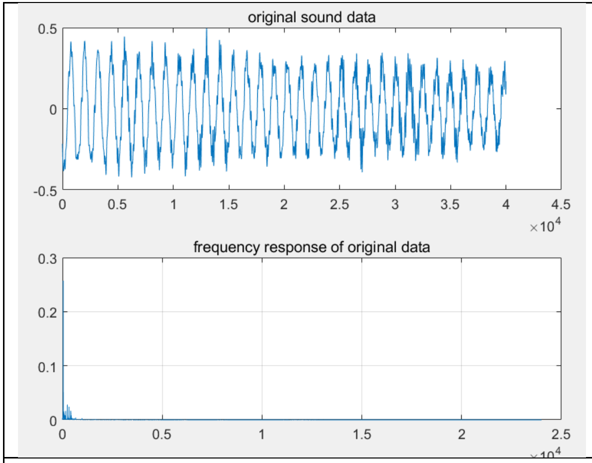
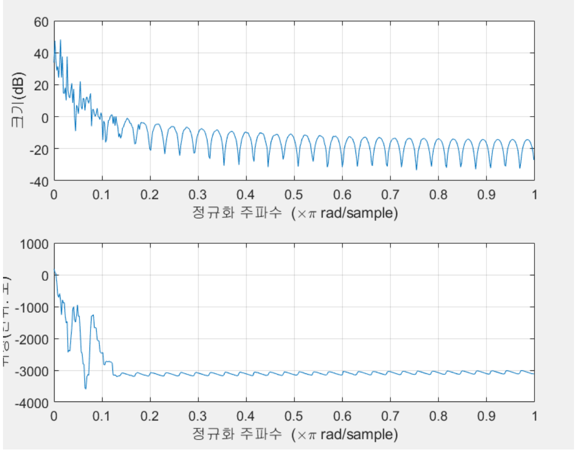
DSP Take Home

융합공학부 20153865 김민석

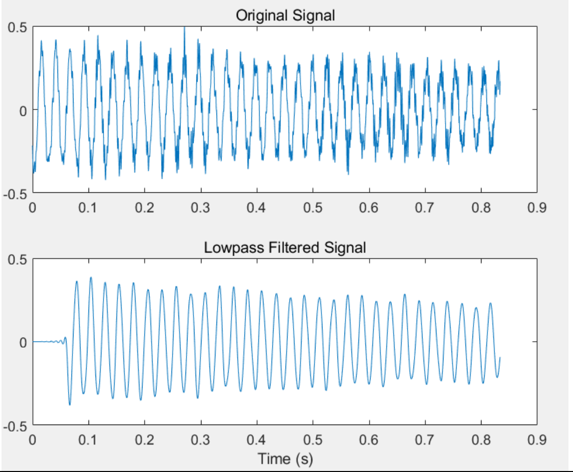
1. Results



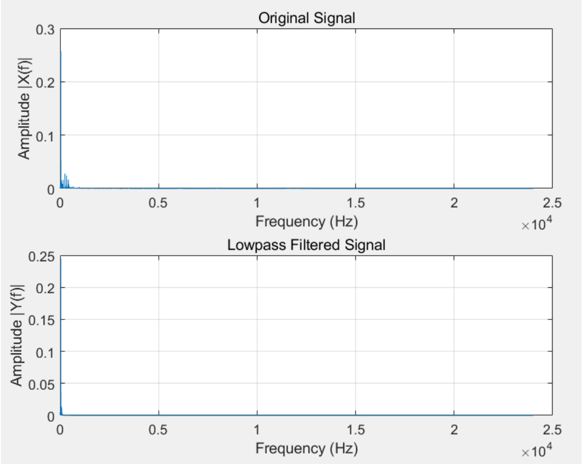
Original sound



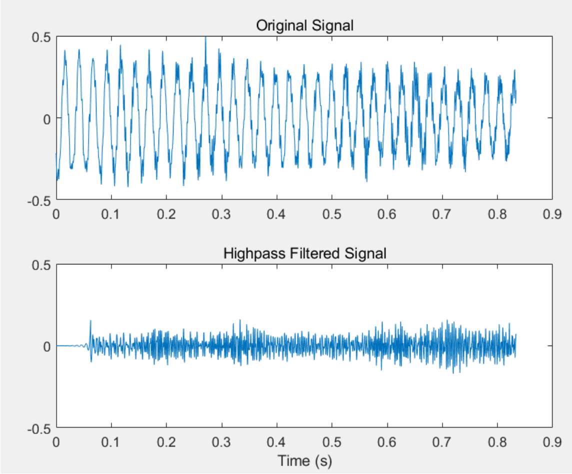
Digital Frequency Response



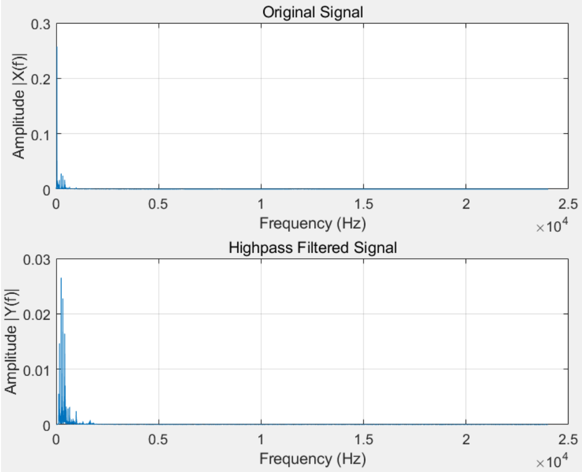
Time domain (Lowpass filter)



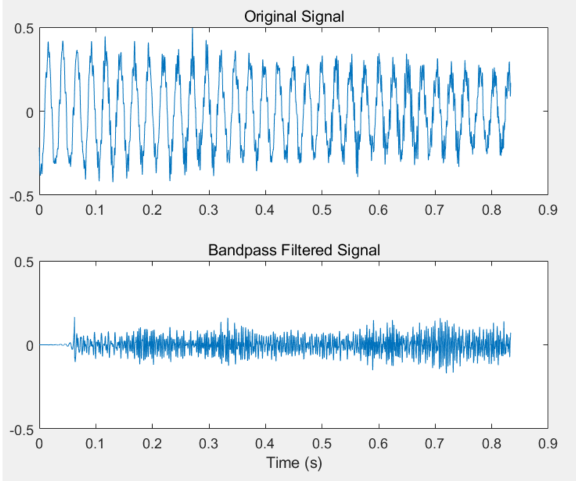
Frequency domain (Lowpass filter)



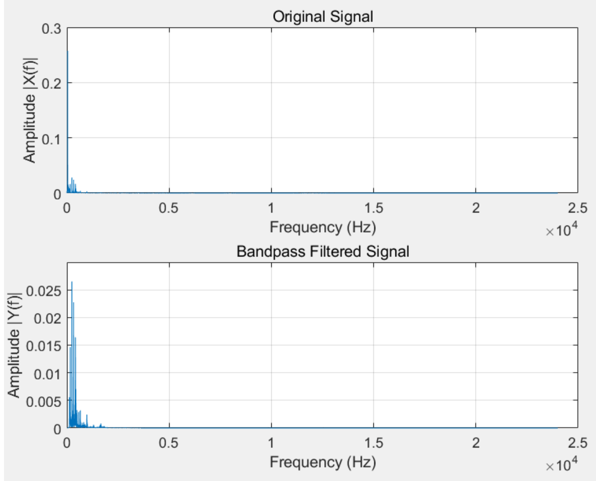
Time domain (Highpass filter)



Frequency domain (Highpass filter)



Time domain (Bandpass filter)



Frequency domain (Bandpass filter)

2. Method / Discussion

=> Alan walker 의 'Legends never die'의 ncs 음악으로 필터링을 진행하였으며, low pass, high pass, band pass 필터를 각각 적용했다. 각 필터의 cutoff 지점을 정하기 위해서, Digital Frequency Response 를 확인했으며, X 가 0.005 인 지점으로 filter 를 만들었다. Hanning 윈도우 필터를 적용했으며, low pass filter 가 가장 소리를 잘 필터링 해주었다. 다른 filter 들은 time domain 을 보았을 때, sin 곡선형태까지는 나오지 않았다.

아마 origianl sound 에서 저주파에서 음의 변화도가 완만해서, 이러한 결과를 가져온것이 아닐까 추측한다.

그리고, 각 필터의 계수를 6001 개를 사용했으며, delay 가 생기게 되었다.

3. CODE

```
%% read mp3 file


[out fs] = audioread('LEGENDSNEVER DIE.mp3');

N = 40000;      % data cut

k = 1200000;

data = out(40000:80000,1);

data2 = out(40000+k:200000+k,1);

%sound(data2,fs);

%sound(data/max(abs(data2)),fs);   % play

filename = 'C:\Users\wonde\Desktop\original.mp4';

audiowrite(filename,data2,fs);


%% Spectrum (one-sided)

N=length(data);

f=[0:N/2]*fs/N;

Axk=2*abs(fft(data))/N;Axk(1)=Axk(1)/2;

figure(1);

subplot(2,1,1);

plot(data);

title('original sound data');

subplot(2,1,2);

plot(f,Axk(1:N/2+1));grid

title('frequency response of original data');


freqz(data,1);


%% Filtering

1. lowpass (Ex 7.8)


figure(3);

blo = fir1(6001,0.005,hann(6002));

outlo = filter(blo,1,data);
```

```
subplot(2,1,1);

t = (0:length(data)-1)/fs;

plot(t,data);

title('Original Signal');

ys = ylim;;

subplot(2,1,2);

plot(t, outlo);

title('Lowpass Filtered Signal');

xlabel('Time (s)');

ylim(ys);


figure(4);

subplot(2,1,2); plot(f,Axk(1:N/2+1));

xlabel('Frequency (Hz)'); ylabel('Amplitude |X(f)|');grid;

subplot(2,1,1); plot(f,Axk(1:N/2+1));

title('Original Signal');

xlabel('Frequency (Hz)'); ylabel('Amplitude |X(f)|');grid;

Ayk = 2*abs(fft(outlo))/N;Ayk(1) = Ayk(1)/2;

subplot(2,1,2);plot(f,Ayk(1:N/2+1));

title('Lowpass Filtered Signal');

xlabel('Frequency (Hz)'); ylabel('Amplitude |Y(f)|');grid;


sound(outlo)


% 2. highpass (Ex 7.9)


figure(5);

blo = fir1(6000,0.005,'high',hann(6001));

outlo = filter(blo,1,data);

subplot(2,1,1);

t = (0:length(data)-1)/fs;

plot(t,data);
```

```
title('Original Signal');
ys = ylim;;
subplot(2,1,2);
plot(t, outlo);
title('Highpass Filtered Signal');
xlabel('Time (s)');
ylim(ys);

figure(6);
subplot(2,1,2); plot(f,Axk(1:N/2+1));
xlabel('Frequency (Hz)'); ylabel('Amplitude |X(f)|');grid;
subplot(2,1,1); plot(f,Axk(1:N/2+1));
title('Original Signal');
xlabel('Frequency (Hz)'); ylabel('Amplitude |X(f)|');grid;
Ayk = 2*abs(fft(outlo))/N;Ayk(1) = Ayk(1)/2;
subplot(2,1,2);plot(f,Ayk(1:N/2+1));
title('Highpass Filtered Signal');
xlabel('Frequency (Hz)'); ylabel('Amplitude |Y(f)|');grid;
```

3. bandpass (Ex 7.10)

```
figure(7);
blo = fir1(6001,[0.005 0.15],'bandpass',hann(6002));
outlo = filter(blo,1,data);
subplot(2,1,1);
t = (0:length(data)-1)/fs;
plot(t,data);
title('Original Signal');
ys = ylim;;
subplot(2,1,2);
plot(t, outlo);
title('Bandpass Filtered Signal');
xlabel('Time (s)');
ylim(ys);

figure(8);
subplot(2,1,2); plot(f,Axk(1:N/2+1));
xlabel('Frequency (Hz)'); ylabel('Amplitude |X(f)|');grid;
subplot(2,1,1); plot(f,Axk(1:N/2+1));
title('Original Signal');
xlabel('Frequency (Hz)'); ylabel('Amplitude |X(f)|');grid;
Ayk = 2*abs(fft(outlo))/N;Ayk(1) = Ayk(1)/2;
subplot(2,1,2);plot(f,Ayk(1:N/2+1));
title('Bandpass Filtered Signal');
xlabel('Frequency (Hz)'); ylabel('Amplitude |Y(f)|');grid;

sound(outlo)

%% Echo
% Prob 6.32
% FIR
R = 7000;
%Difference equation: y[n]=x[n]+ax[n-R]
%Equivalently, by the transfer function H(z)=1+az^(-R)
num=[1,zeros(1,R-1),0.8];
den=[1];
%The output of the FIR filter is computed using the function 'filter'
d1 = filter(num,den,data2);
soundsc(d1,fs);
filename = 'C:\Users\wonde\Desktop\FIR.mp4';
audiowrite(filename,d1,fs);
```

```
% IIR

num=[0,zeros(1,7000-1),1]

den=[1,zeros(1,7000-1),-0.8];

d1=filter(num,den,data2);

soundsc(d1,fs);

filename = 'C:\Users\wonde\Desktop\IIR.mp4';

audiowrite(filename,d1,fs);
```