MATLAB PROGRAMS

1.1 FAST FOURIER TRANSFORMATION

```
close all; clear all;

%clear system of all variables

Clc

%import data to be analyzed

A = importdata('C:\Users\Sudip Paul\Desktop\Control.txt');

%get the number of rows and columns of imported data

[m n]= size(A);

N =n;

M=m;

for i =1:N

X = A(:,i);

Y=X(1:2000);

a = [1 -2.2137 2.9403 -2.1697 0.9606];

Y1= filter(1,a,Y);

%pwelch extracts the power spectrum density from the filtered

%vector Y1

[Pxx,F]=pwelch(Y1,256,[],512,512);

figure(i)

plot(F(1:100),Pxx(1:100))

c(i,:)=Pxx(1:100);

L = c';

end

dlmwrite('after.txt', L, 'delimiter', '\t', 'precision', 3)
```

1.2 LYAPUNOV EXPONENT

```
function lyac2
%Read data set. The user may put his own path where his/her data
%is stored. In mathematics the Lyapunov exponent or Lyapunov
%characteristic exponent of a dynamical system is a quantity
%that characterizes the rate of separation of infinitesimally
%close trajectories.
%Note that the EEG data is read into the vector x. The idea is
%to track the continuous evolution of the system.
%The evolution is measured as the square of the distance from an
%initial point.
%The evolution of the data is measured from the first sample of
%256 points.
x= xlsread('G:\SUDIP EEG SIGNALS\group 24\td8.xls','A3:EA2562');
%We now determine the number of elements of x
N=length(x);
%here embedded dimension is m i.e. equal to 2
m=2;
%In all the experiments the sampling rate was 256
tao = 256;
%This is the number of iterations to use
maxiter = 1000;
%This is the parameter to determine the distance between
%points in the EEG data
```

```
meanperiod = 1.0;
M=N-(m-1)*tao;
Y=psr_deneme(x,m,tao);
for i=1:M
%ones(M,1)is MX1 matrix of ones
%x0 is the initial point from which the distance is computed
%The starting value is the first point of the EEG data
x0=ones(M,1)*Y(i,:);
%the distance is computed by using the formula:
%d = sqrt((x1-x2)^2+(y1-y2)^2)
distance=sqrt(sum((Y-x0).^2,2));
for j=1:M
if abs(j-i)<=meanperiod
distance(j) = 1000;
end
end
%We have so far computed the “distance” of the initial state x0
%from the rest of the points in the EEG data. Out of the set of
%distances so computed we first find the minimum using the
%matlab function min. The index for which this minimum occurs is
%assigned to the vectors neardis and nearpos.
[neardis(i) nearpos(i)]=min(distance);
end
for k=1:maxiter
```

```
maxind=M-k;
evolve = 1;
pnt=0;
for j=1:M
if j<=maxind && nearpos(j)<=maxind
dist_k=sqrt(sum((Y(j+k,:)-Y(nearpos(j)+k,:)).^2,2));
if dist_k~=0
```

%Now the Lyapunov coefficient is defined as $\lambda_i = \xrightarrow{t\to\infty} \frac{1}{t}\log_2 \frac{\left\|\partial_{x_i}(t)\right\|}{\left\|\partial_{x_i}(0)\right\|}$

```
%The program now evaluates the Lyapunov coefficient as per the
%above definition. First the %numerator is computed. Note that
%in Matlab log is taken to the base of 2.
evolve=evolve+log(dist_k);
%The following statement takes care of the x increment which is
%needed to compute the denominator
pnt=pnt+1;
end
end
end
%This code computes the Lyapunov coefficient
if pnt > 0
d(k)=evolve/pnt;
else
d(k)=0;
```

```
end

end

figure

plot(d)

fs=256;

%As per the Lyapunov exponent vs tlinear. Only for the range

%15:78 we find that the Lyapunov exponent is minimum.

tlinear=15:78;

%For linear plots polyfit gives the slope

F = polyfit(tlinear,d(tlinear),1);

%This prints out the lyapunov coefficient

lle = F(1)*fs

function Y=psr_deneme(x,m,tao,npoint)

N=length(x);

if nargin == 4

M=npoint;

else

M=N-(m-1)*tao;

end

%Y=zeros(M,m) is a MXm matrix of zeros. Y is basically the x

%vector i.e. the EEG signal

Y=zeros(M,m);

for i=1:m
Y(:,i)=x((1:M)+(i-1)*tao)';
end
```

1.3 CORRELATION DIMENSION

```
function correlation
%This function calculates the correlation dimension. The
%correlation dimension is given by the slope(ν)of the plot of
%log(C(r) vs log(r)), where C(r) is the correlation function and
%r is a measure of how many points lie within a certain distance
%of a reference point say V (Grssberger et. al (1983)).
for ii = 1:3
%Enter path for control
%The user may enter the path where his/her data set is stored
if (ii == 1)
D = xlsread('C:\Users\Sudip Paul\Desktop\fpc1.xls','A3:FX2562');
end
%Enter path for stroke
%The user may enter the path where his/her data set is stored
if (ii == 2)
D = xlsread('C:\Users\Sudip Paul\Desktop\fps1.xls','A3:FX2562');
end
%Enter path for drug
%The user may enter the path where his/her data set is stored
if (ii == 3)
D = xlsread('C:\Users\Sudip Paul\Desktop\fpd1.xls','A3:FX2562');
end
```

```
x = D(:);
%M contains the number of elements of x
M = numel(x);
tau = 256;
k = 1;
for m = 2:2:20
sumi = 0.0;
for i = 1:M
sumj = 0.0;
for j = 1:m-1
if ((i+j*tau) < M)
sumj = sumj+ abs(x(i+j*tau)-x(i));
end
end
sumi = sumi + (sumj)/M;
sumj = 0.0;
S(i) = sumi;
xx(i) = i;
end
%Note that in MATLAB log is to the base 2. The formula for
%correlation dimension requires log to base of 2
ymax = max(log(S));
ymin = min(log(S));
xmax = max(log(xx));
```

```
xmin = min(log(xx));

slope = (ymax - ymin)/(xmax -xmin);

MM(1,k) = m;

MM(2,k)= slope;

k = k+1;

end

if (ii == 1)

dlmwrite('C:\Users\Sudip Paul\Desktop\outxx',MM);

end

if (ii ==2)

dlmwrite('C:\Users\Sudip Paul\Desktop\outxx',MM,'-

append','newline','pc');

end

if (ii ==3)

dlmwrite('C:\Users\Sudip Paul\Desktop\outxx',MM,'-

append','newline','pc');

end

end

end
```

1.4 AUTOCORRELATION DIMENSION

```
function eegcorr
%Note that the entire EEG of one particular region of any
%condition
%Specify the path of the data in xls format. If the data is in
%some other format (say text) use dlmread
filename = 'C:\Users\Sudip Paul\Desktop\td7.xls';
%read the entire dataset from the excel file
%Note that the data in this case extended from columns A3 to
%FX2562
D = xlsread(filename,'A3:FX2562');
%This statement catenates all the columns into one column
A = D(:);
%This statement gives the number of rows and columns in A
size(A);
%autocorr is a built-in function of MATLAB which determines the
%autocorrelation function of a time series. The first argument
%is the column vector of the time series. The second argument
%is the "lag". This is the extent of the set into which the time
%series is divided and the auto correlation is computed
autocorr(A,100)
end
```

1.5 NEURAL NETWORK

```
function nnmodel
%In this program a feed forward neural network is setup using
%the "newff" function of MATLAB. The feed forward network is
%trained with EEG spectra corresponding to control, stroke
%induced and drug administered cases. The trained neural network
%is tested with target data to test if it is able to recognize
%and classify correctly or not. To characterize an EEG spectra
%we have used the DWT coefficients (D1-D4) and approximation
%(A4). Note that D1-D4 and A4 are represented as maxima, minima,
%mean and standard deviation of the relevant spectra. The input
%data hence contains the above parameters for the relevant
%spectra.
%open the file containing the DWT coefficients of all 3 regions
%rat brain EEG data for one rat.
fid = fopen('C:\Users\Sudip Paul\Desktop\rat1.txt','r');
%The variable numline will contain the number of lines of data
%It is initialized to zero
numline = 0;
%Every call to fgetl reads a line of text from the target data
%file
tline = fgetl(fid);
%We want to compute how many lines of data are present
%The number of lines of data is put in the variable numline
```

```
%The while loop reads as long as there are characters in the
%line. The while loop terminates when it encounters a blank
%line.
while ischar(tline)
tline = fgetl(fid);
numline = numline+1;
end
%Close the data file
fclose(fid);
%Now divide the number of lines into sets with each set
%containing five lines (D1-D4 and A4 corresponding to each EEG
%spectra. Note that corresponding to each row (D1-D4, A4) four
%statistical parameters Maximum, Minimum, Mean and Standard
%Deviation for each spectra were taken.
%Each EEG spectra is characterized by 5 parameter sets (D1-D4
%and A4). Hence we have to know how many such data sets (or sets
%of 5) are present in the input data. This is accomplished via
%the statement "round", which rounds off the result of
numline/5.
nset = round(numline/5);
%The variable x is initialized to null
x =' ';
%Open the data file again
fid = fopen('C:\Users\Sudip Paul\Desktop\rat1.txt','r');
```

```
%Now we want to loop nset times, where nset is the number of
%sets of data, each dataset containing 5 lines of data
for i = 1:nset
%read 5 lines of data
%Note that the data is in ASCII format. Hence it has to be
%trimmed using strtrim which removes leading and trailing
%spaces. fgets reads a line of data. Str2num converts string
%to number
x1 = str2num(strtrim(fgets(fid)));
x2 = str2num(strtrim(fgets(fid)));
x3 = str2num(strtrim(fgets(fid)));
x4 = str2num(strtrim(fgets(fid)));
x5 = str2num(strtrim(fgets(fid)));
%Next the data read so far is catenated horizantally using
%horzcat into one line
%Here x1(1): Maximum; x1(2): Minimum; x1(3): Mean; x1(4):
%Standard Deviation
%We want to concatenate x1(1), x1(2), x1(3), x1(4), x1(5) and
%assign the concatenated variable to I1. This is done below
I1 = horzcat(num2str(x1(1)),'  ',num2str(x1(2)),'
',num2str(x1(3)),'  ',...
num2str(x1(4)),';');
%We similarly concatenate the values of x2(1), x2(2), x2(3),
%x2(4), x2(5) and assign it to I2
```

```
I2 = horzcat(num2str(x2(1)),'  ',num2str(x2(2)),'
',num2str(x2(3)),'  ',...
num2str(x2(4)),';');
%We similarly concatenate the values of x3(1), x3(2), x3(3),
%x3(4), x3(5) and assign it to I3
I3 = horzcat(num2str(x3(1)),'  ',num2str(x3(2)),'
',num2str(x3(3)),'  ',...
num2str(x3(4)),';');
%We similarly concatenate the values of x4(1), x4(2), x4(3),
%x4(4), x4(5) and assign it to I4
I4 = horzcat(num2str(x4(1)),'  ',num2str(x4(2)),'
',num2str(x4(3)),'  ',...
num2str(x4(4)),';');
%We similarly concatenate the values of x5(1), x5(2), x5(3),
%x5(4), x5(5) and assign it to I5
I5 = horzcat(num2str(x5(1)),'  ',num2str(x5(2)),'
',num2str(x5(3)),'  ',...
num2str(x5(4)),';');
I = horzcat(I1,I2,I3,I4,I5);
I = str2num(I);
input = I;
%The input constructed so far is in the form a row. Since the
%input has to be in the form of a column we take the transpose
%of the input
```

```
input = input';

output = [1 2 3 4 5 ];

%Newff Create feed-forward back propagation network.

%newff(P,T,[S1 S2...S(N-l)],{TF1 TF2...TFNl},

%BTF,BLF,PF,IPF,OPF,DDF)

%Here P:input and T: Target

%Here we have used a hidden layer of 20,000 neurons and an

%output layer of one neuron.

%The transfer function for training the neurons of hidden layer

%is 'tansig' and for output layer 'purelin' is used.

net = newff(minmax(input),[20000,1],{'tansig'

'purelin'},'trainscg');

%Number of epochs used to train the neural network. One can

%visually increase/decrease this parameter since a graph of the

%mean square error vs. Training epoch is generated.

net.trainParam.epochs = 500;

%This code actually initiates the training of the network.

%Training of the network means adjustment of the connection

%weights so that optimal results are obtained. The training is

%done by the software itself

net = train(net,input,output);

%Our network is now trained. Now we have to test it with target

%data to test the network to see that it recognizes the data. We
```

```
%chosen stroke condition of the Fronto Parietal region the
%approximation coefficient of the DWT
inp1 = [-669    -7881   -2096   1116;]';
msgbox(strcat('set number: ', num2str(i)),'Current Set')
%The selected input data was used as input to the trained
%network
outputPredicted = sim(net, inp1)
%As output we just want to see the data set number. In our case
%this belongs to data set number 1.
if (round(outputPredicted) == 1)
msgbox(strcat('set:',num2str(i),'data set number:',
num2str(round(outputPredicted))),'Match Found')
break;
elseif(round(outputPredicted) == 2)
msgbox(strcat('set:',num2str(i),'data set number:',
num2str(round(outputPredicted))),'Match Found')
break;
elseif(round(outputPredicted) == 3)
msgbox(strcat('set:',num2str(i),'data set number:',
num2str(round(outputPredicted))),'Match Found')
break;
elseif(round(outputPredicted) == 4)
msgbox(strcat('set:',num2str(i),'data set number:',
num2str(round(outputPredicted))),'Match Found')
```

```
break;

elseif(round(outputPredicted) == 5)

msgbox(strcat('set:',num2str(i),'data set number:',

num2str(round(outputPredicted)),'Match Found')

break;

end

end

end
```