APPENDIX-B
function varargout = trial2(varargin)
% TRIAL2 M-file for trial2.fig
% TRIAL2, by itself, creates a new TRIAL2 or raises the existing
% singleton*
%
% H = TRIAL2 returns the handle to a new TRIAL2 or the handle to
% the existing singleton*
%
% TRIAL2('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in TRIAL2.M with the given input %
% arguments.
%
% TRIAL2('Property','Value',...) creates a new TRIAL2 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before trial2_OpeningFunction gets called. An
% unrecognized property name or invalid value makes property %
% application
% stop. All inputs are passed to trial2_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
% only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIDATA

% Edit the above text to modify the response to help trial2

% Last Modified by GUIDE v2.5 15-Apr-2004 21:49:19

% Begin initialization code - DO NOT EDIT

 gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
 'gui_Singleton', gui_Singleton, ...
 'gui_OpeningFcn', @trial2_OpeningFcn, ...
 'gui_OutputFcn', @trial2_OutputFcn, ...
 'gui_LayoutFcn', [] , ...

203
'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before trial2 is made visible.
function trial2_OpeningFcn(hObject, eventdata, handles, varargin)
  % This function has no output args, see OutputFcn.
  % hObject    handle to figure
  % eventdata  reserved - to be defined in a future version of MATLAB
  % handles    structure with handles and user data (see GUIDATA)
  % varargin   command line arguments to trial2 (see VARARGIN)

  % Choose default command line output for trial2
  handles.output = hObject;

  % Update handles structure
 .guidata(hObject, handles);

  % UIWAIT makes trial2 wait for user response (see UIRESUME)
  % hObject    handle to figure
  uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = trial2_OutputFcn(hObject, eventdata, handles)
  % varargout  cell array for returning output args (see VARARGOUT)
  % hObject    handle to figure
  % eventdata  reserved - to be defined in a future version of MATLAB
  % handles    structure with handles and user data (see GUIDATA)

  % Get default command line output from handles structure
  varargout{1} = handles.output;
% --- Executes on button press in credit_pushbutton.
function credit_pushbutton_Callback(hObject, eventdata, handles)
% hObject handle to credit_pushbutton (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
pos_size = get(handles.figure1,'Position');
user_response = Credits('Title','CREDITS');
switch user_response
    case 'Yes'
        % Prepare to close GUI application window
        delete(handles.figure1)
    end
% --- Executes on button press in execute_pushbutton.
function execute_pushbutton_Callback(hObject, eventdata, handles)
% hObject handle to execute_pushbutton (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
delete(handles.figure1)

% Generation of eeg signal.
clc;
fs=100;
n=256;
% fs = input('Enter the Sampling Frequency (fs): ');
% t=0:1/fs:5;
% f1 = input('Enter starting of frequency : ');
f1 = 0.5
a=f1;
f2=30
% f2 = input('Enter Ending range of frequency: ');
%x=0;
% while(f1 <= f2)
    % x=x+sin(2*pi*f1*t);
% f1=f1+0.5;
%end
%plot(t,x);
load t4.000
m = input('Patient Data Set number: ');
x = t4(:,m);
[r,c]=size(t4);
%t=0:1/fs:255;
t = (0:255)/255*fs;
y = fft(x,256);
abs_y = abs(y);
f3 = 3000; % Noise frequency in Hz.
f4 = 3100; % Noise frequency in Hz.
tn = (0:n-1)/fs; % Time duration of the noise signal.
noise = sin(2*pi*f3*tn) + 0.8*sin(2*pi*f4*tn);
u = abs_y + noise;
plot(t,u);
xlabel('Frequency (Hz)')
ylabel('Amplitude (micro volts)')
%title(['Signal for frequency-band of ',num2str(a),'Hz to ',num2str(f2),'Hz'])
title('EEG signal form Patient data file')
grid
pause

%Detection of K-complex
s_filt= sgolayfilt(x,3,41);
tl=0:63;
plot(tl,s_filt);
xlabel('Frequency (Hz)')
ylabel('Amplitude (micro volts)')
title('The Detected K-Complex in EEG.')
grid
pause

%input parameters to periodogram.
nfft=256;
w=hanning(256);
noverlap1=128;
dflag='none';
format short
[pxx,f]=psd(abs_y,nfft,fs,w,noverlap1,dflag);
\[ k = \frac{\text{sum}(|\text{abs}_y|^2)}{\text{length}(\text{abs}_y)} \]
\[ \text{plot}(f, 10 \times \log_{10}(\text{pxx} \times \text{norm}(w)^2/\text{sum}(w)^2)) \]
\[ \text{xlabel}('\text{frequency(Hz)}') \]
\[ \text{ylabel}('\text{power spectrum magnitude(dB)}') \]
\[ \text{title}('\text{The estimation of power spectral density of EEG signal}') \]
\[ \text{grid} \]
\[ \text{pause} \]

% Classification of eeg signal into alpha, beta, gamma, theta, delta waves.
% delta range = 0.5 to 4 Hz
% theta range = 4 to 8 Hz
% alpha range = 8 to 13 Hz
% beta range = 13 to 22 Hz
% gamma range = 22 to 30 Hz

% Filter design for 0.5 to 4 Hz.
\[ e = 2 \times 0.5/100; \]
\[ e_1 = 2 \times 4/100; \]
\[ s = [e \ e_1]; \]
\[ [b, a] = \text{cheby1}(4, 0.09, s); \]
\[ x_1 = \text{filtfilt}(b, a, \text{abs}_y); \]
\[ \text{subplot}(2, 1, 1); \]
\[ \text{plot}(t, x_1); \]
\[ \text{grid}; \]
\[ \text{title}(['\text{Delta frequencies in the range of 0.5 to 4 Hz}']); \]

% Input parameters to periodogram.
\[ \text{nfft} = 256; \]
\[ w = \text{hanning}(256); \]
\[ \text{noverlap} = 128; \]
\[ \text{dflag} = \text{'none'}; \]

% Calculation of energy of signal.
\[ \text{format short} \]
\[ [\text{pxx}_1, f] = \text{psd}(x_1, \text{nfft}, \text{fs}, w, \text{noverlap}, \text{dflag}); \]
\[ k_1 = \text{sum}(x_1^2)/\text{length}(x_1) \]
% the plot of periodogram.

subplot(2, 1, 2)
plot(f, 10*log10(pxx1*norm(w)^2/sum(w)^2))
xlabel('Frequency(Hz)')
ylabel('Power spectrum magnitude(dB)')
title('The estimation of power spectral density (DELTA BAND)')
grid
pause

% filter design for 4 to 8 Hz.

e=2*4/100;
e1=2*8/100;
s=[e e1];
[b,a]=cheby1(5, .075, s);
x2=filtfilt(b,a,abs_y);
subplot(2, 1, 1);
plot(t, x2);
grid;
title(['Theta frequencies in the range of 4 to 8 Hz']);

% input parameters to periodogram.

nfft=256;
w=hanning(256);
noverlap1=128;
dflag='none';

% calculation of energy of signal.

format short
[pxx2, f] = psd(x2, nfft, fs, w, noverlap1, dflag);
k2 = sum(x2.^2)/length(x2)

% the plot of periodogram.

subplot(2, 1, 2)
plot(f, 10*log10(pxx2*norm(w)^2/sum(w)^2))
xlabel('Frequency(Hz)')

208
ylabel('Power spectrum magnitude(dB)')
title('The estimation of power spectral density (THETA BAND).')
grid
pause

%filter design for 8 to 13 Hz.

e=2*8/100;
e1=2*13/100;
s=[e e1];
[b,a]=cheby1(5,.075,s);
x3=filtfilt(b,a,abs_y);
subplot(2,1,1);
plot(t,x3);

%[t,x3]=contour(peaks);
%colormap(hsv(20)); shading flat;
grid;
title(['Alpha frequencies in the range of 8 to 13 Hz ']);

%input parameters to periodogram.

nfft=256;
w=hanning(256);
noverlap1=128;
dflag='none';

%calculation of energy of signal.

format short
[pxx3,f]=psd(x3,nfft,fs,w,noverlap1,dflag);
k3=sum(x3.^2)/length(x3)

%the plot of periodogram.

subplot(2,1,2)
plot(f,10*log10(pxx3*norm(w).^2/sum(w).^2))
xlabel('Frequency(Hz)')
ylabel('Power spectrum magnitude(dB)')
title('The estimation of power spectral density (ALPHA BAND)')
grid
% filter design for 13 to 22 Hz.

e=2*13/100;
e1=2*22/100;
s=[e e1];
[b,a]=cheby1(5,.12,s);
x4 =filtfilt(b,a,abs_y);
subplot(2,1,1);
plot(t,x4);
grid;
title(['Beta I frequencies in the range of 13 to 22 Hz']);

% input parameters to periodogram.
nfft=256;
w=hanning(256);
noverlap1=128;
dflag='none';

% calculation of energy of signal.

format short
[pxx4,f]=psd(x4,nfft,fs,w,noverlap1,dflag);
k4=sum(x4.^2)/length(x4)

% the plot of periodogram.

subplot(2,1,2)
plot(f,10*log10(pxx4*norm(w)^2/sum(w)^2))
xlabel('Frequency(Hz)')
ylabel('Power spectrum magnitude(dB)')
title('The estimation of power spectral density (BETA BAND I)')
grid
pause

% filter design for 22 to 30 Hz.

e=2*22/100;
el=2*30/100;
s=[e el];
[b,a]=cheby1(5,.09,s);
x5=ffiltfilt(b,a,abs_y);
subplot(2,1,1);
plot(t,x5);
grid;
title(['Beta II frequencies for range of 22 to 30 Hz']);

%ninput parameters to periodogram.

nfft=256;
w=hanning(256);
noverlap1=128;
dflag='none';

%calculation of energy of signal.

format short
[pxx5,f]=psd(x5,nfft,fs,w,noverlap1,dflag);
k5=sum(x5.^2)/length(x5)

%the plot of periodogram.

subplot(2,1,2)
plot(f,10*log10(pxx5*norm(w)^2/sum(w)^2))
xlabel('Frequency(Hz)')
ylabel('Power spectrum magnitude(dB)')
title('The estimation of power spectral density (BETA BAND II).')
grid
pause

% --- Executes on button press in close_pushbutton.
function close_pushbutton_Callback(hObject, eventdata, handles)
% hObject handle to close_pushbutton (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get the current position of the GUI from the handles structure
% to pass to the modal dialog.
RESULT OF MATLAB PROGRAM OF EEG SIGNAL FOR PATIENT DATA FILE NUMBER 15.

Patient Data Set number: 15
Minimum cutoff frequency, \( f_1 = 0.5 \) Hz
Maximum cutoff frequency, \( f_2 = 30 \) Hz

POWER SPECTRUM DENSITY OF:

- **EEG SIGNAL** \( k = 2.2613 \times 10^3 \)
- **DELTA BAND** \( k_1 = 315.4517 \)
- **THETA BAND** \( k_2 = 230.3188 \)
- **ALPHA BAND** \( k_3 = 131.7529 \)
- **BETA I BAND** \( k_4 = 40.3011 \)
- **BETA II BAND** \( k_5 = 6.3707 \)

The graphs for EEG signal and estimation of power spectral density are given in figure B1 and B3. Classification of EEG bands and estimation of PSD is shown in figure B4 to B8.
Fig. B1. EEG signal of patient number 25

Fig. B2. Detected k-complex in EEG signal of patient number 25
Fig. B3. Estimation of Power Spectral Density of EEG signal

Fig. B4. Delta band and Estimation of Power Spectral Density
Fig.B5. Theta band and Estimation of Power Spectral Density

Fig.B6. Alpha band and Estimation of Power Spectral Density
Fig.B7. Beta I band and Estimation of Power Spectral Density

Fig.B8. Beta II band and Estimation of Power Spectral Density