Lab Assignment – 01 – Part 1 – Spring 2020

Signal & Systems

Atharva Deshpande || UG-1 || CSE || S20190010043

Indian Institute of Information Technology, Sri City.

Sri City, Andhra Pradesh.

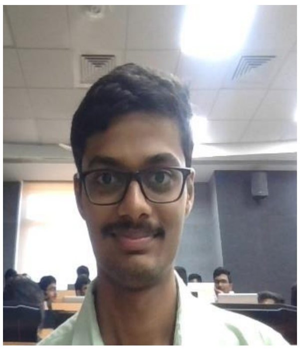
January 23, 2020.

1.Handling Images in MATLAB

- First, we need to take an image (passport size), obtained from any source like a camera in a smart phone or a laptop or any available image in the device.
- Then we have to upload it in the software MATLAB and using appropriate function we read the image.
- Finally, we analyse the different aspects of the uploaded image and based on our observations we report the obtained data from the image.

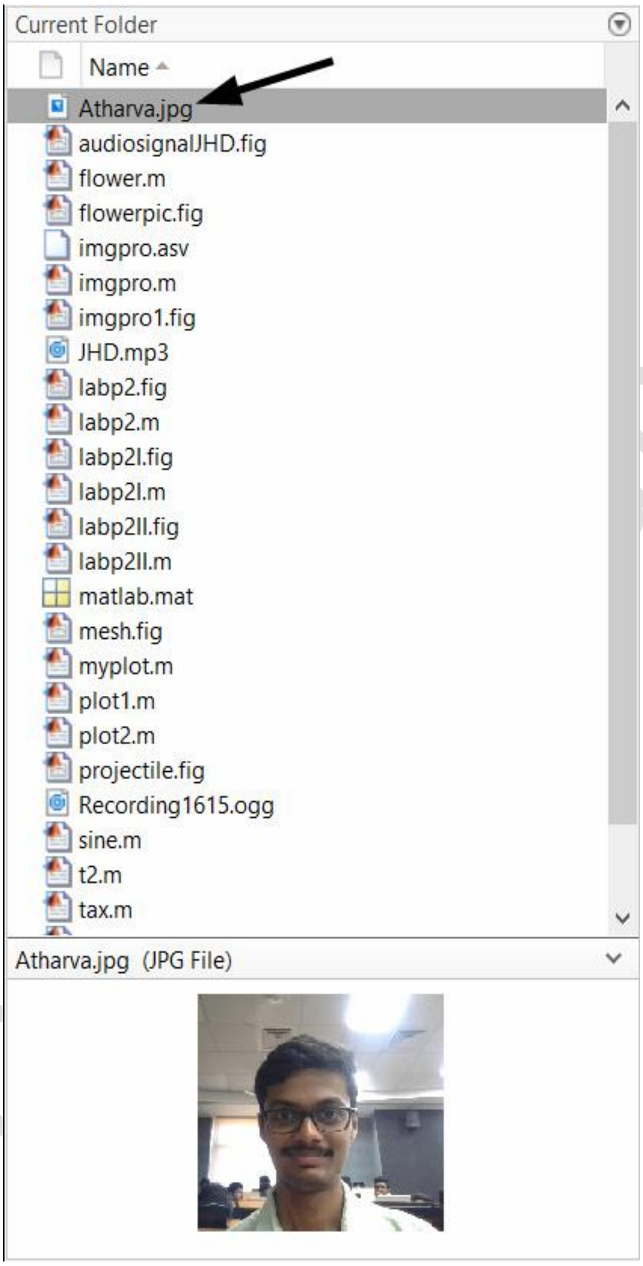
Here we go,

(i) Following the instructions, the image 'Atharva.jpg' was captured from a laptop and its dimensions were reported as 770 x 707.



Source File: 1

(ii) We then copy the image and paste it into the folder given in the column at left.

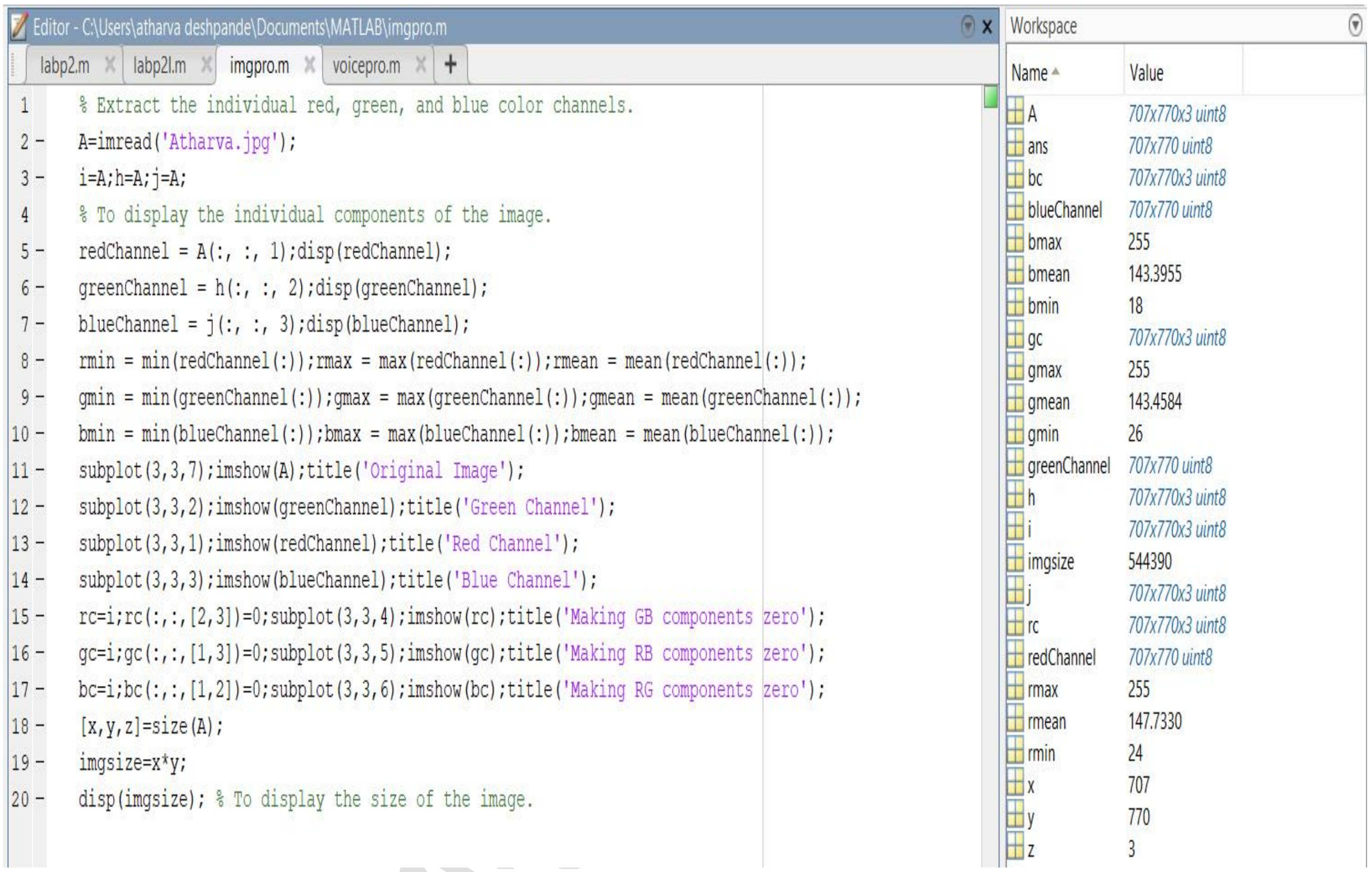


So now, we have uploaded the file and the software is ready to access the image.

(iii) From here onwards, we start coding and first use vector A to store the information of the image using the function *'imread(file name)'*.

Our goal is to extract the red, blue and green channels, exploring the given image and experimenting with its colour gradient.

- Here is the code:

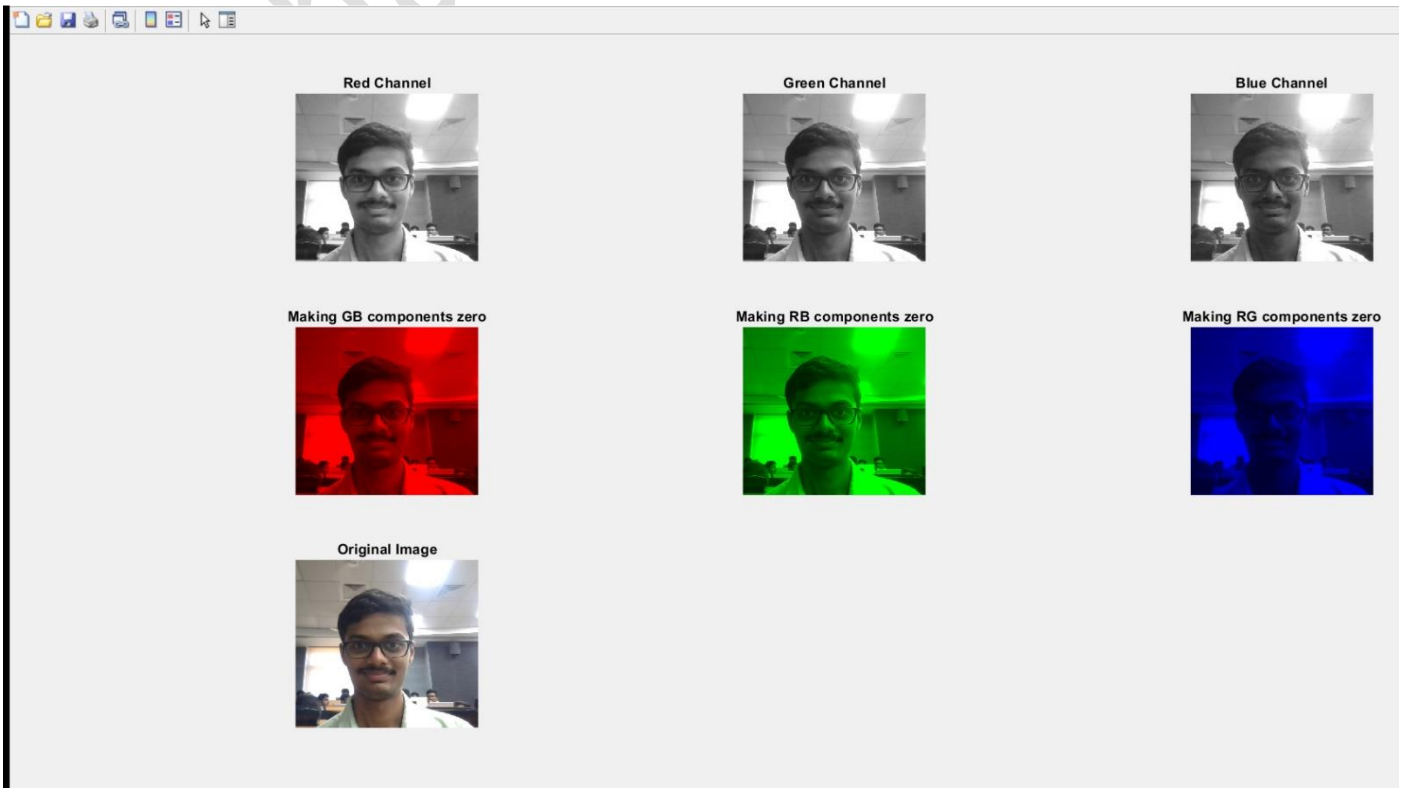



```
% Extract the individual red, green, and blue color channels.
A=imread('Atharva.jpg');
i=A;h=A;j=A;
% To display the individual components of the image.
redChannel = A(:, :, 1);disp(redChannel);
greenChannel = h(:, :, 2);disp(greenChannel);
blueChannel = j(:, :, 3);disp(blueChannel);
rmin = min(redChannel(:));rmax = max(redChannel(:));rmean = mean(redChannel(:));
gmin = min(greenChannel(:));gmax = max(greenChannel(:));gmean = mean(greenChannel(:));
bmin = min(blueChannel(:));bmax = max(blueChannel(:));bmean = mean(blueChannel(:));
subplot(3,3,7);imshow(A);title('Original Image');
subplot(3,3,2);imshow(greenChannel);title('Green Channel');
subplot(3,3,1);imshow(redChannel);title('Red Channel');
subplot(3,3,3);imshow(blueChannel);title('Blue Channel');
rc=i;rc(:,:,[2,3])=0;subplot(3,3,4);imshow(rc);title('Making GB components zero');
gc=i;gc(:,:,[1,3])=0;subplot(3,3,5);imshow(gc);title('Making RB components zero');
bc=i;bc(:,:,[1,2])=0;subplot(3,3,6);imshow(bc);title('Making RG components zero');
[x,y,z]=size(A);
imgsize=x*y;
disp(imgsize); % To display the size of the image.
```

Name	Value
A	707x770x3 uint8
ans	707x770 uint8
bc	707x770x3 uint8
blueChannel	707x770 uint8
bmax	255
bmean	143.3955
bmin	18
gc	707x770x3 uint8
gmax	255
gmean	143.4584
gmin	26
greenChannel	707x770 uint8
h	707x770x3 uint8
i	707x770x3 uint8
imgsize	544390
j	707x770x3 uint8
rc	707x770x3 uint8
redChannel	707x770 uint8
rmax	255
rmean	147.7330
rmin	24
x	707
y	770
z	3

Source File: 2

- Here is the output:



Before, detailing the code and its output we first answer the questions given in the assignment.

a) What is the number of independent variables?

Ans. Two. There are 2 independent variables which are basically the x & y co-ordinates of the image.

b) What is the number of components?

Ans. Three. There are 3 components which are ***Red, Green and Blue*** (RGB).

c) Display the individual components of the image.

Ans. *5th, 6th and 7th lines* of the above code display the components of the image.

d) Determine the minimum, mean and maximum of each component.

Ans. As we can see in the right side of Source File: 2, the workspace of the above code displays the minimum, mean and maximum of each component.

	Red Channel	Green Channel	Blue Channel
Minimum	24	26	18
Maximum	255	255	255
Mean	147.7330	143.4584	143.3955

e) Determine the size of the image.

<u>Ans.</u> The size of the image is given by the variable named as 'imgsize' in the workspace and it displays the size as 544390.

*(544390=770 x 707)

(iv) Since, we did not wish any overlapping of extracting the individual components, it's better to take the copies of image and perform the operations with and every individual component. Hence, variables h, i & j were chosen.

Then as we can see in the output that the darkness of the face of the person in the picture varies significantly in each and every channel.

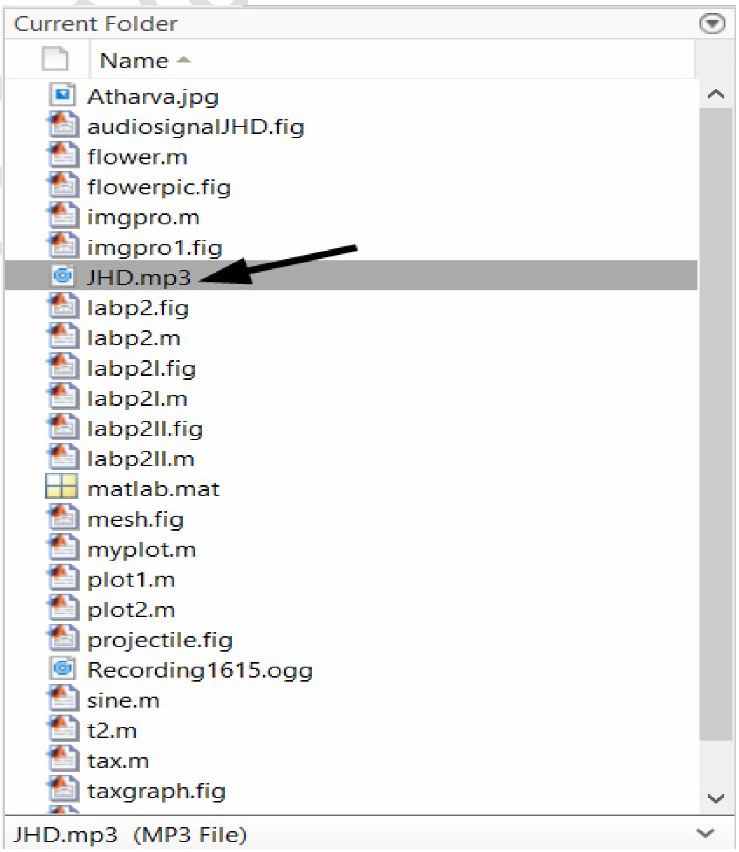
We went one step ahead and tried to find out a way to keep one component in active mode by keeping other two as zero. Lines *15, 16 & 17* of the code in Source File:2 helped in making this happen.

2.Analyzing audio signals in MATLAB

- First, we need to record sound (length ≥ 5 sec) from any device like a smart phone or a laptop or we can also use any audio clip available in the device.
- Then we have to upload it in the software MATLAB and using appropriate function we read the audio clip.
- Finally, we analyse the different aspects of the uploaded sound and based on our observations we report the obtained data from the audio.

(i) Following the instructions, the audio ‘JHD.mp3’ was recorded from a laptop and its length was reported as 5 seconds.

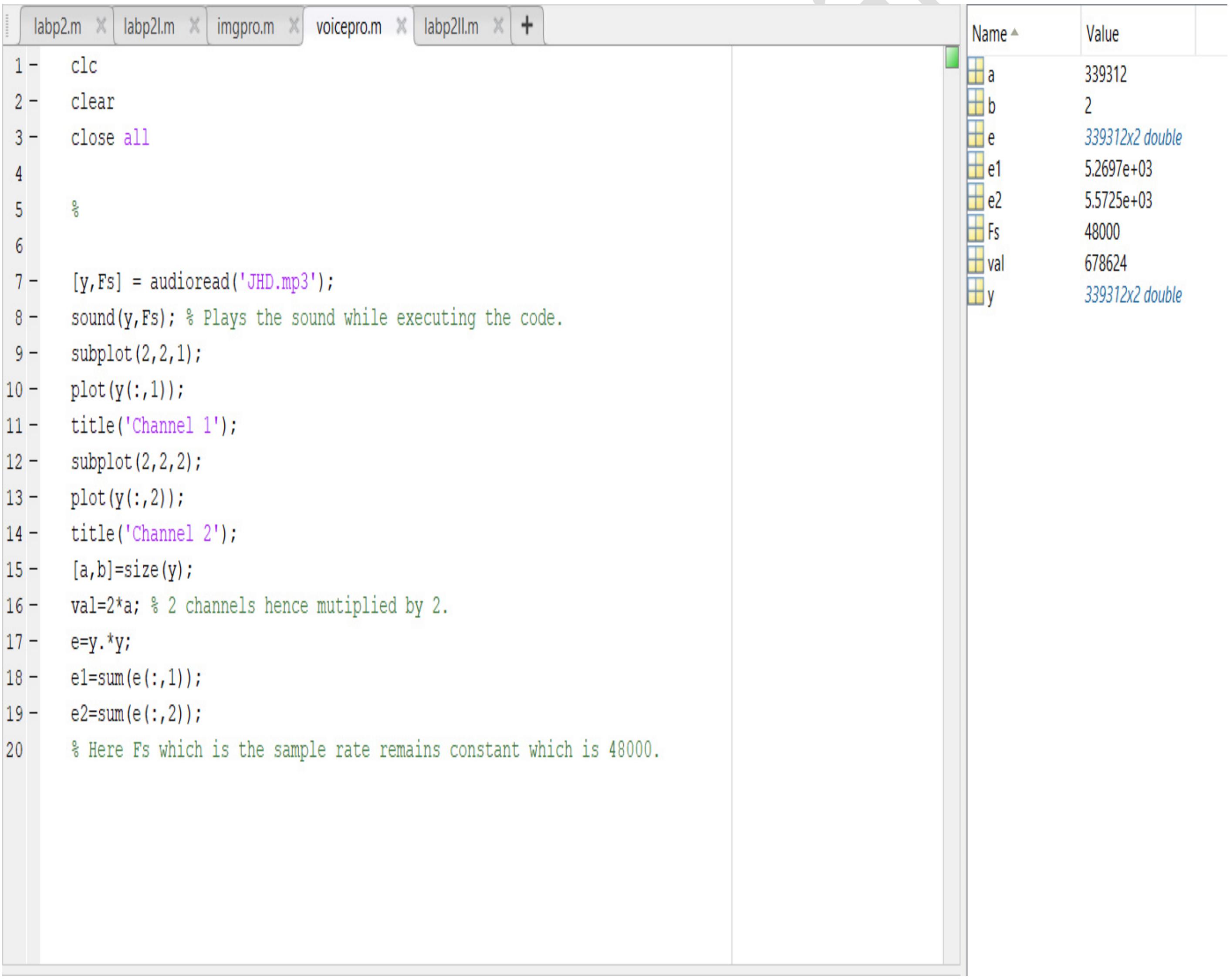
(ii) We then copy ‘JHD.mp3’ and paste it into the folder given in the column at left.



So now, we have uploaded the file and the software is ready to access it.

(iii) From here onwards, we start coding and we first use the function *'audioread(file name)'* to extract all the information from the audio clip and store it in the variable 'y'. The variable 'Fs' here, stores the sample rate of the audio clip.

- Here is the code:



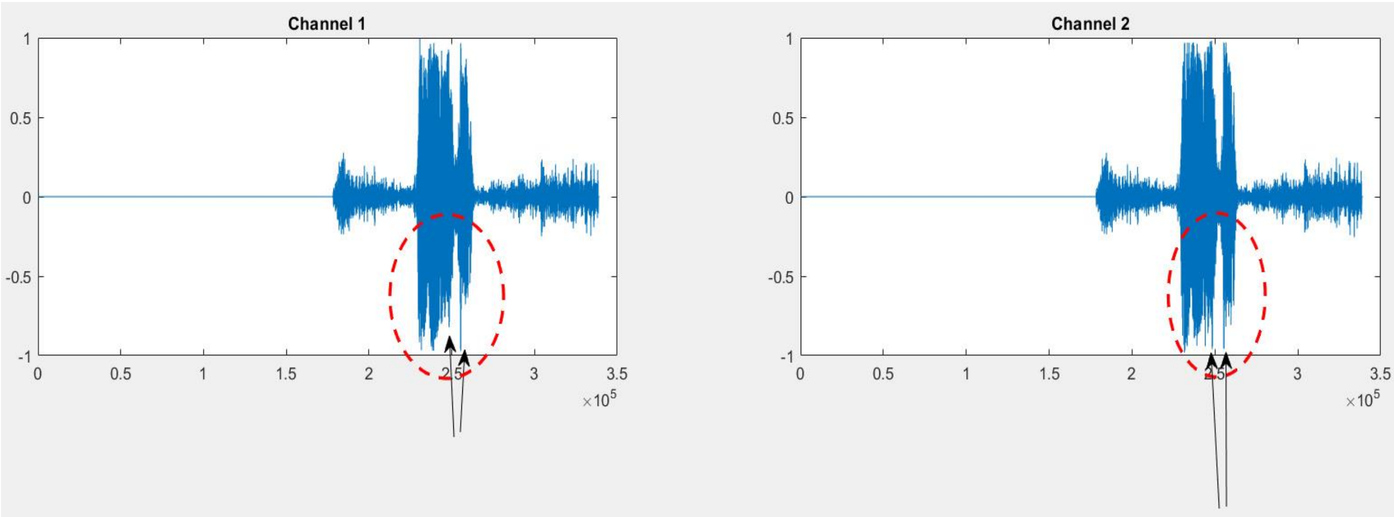

```
clc
clear
close all


%

[y,Fs] = audioread('JHD.mp3');
sound(y,Fs); % Plays the sound while executing the code.
subplot(2,2,1);
plot(y(:,1));
title('Channel 1');
subplot(2,2,2);
plot(y(:,2));
title('Channel 2');
[a,b]=size(y);
val=2*a; % 2 channels hence mutiplied by 2.
e=y.*y;
e1=sum(e(:,1));
e2=sum(e(:,2));
% Here Fs which is the sample rate remains constant which is 48000.
```

Name	Value
a	339312
b	2
e	339312x2 double
e1	5.2697e+03
e2	5.5725e+03
Fs	48000
val	678624
y	339312x2 double

➢ Here is the output from both the channels.



Before, detailing the code and its output we first answer the questions given in the assignment.

a) How many channels are present in the signal?

Ans. Two.

b) Is this signal Digital or Analog?

Ans. The given signal is ***Digital***.

c) What is the number of values in the sequence?

Ans. The total number of values in both the channels is 678624, as shown in the workspace for variable '*val*'.

d) What is the energy of the signal in both the channels?

Ans. Energy of the signal = e1 = 5.2697×10^3 in channel 1

Energy of the signal = e2 = 5.5725×10^3 in channel 2

e) What is the sample rate?

Ans. Sample rate is the total number of samples divided by the total time taken by a signal.

Here Fs is the sample rate which is coming out to be 48000.

(iv) Here, channel 1 and channel 2 are having almost the same readings at different indices but at one instance we can see a significant difference between the two and the same has been represented in the previous image. Also, their energies are almost the same.

Lab Assignment – 01 – Part 2 – Spring 2020

Signal Generation

- Given a signal, we have to plot and display it.
- Moreover, we have to change the time period of the waveform and define another signal for the same.

The given signal is,

$$x(t) = \begin{cases} 1 + t & ; \quad -1 < t < 0 \\ 1 - t & ; \quad 0 < t < 1 \\ 0 & ; \quad \textit{otherwise} \end{cases}$$

(i) First, we have to plot x(t). By correctly constructing the code for the signal, we can easily develop the sketch of the signal x(t).

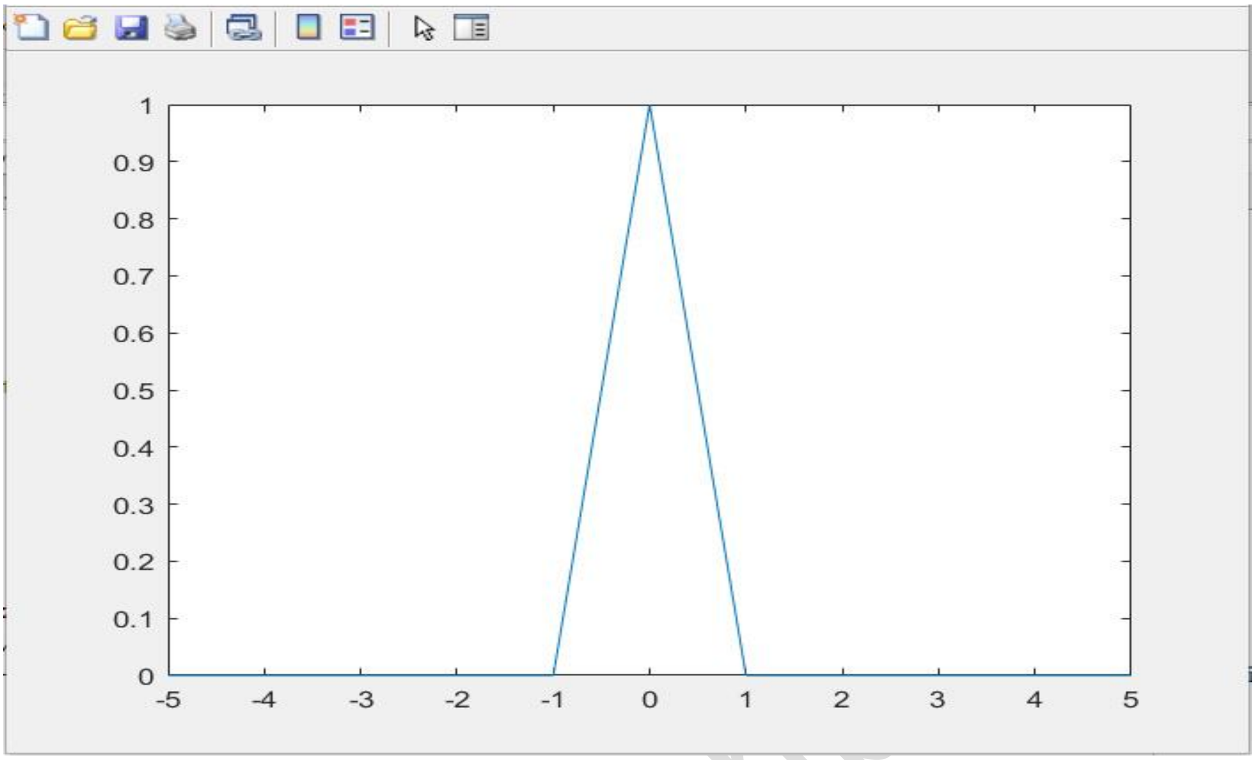
➢ Here is the code:

```
clc
clear
close all

%

m=5;
t=-m:1:m;
x=zeros(size(t));
x(t<=-1 | t>1)=0;
x(t>-1 & t<=0)=1+t(t>-1 & t<=0);
x(t>0 & t<=1)=1-t(t>0 & t<=1);
plot(t,x);
```

- Here is the output:



(ii) The other task assigned to create another signal y(t) which has a fundamental time period of 2.

- One method can be specifying the function for each interval one by one and ultimately displaying the signal. Like, in this way, we code the entire signal x(t) having a fundamental period of 1 and assuming the interval from [-5,5].

imgpro.m | voicepro.m | labp2ll.m | labp2l.m | labp2.m

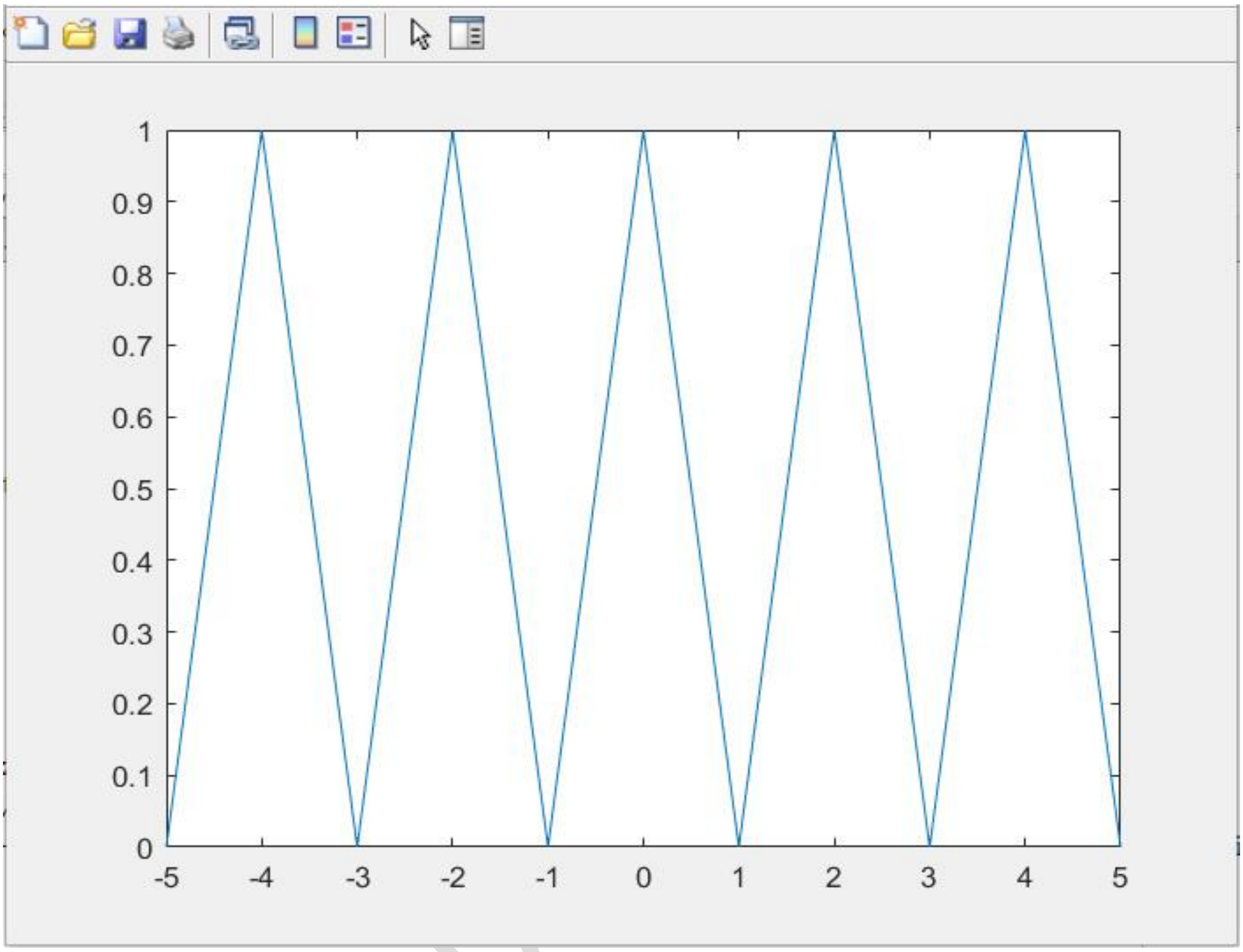
```
clear
close all

%

t=-5:1:5;
x=zeros(size(t));
 x(t>=-5 & t<-4)=5+t(t>=-5 & t<-4);
 x(t>=-4 & t<-3)=5+t(t>=-4 & t<-3);
 x(t>=-3 & t<-2)=3+t(t>=-3 & t<-2);
 x(t>=-2 & t<-1)=3+t(t>=-2 & t<-1);
 x(t>=-1 & t<0)=1+t(t>=-1 & t<0);
 x(t>=0 & t<1)=1-t(t>=0 & t<1);
 x(t>1 & t<=2)=-1+t(t>1 & t<=2);
 x(t>2 & t<=3)=3-t(t>2 & t<=3);
 x(t>3 & t<=4)=-3+t(t>3 & t<=4);
 x(t>4 & t<=5)=5-t(t>4 & t<=5);
% for i=0:length(t)-1
%     x(t>=-n+i & t<-n+i+1)=n+1+t(t>=-n+i & t<-n+i+1);
plot(t,x);
```

Command Window

And this is how the plot looks like:



- The other method is coding the signal y(t) efficiently with the help of for loop. Here is the code for generating the signal y(t) having fundamental period 2.

Editor - C:\Users\atharva deshpande\Documents\MATLAB\labp2ll.m

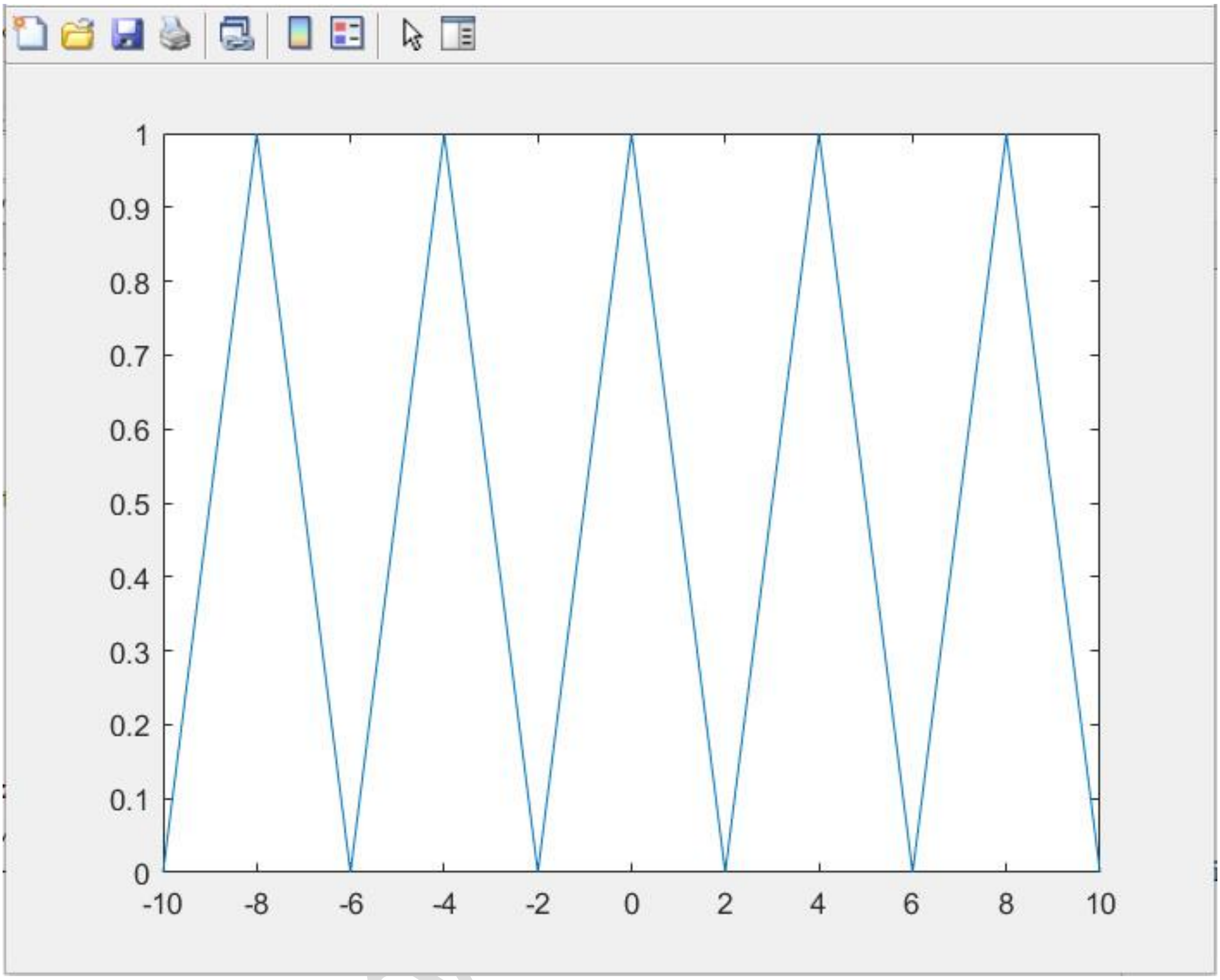
labp2.m | labp2l.m | imgpro.m | voicepro.m | labp2ll.m

```
clc
clear
close all

%
% Note that the value of m = 4n+2 ,n>=0, where m is the required interval.

m=10;
p=2;
k=p-m;
a=k-2;
t=-m:p:m;
y=zeros(size(t));
for i=0:(m/2)-1
    y(t>=k+2*p*i & t<k+(2*p*i)+2)=(-(t(t>=k+2*p*i & t<k+(2*p*i)+2))/2)-(m/2)+2*(i+1);
end
plot(t,y);
```

Here’s the output:



****

END OF FILE

****