


$h_{rL3}[n]$

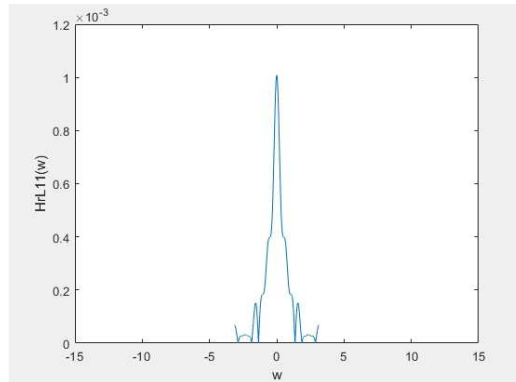


$h_{rL11}[n]$

3-3.  $h_{rL3}[n]$ ,  $h_{rL11}[n]$ 의 Frequency Response  $H_{rL3}(\omega)$ ,  $H_{rL11}(\omega)$ 의 Magnitude를 그리세요.



$H_{rL3}(w)$

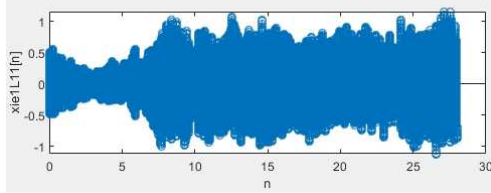
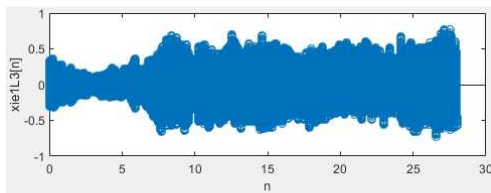


$H_{rL11}(w)$

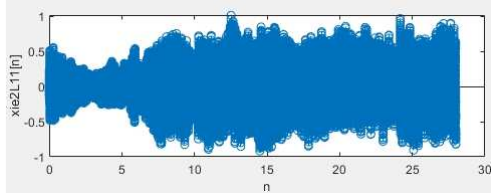
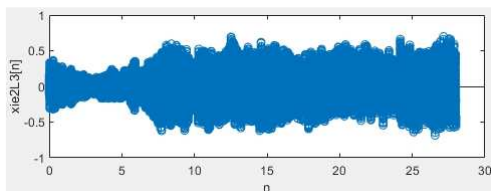
4. Up-sampling된 신호를  $x_i[n]$ 이라 하고, Reconstruction Filter를  $h_r[n]$ 이라고 할 때,  $x_e[n]$ 는 다음과 같은 관계를 갖습니다.

$$x_i[n] = x_e[n] * h_r[n]$$

4-1. Zero-padding된 2개의 신호  $x_{e1}[n]$ ,  $x_{e2}[n]$ 를 위에서 제작한 Reconstruction Filter(3point, 11point)  $h_{rL3}[n]$ ,  $h_{rL11}[n]$ 을 이용하여 time domain에서 filtering한 신호  $x_{ie1L3}[n]$ ,  $x_{ie1L11}[n]$ ,  $x_{ie2L3}[n]$ ,  $x_{ie2L11}[n]$ 의 파형을 그리세요.

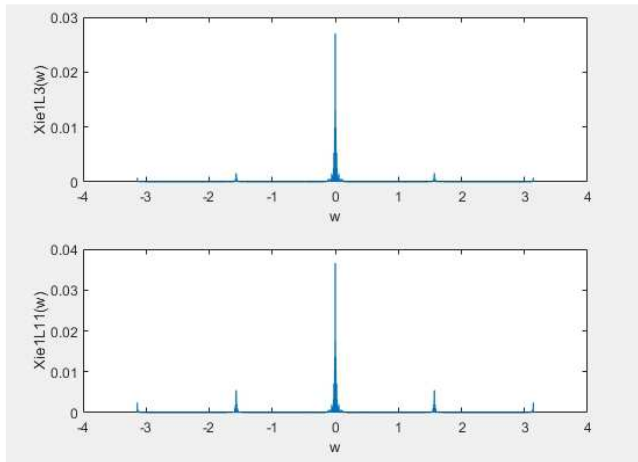


$x_{ie1L3}[n]$  &  $x_{ie1L11}[n]$

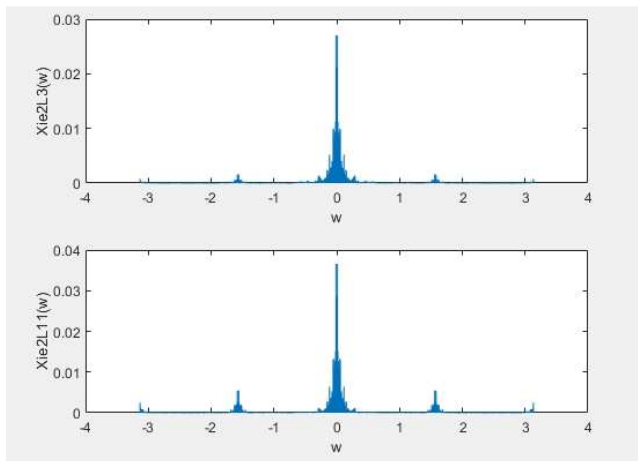


$x_{ie2L3}[n]$  &  $x_{ie2L11}[n]$

**4-2.** 문제 4-1에서 filtering한 신호  $x_i[n]$ 의 Spectrum  $X_i(w)$ 의 magnitude 그래프를 각각 그리세요.



$X_{ie1L3}(w)$  &  $X_{ie1L11}(w)$



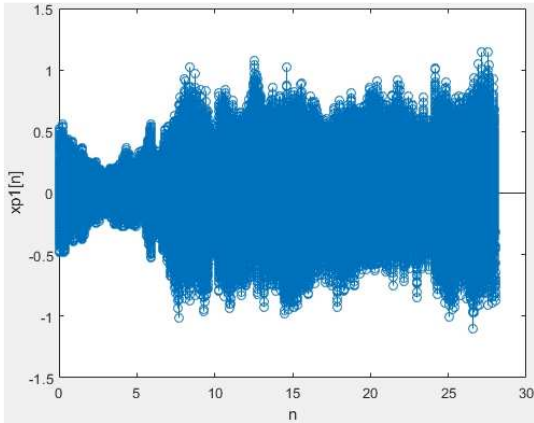
$X_{ie2L3}(w)$  &  $X_{ie2L11}(w)$

**4-3.**  $x_i[n]$ 을 청취해보고 원래 신호  $x_e[n]$ 과 어떤 차이가 있는지 기술하세요.

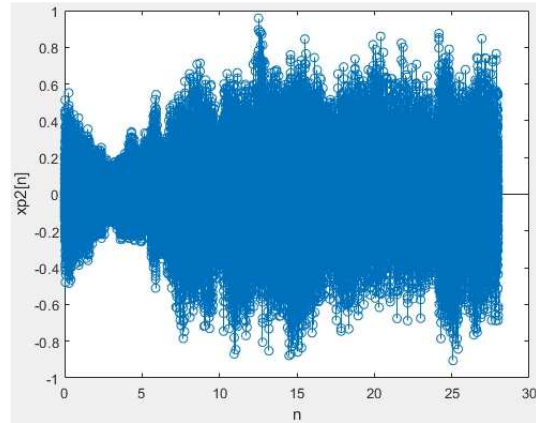
-(11 point 기준)  $x_i[n]$ 의 음량이  $x_e[n]$ 보다 더 크다.

5-1. 문제 4에서 얻은 신호  $x_i[n]$ (11point reconstruction only)에 대해, 아래의 식을 이용하여  $x_p[n]$ 의 파형을 그리세요. (M=5)

$$x_p[n] = \begin{cases} x_i[n] & \text{for } n = Mk \\ 0 & \text{otherwise} \end{cases}$$

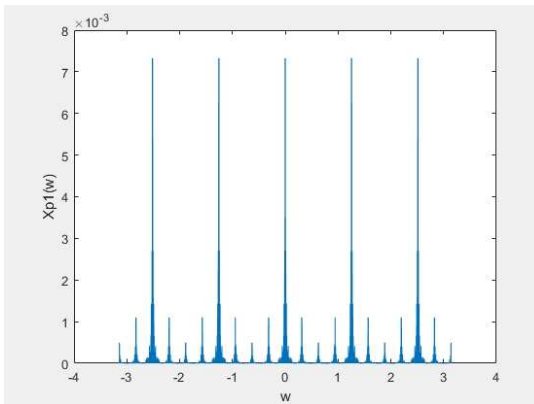


$x_{p1}[n]$

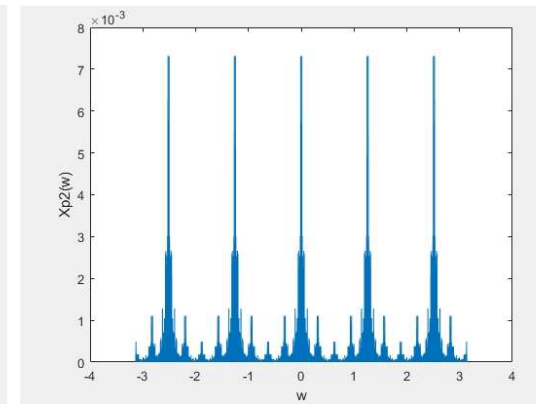


$x_{p2}[n]$

5-2.  $x_p[n]$ 의 Spectrum  $X_p(\omega)$ 의 magnitude 그래프를 그리시오.



$X_{p1}(w)$



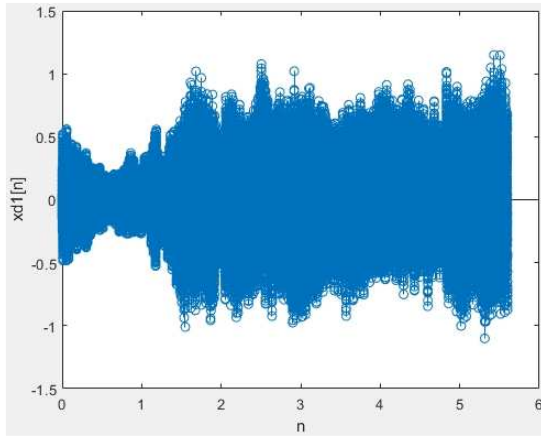
$X_{p2}(w)$

5-3.  $x_p[n]$ 을 청취해보고 원래 신호  $x_i[n]$ 와 어떤 차이가 있는지 기술하세요.

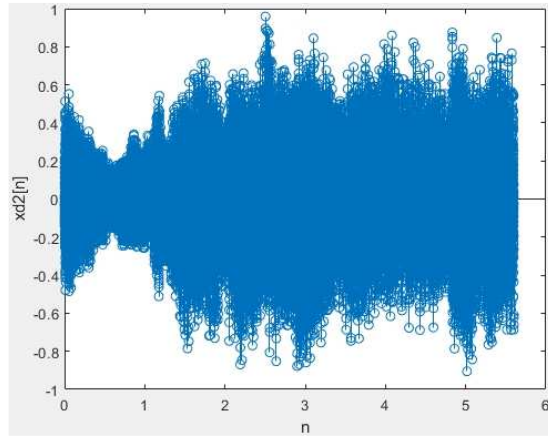
-잡음이 섞여 들리며  $x_{p2}[n]$ 의 경우는 소리가 완전히 깨져서 들린다.

6-1. Down-sampling을 하기 위해 문제 5에서 얻은 신호  $x_p[n]$ 에 대해, 아래의 식을 사용하였습니다.  $x_d[n]$ 의 파형을 그리세요. (M=5)

$$x_d[n] = x_p[Mn]$$

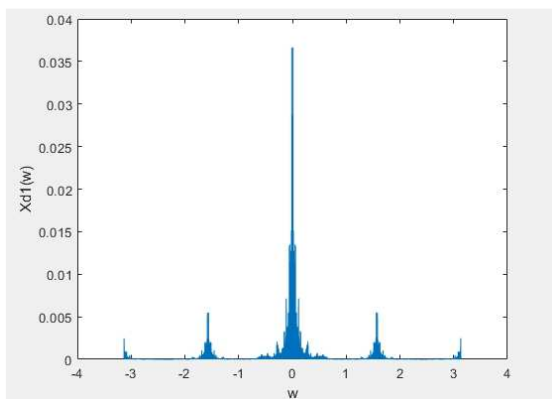


$x_{d1}[n]$

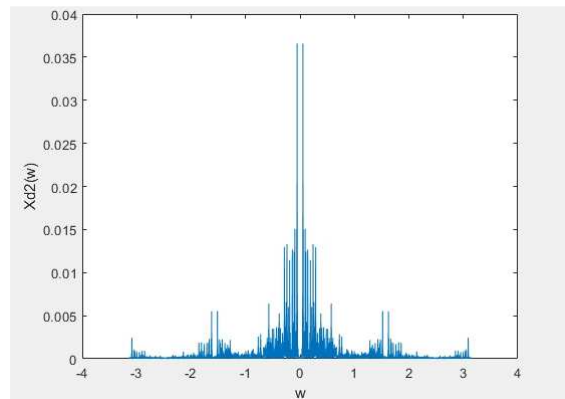


$x_{d2}[n]$

6-2.  $x_d[n]$ 의 Spectrum  $X_d(\omega)$ 의 magnitude 그래프를 그리세요.



$X_{d1}(w)$



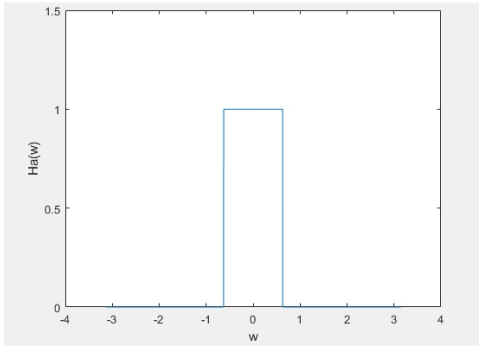
$X_{d2}(w)$

6-3.  $x_d[n]$ 을 청취해보고 원래 신호  $x_p[n]$ 과 어떤 차이가 있는지 기술하세요.

$x_d[n]$ 의 음량이  $x_p[n]$ 보다 더 크다.

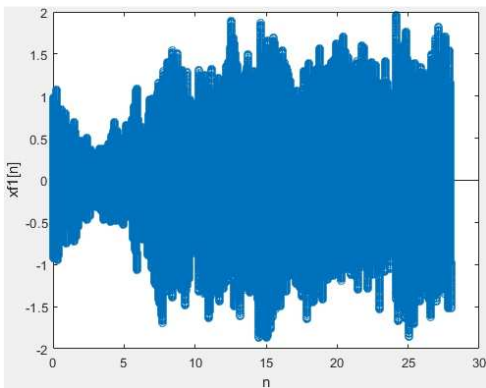


7-1.  $H_a(\omega)$ 의 magnitude 그래프를 그리세요.

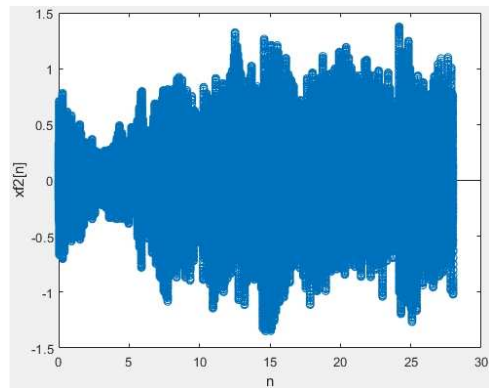


$H_a(\omega)$

7-2. 문제 7-1에서 제작한 Anti-aliasing filter를 이용하여 Down-sampling하려고 합니다. 원래는 time domain에서 windowing을 한 후 filtering을 해야하나, 이번 문제 7-2에서는 신호  $x_i[n]$ 를 **frequency domain**에서 바로 filtering하세요. (Convolution property 이용)  $x_i[n]$ 를 설계한 필터  $h_a[n](H_a(\omega))$ 로 filtering한 후 Down-sampling을 진행한  $x_f[n]$ 의 파형을 그리세요.

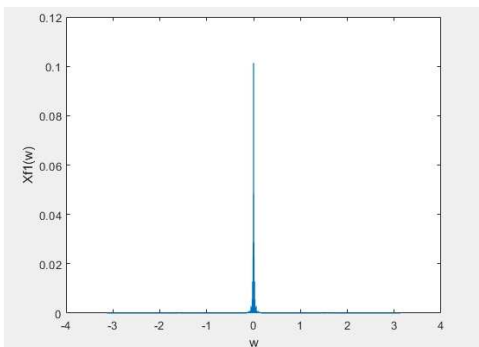


$x_{f1}[n]$

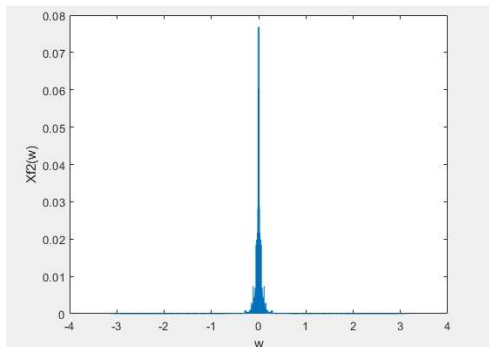


$x_{f2}[n]$

7-3.  $x_f[n]$ 의 Spectrum  $X_f(\omega)$ 의 magnitude를 그리세요.



$X_{f1}(\omega)$



$X_{f2}(\omega)$



### Discussion

입력 신호인 'Sample1.wav', 'Sample2.wav' 각각에 대한 문제 1~4의 결과들을 비교하고 Up-sampling 과정을 자세히 설명하세요. 또한 문제 5~7의 결과들을 비교하여 Down-sampling 과정을 토의하세요.

#### <Up-sampling>

-우선 Up-sampling을 하기 전에 각각의 신호에 zero-padding 과정을 거친다.

$$\text{Zero-padding: } x_e[n] = \begin{cases} x[n/L] & \text{for } n = Lk \\ 0 & \text{otherwise} \end{cases} \quad \text{where } L = 4$$

-그 후에  $x_e[n]$ 을 fourier transform을 했을 때 파형이 4배로 증가한 것을 알 수 있다.

-이상적인 경우라면 원래 신호 사이에 0이 추가되어도 들을 때 차이가 없어야 하는데, 실제로는 어떤 신호 값에서 0으로 변하는 과정이 있기 때문에 noise가 추가되는 것으로 보인다.

-그 후에 reconstruction filter  $h_r[n]$ 를 이의 frequency response인  $H_r(w)$ 을 먼저 설계한 뒤에 이를 inverse fourier transform 과정을 거쳐서 구현한다.

-이때 이상적인 LPF는 시간축에서 무한대의 길이를 요구하기 때문에 이 필터에 rectangular windowing을 해준다.

$$\text{Rectangular windowing: } h_{rL11}[n] = w_{11}[n] * h_r[n] \quad (11\text{points})$$

-Rectangular windowing 과정을 거친 LPF  $h_{rL11}[n]$ 와 zero-padding 과정을 거친  $x_e[n]$ 을 시간축 상에서 convolution한다.

$$x_{ieL11}[n] = x_e[n] * h_{rL11}[n]$$

- $x_{ieL11}[n]$  신호 또한 fourier transform을 했을 때 파형이 4배로 증가한 것을 알 수 있다.

### <Down-sampling>

-먼저  $x_i[n]$  신호의 decimation 과정 중 첫 번째 단계를 거친다.

$$x_p[n] = \begin{cases} x_i[n] & \text{for } n = Mk \\ 0 & \text{otherwise} \end{cases} \quad \text{where } M=5$$

-신호가 1/M로 감소하며, 기존 신호들 중 n이 5의 배수가 아닌 부분이 0으로 바뀔으로써 Up-sampling 과정과 마찬가지로 noise가 발생하는 것을 확인할 수 있다.

- $X_p(w)$ 의 waveform을 확인했을 때 한 주기( $-\pi \sim \pi$ ) 내에서 5개의 파형이 나타난 것을 알 수 있다.

-이후로  $x_i[n]$  신호의 decimation 과정 중 첫 번째 단계를 거친다.

$$x_d[n] = x_p[Mn]$$

-이때 신호의 길이가 1/M만큼 줄어든 것을 확인할 수 있다.

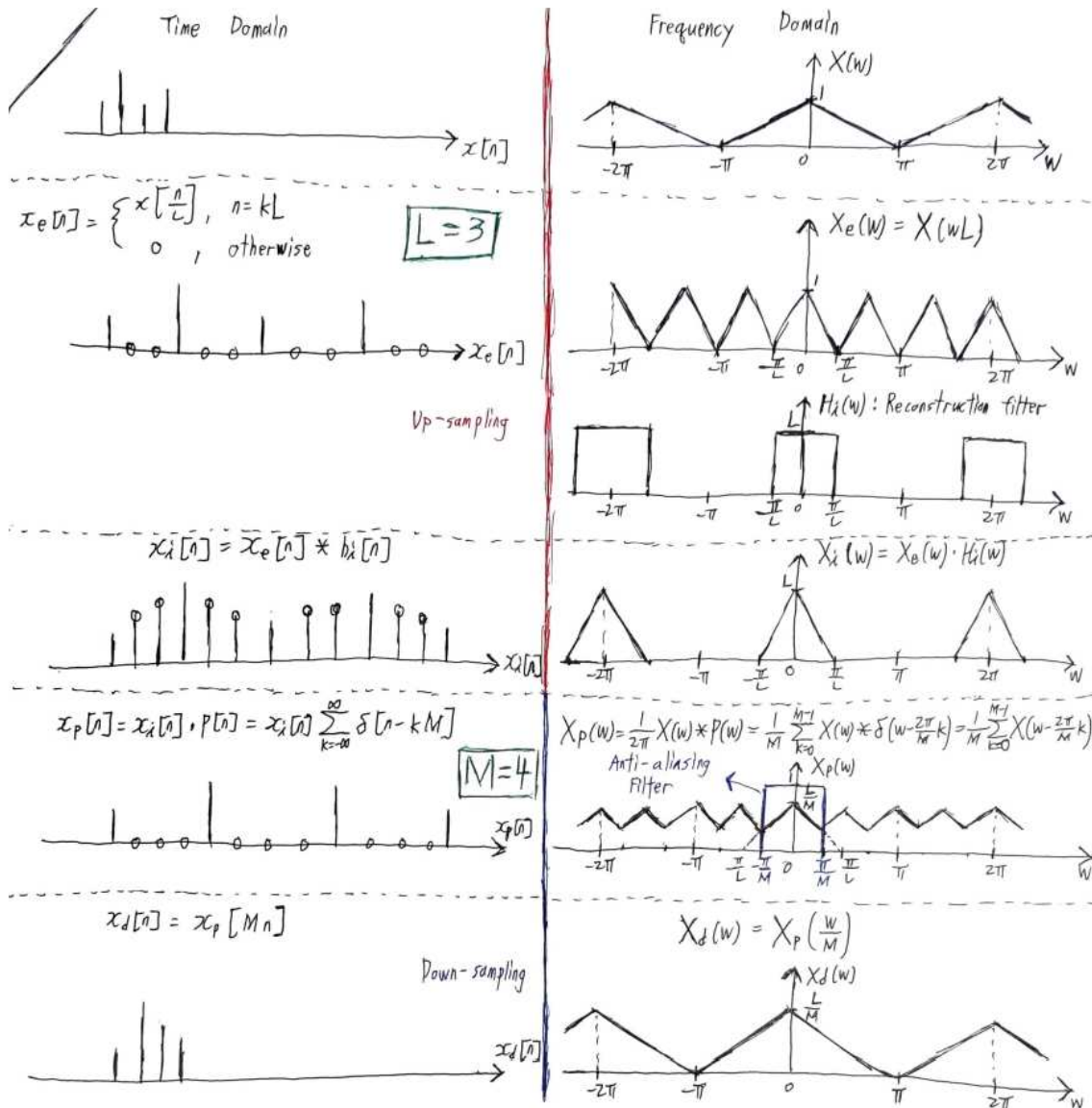
-이와 같은 Down-sampling 과정을 거치면서 aliasing 문제가 발생하므로, 이를 해결하기 위해 Anti-aliasing filter  $h_a[n]$ 의 frequency response  $H_a(w)$ 를 구현한다.

-이때 cutoff frequency는  $\pi/M$  이며, gain은 1이다.

$$w_c = \min(\pi/L, \pi/M) = \pi/M$$

-Down-sampling 과정을 모두 거친 신호가 정상적으로 출력되는 것을 확인할 수 있다.

# <Up & Down-sampling 정리노트>



## <Appendix>

### Q 1-1

```
% Q 1-1
clc;                                % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1);                     % The number of samples in x1
N2=length(y2);                     % The number of samples in x2
n1=(0:N1-1)/Fs1;                   % x-axis in x1
n2=(0:N2-1)/Fs2;                   % x-axis in x2

figure(1)                           % Waveform of x1
stem(n1,y1);
xlabel('n');
ylabel('x1[n]');

figure(2)                           % Waveform of x2
stem(n2,y2);
xlabel('n');
ylabel('x2[n]');
```

### Q 1-2

```
clc;                                % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1);                     % The number of samples in x1
N2=length(y2);                     % The number of samples in x2
n1=(-(N1-1)/2:(N1-1)/2)/Fs1;       % x-axis in x1
n2=(-(N2-1)/2:(N2-1)/2)/Fs2;       % x-axis in x2

X1=fft(y1)./N1;                    % DFT & Normalization of x1
m1=abs(fftshift(X1));              % Rearrangement of X1
f1=Fs1*n1/N1;                      % x-axis in frequency
w1=2*pi*f1;                        % x-axis in X1

X2=fft(y2)./N2;                    % DFT & Normalization of x2
m2=abs(fftshift(X2));              % Rearrangement of X2
f2=Fs2*n2/N2;                      % x-axis in frequency
w2=2*pi*f2;                        % x-axis in X2

figure(1)                           % Waveform of X1
plot(w1,log(m1));
xlabel('w');
ylabel('X1(w)');

figure(2)                           % Waveform of X2
plot(w2,log(m2));
xlabel('w');
ylabel('X2(w)');
```

## Q 1-3

```
clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
sound(y1,Fs1); % Play Sample1
pause(8);
sound(y2,Fs2); % Play Sample2
```

## Q 2-1

```
clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
Y1=zeros(1,N1+L); % xe1
Y2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    Y1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    Y2(j+L+1)=y2(j+1);
end

sound(Y1+L,Fs1+L);
pause(8);
sound(Y2+L,Fs2+L);
```

## Q 2-2

<pre>clc; % Reset clear;  [y1,Fs1]=audioread('Sample1.wav'); [y2,Fs2]=audioread('Sample2.wav'); N1=length(y1); % The number of samples in x1 N2=length(y2); % The number of samples in x2  L=4; % Gain=4 Y1=zeros(1,N1+L); % xe1 Y2=zeros(1,N2+L); % xe2  for i=0:N1-1 % x1 Zero-padding     Y1(i+L+1)=y1(i+1); end  for j=0:N2-1 % x2 Zero-padding     Y2(j+L+1)=y2(j+1); end</pre>	<pre>X1=fft(Y1)./(N1+L); % DFT &amp; Normalization of xe1 m1=abs(fftshift(X1)); % Rearrangement of Xe1 f1=((-N1+L/2):(N1+L/2)-1)/(N1+L); % x-axis in frequency w1=2*pi*f1; % x-axis in Xe1  X2=fft(Y2)./(N2+L); % DFT &amp; Normalization of xe2 m2=abs(fftshift(X2)); % Rearrangement of Xe2 f2=((-N2+L/2):(N2+L/2)-1)/(N2+L); % x-axis in frequency w2=2*pi*f2; % x-axis in Xe2  figure(1) % Waveform of Xe1 plot(w1,log(m1)); xlabel('w'); ylabel('Xe1(w)');  figure(2) % Waveform of Xe2 plot(w2,log(m2)); xlabel('w'); ylabel('Xe2(w)');</pre>
--	---

### Q 3-1

```
clc; % Reset
clear;

fs=1000;
w=-pi:1/fs:pi;
L=4;
H=L.*(-pi/L<=w&w<=pi/L);

figure(1)
plot(w,H);
xlabel('w');
ylabel('H(w)');
ylim([0 5]);
```

### Q 3-2

```
% Q 3-2
clc; % Reset
clear;

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):(W/2-1); % x-axis of hr[n]

w3=1.*(-3<=n&n<=3); % 3 points rectangular window
hr3=w3.*hr; % hrL3[n]=w3[n]*hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hrL11[n]=w11[n]*hr[n]

figure(1) % Waveform of hrL3[n]
stem(n,hr3);
xlim([-15 15]);
xlabel('n');
ylabel('hrL3[n]');

figure(2) % Waveform of hrL11[n]
stem(n,hr11);
xlim([-15 15]);
xlabel('n');
ylabel('hrL11[n]');
```

### Q 3-3

```

clc; % Reset
clear;

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):((W/2)-1); % x-axis of hr[n]
N=length(n);

w3=1.*(-3<=n&n<=3); % 3 points rectangular window
hr3=w3.*hr; % hrL3[n]=w3[n]*hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hrL11[n]=w11[n]*hr[n]

H3=fft(hr3)/N; % DFT & Normalization of hrL3[n]
Hr3=abs(fftshift(H3)); % Rearrangement of HrL3(w)

H11=fft(hr11)/N; % DFT & Normalization of hrL11[n]
Hr11=abs(fftshift(H11)); % Rearrangement of HrL11(w)

w_rad=2*pi*(-N/2:N/2-1)/N; % x-axis of HrL3(w) & HrL11(w)

figure(1) % Waveform of HrL3(w)
plot(w_rad,Hr3);
xlim([-15 15]);
xlabel('w');
ylabel('HrL3(w)');

figure(2) % Waveform of HrL11(w)
plot(w_rad,Hr11);
xlim([-15 15]);
xlabel('w');
ylabel('HrL11(w)');

```



## Q 4-1

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):((W/2)-1); % x-axis of hr[n]

w3=1.*(-3<=n&n<=3); % 3 points rectangular window
hr3=w3.*hr; % hrL3[n]=w3[n]*hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hrL11[n]=w11[n]*hr[n]

xie1L3=conv(xe1,hr3,'same'); % xie1L3[n]=xe1[n]*hrL3[n]
xie1L11=conv(xe1,hr11,'same'); % xie1L11[n]=xe1[n]*hrL11[n]
xie2L3=conv(xe2,hr3,'same'); % xie2L3[n]=xe2[n]*hrL3[n]
xie2L11=conv(xe2,hr11,'same'); % xie2L11[n]=xe2[n]*hrL11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1L3[n] & xie1L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2L3[n] & xie2L11[n]

figure(1)
subplot(2,1,1); % Waveform of xie1L3[n]
stem(k1,xie1L3);
xlabel('n');
ylabel('xie1L3[n]');

subplot(2,1,2); % Waveform of xie1L11[n]
stem(k1,xie1L11);
xlabel('n');
ylabel('xie1L11[n]');

figure(2)
subplot(2,1,1); % Waveform of xie2L3[n]
stem(k2,xie2L3);
xlabel('n');
ylabel('xie2L3[n]');

subplot(2,1,2); % Waveform of xie2L11[n]
stem(k2,xie2L11);
xlabel('n');
ylabel('xie2L11[n]');

```

## Q 4-2

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:(N1+L-1))/Fs1; % x-axis in xe1
n2=(0:(N2+L-1))/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):((W/2)-1); % x-axis of hr[n]

w3=1.*(-3<=n&n<=3); % 3 points rectangular window
hr3=w3.*hr; % hrL3[n]=w3[n]*hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hrL11[n]=w11[n]*hr[n]

xie1_L3=conv(xe1,hr3,'same'); % xie1L3[n]=xe1[n]*hrL3[n]
xie1_L11=conv(xe1,hr11,'same'); % xie1L11[n]=xe1[n]*hrL11[n]
xie2_L3=conv(xe2,hr3,'same'); % xie2L3[n]=xe2[n]*hrL3[n]
xie2_L11=conv(xe2,hr11,'same'); % xie2L11[n]=xe2[n]*hrL11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1L3[n] & xie1L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2L3[n] & xie2L11[n]
K1=length(k1);
K2=length(k2);

Xie1_L3=abs(fftshift(fft(xie1_L3)./K1)); % DFT of xie1_L3[n]
Xie1_L11=abs(fftshift(fft(xie1_L11)./K1)); % DFT of xie1_L11[n]
Xie2_L3=abs(fftshift(fft(xie2_L3)./K2)); % DFT of xie2_L3[n]
Xie2_L11=abs(fftshift(fft(xie2_L11)./K2)); % DFT of xie2_L11[n]

w1_rad=2*pi*(-K1/2:K1/2-1)/K1; % x-axis of Xie1L3(w) & Xie1L11(w)
w2_rad=2*pi*(-K2/2:K2/2-1)/K2; % x-axis of Xie2L3(w) & Xie2L11(w)

figure(1)
subplot(2,1,1); % Waveform of Xie1L3(w)
plot(w1_rad,Xie1_L3);
xlabel('w');
ylabel('Xie1L3(w)');

subplot(2,1,2); % Waveform of Xie1L11(w)
plot(w1_rad,Xie1_L11);
xlabel('w');
ylabel('Xie1L11(w)');

figure(2)
subplot(2,1,1); % Waveform of Xie2L3(w)
plot(w2_rad,Xie2_L3);
xlabel('w');
ylabel('Xie2L3(w)');

subplot(2,1,2); % Waveform of Xie2L11(w)
plot(w2_rad,Xie2_L11);
xlabel('w');
ylabel('Xie2L11(w)');

```

## Q 4-3

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:(N1+L-1))/Fs1; % x-axis in xe1
n2=(0:(N2+L-1))/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):((W/2)-1); % x-axis of hr[n]

w3=1.*(-3<=n&n<=3); % 3 points rectangular window
hr3=w3.*hr; % hrL3[n]=w3[n]*hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hrL11[n]=w11[n]*hr[n]

xie1_L3=conv(xe1,hr3,'same'); % xie1L3[n]=xe1[n]*hrL3[n]
xie1_L11=conv(xe1,hr11,'same'); % xie1L11[n]=xe1[n]*hrL11[n]
xie2_L3=conv(xe2,hr3,'same'); % xie2L3[n]=xe2[n]*hrL3[n]
xie2_L11=conv(xe2,hr11,'same'); % xie2L11[n]=xe2[n]*hrL11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1L3[n] & xie1L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2L3[n] & xie2L11[n]

sound(xe1,Fs1+L); % Play xe1[n]
pause(8);
sound(xie1_L11,Fs1+L); % Play xie1_L11[n]
pause(8);
sound(xe2,Fs2+L); % Play xe2[n]
pause(8);
sound(xie2_L11,Fs2+L); % Play xie2_L11[n]

```

## Q 5-1

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1*L); % xe1
xe2=zeros(1,N2*L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i*L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j*L+1)=y2(j+1);
end

n1=(0:N1*L-1)/Fs1; % x-axis in xe1
n2=(0:N2*L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):(W/2-1); % x-axis of hr[n]

w11=1.*( -11<=n&n<=11); % 11 points rectangular window
hrL11[n]=w11[n]*hr[n];

xie1_L11=conv(xe1,hr11,'same'); % xie1_L11[n]=xe1[n]*hrL11[n]
xie2_L11=conv(xe2,hr11,'same'); % xie2_L11[n]=xe2[n]*hrL11[n]

k1=(0:(N1*L-1))/Fs1; % x-axis of xie1_L11[n]
k2=(0:(N2*L-1))/Fs2; % x-axis of xie2_L11[n]
K1=length(k1);
K2=length(k2);

M=5;
xp1=zeros(1,K1); % xp1
xp2=zeros(1,K2); % xp2

for l=1:K1
    if rem(l,M)==0
        xp1(l)=xie1_L11(l);
    end
end

for J=1:K2
    if rem(J,M)==0
        xp2(J)=xie2_L11(J);
    end
end

p1=(0:K1-1)/Fs1; % x-axis in xp1
p2=(0:K2-1)/Fs2; % x-axis in xp2

figure(1) % Waveform of xp1[n]
stem(p1,xp1);
xlabel('n');
ylabel('xp1[n]');

figure(2) % Waveform of xp2[n]
stem(p2,xp2);
xlabel('n');
ylabel('xp2[n]');

```

## Q 5-2

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=pi*1/fs/pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):(W/2-1); % x-axis of hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hr11[n]=w11[n]*hr[n]

xie1_L11=conv(xe1,hr11,'same'); % xie1_L11[n]=xe1[n]*hr11[n]
xie2_L11=conv(xe2,hr11,'same'); % xie2_L11[n]=xe2[n]*hr11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1_L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2_L11[n]
K1=length(k1);
K2=length(k2);

M=5;
xp1=zeros(1,K1); % xp1
xp2=zeros(1,K2); % xp2

for l=1:K1
    if rem(l,M)==0
        xp1(l)=xie1_L11(l);
    end
end

for J=1:K2
    if rem(J,M)==0
        xp2(J)=xie2_L11(J);
    end
end

p1=(0:K1-1)/Fs1; % x-axis in xp1
p2=(0:K2-1)/Fs2; % x-axis in xp2

Xp1=abs(fftshift(fft(xp1)./K1)); % DFT of xp1[n]
Xp2=abs(fftshift(fft(xp2)./K2)); % DFT of xp2[n]

w1=2*pi*(-K1/2:K1/2-1)/K1; % x-axis of Xp1(w)
w2=2*pi*(-K2/2:K2/2-1)/K2; % x-axis of Xp2(w)

figure(1) % Waveform of Xp1(w)
plot(w1,Xp1);
xlabel('w');
ylabel('Xp1(w)');

figure(2) % Waveform of Xp2(w)
plot(w2,Xp2);
xlabel('w');
ylabel('Xp2(w)');

```

## Q 5-3

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*( -pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(fftshift(h)); % Rearrangement of hr[n]
n=(-W/2):((W/2)-1); % x-axis of hr[n]

w11=1.*( -11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hr11[n]=w11[n]*hr[n]

xie1_L11=conv(xe1,hr11,'same'); % xie1_L11[n]=xe1[n]*hr11[n]
xie2_L11=conv(xe2,hr11,'same'); % xie2_L11[n]=xe2[n]*hr11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1_L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2_L11[n]
K1=length(k1);
K2=length(k2);

M=5;
xp1=zeros(1,K1); % xp1
xp2=zeros(1,K2); % xp2

for l=1:K1
    if rem(l,M)==0
        xp1(l)=xie1_L11(l);
    end
end

for J=1:K2
    if rem(J,M)==0
        xp2(J)=xie2_L11(J);
    end
end

sound(xie1_L11,Fs1+L); % Play xie1_L11[n]
pause(8);
sound(xp1,Fs1+L); % Play xp1[n]
pause(8);
sound(xie2_L11,Fs2+L); % Play xie2_L11[n]
pause(8);
sound(xp2,Fs2+L); % Play xp2[n]

```

## Q 6-1

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:L/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):((W/2)-1); % x-axis of hr[n]

w1=1.*(1<=n&n<=11); % 11 points rectangular window
hr1=w1.*hr; % hr1[n]=w1[n]*hr[n]

xie1_L11=conv(xe1,hr1,'same'); % xie1_L11[n]=xe1[n]*hr_L11[n]
xie2_L11=conv(xe2,hr1,'same'); % xie2_L11[n]=xe2[n]*hr_L11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1_L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2_L11[n]
K1=length(k1);
K2=length(k2);

M=5;
xp1=zeros(1,K1); % xp1
xp2=zeros(1,K2); % xp2

for i=1:K1
    if rem(i,M)==0
        xp1(i)=xie1_L11(i);
    end
end

for j=1:K2
    if rem(j,M)==0
        xp2(j)=xie2_L11(j);
    end
end

d1=(0:(K1/M-1))/Fs1; % x-axis of xd1[n]
d2=(0:(K2/M-1))/Fs2; % x-axis of xd2[n]
D1=length(d1);
D2=length(d2);
xd1=zeros(1,D1); % xd1[n]
xd2=zeros(1,D2); % xd2[n]

for i=1:D1
    xd1(i)=xp1(i+M); % xd1[n]=xp1[Mn]
end

for j=1:D2
    xd2(j)=xp2(j+M); % xd2[n]=xp2[Mn]
end

figure(1) % Waveform of xd1[n]
stem(d1,xd1);
xlabel('n');
ylabel('xd1[n]');

figure(2) % Waveform of xd2[n]
stem(d2,xd2);
xlabel('n');
ylabel('xd2[n]');

```



## Q 6-2

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):(W/2-1); % x-axis of hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hr11[n]=w11[n]*hr[n]

xie1_L11=conv(xe1,hr11,'same'); % xie1_L11[n]=xe1[n]*hr11[n]
xie2_L11=conv(xe2,hr11,'same'); % xie2_L11[n]=xe2[n]*hr11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1_L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2_L11[n]
K1=length(k1);
K2=length(k2);

M=5;
xp1=zeros(1,K1); % xp1
xp2=zeros(1,K2); % xp2

for i=1:K1
    if rem(i,M)==0
        xp1(i)=xie1_L11(i);
    end
end

for j=1:K2
    if rem(j,M)==0
        xp2(j)=xie2_L11(j);
    end
end

d1=(0:(K1/M-1))/Fs1; % x-axis of xd1[n]
d2=(0:(K2/M-1))/Fs2; % x-axis of xd2[n]
D1=length(d1);
D2=length(d2);
xd1=zeros(1,D1); % xd1[n]
xd2=zeros(1,D2); % xd2[n]

for i=1:D1
    xd1(i)=xp1(i*M); % xd1[n]=xp1[Mn]
end

for j=1:D2
    xd2(j)=xp2(j*M); % xd2[n]=xp2[Mn]
end

w1=2*pi*(-D1/2:D1/2-1)/D1; % x-axis of Xd1(w)
w2=2*pi*(-D2/2:D2/2-1)/D2; % x-axis of Xd2(w)
Xd1=abs(fftshift(fft(xd1)./D1)); % DFT of xd1[n]
Xd2=abs(fftshift(fft(xd2)./D2)); % DFT of xd2[n]

figure(1) % Waveform of Xd1(w)
plot(w1,Xd1);
xlabel('w');
ylabel('Xd1(w)');

figure(2) % Waveform of Xd2(w)
plot(w2,Xd2);
xlabel('w');
ylabel('Xd2(w)');

```

## Q 6-3

```

clc; % Reset
clear;

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

h=ifft(H); % IDFT of H(w)
hr=abs(ifftshift(h)); % Rearrangement of hr[n]
n=(-W/2):(W/2-1); % x-axis of hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hrL11=w11.*hr; % hrL11[n]=w11[n]*hr[n]

xie1_L11=conv(xe1,hrL11,'same'); % xie1_L11[n]=xe1[n]*hrL11[n]
xie2_L11=conv(xe2,hrL11,'same'); % xie2_L11[n]=xe2[n]*hrL11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xie1_L11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xie2_L11[n]
K1=length(k1);
K2=length(k2);

M=5;
xp1=zeros(1,K1); % xp1
xp2=zeros(1,K2); % xp2

for i=1:K1
    if rem(i,M)==0
        xp1(i)=xie1_L11(i);
    end
end

for j=1:K2
    if rem(j,M)==0
        xp2(j)=xie2_L11(j);
    end
end

d1=(0:(K1/M-1))/Fs1; % x-axis of xd1[n]
d2=(0:(K2/M-1))/Fs2; % x-axis of xd2[n]
D1=length(d1);
D2=length(d2);
xd1=zeros(1,D1); % xd1[n]
xd2=zeros(1,D2); % xd2[n]

for i=1:D1
    xd1(i)=xp1(i*M); % xd1[n]=xp1[Mn]
end

for j=1:D2
    xd2(j)=xp2(j*M); % xd2[n]=xp2[Mn]
end

sound(xp1,Fs1*L); % Play xp1[n]
pause(8);
sound(xd1,Fs1*L/M); % Play xd1[n]
pause(8);
sound(xp2,Fs2*L); % Play xp2[n]
pause(8);
sound(xd2,Fs2*L/M); % Play xd2[n]

```

## Q 7-1

```

clc;
clear;

fs=1000;
w=-pi:1/fs:pi;
M=5;
Ha=1.*(-pi/M<=w&w<=pi/M);

figure(1)
plot(w,Ha);
xlabel('w');
ylabel('Ha(w)');
ylim([0 1.5]);

```

## Q 7-2

```

clc;
clear;

fs=1000;
wa=-pi:1/fs:pi;
W=length(wa);
M=5;
Ha=1.*(-pi/M<=wa&wa<=pi/M); % Ha(w)
ha=abs(ifftshift(ifft(Ha))); % IDFT of Ha(w)

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:N1+L-1)/Fs1; % x-axis in xe1
n2=(0:N2+L-1)/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)
hr=abs(ifftshift(ifft(H))); % IDFT of H(w)
n=(-W/2):((W/2)-1); % x-axis of hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hr11[n]=w11[n]*hr[n]

xi1=conv(xe1,hr11,'same'); % xi1[n]=xe1[n]*hr11[n]
xi2=conv(xe2,hr11,'same'); % xi2[n]=xe2[n]*hr11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of xi1[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of xi2[n]
K1=length(k1);
K2=length(k2);

xf1=conv(xi1,ha,'same'); % xf1[n]=xi1[n]*ha[n]
xf2=conv(xi2,ha,'same'); % xf2[n]=xi2[n]*ha[n]

figure(1) % Waveform of xf1[n]
stem(k1,xf1);
xlabel('n');
ylabel('xf1[n]');

figure(2) % Waveform of xf2[n]
stem(k2,xf2);
xlabel('n');
ylabel('xf2[n]');

```

## Q 7-3

```

clc;
clear;

fs=1000;
wa=-pi:1/fs:pi;
W=length(wa);
M=5;
Ha=1, +(-pi/M<=wa&wa<=pi/M); % Ha(w)
ha=abs(ifftshift(ifft(Ha))); % IDFT of Ha(w)

[y1,Fs1]=audioread('Sample1.wav');
[y2,Fs2]=audioread('Sample2.wav');
N1=length(y1); % The number of samples in x1
N2=length(y2); % The number of samples in x2

L=4; % Gain=4
xe1=zeros(1,N1+L); % xe1
xe2=zeros(1,N2+L); % xe2

for i=0:N1-1 % x1 Zero-padding
    xe1(i+L+1)=y1(i+1);
end

for j=0:N2-1 % x2 Zero-padding
    xe2(j+L+1)=y2(j+1);
end

n1=(0:(N1+L-1))/Fs1; % x-axis in xe1
n2=(0:(N2+L-1))/Fs2; % x-axis in xe2

fs=1000;
w=-pi:1/fs:pi;
W=length(w);
L=4;
H=L.*(-pi/L<=w&w<=pi/L); % H(w)

hr=abs(ifftshift(ifft(H))); % IDFT of H(w)
n=(-W/2):(W/2-1); % x-axis of hr[n]

w11=1.*(-11<=n&n<=11); % 11 points rectangular window
hr11=w11.*hr; % hr11[n]=w11[n]*hr[n]

x11=conv(xe1,hr11,'same'); % x11[n]=xe1[n]*hr11[n]
x12=conv(xe2,hr11,'same'); % x12[n]=xe2[n]*hr11[n]

k1=(0:(N1+L-1))/Fs1; % x-axis of x11[n]
k2=(0:(N2+L-1))/Fs2; % x-axis of x12[n]
K1=length(k1);
K2=length(k2);

xf1=conv(x11,ha,'same'); % xf1[n]=x11[n]*ha[n]
xf2=conv(x12,ha,'same'); % xf2[n]=x12[n]*ha[n]

Xf1=abs(fftshift(fft(xf1)./K1)); % DFT of x11[n]
Xf2=abs(fftshift(fft(xf2)./K2)); % DFT of x12[n]

w1=2*pi*(-K1/2:K1/2-1)/K1; % x-axis Xf1(w)
w2=2*pi*(-K2/2:K2/2-1)/K2; % x-axis Xf2(w)

figure(1) % Waveform of Xf1(w)
plot(w1,Xf1);
xlabel('w');
ylabel('Xf1(w)');

figure(2) % Waveform of Xf2(w)
plot(w2,Xf2);
xlabel('w');
ylabel('Xf2(w)');

```