LAB : 04
Signals And Systems

Umar Habib

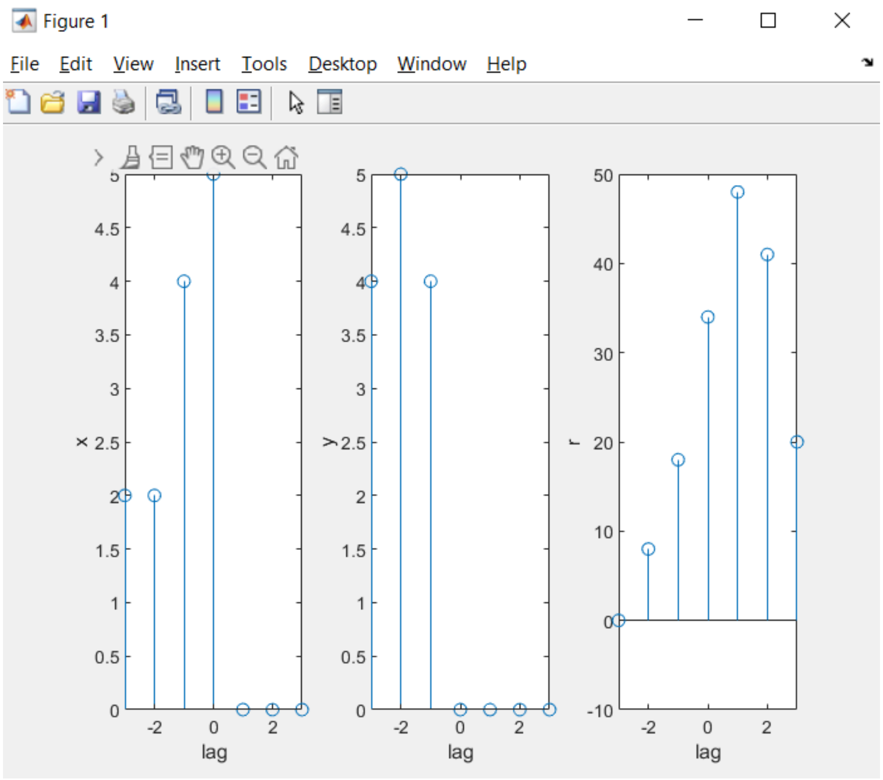
Task:1

- On the correlation graph, the peaks are higher where the inputs have similar values. For instance both the inputs have 4 on the third index, and the output graph shows a high peak on the right side. This shows that the correlation will be higher when there is higher similarity.

CODE:

```
clc
close all
clear all
x = [2 2 4 5];
y = [4 5 4 0];
[r,k]= xcorr(x,y);
x = [x 0 0 0];
y = [y 0 0 0];
subplot(1,3,1)
stem(k,x)
xlabel('lag')
ylabel('x')
subplot(1,3,2)
stem(k,y)
xlabel('lag')
ylabel('y')
subplot(1,3,3)
stem(k,r)
xlabel('lag')
ylabel('r')
```

SNAP:



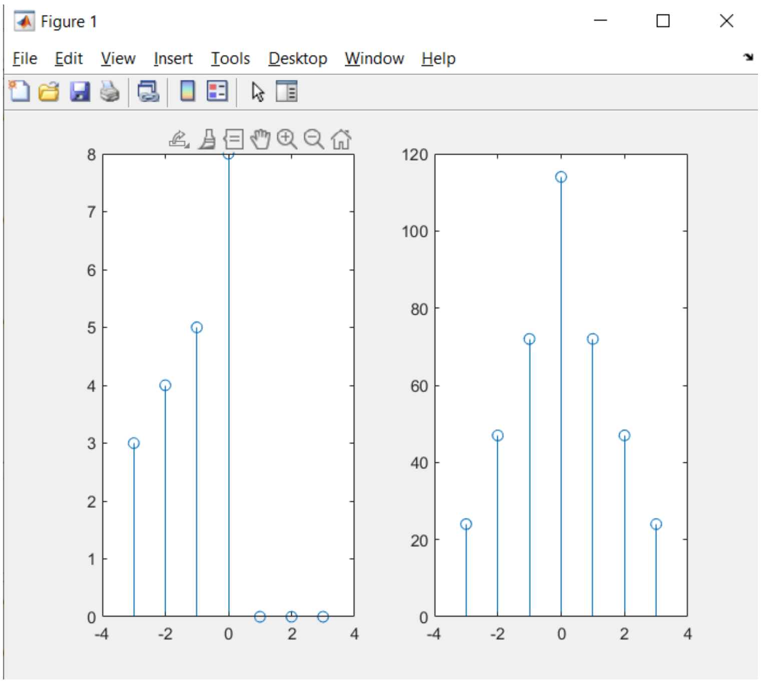
Task:2

- The autocorrelation looks like a bell curve plotted in discrete time. It has the highest peak at 0, which means that both the signals are the same.

CODE:

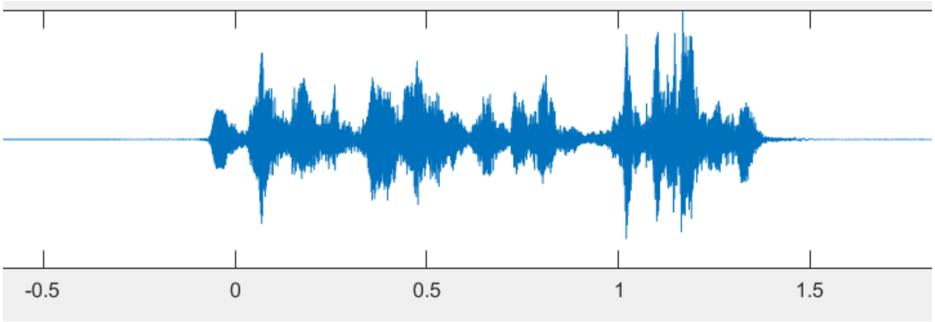
```
x = [3 4 5 8];
[r,k] = xcorr(x,x)
x = [x 0 0 0];
subplot 121
stem(k,x)
subplot 122
stem(k,r)
```

SNAPSHOT:

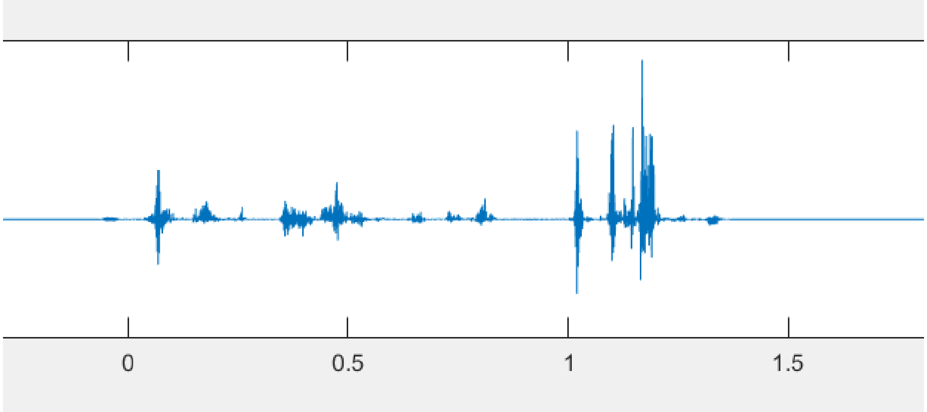


Task:3

Without peak enhancing:



With peak enhancing:



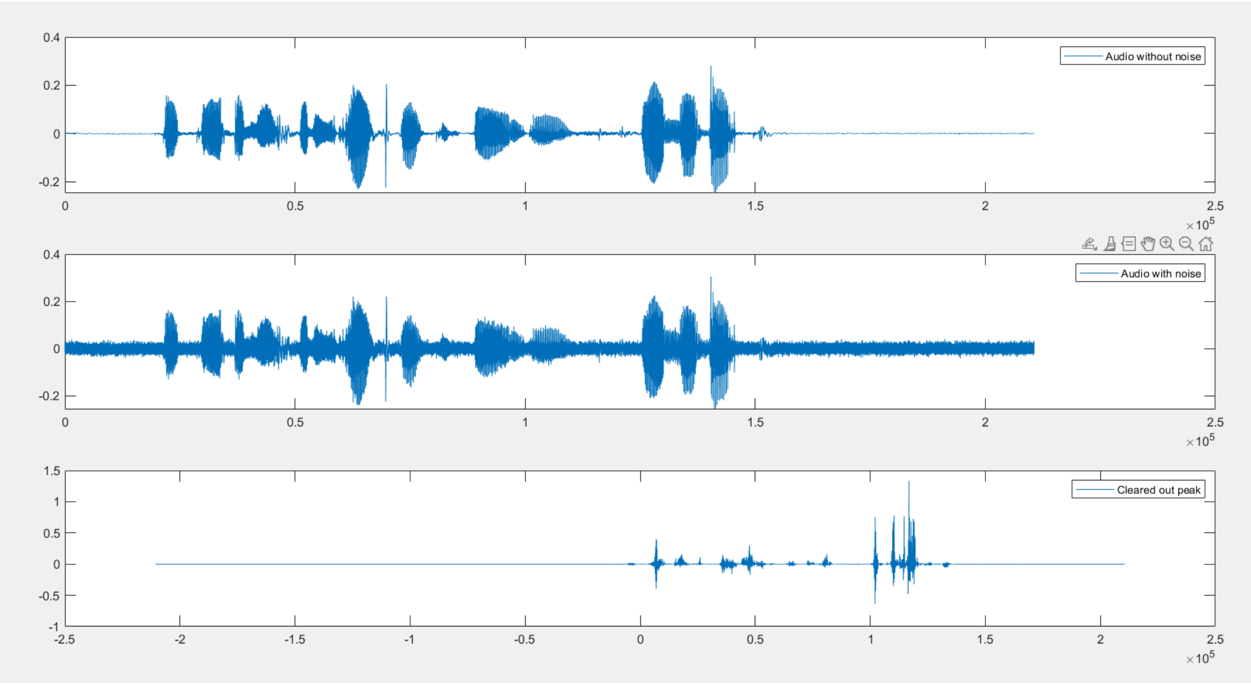
- It can be seen from the above plots that enhancing the peak gives us a clearer view of when the audios are matching. In the given audios, the common word was Islamabad which comes at the end of the first audio, which is why the correlation has a high peak towards the end. The high peak is clearer to detect after enhancement.

- After detecting the peak of correlation and playing it, we have extracted the part where both the sounds are the same. The lag() function returns to us the exact value of the lag on the specified index.

CODE:

```
clc
close all
clear all
[y,Fs_y] = audioread('Speech_segment.wav');
[x,Fs_x] = audioread('speech.wav');
sound(y,Fs_y)
pause(2)
sound(x,Fs_x)
[correlation,lag] = xcorr(x,y);
cmax = max(abs(correlation));
Nonlinear_Effect = (1.1*correlation/cmax).^3;
subplot 311
plot(x)
subplot 312
plot(y)
subplot 313
plot(lag,Nonlinear_Effect)
maxval = max(Nonlinear_Effect);
index = find(Nonlinear_Effect == maxval);
location = lag(index)
pause(4)
sound(x(location:end), Fs_x)
```

Task:4



CODE:

```
x_noise = randn(size(x));
x_new = x + x_noise/100;
sound(x_new,Fs_x)
%detecting the same segment with noisy audio
[correlation,lag] = xcorr(x_new,y);
cmax = max(abs(correlation));
Nonlinear_Effect = (1.1*correlation/cmax).^3;
%Plotting the graphs
subplot 311
plot(x)
legend('Audio without noise')
subplot 312
plot(x_new)
legend('Audio with noise')
subplot 313
plot(lag,Nonlinear_Effect)
legend('Cleared out peak')
%finding the peak
maxval = max(Nonlinear_Effect);
index = find(Nonlinear_Effect == maxval);
location = lag(index)
pause(4)
sound(x_new(location:end), Fs_x)

%—PART B—


x_noise = randn(size(x));
x_new = x + x_noise/2;
powX = mean(x.^2);
powN = mean(x_noise.^2);
SNR = powX/powN
[correlation,lag] = xcorr(x_new,y);
cmax = max(abs(correlation));
Nonlinear_Effect = (1.1*correlation/cmax).^3;
%Plotting the graphs
subplot 311
plot(x)
legend('Audio without noise')
subplot 312
plot(x_new)
legend('Audio with noise')
subplot 313
plot(lag,Nonlinear_Effect)
legend('Cleared out peak')
%finding the peak
maxval = max(Nonlinear_Effect);
index = find(Nonlinear_Effect == maxval);
location = lag(index)
sound(x_new(location:end), Fs_x)
```

- It can be seen that the second plot is the same as the first plot but with noises.
- In the third plot we have a peak at the farther right of the graph, indicating that the audios are similar towards the end.
- This method of correlation between two audios is effective and useful because it is able to detect similarity between audios even with continuous noise.
- SNR = 8.4931e-04 seems to be the limit of the detection of matlab. So matlab seems to detect the correlation even if I just reduce the amplitude of the noise by half.

Task:5(POSTLAB)

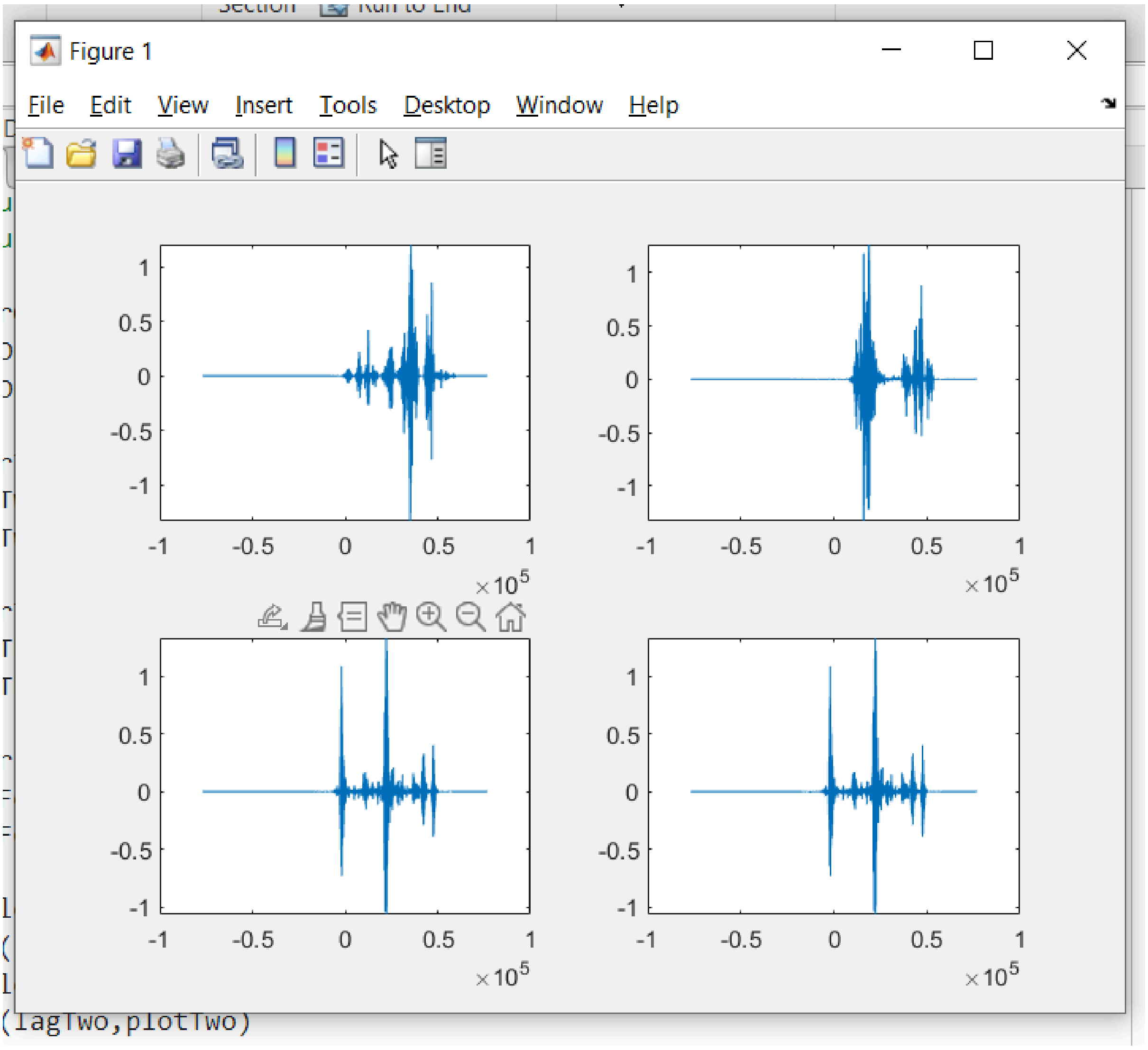
- The correlation peak merged with the original sound does not end up at the exact point where the speaker says “okay” but instead it is a little bit in advance.
- This is not a good password detector since correlation can be found even when the tone and frequency is the same

CODE:

```
clc
clear all
close all
[mainAudio,Fs_mainAudio] = audioread('main_audio.wav');
[audioOne,Fs_audioOne] = audioread('okay_1.wav');
[audioTwo,Fs_audioTwo] = audioread('okay_2.wav');
[audioThree,Fs_audioThree] = audioread('okay_3.wav');
[audioFour,Fs_audioFour] = audioread('okay_4.wav');
% sound(mainAudio,Fs_mainAudio)
% pause(2)
% sound(audioOne,Fs_audioOne)
% pause(2)
% sound(audioTwo,Fs_audioTwo)
% pause(2)
% sound(audioThree,Fs_audioThree)
% pause(2)
% sound(audioFour,Fs_audioFour)
[corrOne,lagOne] = xcorr(mainAudio,audioOne);
cmaxOne = max(abs(corrOne));
plotOne = (1.1*corrOne/cmaxOne).^3;
[corrTwo,lagTwo] = xcorr(mainAudio,audioTwo);
cmaxTwo = max(abs(corrTwo));
plotTwo = (1.1*corrTwo/cmaxTwo).^3;
[corrThree,lagThree] = xcorr(mainAudio,audioThree);
cmaxThree = max(abs(corrThree));
plotThree = (1.1*corrThree/cmaxThree).^3;
[corrFour,lagFour] = xcorr(mainAudio,audioThree);
cmaxFour = max(abs(corrFour));
```

```
plotFour = (1.1*corrFour/cmaxFour).^3;
subplot 221
plot(lagOne,plotOne)
subplot 222
plot(lagTwo,plotTwo)
subplot 223
plot(lagThree,plotThree)
subplot 224
plot(lagFour,plotFour)
maxOne = max(plotOne)
indexOne = find(plotOne == maxOne);
location = lagOne(indexOne)
% pause(2)
% sound(mainAudio(location:end),Fs_mainAudio)
maxTwo = max(plotTwo);
indexTwo = find(plotTwo == maxTwo);
location = lagTwo(indexTwo)
% pause(2)
% sound(mainAudio(location:end),Fs_mainAudio)
maxThree = max(plotThree);
indexThree = find(plotThree == maxThree);
location = lagThree(indexThree);
% pause(2)
% sound(mainAudio(location:end),Fs_mainAudio)
maxFour = max(plotFour);
indexFour = find(plotFour == maxFour);
location = lagFour(indexFour);
% pause(2)
% sound(mainAudio(location:end),Fs_mainAudio)
threshold = min([cmaxOne cmaxTwo cmaxThree cmaxFour])
```

SNAP:



```
(lagTwo,plotTwo)
```