BIOMEDICAL SIGNAL PROCESSING

NAME:-SWATI KUMRI

ROLL NO.:-1804146

BRANCH:-ECE -A

ASSIGNMENT NO:-6

TOPIC:-SPIKES AND WAVE COMPLEX DETECTION IN EEG SIGNAL USING TEMPLATE MATCHING

Electroencephalography (**EEG**) is an efficient modality which helps to acquire brain signals corresponds to various states from the scalp surface area. These signals are generally categorized as delta, theta, alpha, beta and gamma based on signals frequencies ranges from 0.1 Hz to more than 100 Hz.

The electroencephalogram (EEG) is a recording of the electrical activity of the brain from the scalp. The recorded waveforms reflect the cortical electrical activity.

Signal intensity: EEG activity is quite small, measured in microvolts (μV).
Signal frequency: the main frequencies of the human EEG waves are:

Delta: has a frequency of 3 Hz or below.

Theta: has frequency of 3.5 to 7.5 Hz and is classified as "slow" activity.

Alpha: has a frequency between 7.5 and 13 Hz.

Beta: beta activity is "fast" activity. It has a frequency of 14 and greater Hz.

SPIKES AND WAVES IN EEG SIGNAL :-

Spikes and waves is a pattern of the electroencephalogram (**EEG**) typically observed during epileptic seizures. A spike-and-wave discharge is a regular, symmetrical, generalized EEG pattern seen particularly during absence epilepsy, also known as 'petit mal' epilepsy.

Spikes or sharp waves are terms commonly seen in EEG reports. If these happen only once in a while or at certain times of day, they may not mean anything. If they happen frequently or are found in specific areas of the brain, it could mean there is potentially an area of seizure activity nearby.

TEMPLATE MATCHING ALGORITHM:-

Template matching is a technique used to isolate certain features in an image .These features can be single pixels ,lines ,edges or complete objects .It is easier to look upon an image ,convolved with a template window ,as a correlation between an image and that window .The result will be an image with high values where there is a strong correlation and low values elsewhere.

Template matching is the term given to the process of detecting an event buried in a signal by comparing it to a predefined template. This template is got by manual detection of peak in the EEG .

Even though, there are many others methods template matching technique like stereo matching ,image registration and scale -invariant feature transform. We have selected template matching technique because complexity is less and marking of spikes and waves can be done easily.

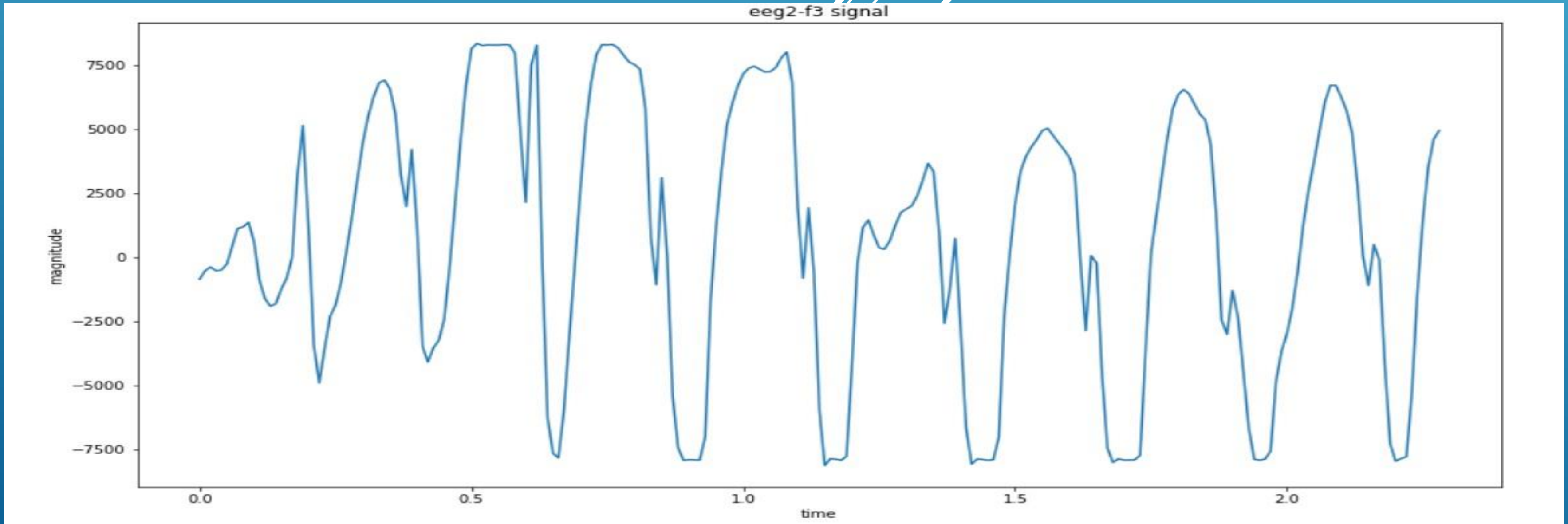
TO DETECT SPIKES AND WAVES IN EEG SIGNAL BY TEMPLATE MATCHING:-

1.A spike and wave is a well defined event in EEG signal.

2.The complex is composed of sharp-spikes followed by wave with a frequency of about 3Hz.

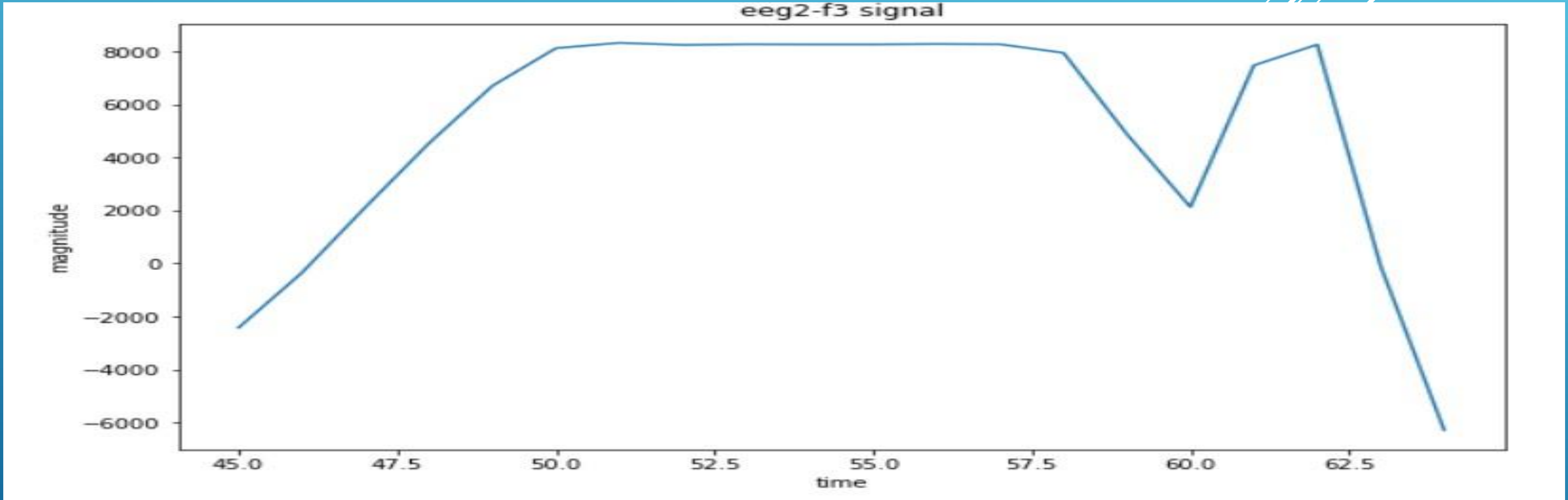
3.The template will be correlated with same channel from which it was extracted to detect similar events that appear at latter time or with another channel to search for similar events.

4.Figure of f3 channel:-



CONTINUED..

5.The spikes and waves complex between 0.45 and 0.65 in the signal is selected to use as the template.



CONTINUED…

6.The template matching was performed with the same channel signal using the equation (Normalize correlation-coefficient):

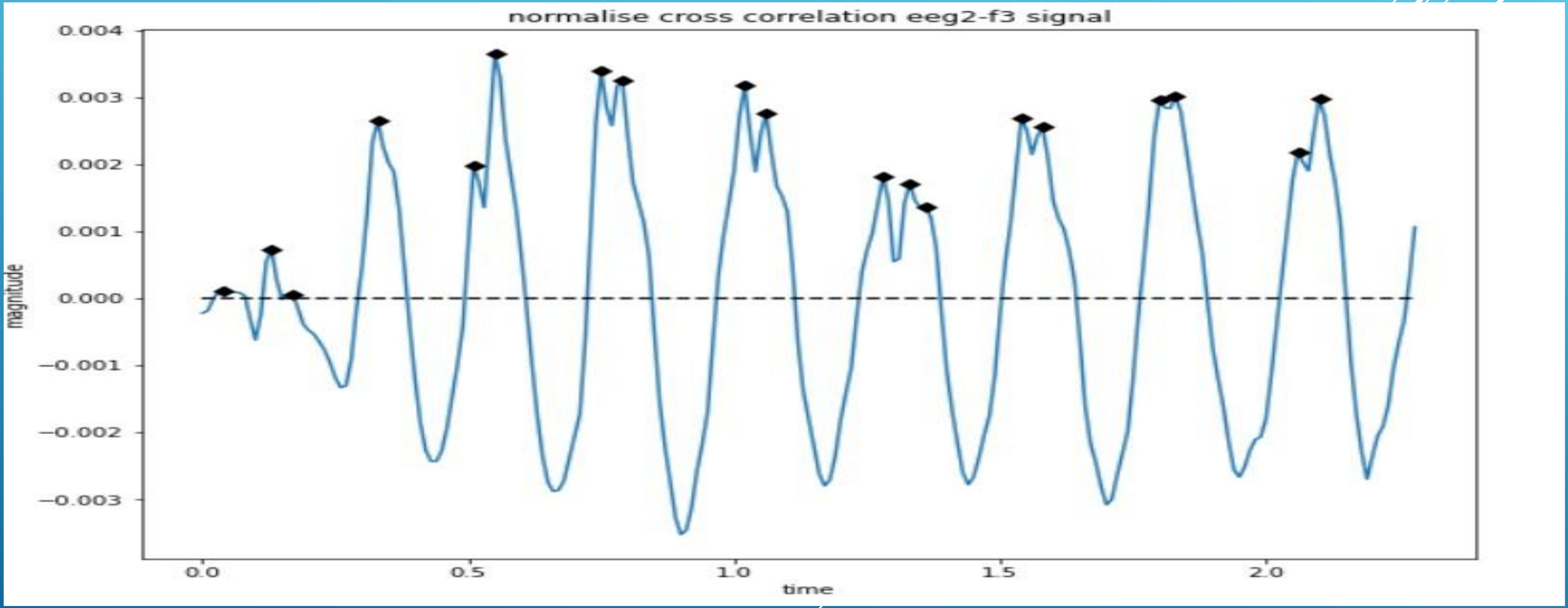
$$R(x,y)=\frac{\sum_{x',y'}\left(I(x+x',y+y')-\bar{I}(x,y)\right)\cdot\left(t(x',y')-\bar{t}\right)}{\sqrt{\sum_{x',y'}\left(I(x+x',y+y')-\bar{I}(x,y)\right)^2\cdot\sum_{x',y'}\left(t(x',y')-\bar{t}\right)^2}}$$

7.The result demonstrates strong and clear peaks at each occurrances of the spike-and –wave complex in the EEG signal.

8.The peaks occur at the same instant of time as the corresponding spike-and-wave complexes.

CONTINUED…

Result of template matching:-



CONCLUSION:-

In this ,we introduces a technique based on Template matching method for spike and wave complex detection.

We correlated template of channel with same channel as well as other channel and obtained the desired result.

We learn to process EEG signal using python programming.

Code:-

```
#importing libraries
#pandas->for reading the dataset
#numpy->for storing data as array
#matplotlib->for displaying th graph


import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


#reading dataset using panda library.
# we have 10 channels of eeg signals

c3 = pd.read_csv('eeg2-c3.dat')
c4 = pd.read_csv('eeg2-c4.dat')
f3 = pd.read_csv('eeg2-f3.dat')
f4 = pd.read_csv('eeg2-f4.dat')
o3 = pd.read_csv('eeg2-o1.dat')
o4 = pd.read_csv('eeg2-o2.dat')
p3 = pd.read_csv('eeg2-p3.dat')
p4 = pd.read_csv('eeg2-p4.dat')
t3 = pd.read_csv('eeg2-t3.dat')
t4 = pd.read_csv('eeg2-t4.dat')

#reading values of each channel

eeg2_c3 = c3.values
eeg2_c4 = c4.values
eeg2_f3 = f3.values
eeg2_f4 = f4.values
eeg2_o3 = o3.values
eeg2_o4 = o4.values
eeg2_p3 = p3.values
eeg2_p4 = p4.values
eeg2_t3 = t3.values
eeg2_t4 = t4.values

#sampling frequency of 100Hz for eeg analysis and corresponding time in
terval

fs = 100
t = np.arange(0,len(eeg2_f3))/fs

#plotting all the channel signal with specified fig size

plt.figure(figsize=(20,50))
```

```
# C3 CHANNEL
plt.subplot(5,2,1)
plt.plot(t,eeg2_c3)
plt.title('eeg2-c3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# C4 CHANNEL
plt.subplot(5,2,2)
plt.plot(t,eeg2_c4)
plt.title('eeg2-c4 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# F3 CHANNEL
plt.subplot(5,2,3)
plt.plot(t,eeg2_f4)
plt.title('eeg2-f3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# F4 CHANNEL
plt.subplot(5,2,4)
plt.plot(t,eeg2_f4)
plt.title('eeg2-f4 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# O3 CHANNEL
plt.subplot(5,2,5)
plt.plot(t,eeg2_o3)
plt.title('eeg2-o3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# O4 CHANNEL
plt.subplot(5,2,6)
plt.plot(t,eeg2_o4)
plt.title('eeg2-o4 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# P3 CHANNEL
plt.subplot(5,2,7)
plt.plot(t,eeg2_p3)
plt.title('eeg2-p3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# P4 CHANNEL
plt.subplot(5,2,8)
plt.plot(t,eeg2_p4)
plt.title('eeg2-p4 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
```

```
# T3 CHANNEL
plt.subplot(5,2,9)
plt.plot(t,eeg2_t3)
plt.title('eeg2-t3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
# T4 CHANNEL
plt.subplot(5,2,10)
plt.plot(t,eeg2_t4)
plt.title('eeg2-t4 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
plt.show()


#getting a template from f3 channel

template = eeg2_f3[45:65]
plt.figure(figsize=(15,8))
plt.plot(template)
plt.show()

#correlating template with same channel ,that is ,f3

from scipy.signal import correlate
ccf = correlate(eeg2_f3,template)
plt.figure(figsize=(20,10))
plt.plot(ccf)
plt.show()

ccf = ccf.flatten()
ccf.shape

t2 = np.arange(0,ccf.size)/fs
#simple cross correlation of f3 signal

from scipy.signal import find_peaks
peaks, _ = find_peaks(ccf, height=0)
plt.figure(figsize=(20,10))
plt.plot(t2,ccf)
plt.plot(t2[peaks],ccf[peaks], "D",color='black')
plt.plot(t2,np.zeros_like(t2), "--", color="black")
plt.title('simple cross correlation eeg2-f3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
plt.show()
```

```python
a=(eeg2_f3-np.mean(eeg2_f3))/(np.std(eeg2_f3)*len(eeg2_f3))
b=(template-np.mean(template))/(np.std(template)*len(template))

from scipy.signal import correlate
ccf_f3 = correlate(a,b,'same')
plt.figure(figsize=(15,8))
plt.plot(ccf_f3)
plt.show()

ccf_f3 = ccf_f3.flatten()
ccf_f3.shape

t3 = np.arange(0,ccf_f3.size)/fs

#normalise cross orrelation

from scipy.signal import find_peaks
peaks, _ = find_peaks(ccf_f3, height=0)
plt.figure(figsize=(10,8))
plt.plot(t3,ccf_f3)
plt.plot(t3[peaks],ccf_f3[peaks], "D",color='black')
plt.plot(t3,np.zeros_like(t3), "--", color="black")
plt.title('normalise cross correlation eeg2-f3 signal')
plt.xlabel('time')
plt.ylabel('magnitude')
plt.show()
```