Assignment 1

Name: Meng Nan

UID: 3030036376

(i) Properties:

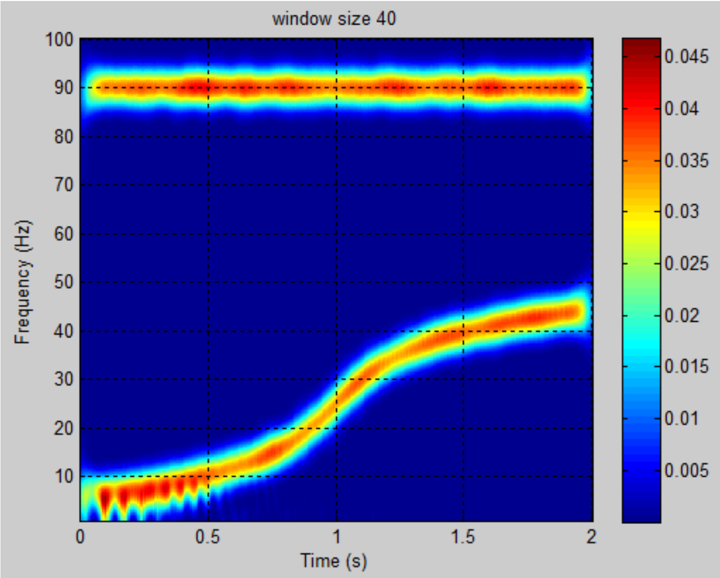
1. The signal contains a sinusoid of amplitude of 1.5 and another sinusoid which has higher frequency.

2. The frequency of sinusoid whose amplitude is 1.5 has become higher and higher.

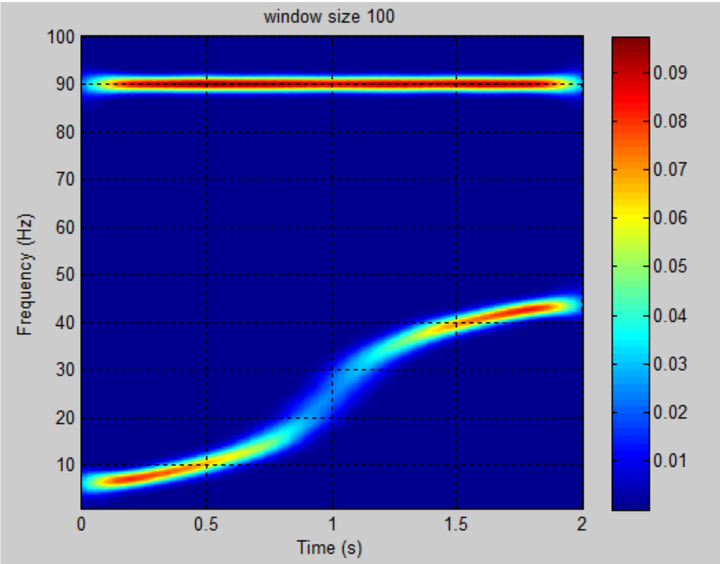
The frequency of the signal at t = 0, 1, and 2 s:

1. t = 0s low frequency = 6.0 Hz high frequency = 90Hz
2. t = 1s low frequency = 25.0 Hz high frequency = 90Hz
3. t = 2s low frequency = 44.0 Hz high frequency = 90Hz

(ii) The results of recalculating the spectrogram



The spectrogram recalculated with the window size of length 40. (window length = 0.2)



The spectrogram recalculated with the window size of length 100. (window length = 0.5)

The smaller the window size is, the higher resolution of the time and frequency of spectrogram is. The second plot with the window size of length 100 has lower resolution than the plot with the window size of length 40. (For the window size determines of the resolution. For a window size of length N and a sampling frequency of Fs, your frequency resolution is Fs/N.)

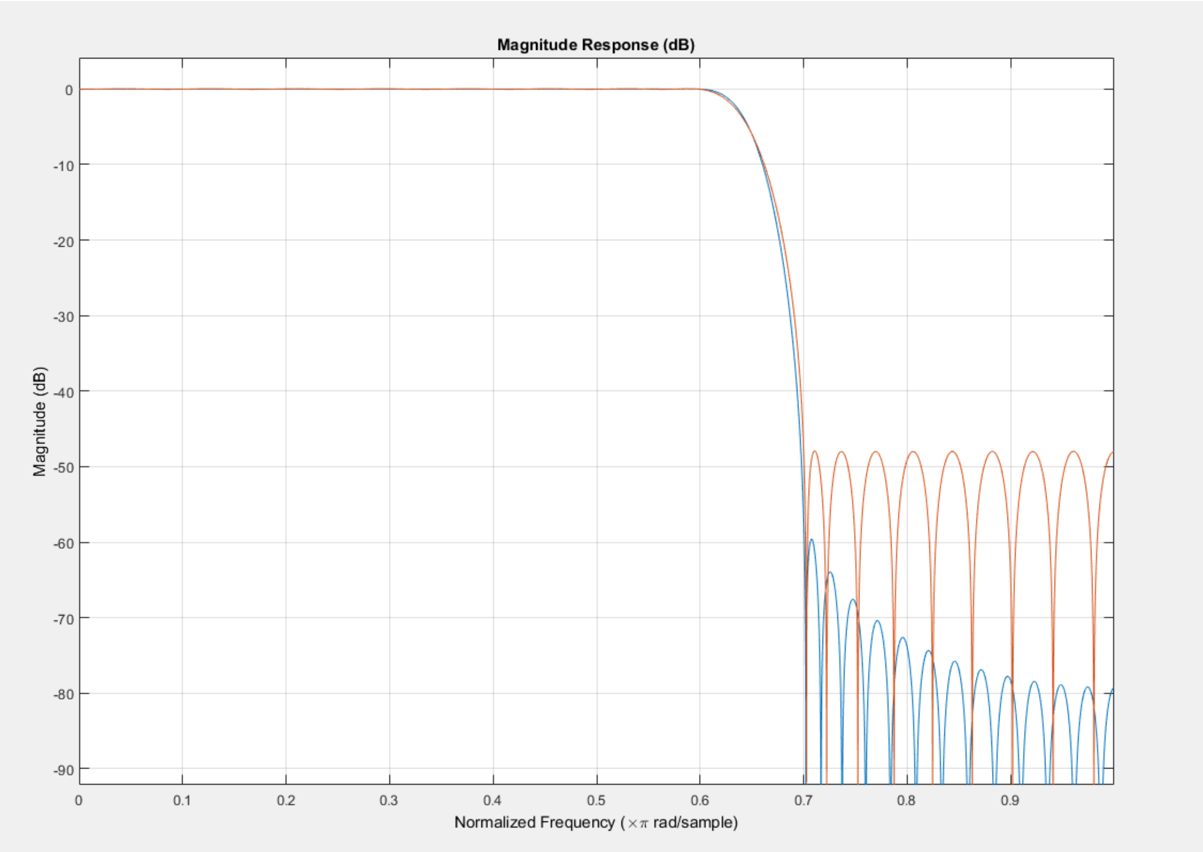
(iii) The estimation of the filter is:

$$M = \frac{A-8}{2.285\Delta\omega} = \frac{-20log_{10}\delta - 8}{2.285\times(\omega_s-\omega_p)} = \frac{-20log_{10}0.001-8}{2.285\times(0.7\pi-0.6\pi)} = 72.4381 \approx 73$$

(iv) The estimation of the filter is:

$$M = \frac{-10log_{10}(\delta_1\delta_2)-13}{2.324\Delta\omega} = \frac{-10log_{10}(0.01\times0.001)-13}{2.324\times(0.7\pi-0.6\pi)} = 50.6776 \approx 51$$

(v) Plot of the frequency response of filters designed in (iii) and (iv):



Note:

The **blue** line represent the frequency response of filter using **Kasier Windows method**

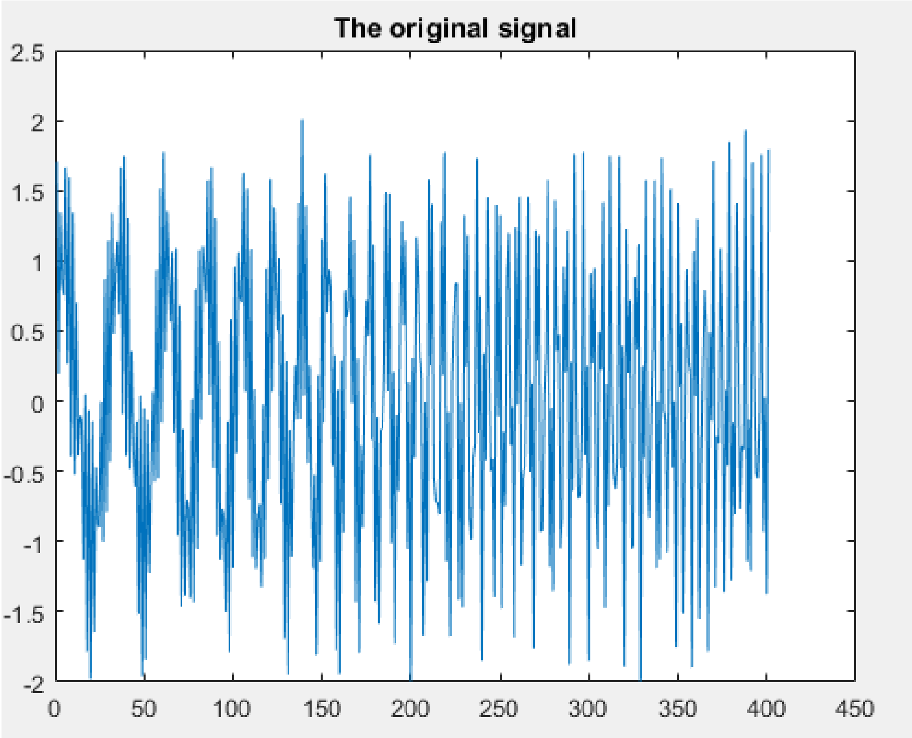
The **orange** line represent the frequency response of filter using **Parks-McClellan method**

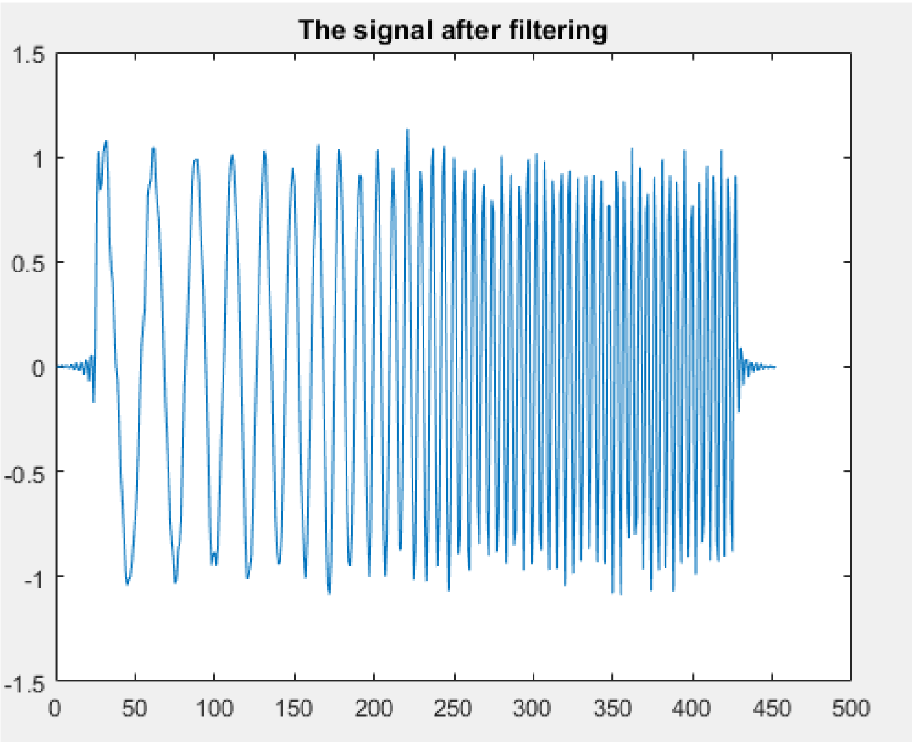
From the plot we can see that the blue one is much better, since it's magnitude response decays faster in the high frequency part. Thus the filtering effect of the blue one is much better than the orange one.

(vi) Comparing the performance of these two filters, we note that the lowpass FIR filter design requires filter order M = 72 using the Kaiser window method, while the Parks-McClellan design method only requires the filter order M = 50 to meet the same filter specifications. The lowpass FIR filter order has up to 30.6% less computational complexity using the Parks-McClellan equiripple design method.

Window method can only produce approximately equal maximum errors in the passband and stopband. However, the Parks-McClellan design method can weight the error differently.

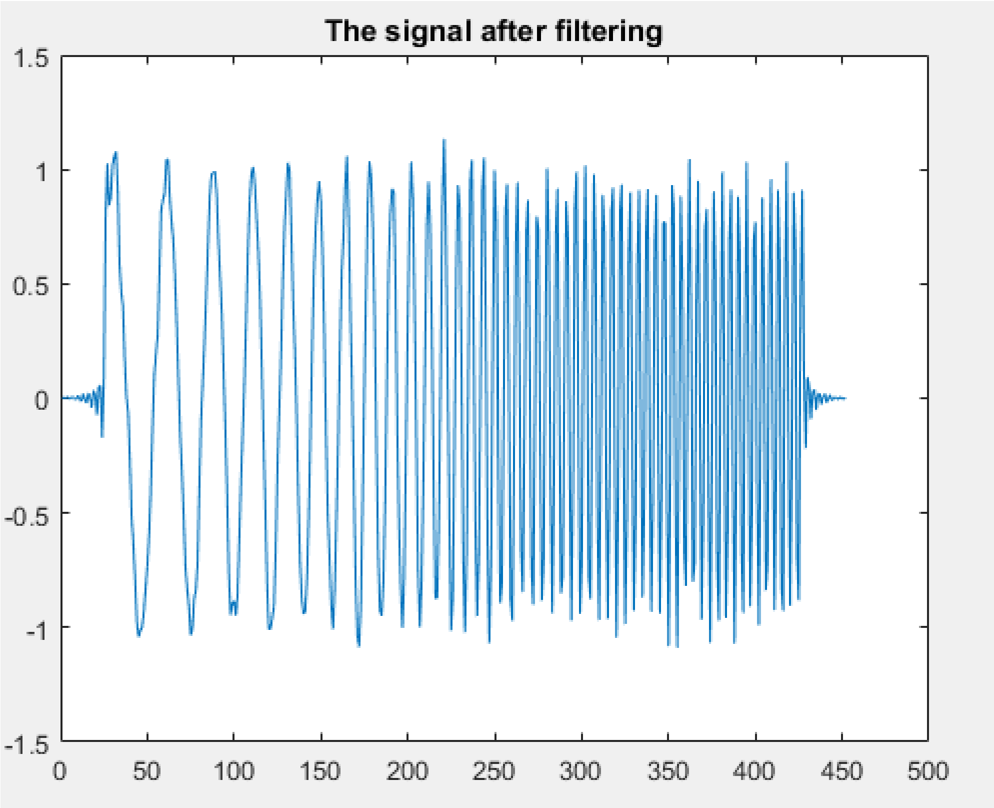
(vii) The plot of the original signal and filtered signal using Parks-McClellan method (by linear convolution)



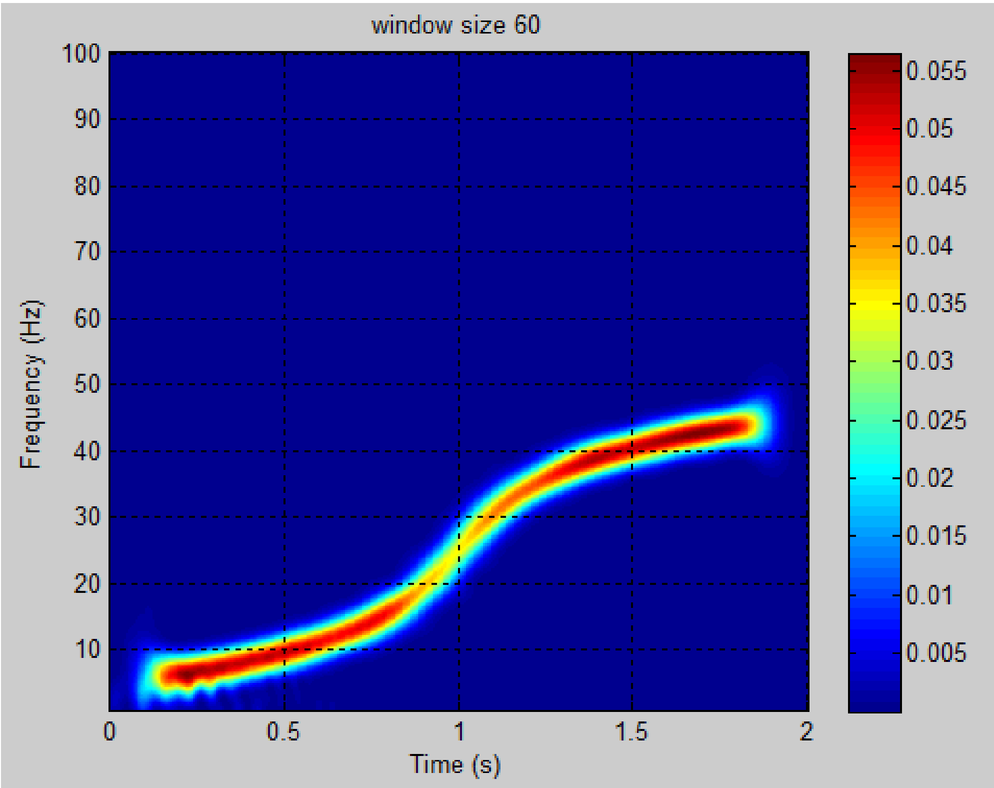


The output signal of filter designed by Parks-McClellan method is the same as the signal after linear convolution.

(viii) The plot of the filtered signal and its spectrogram



The filtered signal



The spectrogram of filtered signal

(ix) The spectrogram of the original signal has both a lower part (the curve line) and a higher part (the straight line), while the spectrogram of the filtered signal has only the lower part (the curve line) and its higher part has been filtered out.

Appendix

ass1_main.m

```
clear all;
close all;

%% ======================================================== %%
% ELEC 2204 Digital Signal Processing
% Assignment 1 (Ver: Oct. 22, 2015)
% Prof. S. C. Chan, Dr. H. C. Wu
% Special thanks to Dr. Z. G. Zhang for providing the signals and codes for
analysis.


%% Load the synthetic signal
load Signal.mat
Fs=200; %% Sampling Frequency
Ts=1/Fs; %% Sampling Period
t=0:Ts:2;
t=t';
%% Plot the signal
figure
plot(t,x)
xlabel('Time (s)')
ylabel('Amplitude')
ylim([-1.5 1.5])

%% perform STFT
win_len=0.3; %0.5 or 0.2 %% default setting for window length;
winsize =win_len*Fs; % actual window size for STFT;
nfft = 1024;    % # FFT points
[P, f] = stft(x, winsize, nfft, Fs);

%% display spectrogram
figure
imagesc(t,f,P)
colorbar
xlabel('Time (s)')
ylabel('Frequency (Hz)')
axis xy
grid on
set(gca,'ylim',[1 100]) % set the limits of frequency in the plot
title(['window size ' num2str(winsize)]);
```

ass1.m

```
%% parameter

omiga_p = 0.6 * pi;
omiga_s = 0.7 * pi;
delta1 = 0.01;
delta2 = 0.001;
```

```
domiga = omiga_s - omiga_p;
delta = min(delta1,delta2);

A = -20*(log10(delta));

beta = [];
if A > 50
    beta = 0.1102 * (A-8.7);
elseif (A >= 21 && A <= 50)
    beta = 0.5842 * (A - 21)^0.4 + 0.07886 * (A - 21);
else
    beta = 0;
end

M = (A - 8)/(2.285*domiga);

M_pc = (-10*log10(delta1*delta2)-13)/(2.324*domiga);

%% (v)
n1 = ceil(M);
Wn = (omiga_p + omiga_s)/(2*pi);% 0 <= Wn <= 1
b1 = fir1(n1,Wn,kaiser(n1+1,beta));
fvtool(b1,1)

n2 = ceil(M_pc);
f = [0, omiga_p/pi, omiga_s/pi, 1];
m = [1 1 0 0];
b2 = firpm(n2,f,m);

fvtool(b1,1,b2,1);% b1 in blue; b2 in orange

%% (vii)
result1 = filter(b2,1,[x ; zeros(length(b2)+length(x)-1-length(x), 1)])
result2 = conv(b2,x)

all(abs(result1 - result2) < 0.001)

% plot result
plot(x);
title('The original signal');
figure;
plot(result1);
title ('The signal after filtering');

%% (viii)

% %% perform STFT
win_len=0.3; %% default setting for window length;
winsize =win_len*Fs; % actual window size for STFT;
nfft = 1024;    % # FFT points
[P, f] = stft(result1, winsize, nfft, Fs);

% %% display spectrogram
figure
imagesc(t,f,P)
colorbar
xlabel('Time (s)')
```

```
ylabel('Frequency (Hz)')
axis xy
grid on
set(gca,'ylim',[1 100]) % set the limits of frequency in the plot
title(['window size ' num2str(winsize)]);
```