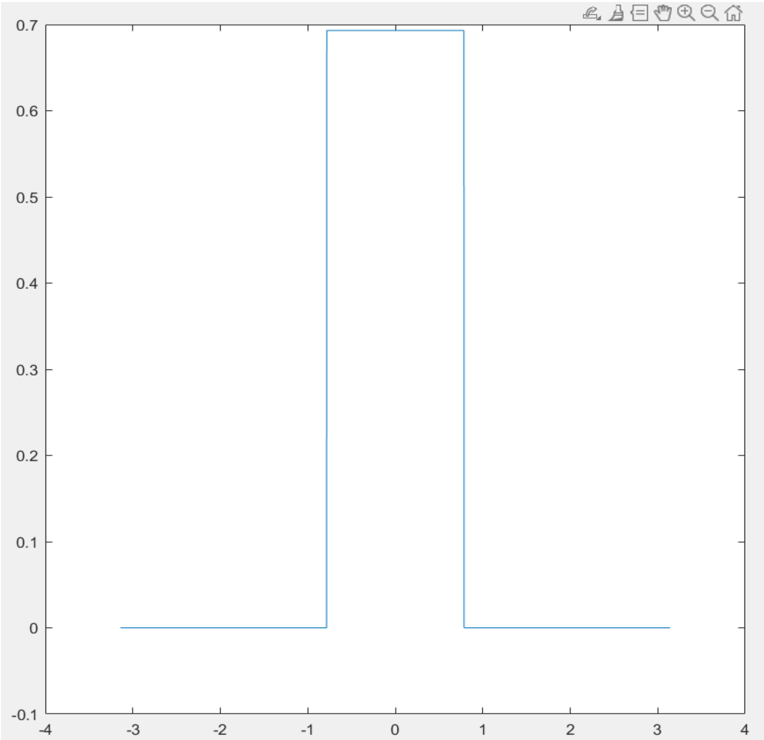
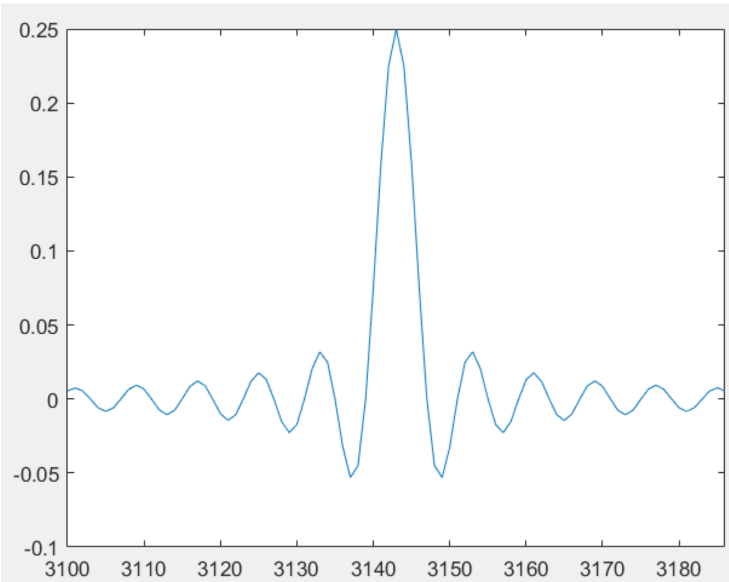
1.Ideal Low pass Filter & Windowing

1-1

주파수축에서의 LPF Magnitude (-pi~pi 파형)

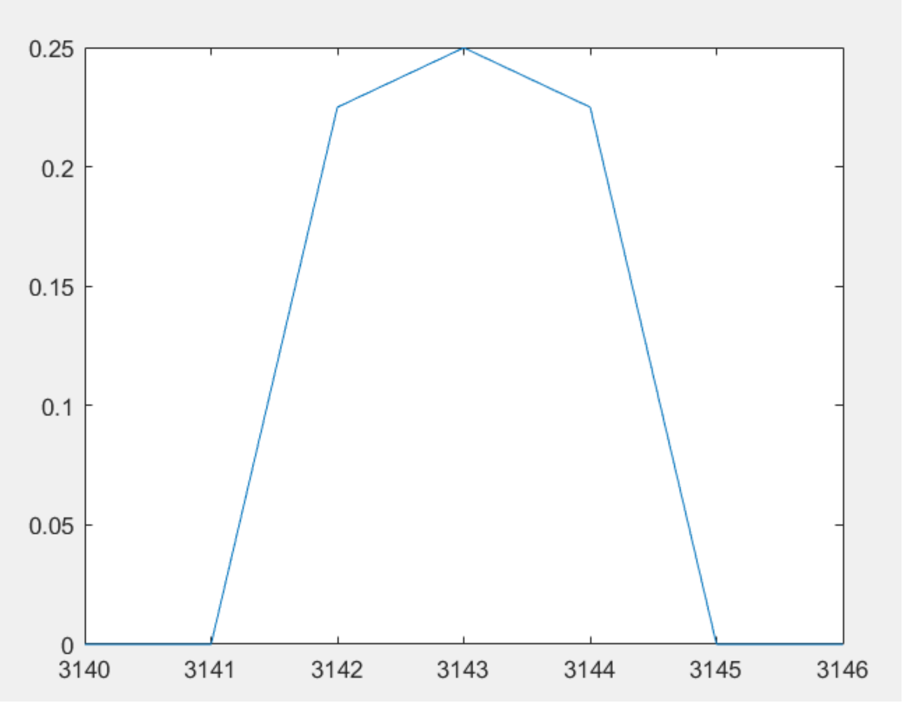


시간축에서의 LPF Magnitude (-pi~pi 파형)

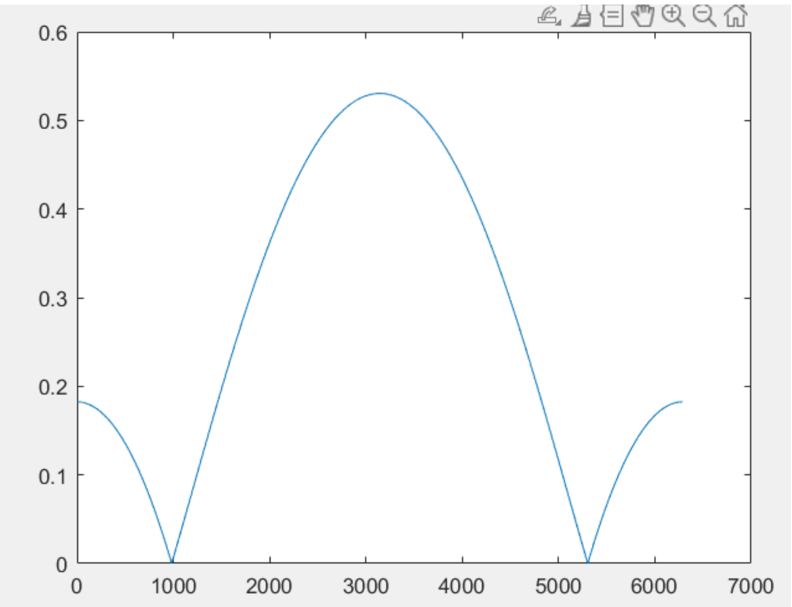


1-2

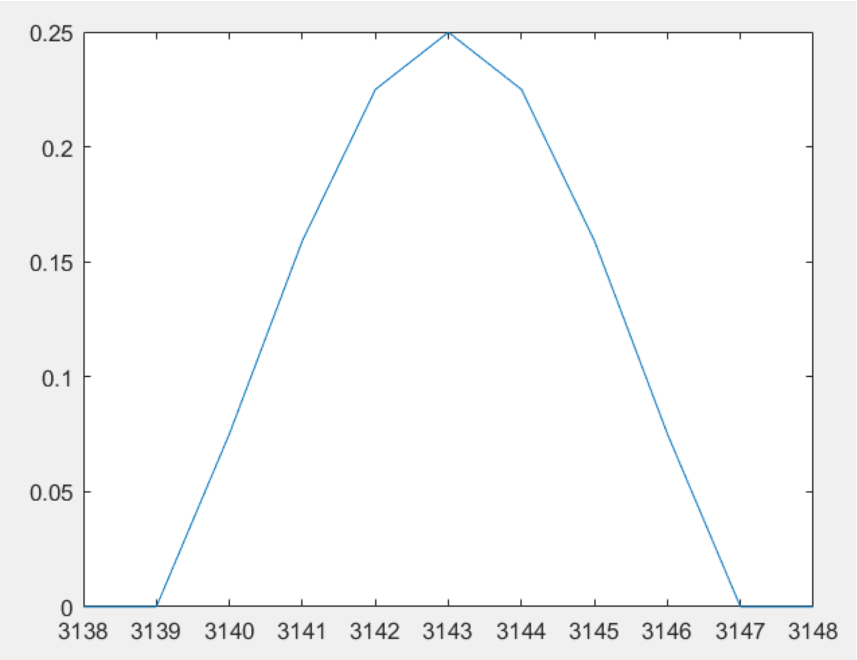
3 point 만 window 했을때의 시간축의 lpf (h3[n])



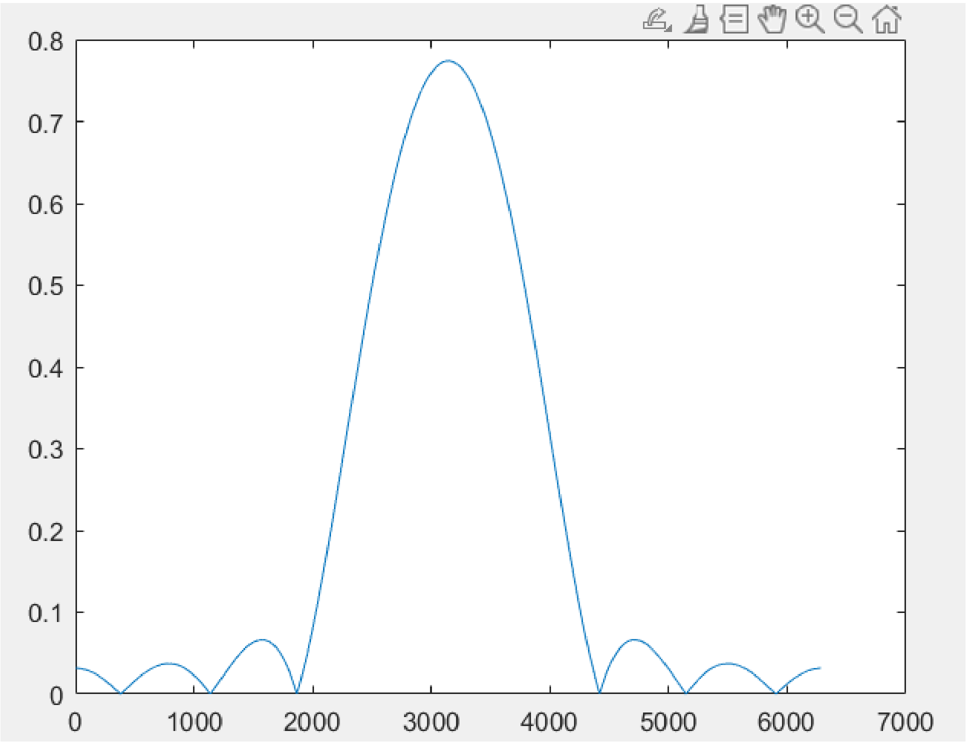
3 point 만 window 했을때의 주파수축의 lpf (H3[w])



7 point 만 window 했을때의 시간축의 lpf (h7[n])



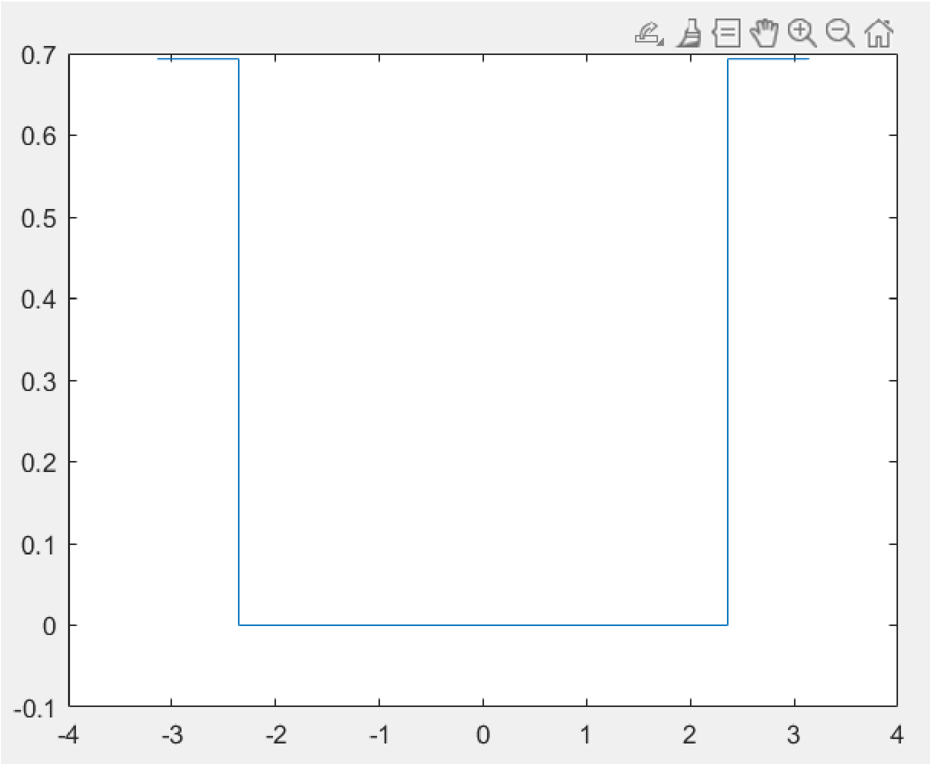
7 point 만 window 했을때의 주파수축의 lpf (H7[w])



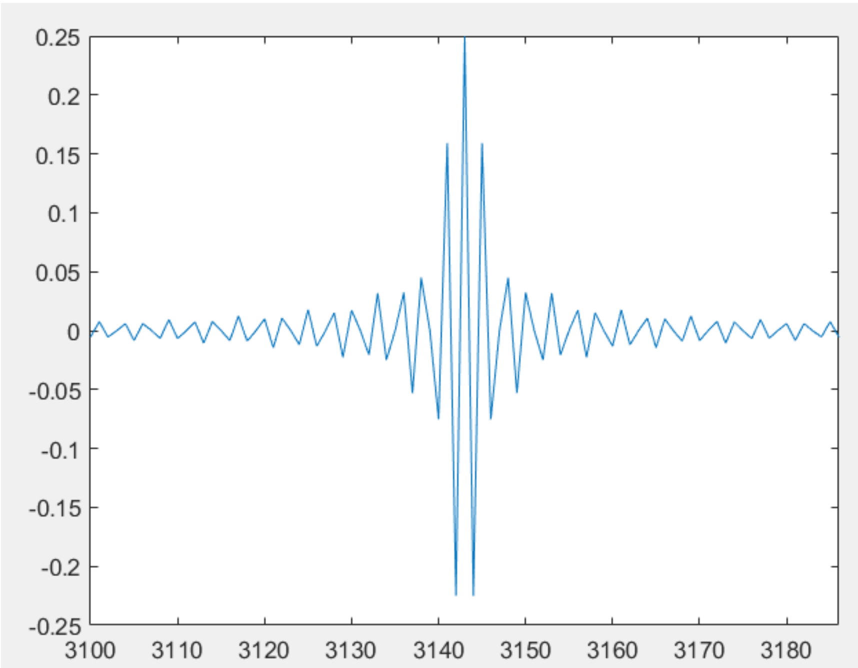
2.Ideal High pass Filter & Windowing

2-1

주파수축에서의 HPF Magnitude (-pi~pi 파형)

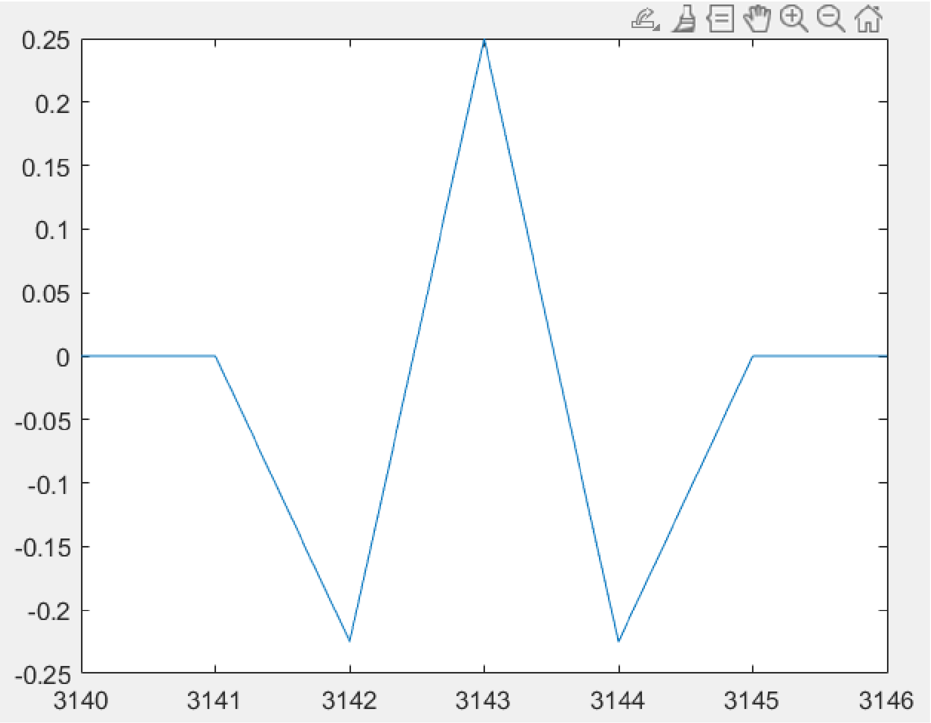


시간축에서의 HPF Magnitude (-pi~pi 파형)

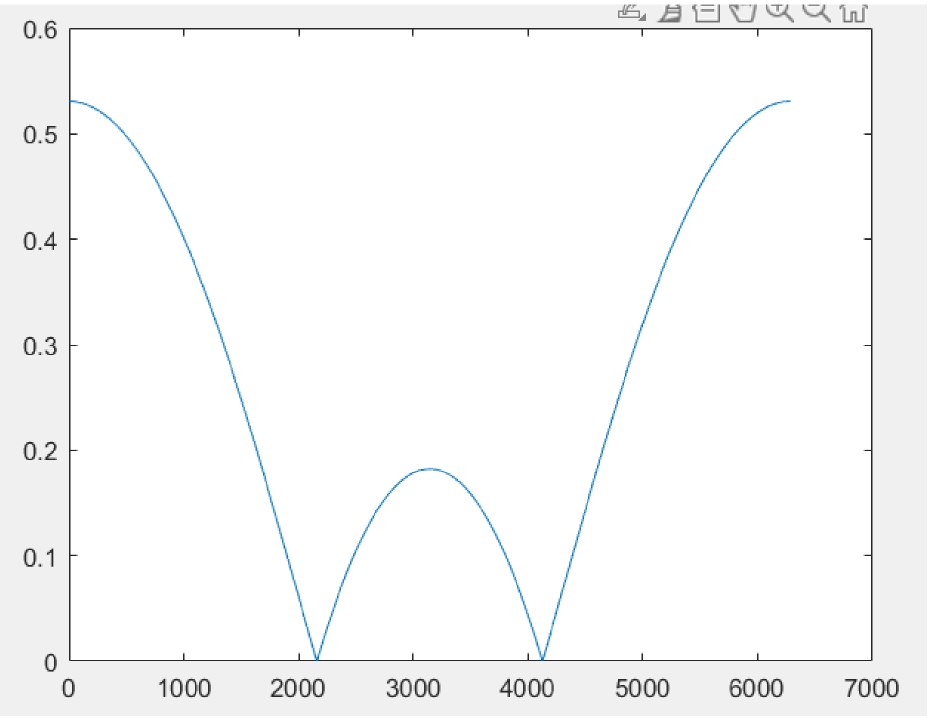


2-2

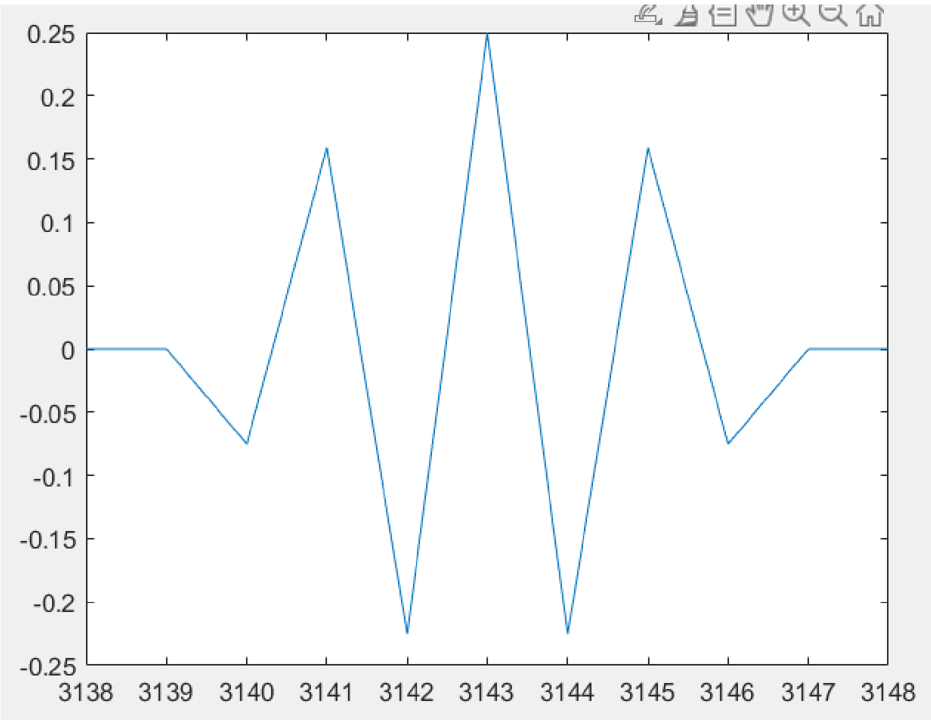
3 point 만 window 했을때의 시간축의 hpf (h3[n])



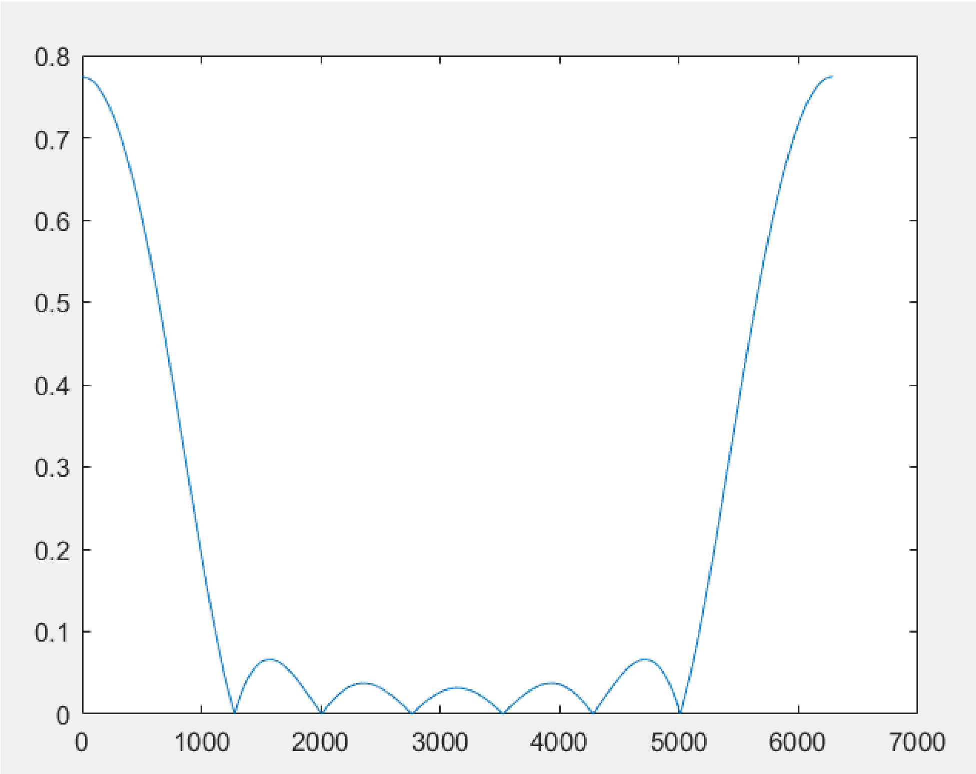
3 point 만 window 했을때의 주파수축의 hpf (H3[w])



7 point 만 window 했을때의 시간축의 hpf (h7[n])

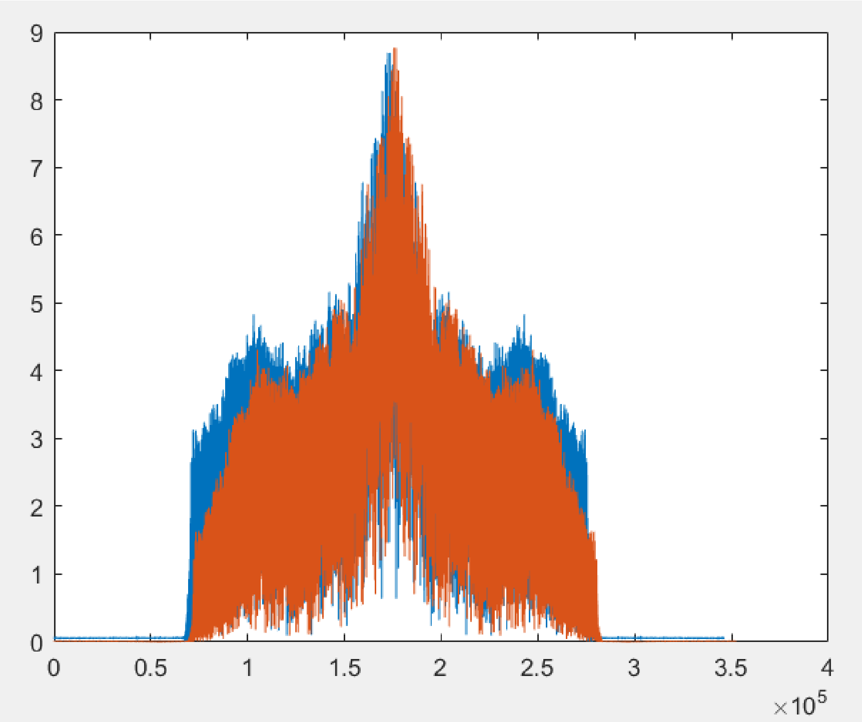


7 point 만 window 했을때의 주파수축의 hpf (H7[w])

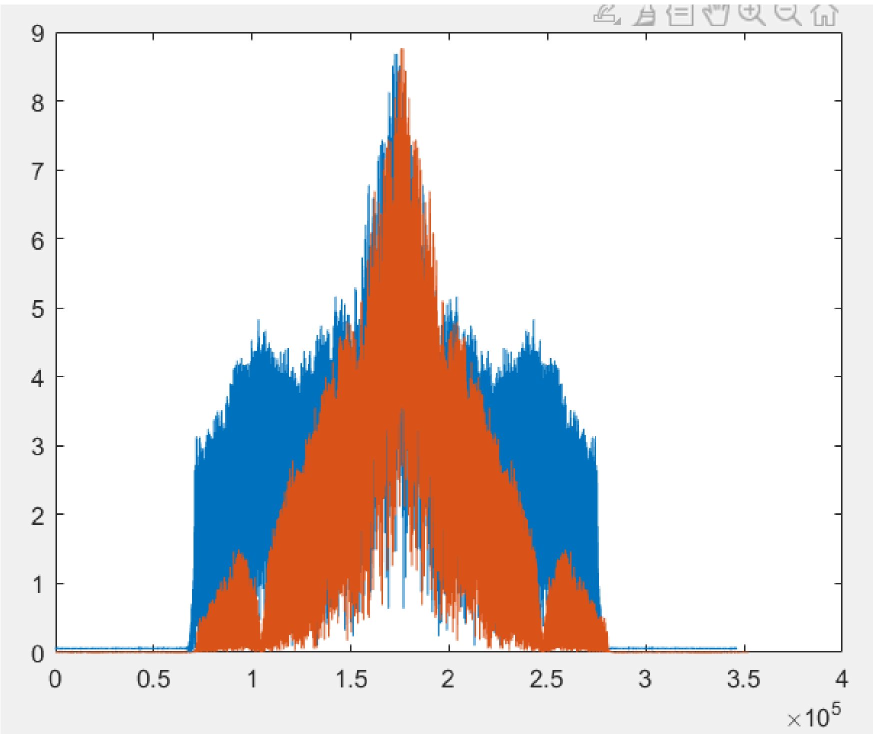


3.Filterting (뒤의 파랑색이 원래 신호)

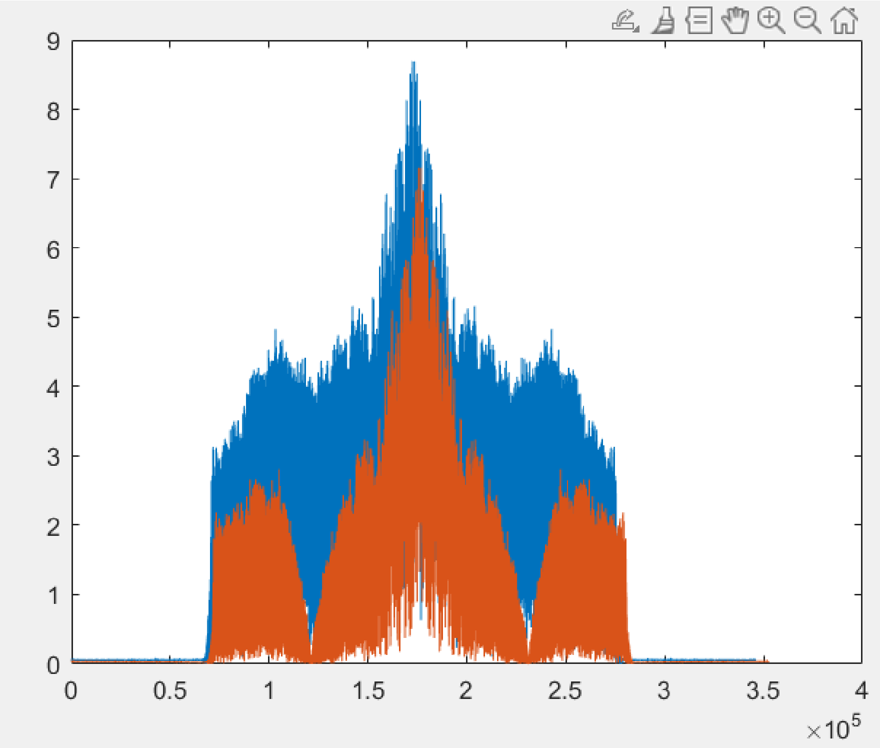
3 point LPF의 convolution 결과



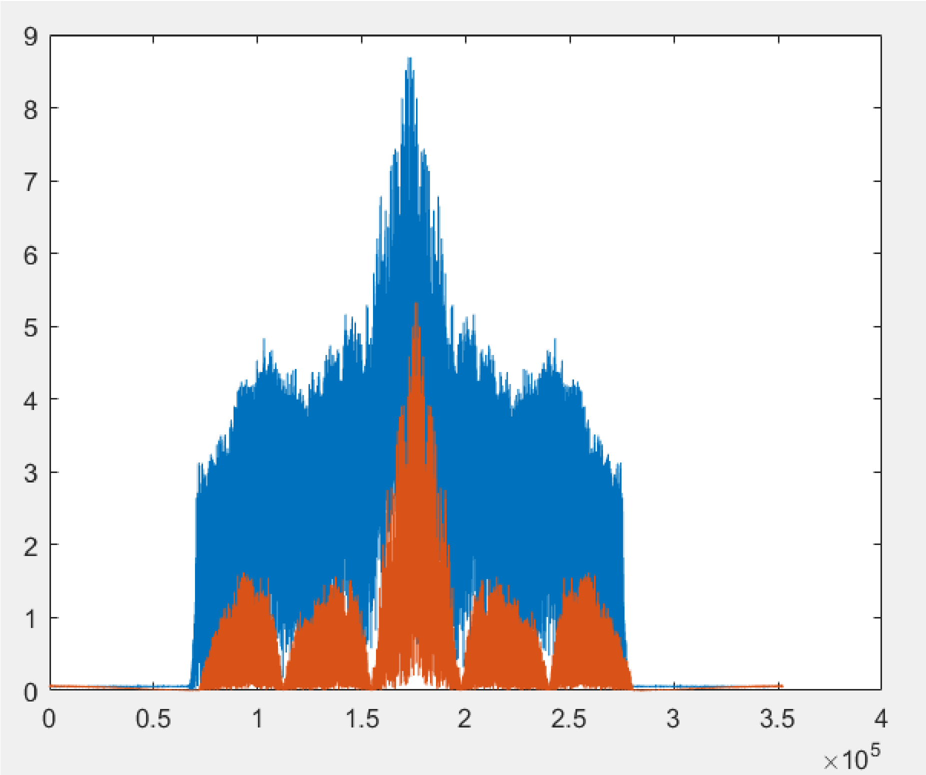
7 point LPF의 convolution 결과



3 point HPF의 convolution 결과



7 point LPF의 convolution 결과



4. Discussion

4-1

3에서 한 Magnitude를 보면 알수 있듯이, 파형에서 중심쪽이 0에 가까운쪽으로 저 주파수 대역, 중심에서 멀어질수록 고 주파수 대역이다.

LPF 결과를 보면. 중심쪽은 거의 그대로 Pass 시키지만. 중심에서 멀어질수록 점점 Magnitdue가 작아져 고주파수쪽은 차단 시킴을 볼 수 있다.

HPF의 경우 중심쪽은 차단 시키고 중심에서 멀어질수록 Pass 시킬 것이다. 하지만. HPF의 Cutoff를 3pi/4 로 두었는데 이는 꽤 높은 주파수 대역만 통과 시키다는 것을 알 수 있다. 3번결과의 원래 신호의 Magnitude를 보면 알 수 있듯이, 중심에서 많이 멀어지면 아예 Magnitude가 0이 됨을 볼 수 있는데, 즉 많이 높은 주파수 신호는 가지지 않음을 알 수 있다. 이에 따라, 비록 HPF를 사용하였지만, Cutoff를 상당히 높은 값을 채택하였기 때문에, 전체적으로 다 차단 시켜 Magnitude가 전체적으로 낮아짐을 볼수 있었다.

이는 필터링한 신호를 들었을 때, LPF의 경우, 사실 그다지 뚜렷한 차이는 보이지 않았지만, 약간 음의 높이가 낮아진듯한 느낌이 들었는데, 이는 저 주파수대역이 강조되었기 때문이라고 볼 수 있다.

HPF의 경우, 결과 Magnitude를 봐도 알수 있듯이., 전체적으로 Magnitude가 줄어들어, 음향의 크기 자체가 줄어들어 소리가 매우 작았다.

4-2

3 point의 경우, 결과 파형을 보아도, 직접 신호를 들어봐도 크게 원본 신호와 큰 차이를 느끼지 못하였다. 하지만 7 point의 경우, 확실히 3 point 의 결과와는 비교했을 때 (특히 lpf 의 경우), Filter의 변화로 인하여 음의 높낮이 나 크기가 확실히 변한 것을 느낄수 있었다.

7 point과 3 point 보다는 더 확실히, 필터링 해 줌을 알 수 있었다.

4-3

이번에 경우 대략적으로 2가지 방법을 이용하여 LPF 와 HPF를 구현하였다.

일반적으로 LPF 에서 HPF를 구현할 때, 주파수 상으로 보았을때는 단순히 Shift 만 시키면 LPF에서 HPF로 바꿀수 있다. 이번 Simulation의 경우, pi 만큼 frequency shift를 시켰다.

이는 Shifting Property에 의하여 주파수 상에서의 shift는 시간축 상에서는 exponential의 곱으로 나타내줄수 있다.

Frequency Shift	$e^{j\gamma n}x[n] \xleftrightarrow{DTFT} X(e^{j(\omega-\gamma)})$

이를 이용하여. 시간축 상에서의 lpf에 $e^{j\pi n}$ 을 곱하면 되는데 이는 $(-1)^n$ 이다.

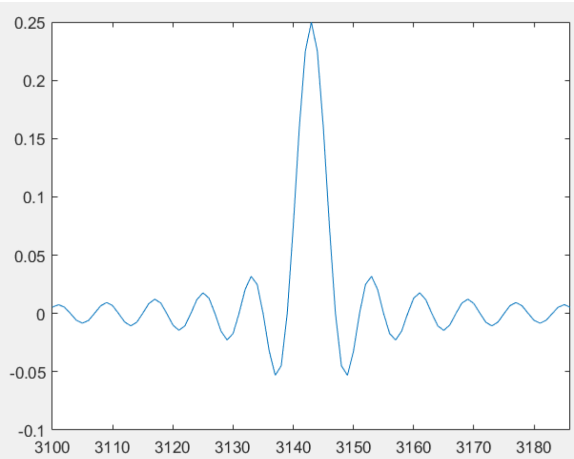
$$\begin{aligned} h_{iH}[n] &= e^{-j\pi n} \cdot h_{iL}[n] \\ &= (e^{-j\pi})^n \cdot h_{iL}[n] \\ &= (-1)^n \cdot h_{iL}[n] \end{aligned}$$

위를 바탕으로 matlab 에서 구현하면, 아래와 같으며 이때 n 말고 n+1를 넣은 이유는 중심축에서의 홀 짝을 맞춰주기 위함이다.

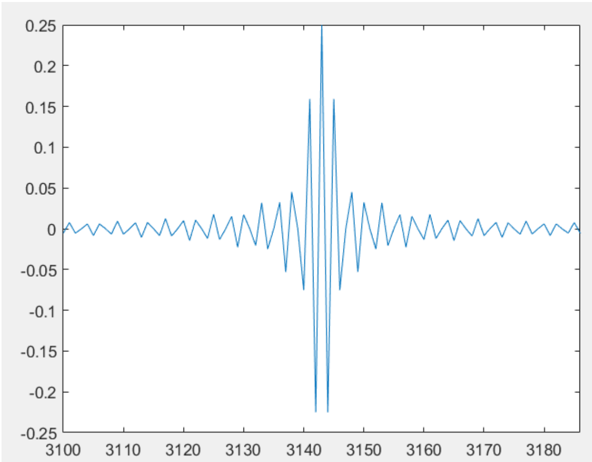
```
%{
c=[1:1:l];
hh=real(hl).*((-1).^(c+1));
%}
```

식들을 보면, 시간축에서의 high pass filter가 더 lpf보다 flunctuate함을 알수 있으며 이는 1번에서 첨부한 결과를 봐도 알수 있다.

LPF

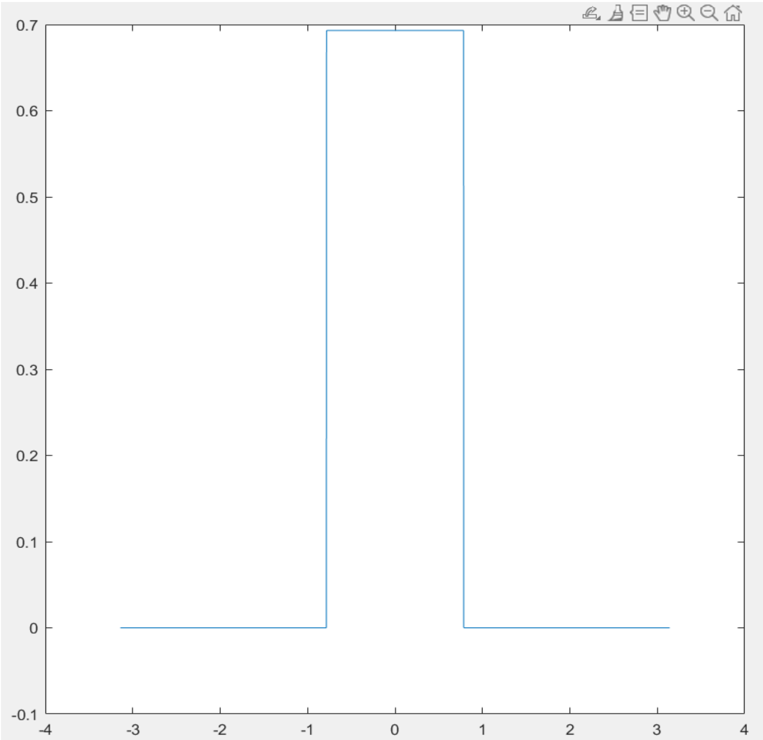


HPF



이 방법외에도, Matlab 에서 fftshift 나 ifftshift 을 시키면 0~2pi 파형이 -pi~pi 파형으로 된다는 점을 이용하여서 LPF 와 HPF 파형을 구 할수 있다.

맨처음에 구현한 필터의 파형은 아래와 같다.



이는 주파수 축을 어떻게 보냐에 따라, HPF 가 될수도 있고 LPF 가 될수도 있다.
주파수 축을 0~2pi 로 보면. 위의 파형은 HPF 가 되고.
-pi~pi 로 보면 위의 파형은 LPF 가된다.
이를 바탕으로 위의 그래프를 바로 ifft 하면 시간 축상에서의 HPF 가 나올 것임을 알 수 있다.
이 방법을 이용하여 HPF 를 구현할수 있으며. 이후 fftshift 나 ifftshift 를 사용하여 축을 변경시키면 LPF 로도 구현할수 있다.

Appendix

project2-1

```
k = [-pi:0.001:pi];
f=1*(k>=-pi/4 & k<=pi/4);
% iff시 0 2pi의 범위로 f를 보고 iff가 될것이다. 따라서 iff시 time domain 상에서는
% hpf가 나올것이다.
hh=ifft(f);


HL=fft(real(hh));% fft결과 -pi~pi의 LPF가 구현
HLS=fftshift(HL) %다시 fft를 한후 shift 해주었기때문에 0 ~ 2pi의 LPF가 구현

% 0~2pi 범위에서의 HL를 다시 ifft 하여 0 2pi 범위의 time domain 의 lpf 구현
hl=ifft(real(HLS));
% shift를 통하여 -pi ~ pi 의 범위로 time domain 의 lpf로 바꾸고 window 진행
hl=ifftshift(hl)


% 1-1 PLOT

% 1. -pi~pi 파형을 가지는 주파수축 LPF
% plot(k,log(abs(HL+1)))

% 2. -pi~pi 파형의 시간축 LPF
%  plot(hl)
%  xlim([3100,3186])

l=length(hl);
a=[1:1:l];
b=[1:1:l];

Winone=1*(a>=l/2 & a<=l/2+2);
Wintwo=1*(b>=l/2-2 & b<=l/2+4);

hl3=hl.*Winone;
hl3=real(hl3);
% 1-2 PLOT 3 point 시간축
% plot(real(hl3)) %hl3[n] 파형 plot
% xlim([3140,3146])
nhl3=hl3/abs(sum(hl3)); % normalise 시켜준다

hl7=hl.*Wintwo;
% 1-2 PLOT 7 point 시간축
% plot(real(hl7)) %hl7[n] 파형 plot
% xlim([3138,3148])
hl7=real(hl7);
```

```
hl7=hl.*Wintwo;
% 1-2 PLOT 7 point 시간축
% plot(real(hl7)) %hl7[n] 파형 plot
% xlim([3138,3148])
hl7=real(hl7);
nhl7=hl7/abs(sum(hl7));


HL3=fft(hl3,l);
HL7=fft(hl7,l);
HL3=fftshift(HL3);
HL7=fftshift(HL7);
% 1-2 PLOT  3, 7 point 주파수축
% plot(log(abs(HL3)+1))  % HL3(w) Magnitude
% plot(log(abs(HL7)+1))  % HL7(w) Magnitude
```

project 2-2

```
k = [-pi:0.001:pi];
f=1*(k>=-pi/4 & k<=pi/4);
% 위의 f 가 0~2pi 범위에서의  frequency domain의 hpf 임으로 그대로 hh 해줘도 무방하다
hs=ifft(f);
hh=real(ifftshift(hs));
l=length(hh);
HH=fft(hs);
HH=fftshift(HH);

%혹은 1에서 구한 time domain low pass filter 에서 hh=real(hl).*((-1).^(c+1))의 식을
%사용하여 해줘도 똑같이 결과가 나옴을 확인 할수 있다.

%{
c=[1:1:l];
hh=real(hl).*((-1).^(c+1));
%}

%위의 주석안의 방법으로 만든 hh와 맨 앞에서 ifft로 구한 hh의 결과는 같다


% 2-1 PLOT
% 1. -pi~pi 파형을 가지는 주파수축 LPF
%  plot(k,log(abs(HH+1)))

% 2. -pi~pi 파형의 시간축 LPF
%  plot(hh)
%  xlim([3100,3186])


a=[1:1:l];
b=[1:1:l];


Winone=1*(a>=l/2 & a<=l/2+2);
Wintwo=1*(b>=l/2-2 & b<=l/2+4);



hh3=hh.*Winone;
hh3=real(hh3);
% 2-2 PLOT 3 point 시간축
%  plot(real(hh3))
```

```
hh3=hh.*Winone;
hh3=real(hh3);
% 2-2 PLOT 3 point 시간축
%  plot(real(hh3))
%  xlim([3140,3146])

hh7=hh.*Wintwo;
hh7=real(hh7);

% 2-2 PLOT 7 point 시간축
%  plot(real(hh7))
%  xlim([3138,3148])

HH3=fft(hh3,l);
HH7=fft(hh7,l);
HH3=fftshift(HH3)
HH7=fftshift(HH7)
% 2-2 PLOT  3, 7 point 주파수축
%  plot(log(abs(HH3)+1))  % HL3(w) Magnitude
%  plot(log(abs(HH7)+1))  % HL7(w) Magnitude
```

project 3-3

```
[y, Fs] = audioread('input.wav');

y1=y.';
YF=fft(y1);
YF=fftshift(YF);
hlf3=conv(nhl3,y1);
hlf7=conv(nhl7,y1);


HLF3=fft(hlf3);
HLF3=fftshift(HLF3);
HLF7=fft(hlf7);
HLF7=fftshift(HLF7);

%{
% 3 Point의 LPF CONVOLUTION 결과
plot(log(abs(YF)+1))
hold on
plot(log(abs(HLF3)+1))
%}


%{
% 7 Point의 LPF CONVOLUTION 결과
plot(log(abs(YF)+1))
hold on
plot(log(abs(HLF7)+1))
%}

hhf3=conv(y1,hh3);
hhf7=conv(y1,hh7);

HHF3=fft(hhf3);
HHF3=fftshift(HHF3);
HHF7=fft(hhf7);
HHF7=fftshift(HHF7);

%{
% 3 Point의 HPF CONVOLUTION 결과
plot(log(abs(YF)+1))
hold on
plot(log(abs(HHF3)+1))
```

```
%{
% 3 Point의 HPF CONVOLUTION 결과
plot(log(abs(YF)+1))
hold on
plot(log(abs(HHF3)+1))
%}

%{
% 7 Point의 HPF CONVOLUTION 결과
plot(log(abs(YF)+1))
hold on
plot(log(abs(HHF7)+1))
%}

%  sound(hlf7,Fs)
```