CA6 - Upsampling & Downsampling

Thursday, March 17, 2022 1:08 AM

1. Draw a block diagram of the five stage sample rate conversion system that converts from $F_x = 11025$ S/s to $F_y = 8000$ S/s.

2. Label the block diagram to indicate the sample rate at the output of each block in the system diagram.

3. Suppose the input signal contains a signal at normalized frequency $f_0 = 1000$ Hz. Label the block digram to indicate the frequency of this signal at the output of each block in the system diagram.

4. Write a Matlab function that performs one stage of sample rate conversion (SRC). The signature of the function should be:

```
[y,fp,fm,Fs] = src(x,fp,fm,U,D,Fs)
```

where x is the input signal, f_p is the pass band edge frequency for the input signal, f_{max} is the maximum frequency in the input signal, U and D are the upsampling and downsampling factors, F_s is the input sample rate, and y is the output signal. Note that f_p, f_m, F_s are all input and output variables. The output frequencies should be scaled D/U by the input frequencies and the output sample rate F_s should be scaled U/D by the input sample rate.

Your function should design and do all the processing for the sample rate conversion for that one stage.

Your function should make plots like those shown above.

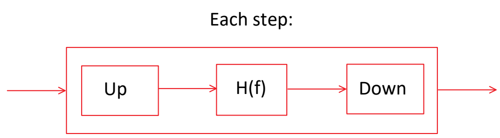
5. Call your function five times to perform the five stage sample rate conversion on this audio file.

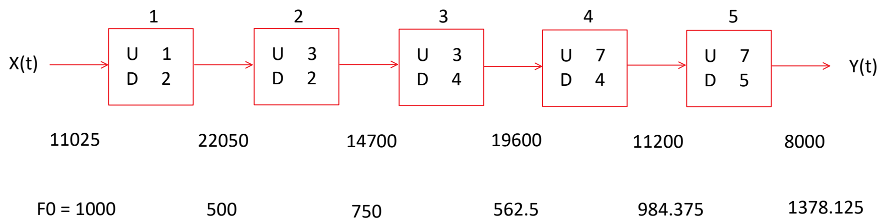
6. Turn in your plots of the signal spectra for each stage.

7. Turn in a table showing the calculation of the critical frequencies f_p, f_m and f_s for each stage.

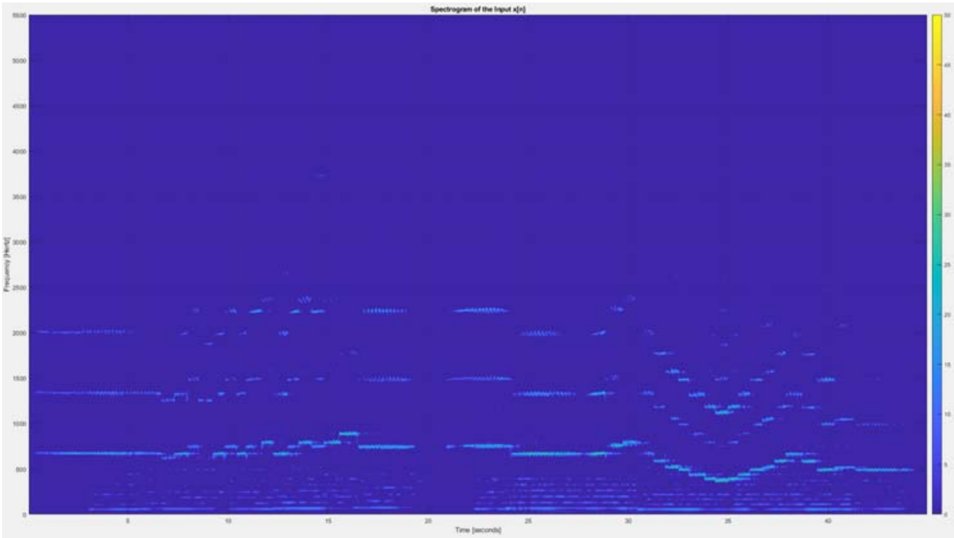
8. Include a spectrogram plot of the input signal and the final output signal.

9. Turn in your code.

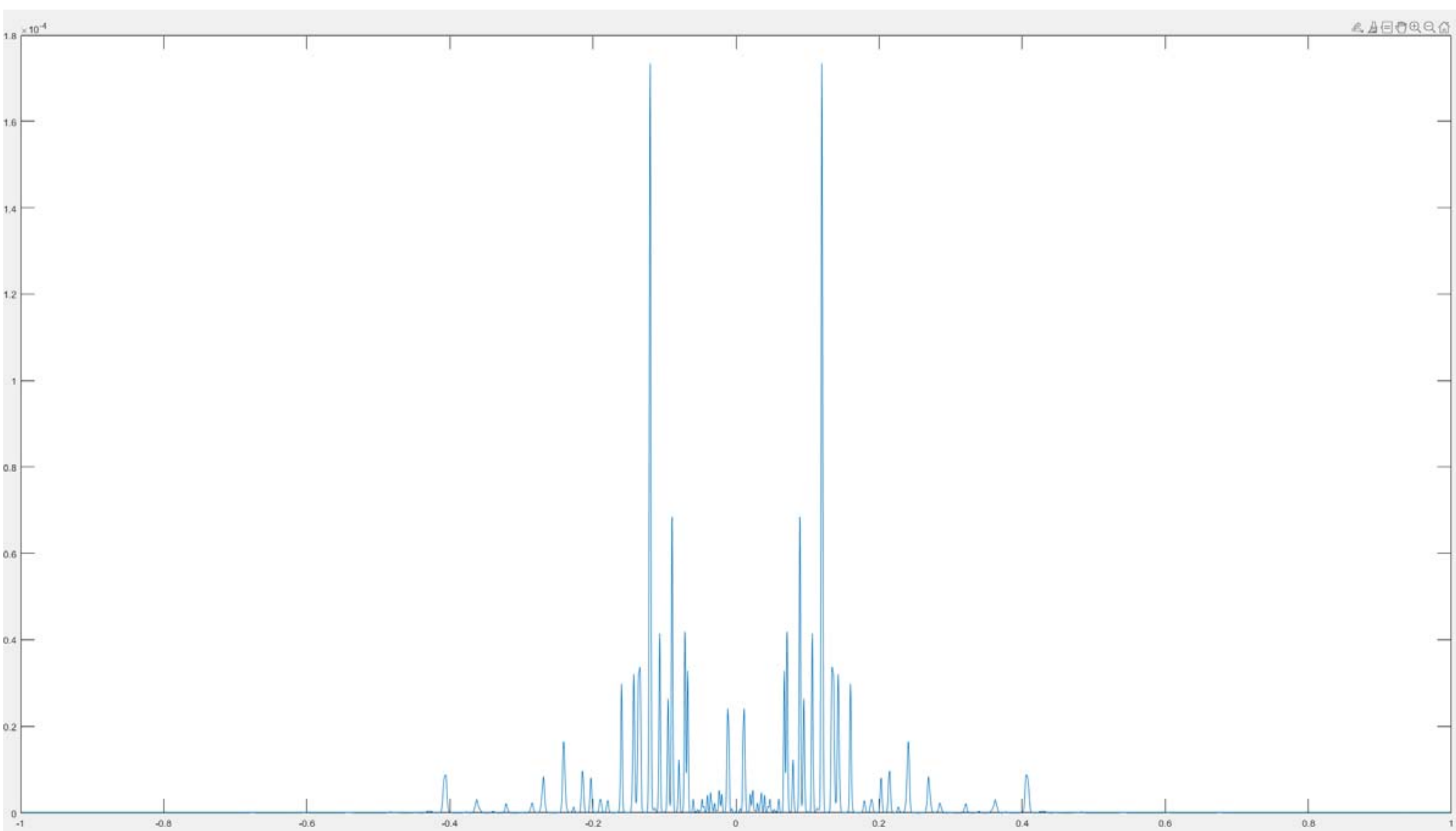


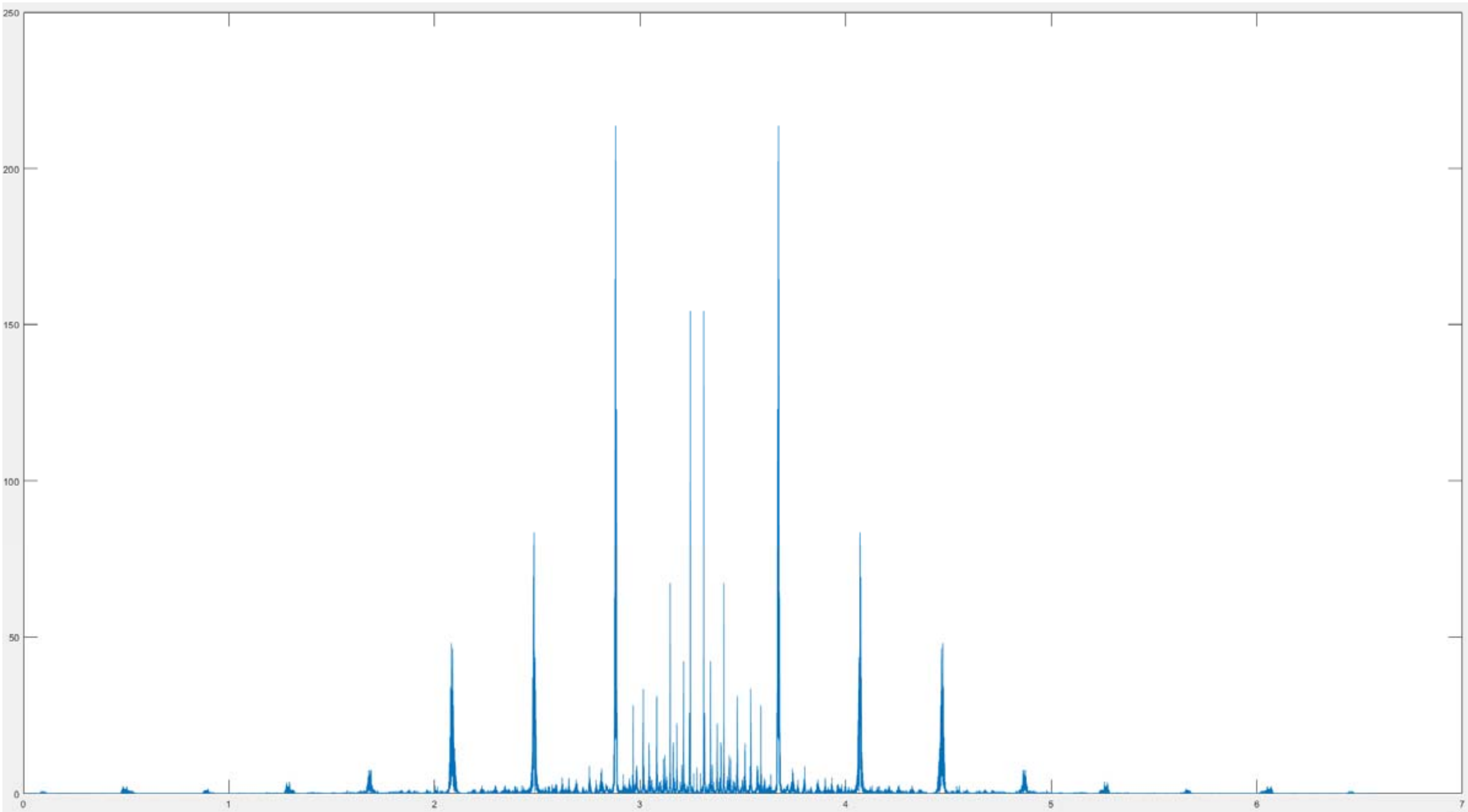


	Input	Step 1	Step 2	Step 3	Step 4	Output
Fs	11025	22050	14700	19600	11200	8000
fp	.3447	0.17244	.2585	.1939	.3393	.4750
fm	.5	.25	.375	.2812	.4922	.5

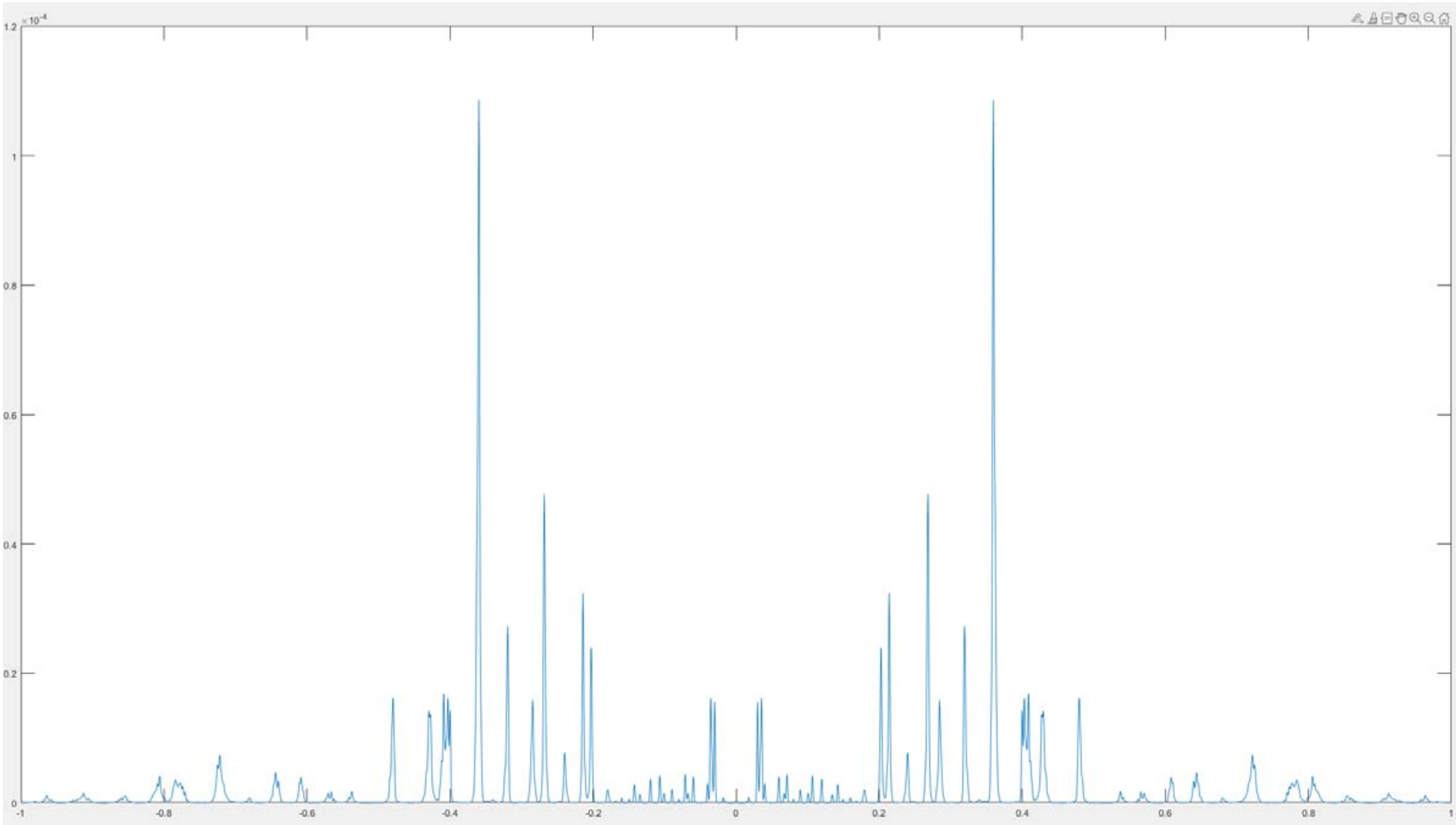

Spectrogram of the Input x[n]
Frequency [Hertz]
Time [seconds]

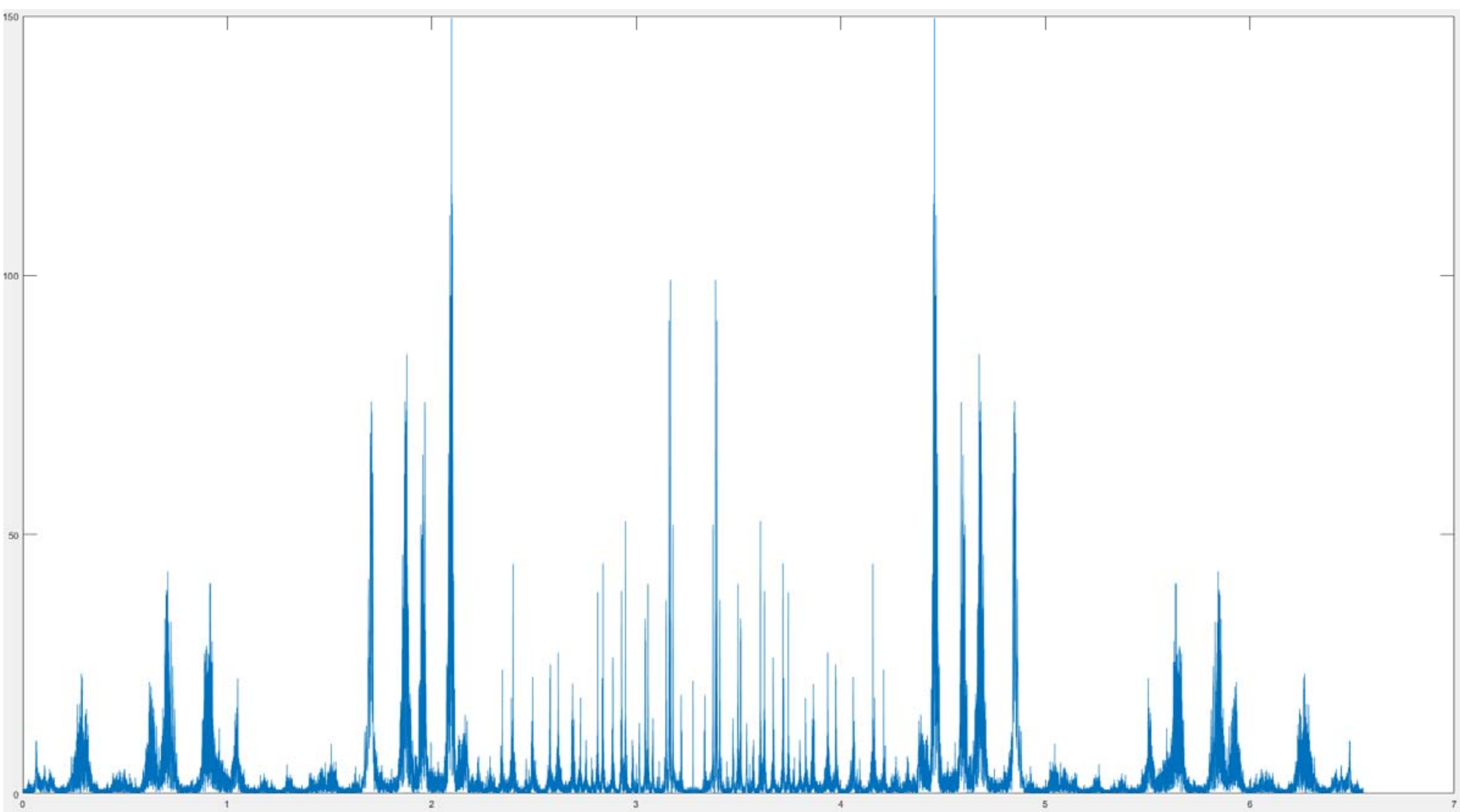
Stage 1


$\times 10^{-4}$
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

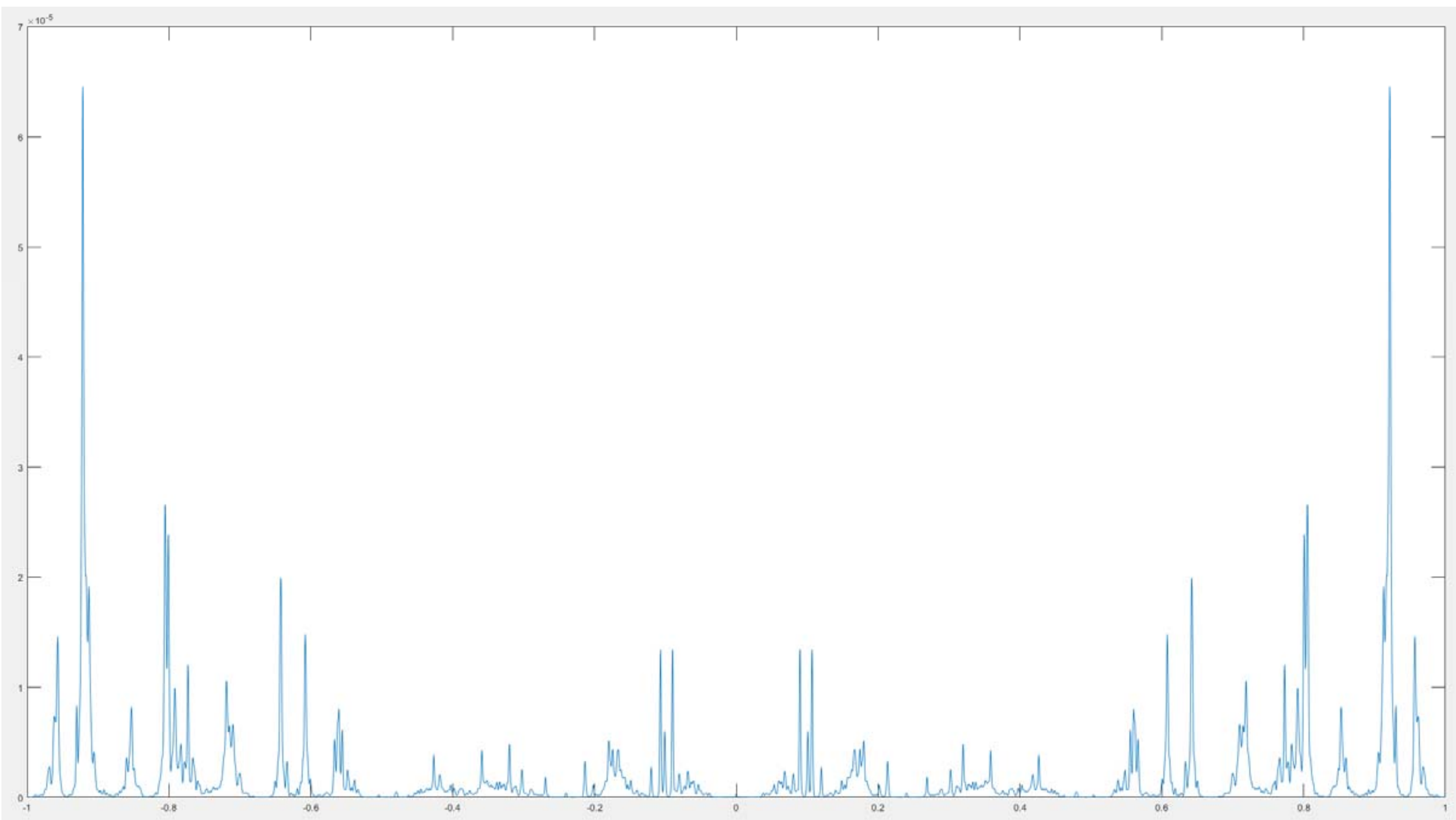


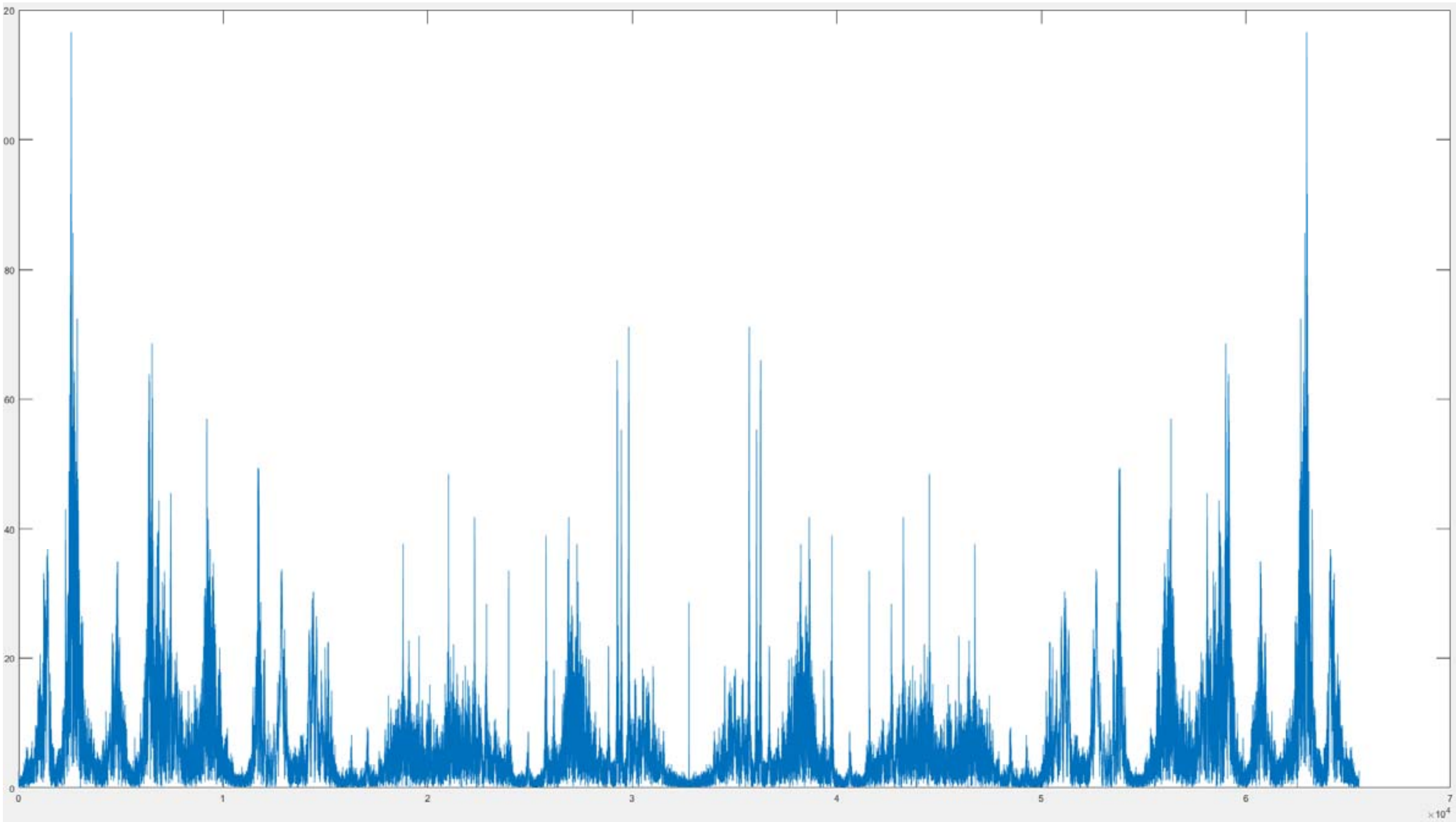
Stage 2


$\times 10^{-4}$
1.2
1
0.8
0.6
0.4
0.2
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

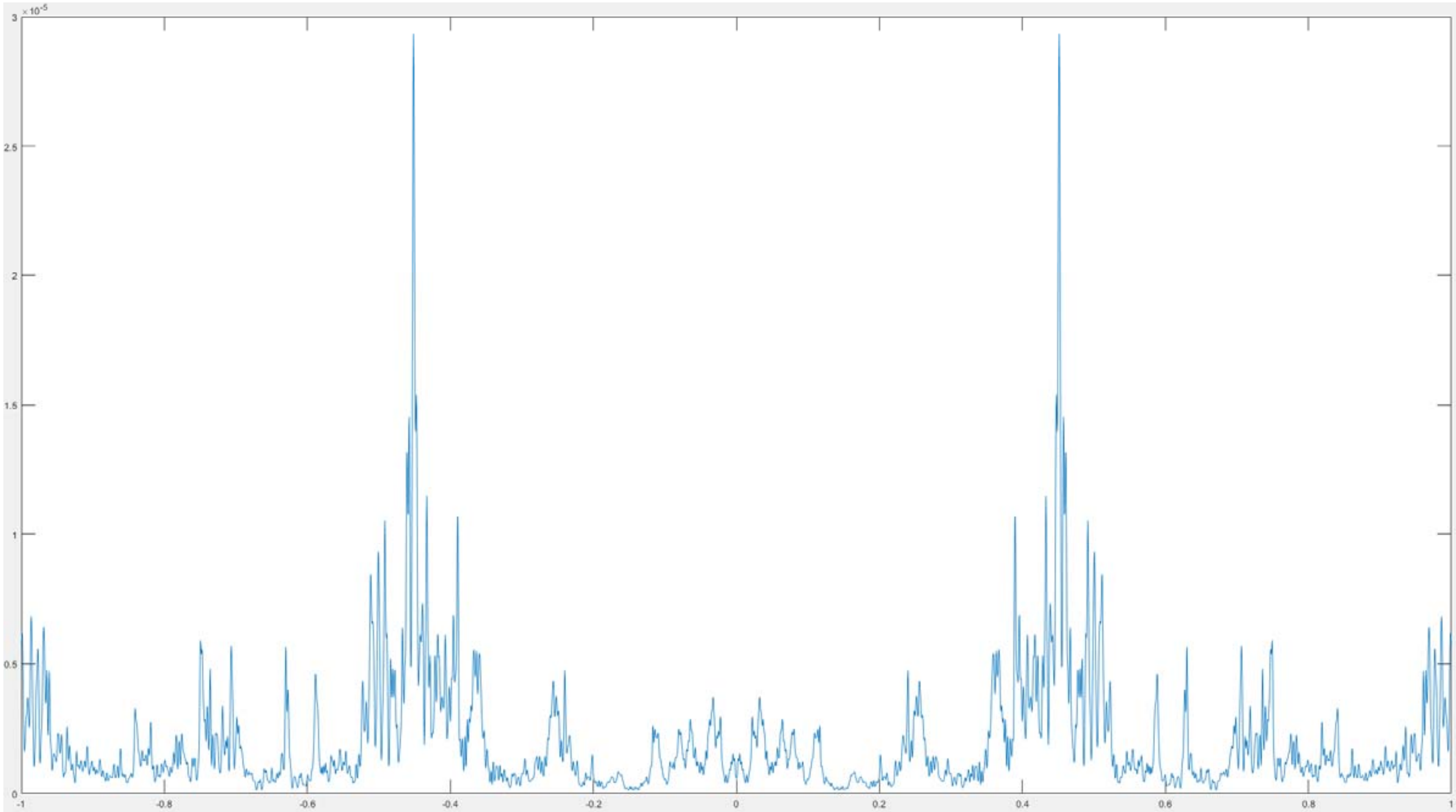

150
100
50
0
0
1
2
3
4
5
6
7
$\times 10^4$

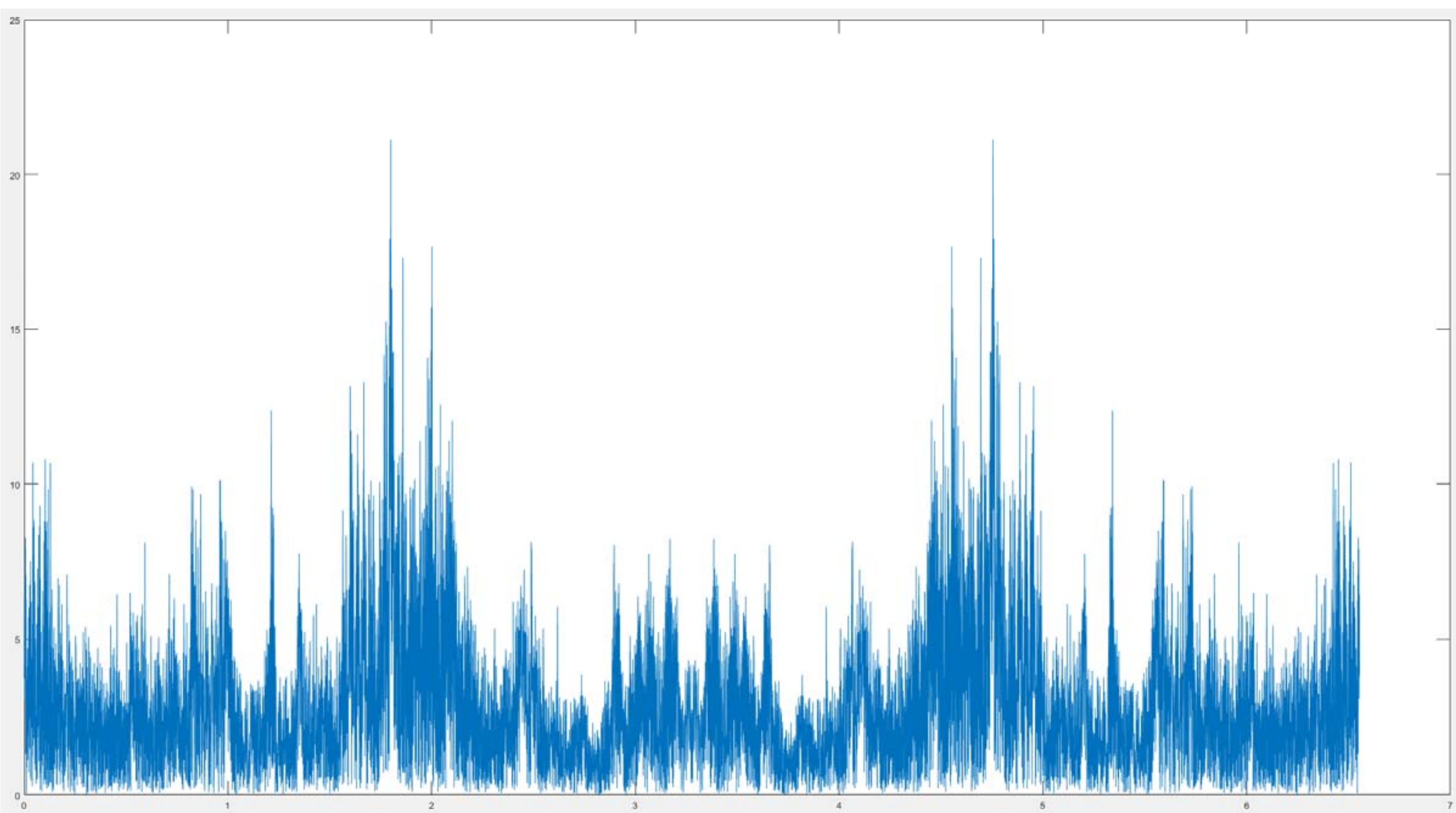
Stage 3



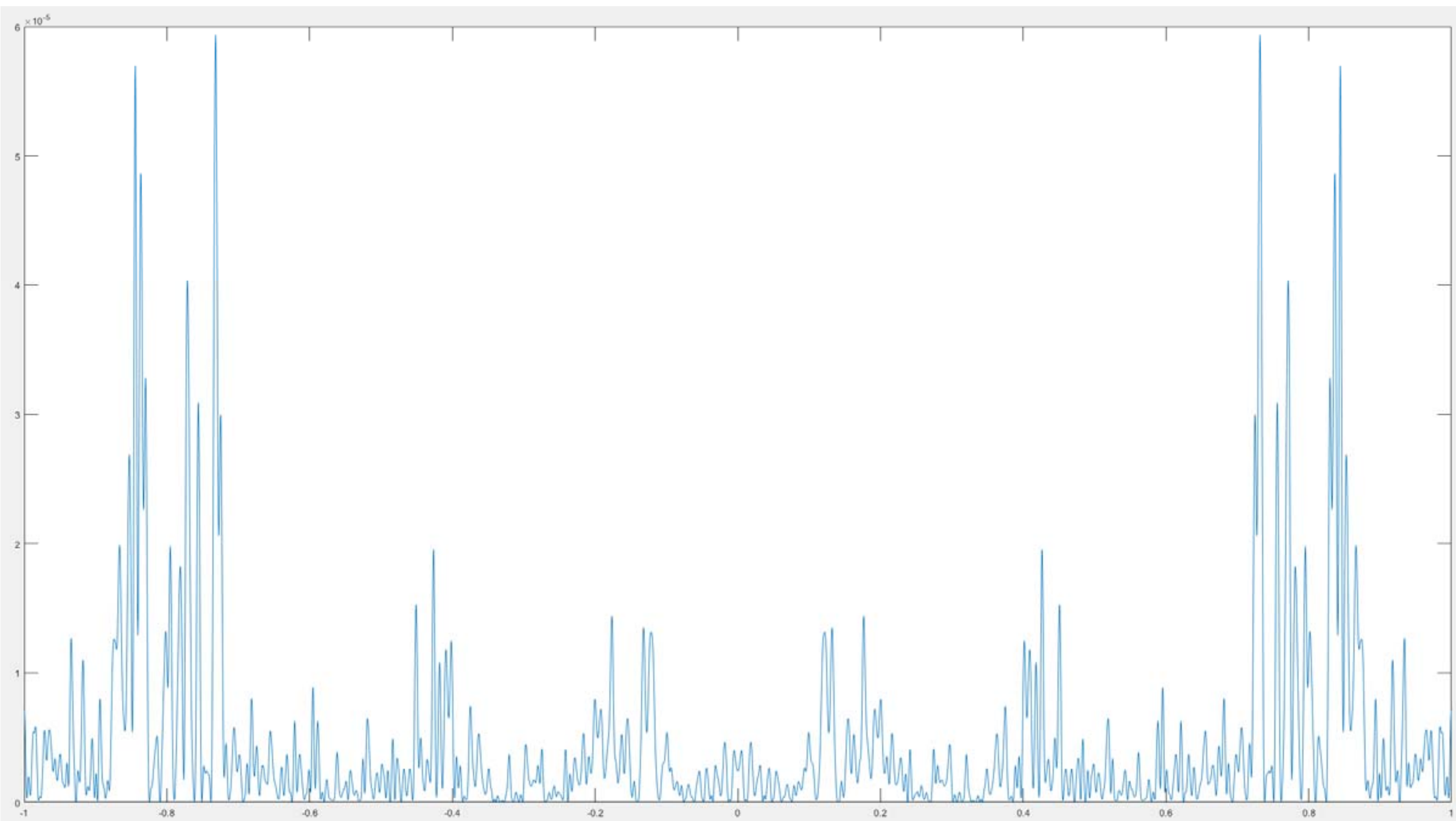


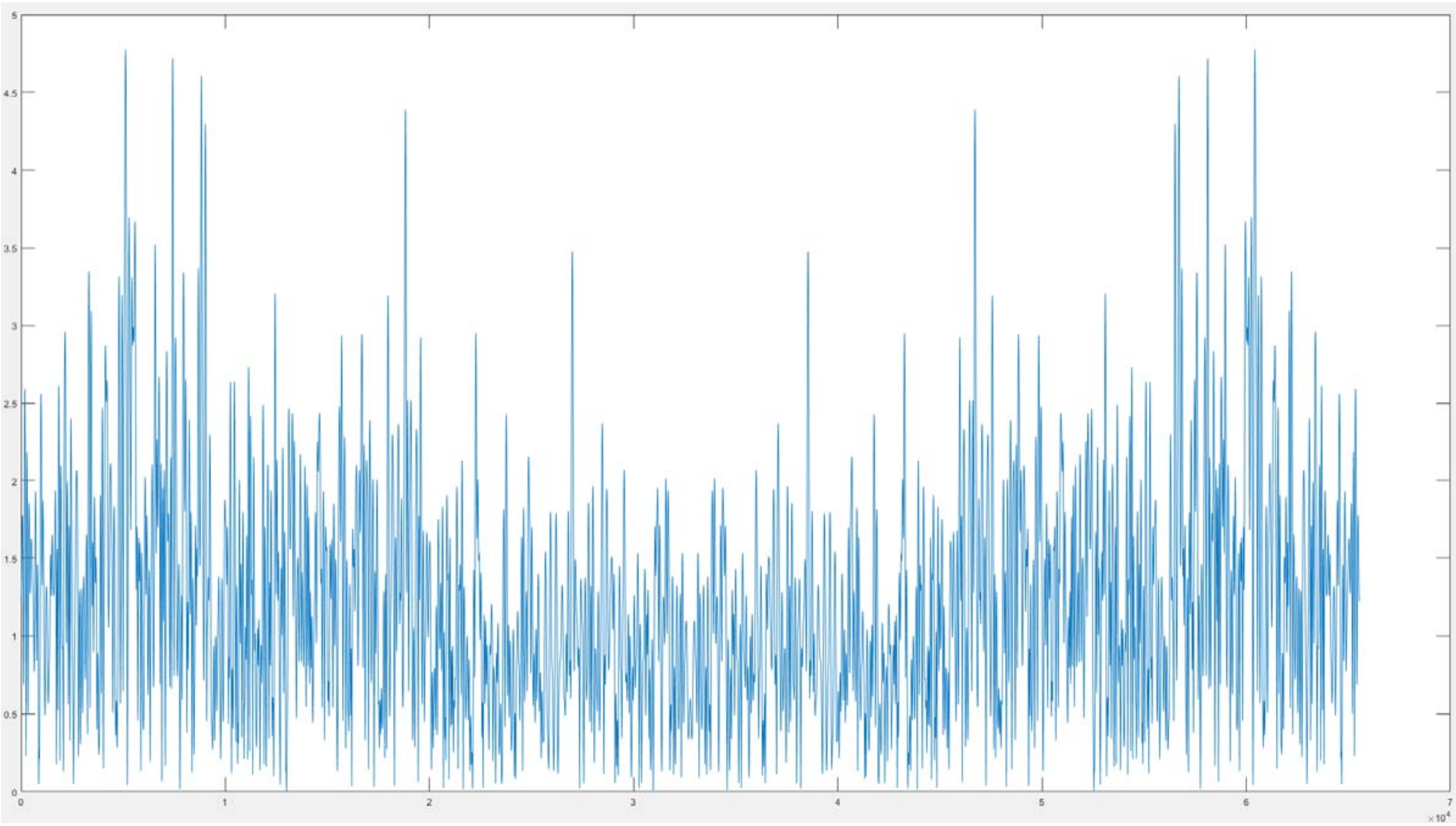
Stage 4

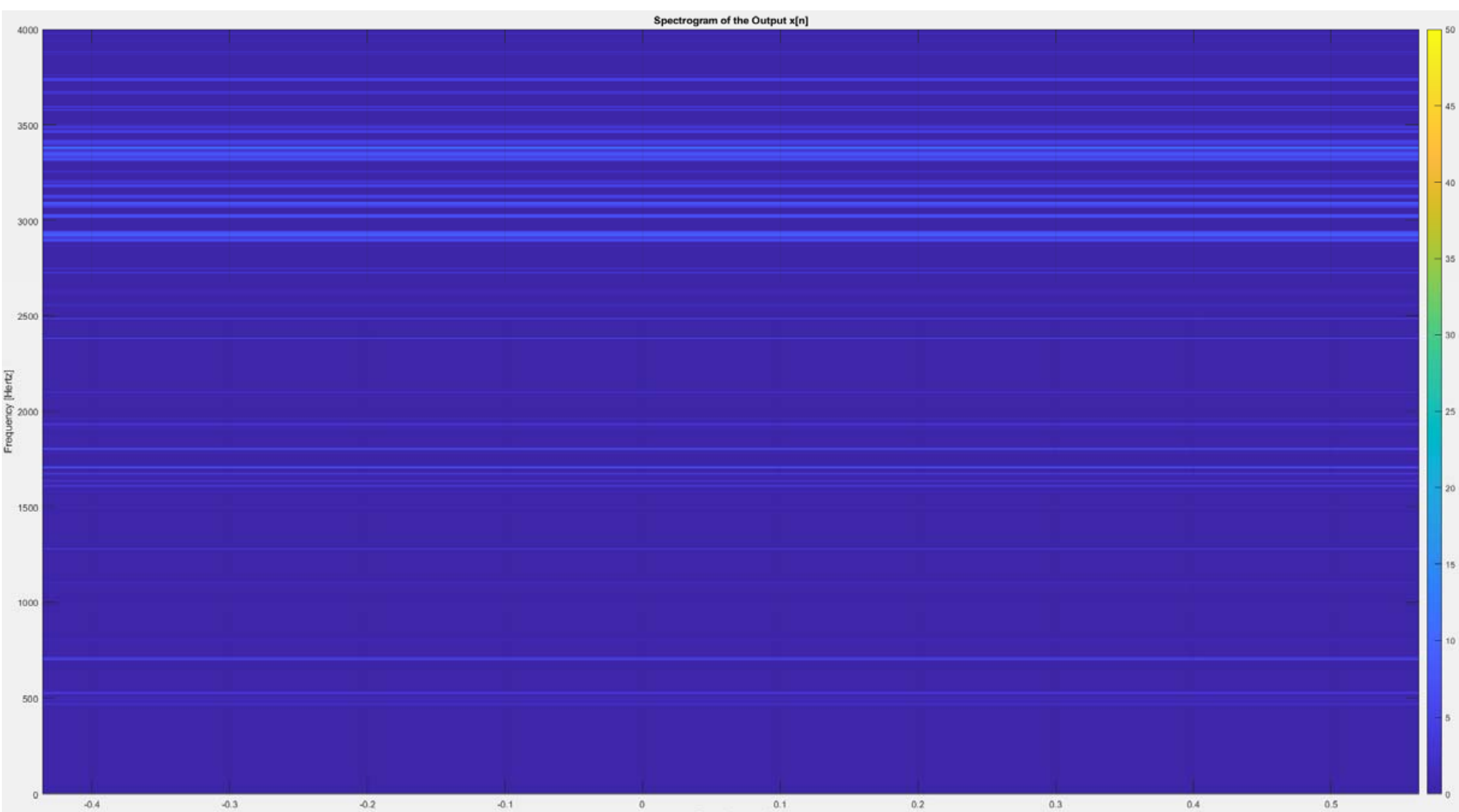

×10^-5
3
2.5
2
1.5
1
0.5
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


25
20
15
10
5
0
0
1
2
3
4
5
6
7
×10^4

Stage 5






Spectrogram of the Output x[n]
Frequency [Hertz]
Time [seconds]

```
function [y,fp,fm,Fs] = src(x,fp,fm,U,D,Fs)

if size(x,1) == 1
  x = transpose(x);
end

% upsample the input
xu = zeros(U*size(x,1),1);
xu(1:U:end) = x;


%create the filter


fm = min([(D/U)*fm,1/2]);
% Filter order
N = 101;

f1 = (fm + fp)/2;
f2 = (fm - fp)/2;
L = (N-1)/2;
n = [-L:L];
h = (sin(2*pi*f1*n) .* sin(2*pi*f2*n)) ./ ((pi*n) .* (2*pi*f2
*n));
h(L+1) = 2*f1;


%filter the sampled input

for n = 1:length(xu)
  z(n) = myFIRfilter(h,xu(n));
end

   % Calculate new vars

fp = (D/U)*fp;

Fs = (U/D)*Fs;

sigs = [fp, fm]
Fs

% downsample the input
p = zeros(size(x));
p(1:D:end) = 1;
z = x.*p;
y = z(1:D:end);
plot(abs(fftshift(fft(y,2^16))));% hold off;
shg;
end
```

```
% script for CA6 to change sample rate from 11025 to
8000 via 5 stages

[x,Fs] = audioread('galway11_mono_45sec.wav');

%   Stage 1: U=2, D=1
[y, fp, fm, Fs] = src(x,0.3447,0.5,2,1,Fs);

% [p,f] = pspectrum(y,'TwoSided',true);
% plot(f/pi, abs(p));

%  Stage 2: U=2, D=3
[y, fp, fm, Fs] = src(y,fp,fm,2,3,Fs);

% [p,f] = pspectrum(y,'TwoSided',true);
% plot(f/pi, abs(p));

% Stage 3: U=4, D=3
[y, fp, fm, Fs] = src(y,fp,fm,4,3,Fs);
```

```
% [p,f] = pspectrum(y,'TwoSided',true);
% plot(f/pi, abs(p));

% Stage 4: U=4, D=7
[y, fp, fm, Fs] = src(y,fp,fm,4,7,Fs);
%
% [p,f] = pspectrum(y,'TwoSided',true);
% plot(f/pi, abs(p));

% Stage 5: U=5, D=7
[y, fp, fm, Fs] = src(y,fp,fm,5,7,Fs);

% [p,f] = pspectrum(y,'TwoSided',true);
% plot(f/pi, abs(p));

audiowrite('outputca6.wav',y,floor(Fs));




[x,Fs] = audioread('outputca6.wav');
N = length(x);


% Plot spectrograms of input and output signals
figure();
[S,F,T] = spectrogram(x,hamming(NFFT),round(0.9
*NFFT),NFFT,Fs);
imagesc(T,F,20*log10(abs(S)),[0 50]);
colorbar;
set(gca,'YDir','normal');
ylim([0 4000]);
xlabel('Time [seconds]');
ylabel('Frequency [Hertz]');
title('Spectrogram of the Output x[n]');
grid on;
```