Digital Signal Processing HW10 MATLAB Part

Name: Jingyang Zhang
Net ID: jz2807

1. **Speech filtering exercise, same as in HW 9, but this time design an FIR filter using Windowing. You should try a few different windows. Compare to your solutions with different filters obtained in previous homework, both by listening to the filtered signals as well as spectrum plots.**

Solution:

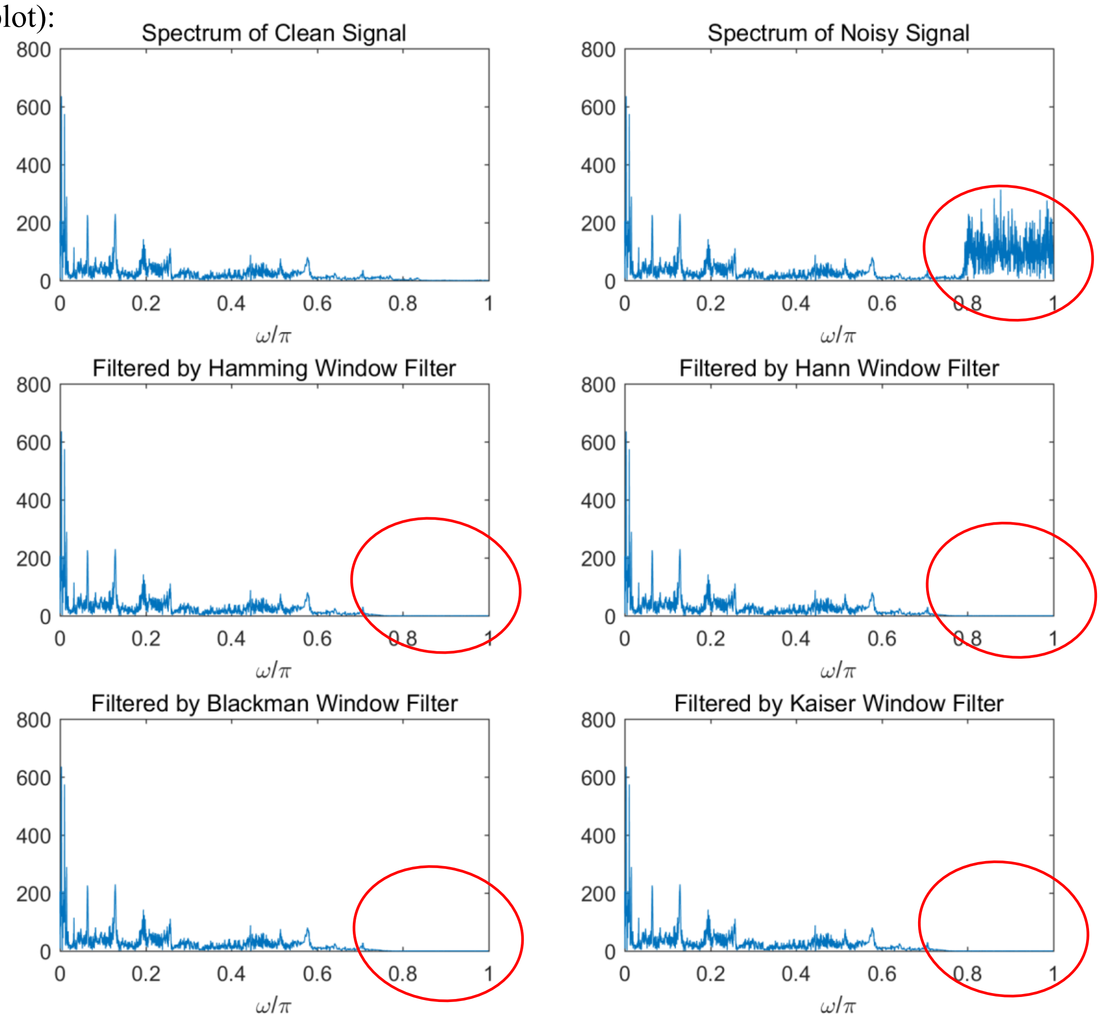
.m file(s): jyz_HW10_1.m

Code:

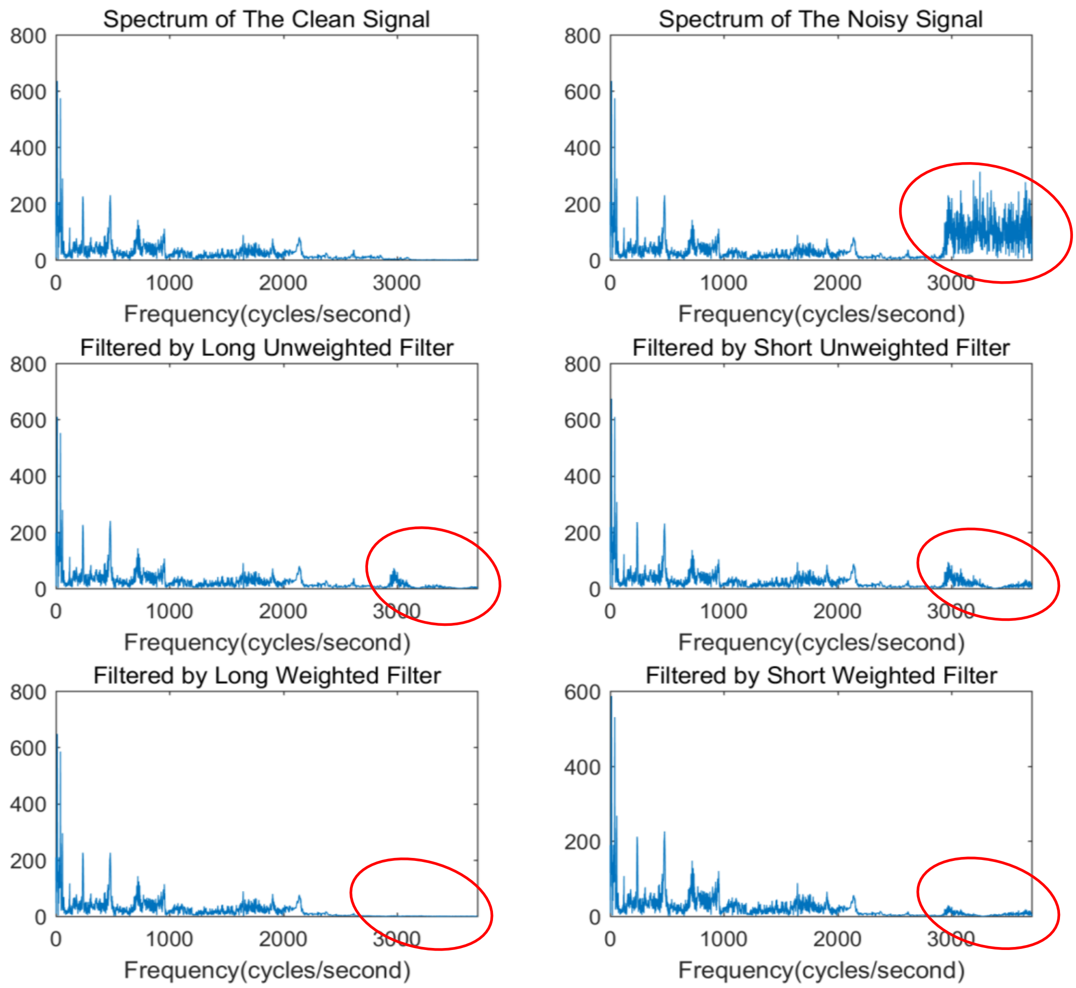
```
close all
clear

%Load signals and basic processing ↓
load NoisySpeech.txt
load mtlb
Noi = NoisySpeech;
Noif = fft(Noi, 4096);
mtlbf = fft(mtlb, 4096);
Ap = 0.05;                 % passband attenuation
As = 60;                   % stopband attenuation
delp = 1 - 10^(-Ap/20);     % passband deviation
dels = 10^(-As/20);        % stopband deviation
L = 4096;
k = 0 : L-1;
w = k*2*pi/L;
W = exp(1j*2*pi/L);
wp = (2500/(Fs/2))*pi; fp = wp/pi;%pass_band
ws = (2900/(Fs/2))*pi; fs = ws/pi;%stop_band
wc = (wp + ws) / 2; fc = wc/pi;   %cut-off frequency
%determine the length of the filter ↓
N = kaiserord([fp fs],[1 0],[delp dels]) + 1;
M = (N - 1) / 2;
n = 0 : N -1;

dCausal = wc / pi * sinc(wc / pi * (n - M));
hHamming = dCausal .* hamming(N)';
hHann = dCausal .* hann(N)';
hBlackman = dCausal .* blackman(N)';
beta = 0.1102 * (As - 8.7);
hKaiser = dCausal .* kaiser(N, beta)';

HHamming = fft(hHamming, L);AHamming = HHamming .*W.^(M*k);
HHann = fft(hHann, L);AHann = HHann .*W.^(M*k);
```

```
HBlackman = fft(hBlackman, L);ABlackman = HBlackman .*W.^(M*k);
HKaiser = fft(hKaiser, L);AKaiser = HKaiser .*W.^(M*k);

%filtering↓
YHamming = abs(Noif' .* AHamming);
YHann = abs(Noif' .* AHann);
YBlackman = abs(Noif' .* ABlackman);
YKaiser = abs(Noif' .* AKaiser);

subplot(3,2,1)
plot(k(1:2048)/2048, abs(mtlbf(1:2048)))
xlabel('\omega/\pi');title('Spectrum of Clean Signal')
subplot(3,2,2)
plot(k(1:2048)/2048, abs(Noif(1:2048)))
xlabel('\omega/\pi');title('Spectrum of Noisy Signal')
subplot(3,2,3)
plot(k(1:2048)/2048, YHamming(1:2048))
xlabel('\omega/\pi');title('Filtered by Hamming Window Filter')
subplot(3,2,4)
plot(k(1:2048)/2048, YHann(1:2048))
xlabel('\omega/\pi');title('Filtered by Hann Window Filter')
subplot(3,2,5)
plot(k(1:2048)/2048, YBlackman(1:2048))
xlabel('\omega/\pi');title('Filtered by Blackman Window Filter')
subplot(3,2,6)
plot(k(1:2048)/2048, YKaiser(1:2048))
xlabel('\omega/\pi');title('Filtered by Kaiser Window Filter')

a = zeros(1,N);
a(1) = 1;

yHamming = filter(hHamming, a, Noi);
yHann = filter(hHann, a, Noi);
yBlackman = filter(hBlackman, a, Noi);
yKaiser = filter(hKaiser, a, Noi);

disp('Press Enter to Play Next Audio(Hamming)')
pause
soundsc(yHamming, Fs);
disp('Press Enter to Play Next Audio(Hann)')
pause
soundsc(yHann, Fs);
disp('Press Enter to Play Next Audio(Blackman)')
pause
soundsc(yBlackman, Fs);
disp('Press Enter to Play Next Audio(Kaiser)')
pause
soundsc(yKaiser, Fs);
disp('Program Finished!')
```

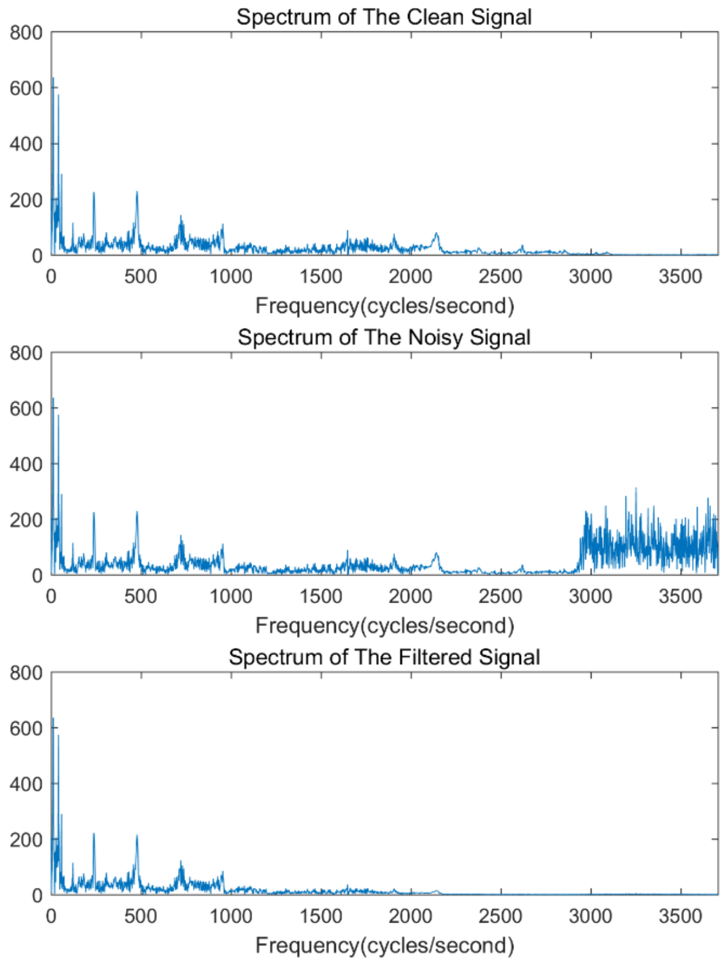
Result(plot):



In HW9:



In HW8:



Comments: In this assignment, all the 4 filters have done their job well, the noise has been annihilated while the intensity of signal remains almost the same. In HW9, only the long weighted filter has a good noise-removing effect, in the other 3 signals filtered by short weighted, long unweighted filter, short unweighted filters, there are still noise left. In HW8, the frequency band of the noise is almost annihilated, but there is some amplitude loss due to the filtering.

In general, after listening to the filtered signals as well as spectrum plots, it could be concluded that all these filters have good influence on the noise removing. The output signals sound considerably clean.

2. Consider a signal consisting of three sinusoidal components,

$$x(n) = A_1\sin(2\pi f_1 n) + A_2\sin(2\pi f_2 n) + A_3\sin(2\pi f_3 n)$$

where the frequencies fi are unknown. On the class webpage, a 100-point signal (signal2.txt) of this form is available. Using appropriate windows, estimate the three frequencies f_i.

Solution:

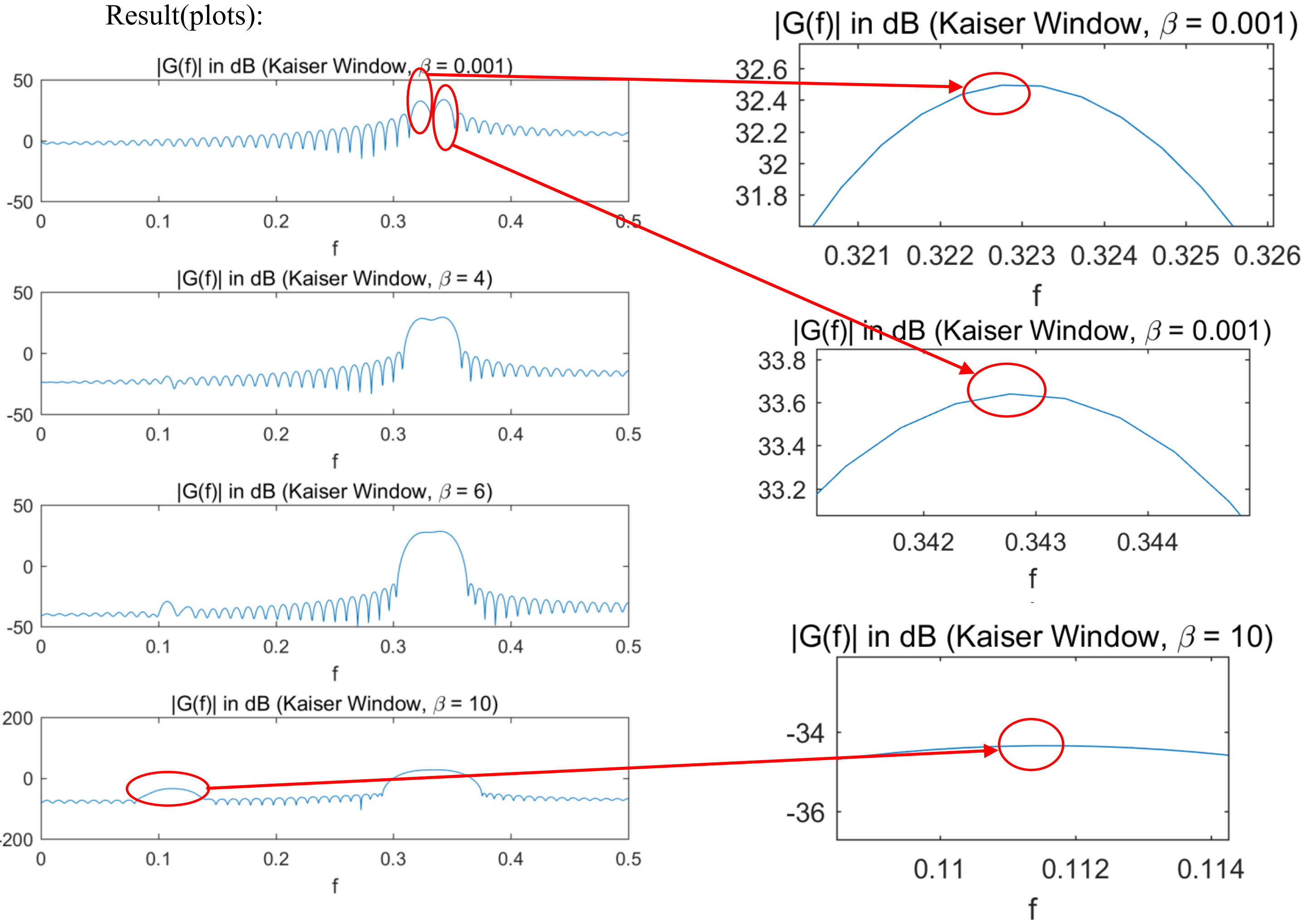
.m file(s): jyz_HW10_2

Code:

```
close all
clear

load signal2.txt;
x = signal2;
N = length(x);
```

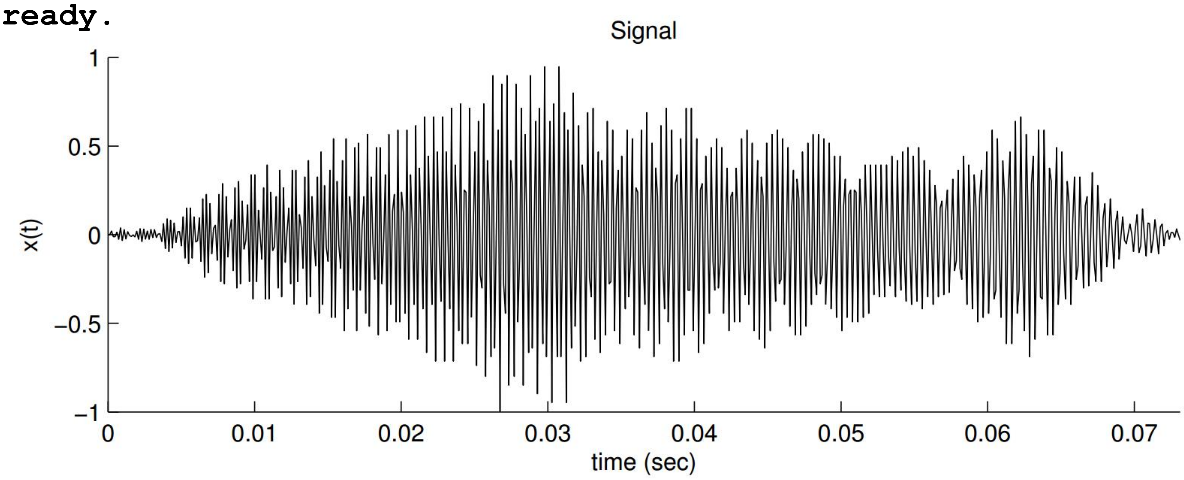
```
X = fft(x, 128);

g1 = x .* kaiser(N,.001);
g4 = x .* kaiser(N,4);
g6 = x .* kaiser(N,6);
g10 = x .* kaiser(N,10);
G1 = fft(g1, 2048);
G4 = fft(g4, 2048);
G6 = fft(g6, 2048);
G10 = fft(g10, 2048);
subplot(4,1,1)
plot(0:1/2048:1/2,20*log10(abs(G1(1:2048/2+1))))
title('|G(f)| in dB (Kaiser Window, \beta = 0.001)')
xlabel('f')
subplot(4,1,2)
plot(0:1/2048:1/2,20*log10(abs(G4(1:2048/2+1))))
title('|G(f)| in dB (Kaiser Window, \beta = 4)')
xlabel('f')
subplot(4,1,3)
plot(0:1/2048:1/2,20*log10(abs(G6(1:2048/2+1))))
title('|G(f)| in dB (Kaiser Window, \beta = 6)')
xlabel('f')
subplot(4,1,4)
plot(0:1/2048:1/2,20*log10(abs(G10(1:2048/2+1))))
title('|G(f)| in dB (Kaiser Window, \beta = 10)')
 xlabel('f')
```

Result(plots):



Comments: Use Kaiser window with different βs to design filters, after the signal is filtered, we can see that the first peak is small in amplitude, when β comes to 10, we can distinguish it from the plot. After zooming in the plot , we can determine $f_1 \approx 0.1118$. Similarly ,when β = 0.001, other 2 peaks is clear, we can get $f_2 \approx 0.3228$, $f_3 \approx 0.3428$.

3. Look at the Matlab documentation for the *specgram* function (use the help command). Make a spectrogram plot of the signal in the data file signal1.txt that was used in a previous problem. Try it with several different block lengths, amount of overlap, etc., until you get a spectrogram that best reveals the structure of the signal. Compare and contrast the spectrogram with the spectrum of this signal you got already.



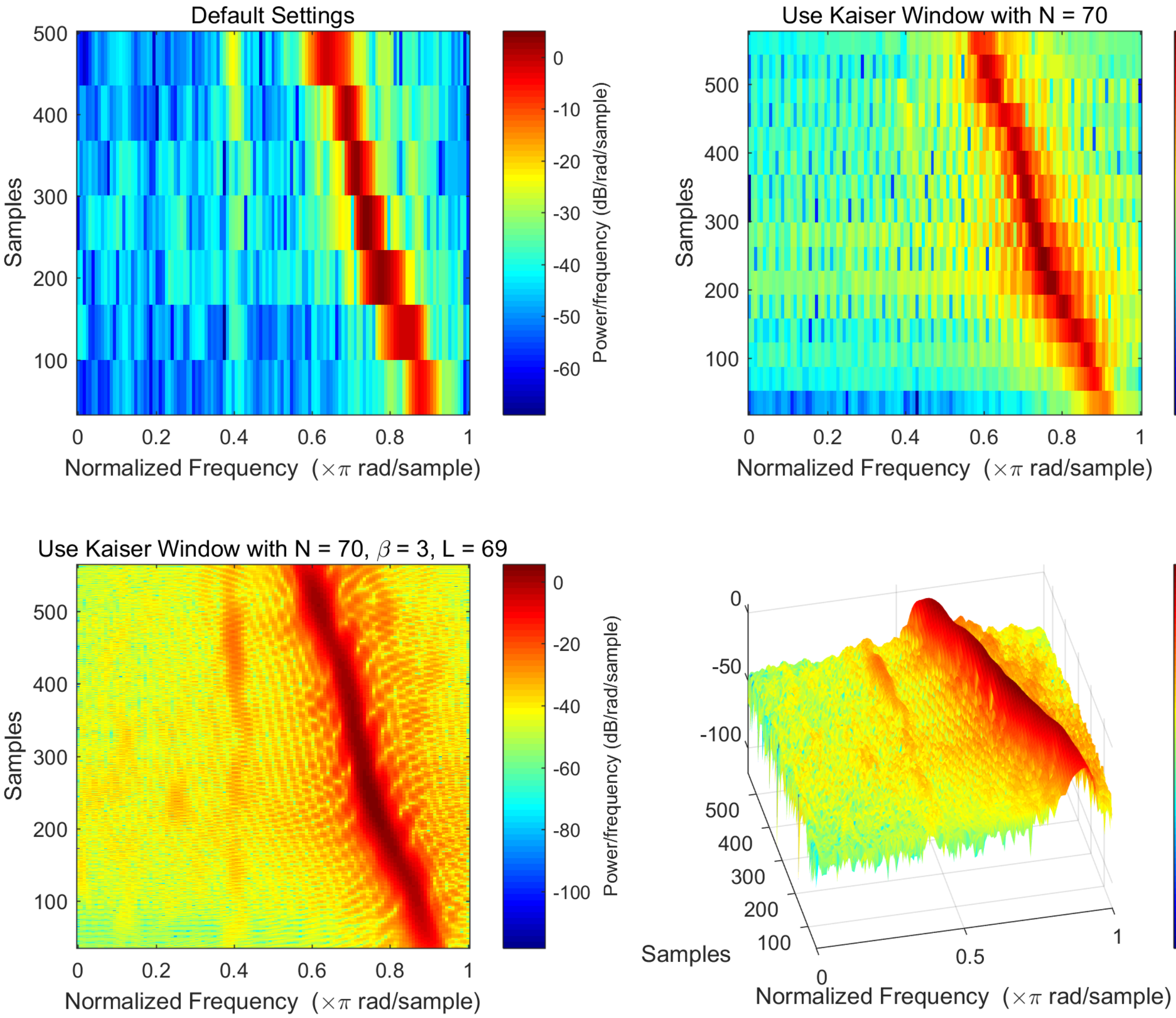
Solution:

.m file(s): jyz_HW10_3.m

Code:

```
close all
clear

load signal1.txt;
x = signal1;
subplot(2,2,1)
spectrogram(x)
title('Default Settings')
subplot(2,2,2)
spectrogram(x,kaiser(70))
title('Use Kaiser Window with N = 70')
subplot(2,2,3)
spectrogram(x,kaiser(70,3),69)
title('Use Kaiser Window with N = 70, \beta = 3, L = 69')
subplot(2,2,4)
spectrogram(x,kaiser(70,3),69)
title('3D View')
view(-13,47)
colormap jet
```

Result(Plots):



Comments: As we can see through these plots, with proper option is used, it is more and more convenient to reveals the structure of the signal. Properties also becomes more intuitive to us.