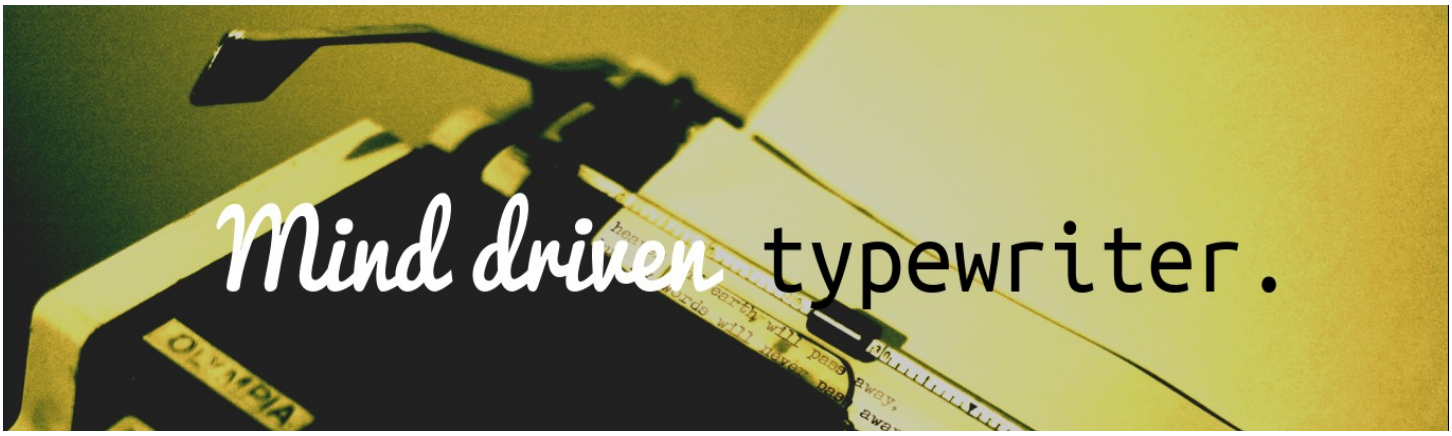
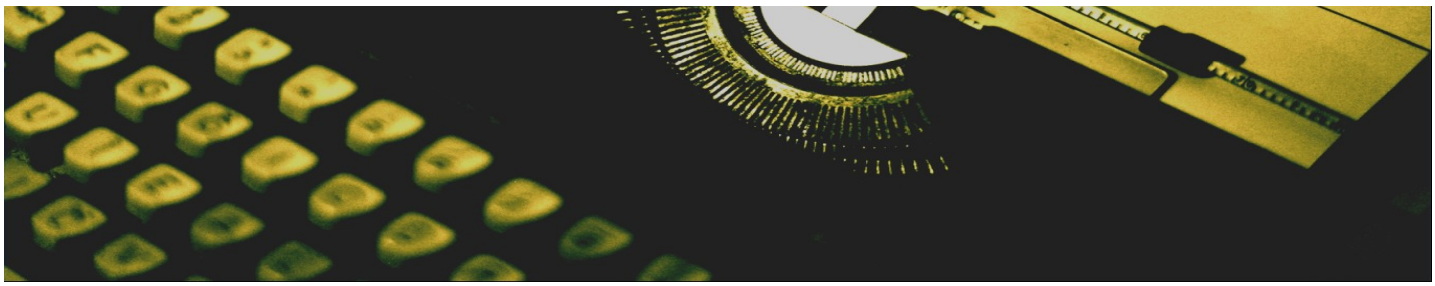


Name – Tirthankar Mittra.

Roll No – 001410701038.

4th yr 2nd semester.
UG-BETCE, Jadavpur University.
(final year seminar report).



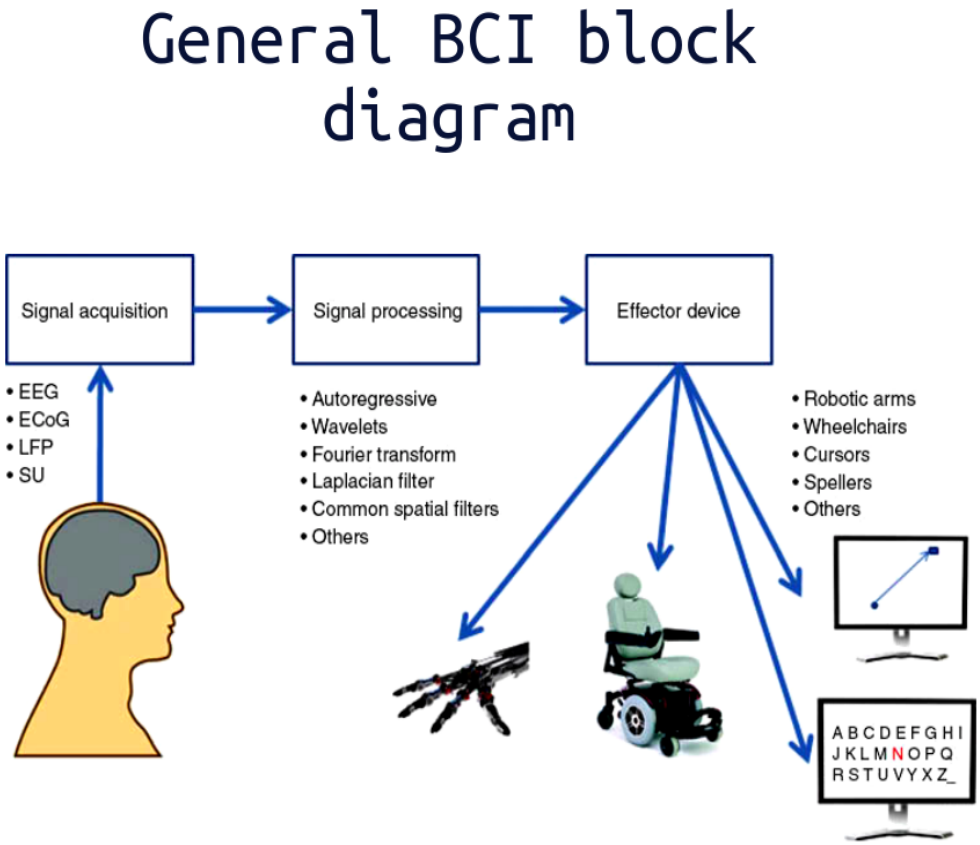
Acknowledgement

I am grateful to Ananda Shankar Chowdhury sir and Mrinal Kanti Naskar sir for letting me present this topic in a seminar.
I am also grateful to Amit Konar sir for guiding us throughout the project.

Contents

1. Introduction to BCI.

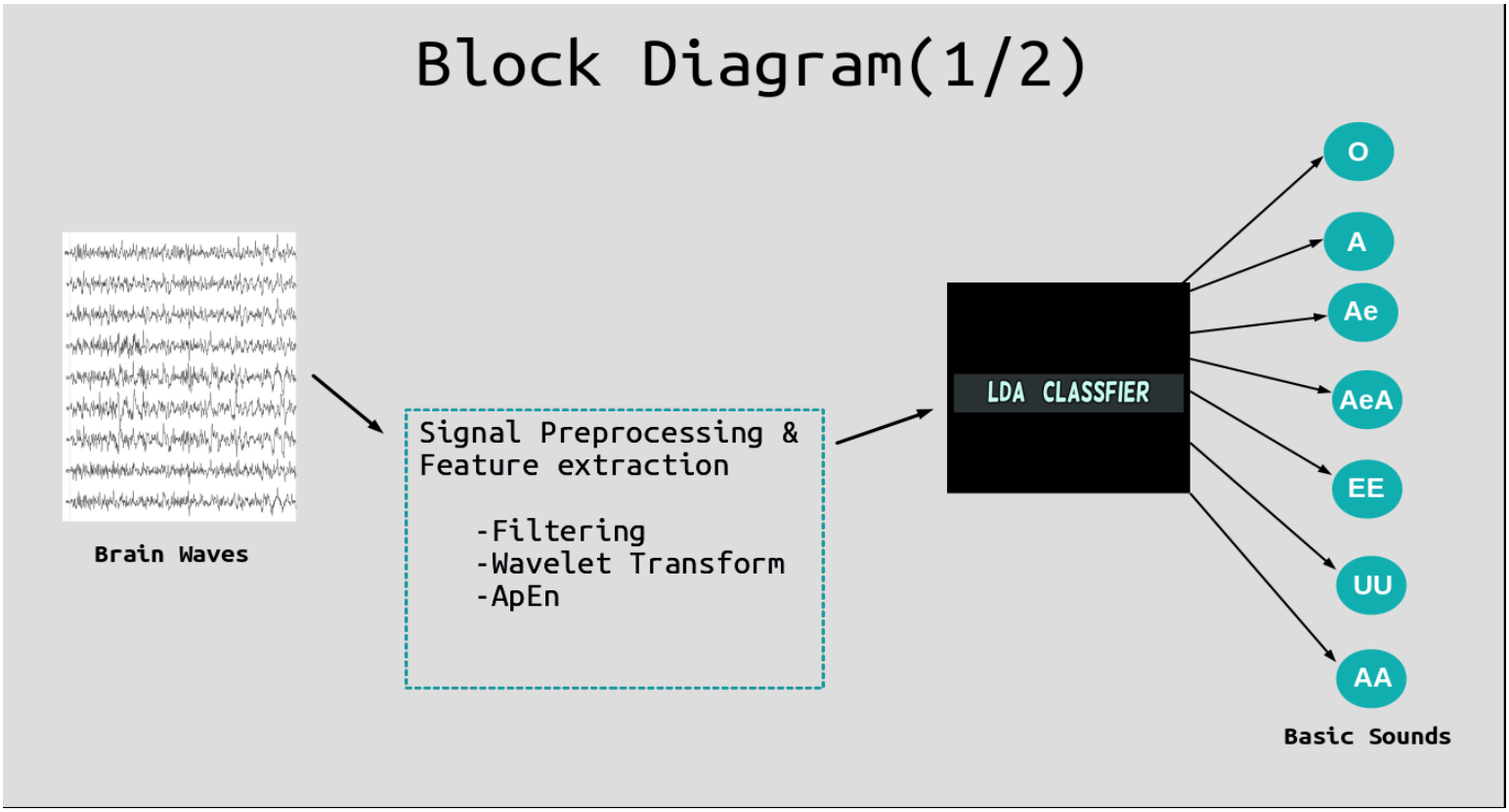
BCI generally stands for **B**rain **C**omputer **I**nterface. It can be said as a collaboration between the brain and a device that enables a device to function through the brain signal.
Example: A prosthetic Limb or a cursor over the screen.



The general steps in a BCI system is represented by the above diagram.
The first step is to acquire the brain signals using methods like EEG and EcoG.
The second step is signal processing, the acquired brain signals are passed through filters, techniques such as wavelet transform, fourier transform are utilized to extract extra information from the EEG signals.
The third step is to use the processed signals to drive electronic devices. **In my case the third step is to drive the typewriter using pre-processed brain signals.**

2. The basic idea.

In this section I'll discuss the basic overview of what I have done using block diagrams.

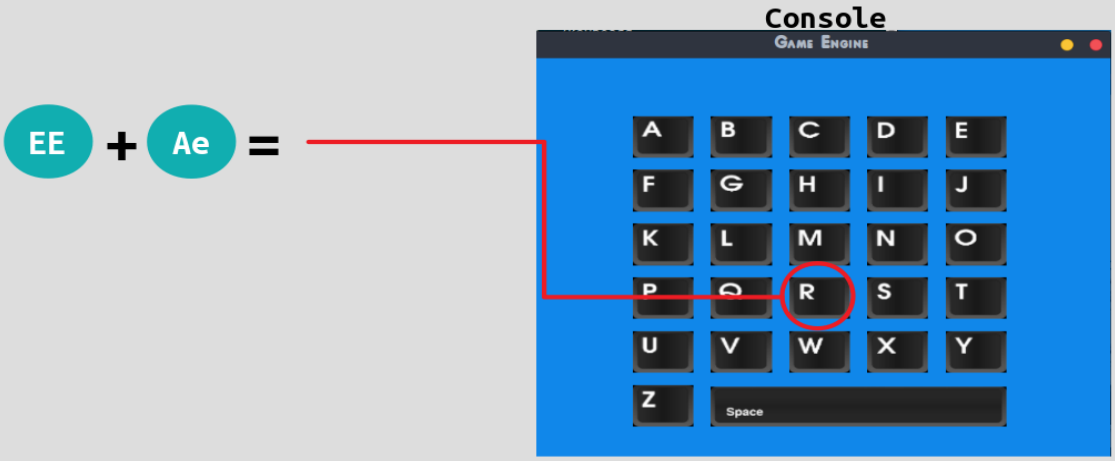


As we can see from the above block diagram, first the brain signals were acquired(EEG has been used here to accomplish this task). Brain signals corresponding to seven basic sounds were extracted these basic sounds are represented by('A','AA','Ae','AeA','EE','UU','O') in the above diagram. In other words the subject would sit infront of a computer screen and when any of these 7 words would appear on the screen the subject would utter the sound silently, In this manner brain signals were acquired.

The second step is the signal pre-processing technique (only filtering was done in this step).

The third step in the block diagram is the correct classification of these seven sounds and for this purpose I have used a LDA classifier.

Block Diagram(2/2)

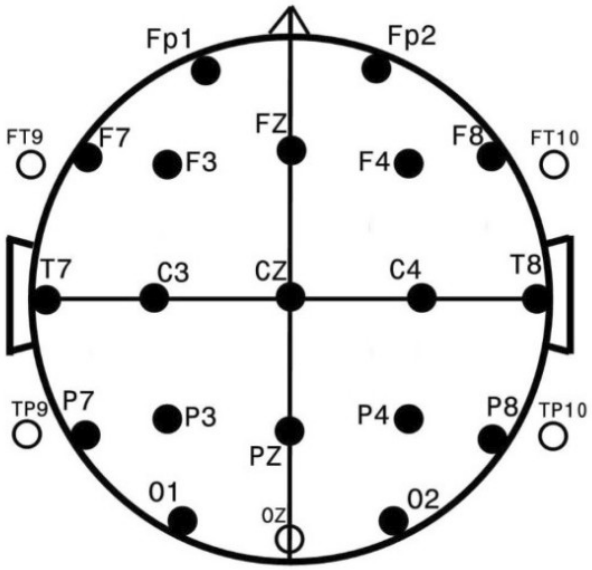


0 Has been reserved for punctuations and other functionalities.

After correct classification, the basic sounds are used to created a console keyboard as shown above. There are seven sounds but we can use six of these to create the above keyboard (6P2 > 27{no. of keyboard elements}) so we can reserve one sound let’s say ‘0’ for future expansion of the above keyboard.

3. EEG and brain parts.

The data was collected on two subjects using EEG electrodes. The 10-20 EEG system was used as shown below:-



Sampling frequency of the device was 500Hz and each subject sat for 2mins for one round of data.
(One round includes all the seven sounds shown on the computer screen in succession with rest period of 10sec in between).
Therefore the total amout of data that collected was per round was huge.(500*120*19 =~10^6).
So our immediate objective becomes to reduce the amount of data.
We can exclude many EEG electrodes based on the various functions of brain lobes which were previously identified.

1.It has been known for about a century that inner speech is accompanied by tiny muscular movements in the larynx, detectable by a technique known as electromyography. In the 1990s, neuroscientists used functional neuroimaging to demonstrate that areas such as the left inferior frontal gyrus (Broca's area), which are active when we speak out loud, are also active during inner speech.
So we take F3 and F7 electrodes(these cover the Broca's area).

2.Brain signals from the cortical region can be used to detect slight muscle movememts when the words are silently uttered. **So we take electrodes C3 C4 and Cz**

3.We also take parietal lobe because it is used for language processing.**(EEG electrodes P3 P4 and Pz were taken)**
In conclusion we are taking electrodes F3 F7 C3 C4 Cz P3 P4 Pz.

Finally the output form the EEG will look something like(as shown below) in a standard text editor.

```
12.41   -0.51   7.74   -8.97   3.50   -5.62   -3.44   -3.44   -12.18   -7.16   9.20   -0.75   7.65   -4.30   -6.07   -10.95   0.44   1.
02   -6.76
12.91   -1.71   8.14   -10.24   3.72   -5.08   -1.51   -2.94   -10.24   -6.48   7.42   -0.96   8.16   -4.07   -4.96   -11.39   0.53   1.
77   -5.24
13.77   -2.95   8.93   -11.11   3.85   -4.84   0.15   -3.03   -9.04   -6.53   6.46   -1.66   8.35   -4.14   -4.55   -12.34   0.76   2.
21   -4.23
13.95   -4.32   9.26   -11.58   4.30   -4.57   1.29   -3.12   -7.82   -6.41   6.15   -2.94   8.07   -4.43   -4.26   -13.04   0.83   2.
39   -3.48
14.26   -5.76   9.65   -11.87   5.00   -4.36   2.72   -2.84   -5.73   -5.84   5.53   -3.82   7.99   -4.38   -3.31   -13.29   0.86   2.
92   -2.41
14.09   -6.95   9.68   -12.04   5.19   -4.56   3.76   -2.69   -3.61   -5.17   4.81   -4.49   7.78   -4.29   -2.33   -13.35   0.88   3.
24   -1.56
13.82   -7.16   9.58   -11.61   5.62   -4.38   4.15   -2.46   -2.15   -4.50   4.86   -4.78   7.32   -3.97   -1.38   -13.31   1.18   3.
84   -1.12
13.17   -6.98   9.24   -11.00   6.52   -3.70   4.15   -2.15   -1.02   -3.52   5.35   -4.81   6.88   -3.14   -0.92   -13.04   1.83   4.
78   -1.09
12.47   -7.05   8.63   -11.20   6.61   -4.08   3.32   -2.82   -1.75   -3.67   4.45   -5.52   6.01   -2.91   -2.10   -13.48   1.87   4.
78   -1.99
11.65   -6.77   8.03   -10.95   6.61   -4.32   2.76   -2.97   -2.27   -3.67   3.74   -5.74   5.58   -2.45   -2.94   -13.14   1.99   4.
75   -2.53
10.97   -6.79   7.31   -11.17   6.63   -4.54   2.31   -2.97   -2.32   -3.81   3.16   -5.71   4.98   -1.86   -3.44   -12.78   1.81   4.
82   -3.20
```

4. Signal Preprocessing

The data collected from EEG was corrupted by background sounds, eye blinking, eye movement and small muscle twitching while the data was collected. The noise contributed from eye blinking and eye movement was removed by passsing the wave through a band-pass filter of order 10.

CODE FOR 'FILTERING' FUNCTION

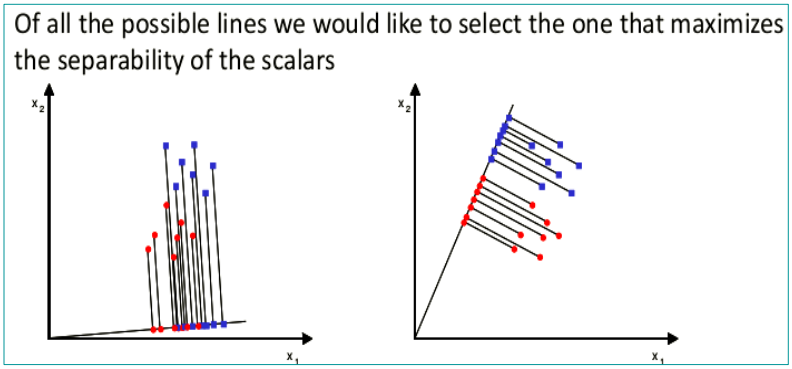
```
% This function returns the filtered output of a
given (input) signal
% Fs = 500 Hz, Fpass1 = 3 Hz, Fpass2 = 13 Hz,
Apass = 1, Order = 10

function [y] = filtering(x)
x=data';
BandPassSpecObj = fdesign.bandpass('N,Fp1,Fp2,Ap',
10, 13, 30, 1, 128);
BandPassFilt = design(BandPassSpecObj, 'cheby1');
y = filter(BandPassFilt, x);
plot(y)
return
```

There was an attempt to reduce the background noise as much as possible when the data was recorded and the subject was told to remain calm.

5. LDA Classifier.

LDA seeks to reduce the dimensionality while preserving much of the class discriminatory information. LDA can also be used as a normal classifier simultaneously plus its simple to code, so initially we have decided to use this classifier.



From the above diagram we can see that the points dependent on two dimensions are projected on one dimension. In the second case the class separation is maintained and yet we have also reduced the dimension of the input data. To acheive this our objective becomes to minimize the separation of data within each class and maximize the separation between any two classes after projection on a lower dimension. Based on the above logic our cost function becomes.

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|}$$

SB(bar)= W*SB*W'.
SW(bar)= W*SW*W'.
[where SW and SB are within class and between class covariance.]

Note:
To learn more about LDA follow the link
http://www.facweb.iitkgp.ernet.in/~sudeshna/courses/ml08/lda.pdf

My multiclass LDA classifier follows the algorithm shown below which I have developed:

Algorithm:

-Find covariance matrix Si for i/p vectors of K classes.

-Find (within class scatter) $S_W = \sum_{i=1}^{C} S_i$

-Find covariance matrix Sb for o/p vector y.

-Solution of $\frac{d}{dw}[J(w)] = 0$ is (Sw^-1*Sb)'s eigen vectors.

-Take the largest "C-1" eigen value's eigen vector $W = [w_1|w_2|\dots|w_{C-1}]$

-After training, let "x" be a new i/p and "yi" be the corresponding o/p which has been found using the formula given below:

$$y_i = w_i^T x \Rightarrow y = W^T x$$

-Assign class "j" to i/p "x" if |yi- uj| is minimum for a "j".

In order to make a successful classification:-

1. Extract the data and reduce the size of the input data by sending it to the function cutData( ), then call train( ) to train the classifier and after that we call the predict( ) to display the result.

Code:

```
clear;close all;clc;
A=cutData(csvread('A3.xls'));
AA=cutData(csvread('AA3.xls'));
Ae=cutData(csvread('Ae3.xls'));
AeA=cutData(csvread('AeA3.xls'));
EE=cutData(csvread('EE3.xls'));
UU=cutData(csvread('UU3.xls'));
O=cutData(csvread('O3.xls'));
[ideal_y,W]=train();
disp('A->1   AA->2   Ae->3   AeA->4   EE->5   UU->6   O->7');
no=predict(ideal_y,W'*mean(O)');
disp(no);
```

2. cutData( ) selects data from columns **F3 F7 C3 C4 Cz P3 P4 Pz**, it also averages every 10 samples from the original input file. If for example the dimension of the input file is 7500*19 after passing through cutData the dimension of the input file reduces to 750*8.

Code:

```
function ret=cutData(A)
  ret1=A(:,[3,5,6,7,8,11,18,19]);
  n=size(A,1)-1;
  su=0;
  f=0;
  for i=1:n
    if rem(i,10)!=0
         su=su+ret1(i,:);
    end
    if rem(i,10)==0
      su=su+ret1(i,:);
      su=su/10;
      if f==1
        ret=[ret;su];
      end
      if f==0
        ret=su;
        f=1;
      end
      su=0;
    end
  end
endfunction
```

3. Taking data from subjects is a time consuming activity so for training the classifier we have taken two sets of data from the same subject and genrated new data by getting a random number between these limits **(in future we are clearly planning to change this shortcoming but recording lots of data).** The training is done in accordance the theory previously discussed. From the train function we get the **W** matrix and the **ideal_y(self explanatory name)** matrix.

Code:

```
function [ideal_y,W]=train()
  A=cutData(csvread('A1.xls'));
  A1=cutData(csvread('A.xls'));
  AA=cutData(csvread('AA1.xls'));
```

```
AA1=cutData(csvread('AA.xls'));
Ae=cutData(csvread('Ae1.xls'));
Ae1=cutData(csvread('Ae.xls'));
AeA=cutData(csvread('AeA1.xls'));
AeA1=cutData(csvread('AeA.xls'));
EE=cutData(csvread('EE1.xls'));
EE1=cutData(csvread('EE.xls'));
UU=cutData(csvread('UU1.xls'));
UU1=cutData(csvread('UU.xls'));
O=cutData(csvread('O1.xls'));
O1=cutData(csvread('O.xls'));
for h=1:15
  for i=1:8
    a=floor(mean(A1)(i));
    b=floor(mean(A)(i));
    c=max(a,b);
    a=min(a,b);
    c1(h,i)=randi([a,c])+rand();
  end
  for i=1:8
    a=floor(mean(AA1)(i));
    b=floor(mean(AA)(i));
    c=max(a,b);
    a=min(a,b);
    c2(h,i)=randi([a,c])+rand();
  end
  for i=1:8
    a=floor(mean(Ae1)(i));
    b=floor(mean(Ae)(i));
    c=max(a,b);
    a=min(a,b);
    c3(h,i)=randi([a,c])+rand();
  end
  for i=1:8
    a=floor(mean(AeA1)(i));
    b=floor(mean(AeA)(i));
    c=max(a,b);
    a=min(a,b);
    c4(h,i)=randi([a,c])+rand();
  end
  for i=1:8
    a=floor(mean(EE1)(i));
    b=floor(mean(EE)(i));
    c=max(a,b);
    a=min(a,b);
    c5(h,i)=randi([a,c])+rand();
  end
  for i=1:8
    a=floor(mean(UU1)(i));
```

```
      b=floor(mean(UU)(i));

      c=max(a,b);
       a=min(a,b);
       c6(h,i)=randi([a,c])+rand();
    end
    for i=1:8
      a=floor(mean(O1)(i));
      b=floor(mean(O)(i));
      c=max(a,b);
      a=min(a,b);
      c7(h,i)=randi([a,c])+rand();
    end
  end
  N=15;%15 training examples for each class.

meuVV=[mean(c1);mean(c2);mean(c3);mean(c4);mean(c5);mean(c6);
mean(c7)];

Sw=cov(c1,c1,1)+cov(c2,c2,1)+cov(c3,c3,1)+cov(c4,c4,1)+cov(c5
,c5,1)+cov(c6,c6,1)+cov(c7,c7,1);
  meuAll=mean(meuVV);
  Sb=zeros(8,8);
  for i=1:7
      tt=((meuVV(i,:)-meuAll)'*(meuVV(i,:)-meuAll));
      tt=N.*tt;
      Sb=Sb+tt;
   end
  tm=inv(Sw)*Sb;
  [W lambda]=eig(tm);
  ideal_y=zeros(7,8);
  for i=1:7
    ideal_y(i,:)=W'*meuVV(i,:)';%row 1 is class1, row 2 is
class2...
  end
endfunction
```

4. In the final step we predict the test data. The following line: `no=predict(ideal_y,W'*mean(O)');` does the job. ‘O’ in the above code line is the user input. The predict() compares to which class W’*mean(O)’ is most closely related to. The result is returned in variable no.

Code:

```
function no=predict(ideal_y,y)
  res=0;
  for i=1:7
      dist=0;
      for j=1:8
        dist=dist+abs(ideal_y(i,j)-y(j));
      end
      if(i==1)
        res=dist;
      end
      if(res>=dist)
        res=dist;
        no=i;
      end
  end
endfunction
```

The ouput corresponding to `no=predict(ideal_y,W'*mean(O)');` is shown below:

```
A->1   AA->2   Ae->3   AeA->4   EE->5   UU->6   O->7
 7
>>
```

The input ‘O’ has been correctly identified to class 7.

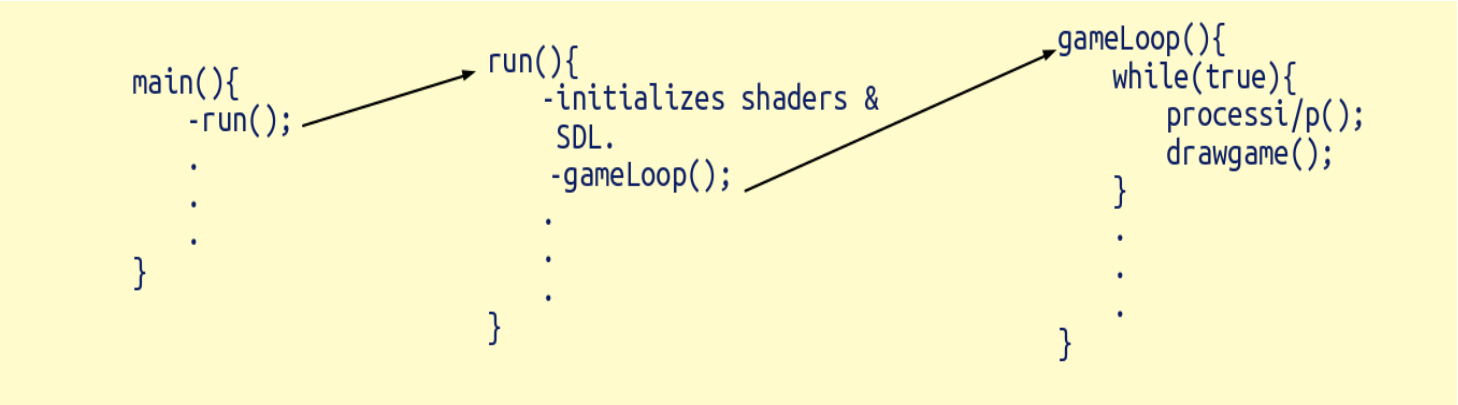
5. Interactive Console Program

The interactive cosole or keyboard was designed using SDL and OpenGL.

-Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D.

-Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics.

The Program:-



The above diagram shows the flow of control in the console program(i.e. how the keyboard operates). The gameLoop() has an infinite loop which continuously calls two functions processi/p() and drawgame(). processi/p() looks for the output from LDA classifier which is generated after the subject utters the sound. After the utterance the modified keyboard is drawn by drawgame().
drawgame() gives the effect of letters blinking on the live keyboard. The result is also stored in a text file.

7. What’s next? How can the typewriter be improved?

The next step involves measuring EEG signals using more number of electrodes and also changing the classifier to a Neural Network.

8. Conclusion

When the experiment using the above setup was conducted it was noted that classification for different subjects were different and for each subject sounds EE and O were identified with high accuracy, sounds A and AA had moderate classification accuracy and the rest of the sounds did a little better than chance(i.e no practical imporatnce for the rest of the sounds).

References

1. https://www.theguardian.com/science/blog/2014/aug/21/science-little-voice-head-hearing-voices-inner-speech

2. This project was implemented using ideas drawn from a paper written previously by Prof. Amit Konar of Jadavpur University(paper’s name -> Mind Driven Type-writer) and also from other knowledge repositories on the web.