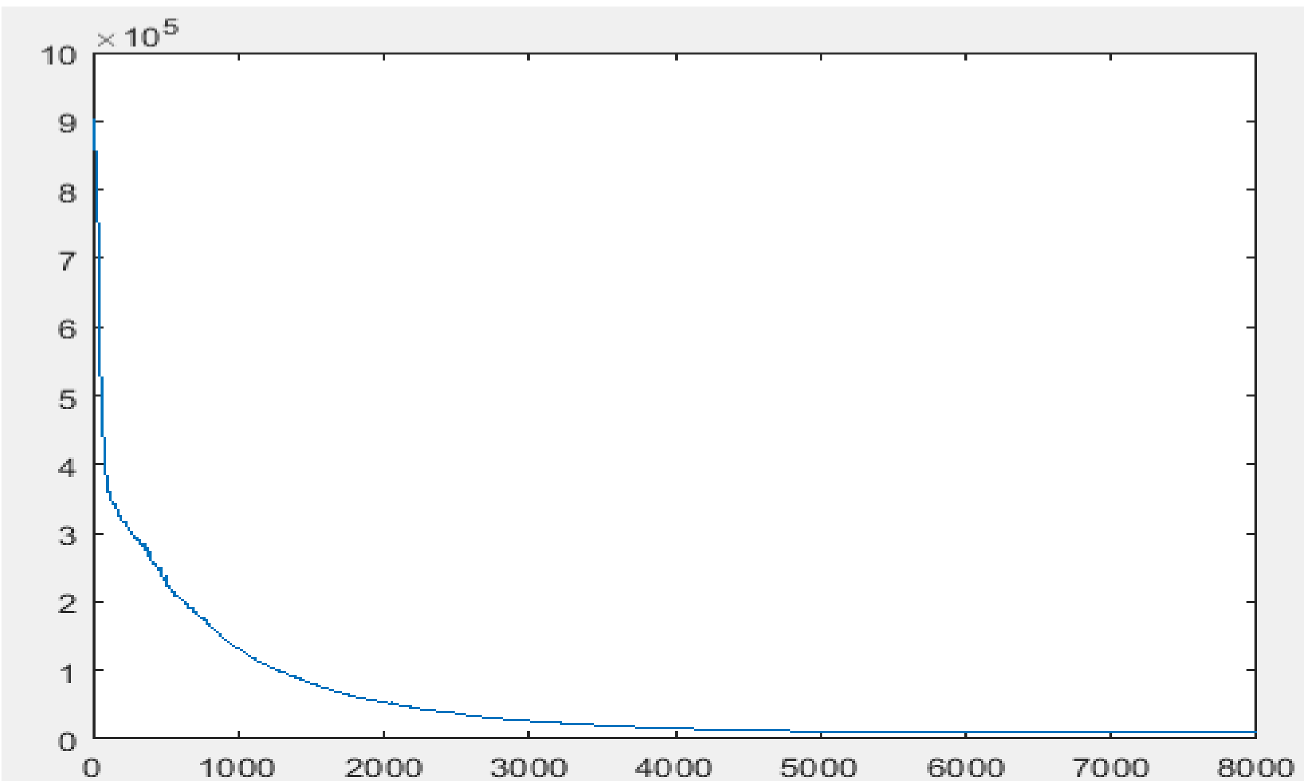
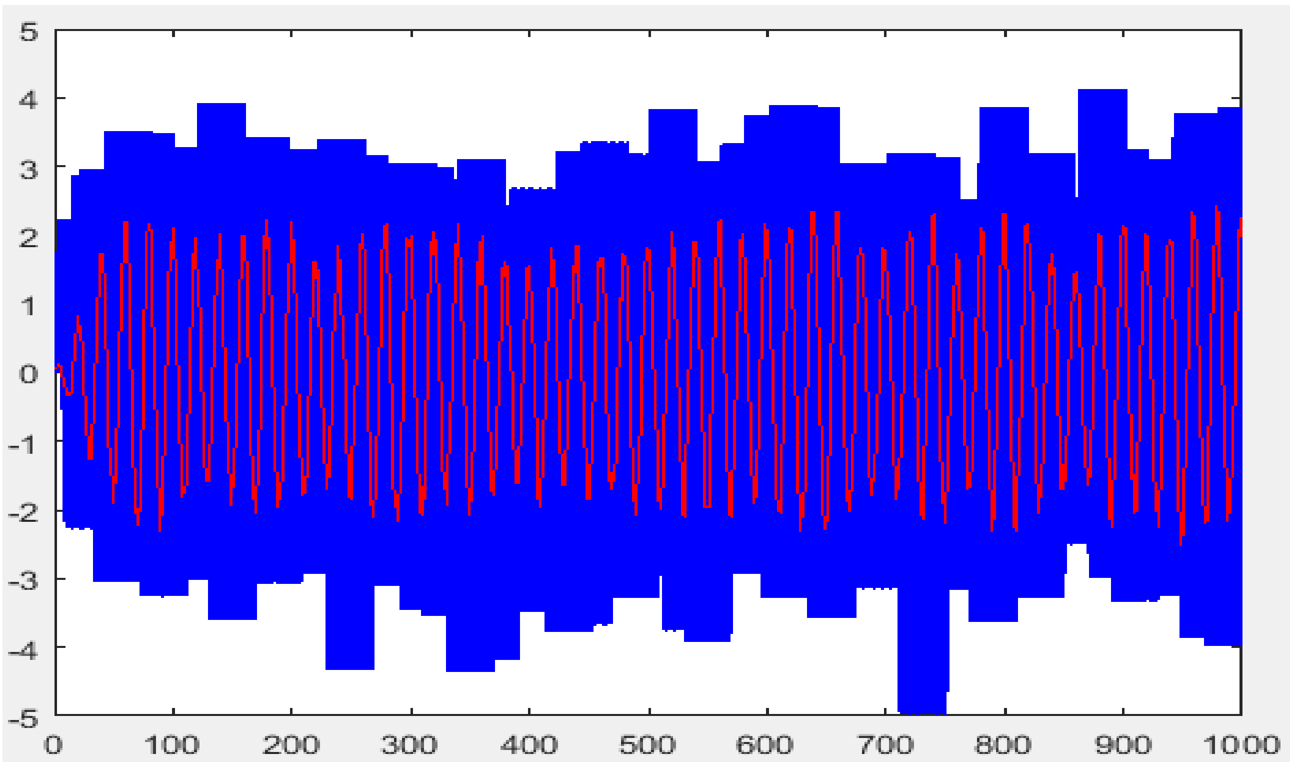
```
diary on
format compact
%Johnny Li
%EEL3135 Fall 2018
%Lab 5 Part 2

type clean

%clean Use LMS to remove white noise out of a signal consists of white
%noise and a sinusoid of frequency 400Hz.
%Script for 2.1

%2.1.1

%Code given.
%Filter length
L=40;
%Sample Frequency
fs = 8000;
%Time Interval
tt = (0:1/fs:1).';
%Coefficient vector
K = length(tt);
%Delay by delta samples to obtain the training signal
delta = 2;
%Full signal (delayed by delta samples)
x = randn(K,1) + 2*cos(400*2*pi*tt);
%Advanced by delta samples and hence get the undelayed full signal
d = x(delta+1:end);
%Signal
dd = d(1:K-2);

%Generate input matrix xx out of x using the same method discussed in part
%1.1.
%Storage vector
xx = zeros((K+L-1),L);
%Generate input matrix xx
for i=1:L
    xx((i:(K+i-1)),i) = x;
end
%Final form
xx = xx(1:K-delta,:);

%2.1.2

%Generate filter weights vector ww of length 40 consists of random filter
%weight using randn().
%Random weights
ww = randn(40,1);

%2.1.3

%Using LMS, adapt a filter with step size u=0.0005. Desired output is d and
%input is matrix xx.
%Step size
```

```
u = .0005;
%Storage Vector
J = zeros(K-delta, 1);
%Loop to update the filter using b(n+1)=b(n)+ue[n]x[n].
for n = 1:(K-delta)
    yy = xx*ww;
    error = dd-yy;
    ww = ww+u*error(n)*xx(n,:)';
    J(n)=error'*error;
end

%Plot Learning Curve
plot(J);
```




```
%2.1.4

%Plot the original signal and the filtered signal (make this red) on the
%same plot using hold on and hold off. Make sure you zoom in until length of
%x-axis is about 1000 samples.
%Plot range 0 to 1000
plot1 = n(1:1000);
%Output plot
plot2 = yy(1:1000, :);
%Input plot
plot3 = xx(1:1000, :);

%Plot blue
```

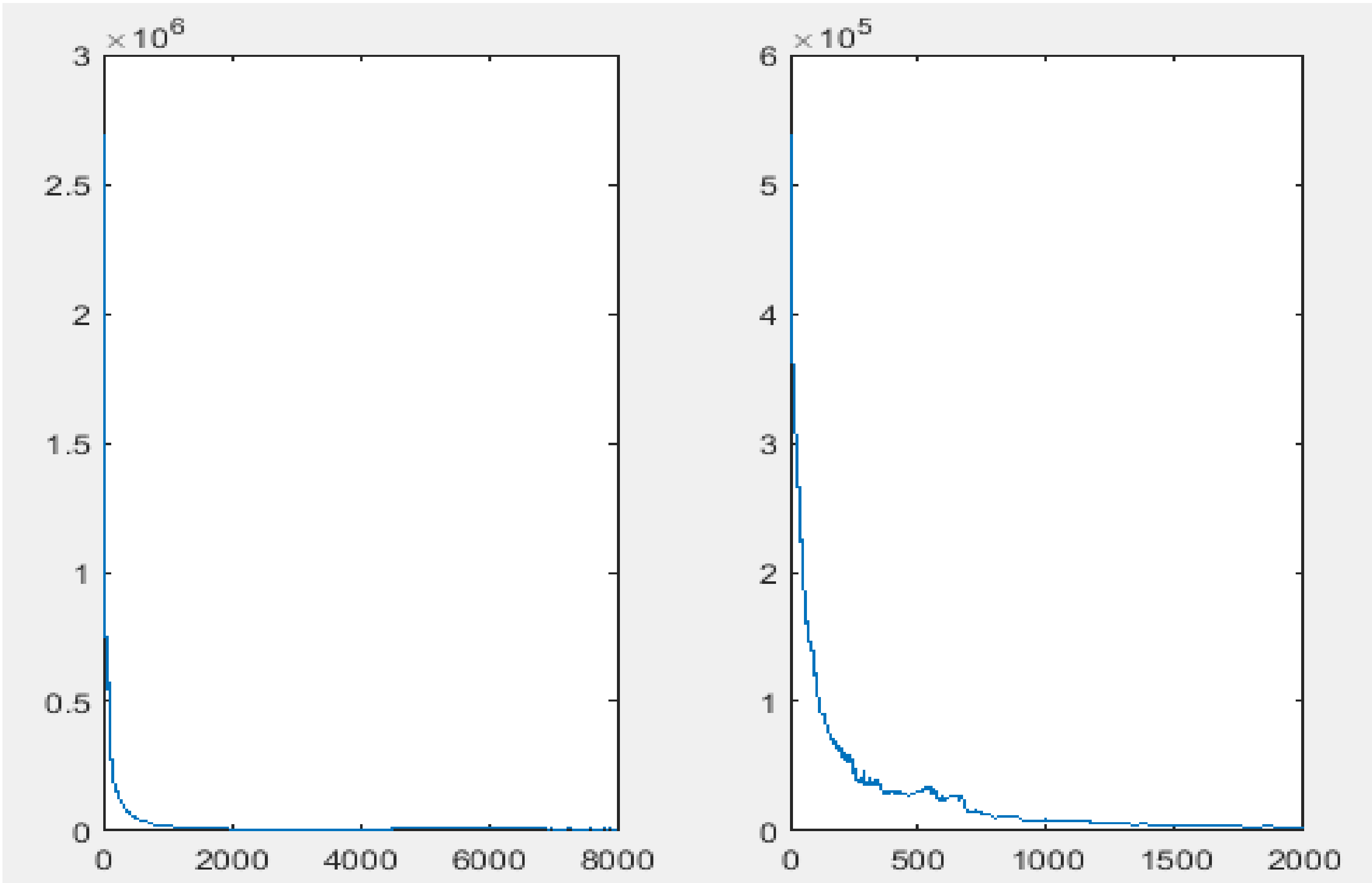
```
plot(plot1, plot3, 'b-');
hold on

%Plot red
plot(plot1, plot2, 'r-');
hold off
```




```
%2.1.5

%Plot the magnitude response of the filter using freqz() iteration where bk
%is the filter weights vector, N is desired resolution of H, Fs is the
%sampling frequency used.
%Magnitude Response of the Filter
[H,f] = freqz(ww,1,512,fs);
%Plot
plot(f,abs(H));

%Question: What kind of filter is this (i.e. highpass, bandpass, or
%stopband)? Estimate the center frequency of the pass-band using data
%cursor.
%Since the plot is near evenly distributed the filter is a bandpass. The
%center frequency is estimated to be 425.

%2.2
%Coefficient
K = 70000;
%Filter length
L = 40;
%Storage vector
```

```
xx = zeros((K+L-1),L);

%Generate input matrix xx
for i = 1:L
    xx((i:(K+i-1)),i) = reference;
end

%Final form
xx = xx(1:K,:);
%Random weights
ww = randn(L,1);

%Step size
u = 0.005;

%Loop to update the filter using b(n+1)=b(n)+ue[n]x[n].
for n = 1:4500
    yy = xx(1:45999,:)*ww;
    error = (primary(1:45999)')-yy;
    ww = ww + u*error(n)*xx(n,:)';
    J(n) = error'*error;
end

%The first sound sinusoid
sound1 = primary(1:45999)'-xx(1:45999,:)*ww;

%Random weights set 2
w2 = randn(L,1);

%Second loop to update the filter using b(n+1)=b(n)+ue[n]x[n].
for m = 46000:1:48000
    yy2 = xx(46000:1:70000,:)*w2;
    e2 = (primary(46000:1:70000)')-yy2;
    w2 = w2 + u*e2(m-45999)*xx(m,:)';
    J2(m-45999) = e2'*e2;
end

%Plot the two signals to compare
subplot(1,2,1);
plot(J);
subplot(1,2,2);
plot(J2);

%The first sound sinusoid
sound2 = primary(46000:70000)'-xx(46000:70000,:)*w2;

%Final signal
v = vertcat(sound1, sound2);

%For clipping
v = v/max(abs(v));

%Create autofile
audiowrite('clean2.wav', v, 19000);
```

%Show learning curves and submit the clean speech signal as .wav file.



%Question: What does the woman say?
%I will not condone a course of action that lead us to war.