Georg-August-Universität Göttingen

Summer Internship Report

Use of tDCS for stroke rehabilitation

Author:
Athira Jane Jacob

Supervisor(s):
Prof.Dr. Michael Nitsche
Dr. Anirban Dutta

August 2, 2014

Contents

Acknowledgments

I express my sincere gratitude to Prof. Michael Nitsche for giving me this opportunity to do an internship with the university. It was a wonderful opportunity for learning and self-development. I also thank Dr. Anirban Dutta, who in spite of being busy with his duties, took time out to hear, guide and keep me on the correct path, allowing me to carry out my project work during the period.
I would also like to thank all my lab mates in the university. With their patience and openness they created an enjoyable working environment. A special thanks to Ms. Aguida Foerster, for being both a friend and guide during my time there. It was a pleasure working with everyone.
Finally, my deepest gratitude to Prof. Asokan at IIT Madras for giving me a chance at this internship and kick-starting the whole process.

Chapter 1

Introduction

The electrical activity of active neurons in the brain can be recorded as voltage differences on the scalp, as the electroencephalogram (EEG). EEG reflects brain electrical activity with millisecond temporal resolution and is one of the most direct correlate of on-line brain processing obtainable non-invasively. EEG activity reflects the summation of the synchronous activity of thousands or millions of neurons that have similar spatial orientation. Because voltage fields fall off with the square of distance, activity from deep sources is more difficult to detect than currents near the skull.

Transcranial direct current stimulation (tDCS), is a non-invasive brain stimulation method that uses direct electrical currents to stimulate specific parts of the brain. A constant, low intensity direct current is passed through two electrodes placed over the head which modulates neuronal activity. There are two types of stimulation with tDCS: anodal and cathodal stimulation. Anodal stimulation excites neuronal activity while cathodal stimulation inhibits neuronal activity. It can potentially be used to help patients with brain injuries like strokes. Studies have indicated that it can enhance cognitive performance in variety of tasks, depending upon the region of brain being stimulated [1]. Generally, tDCS protocols utilize two surface electrodes, one serving as the anode and the other one as the cathode or reference. The position of the electrodes is critical for the spatial distribution and direction of the flow of the current which may determine the effectiveness of the stimulation. It is generally agreed that anodal tDCS has an excitatory effect on the local cerebral cortex by depolarizing neurons, while the converse applies to cathodal stimulation through the process of hyperpolarization. Typically these electrodes have relatively large surfaces of 2035 mm^2 that limit the focus of stimulation. On the other hand, the large surface allows the use of low current densities, which is critical for patient safety.

Another commonly used method of brain stimulation is transcranial magnetic stimulation (TMS). This technique of brain stimulation utilizes an electric coil held above the region of interest on the scalp that uses rapidly changing magnetic fields to induce small electrical currents in the brain. This can cause activity in specific or general parts of the brain with little discomfort, allowing for study of the brain's functioning and interconnections. Increased neuronal activity is induced in repetitive TMS by using a higher frequency and decreased neuronal activity is induced by using a lower frequency.

An electromyograph detects the electrical potential generated by muscle cells when these cells are electrically or neurologically activated. The signals can be analyzed to detect medical abnormalities, activation level, or recruitment order or to analyze the biomechanics of human or animal movement. The activation and force output of a muscle contraction can be assessed through the use of surface EMG electrodes. This is a non-invasive practice to quantify the relationship between a specific movement and the activation of the underlying muscle group(s). The specific amount of force generated by a muscle can also be studied using the EMG.

Chapter 2

Experimental setup

2.1 Introduction

Stroke is one of the leading causes of adult disability in the western world[2]. Changes in synaptic function after stroke, such as reduced excitability, formation of aberrant connections, delays in initiation and termination and deregulated plastic modifications, have been postulated to impede recovery from stroke. This can be treated at the central nervous system (CNS) level with transcranial direct current stimulation (tDCS) thereby facilitating re-learning and retaining of normative muscle activation patterns . Anodal tDCS has been shown to increase cortical excitability and improve motor learning and function[3] [4]. Stroke patients often suffer from drop foot which affects their ability to lift their foot at the ankle. This causes the toes to drag along the ground while walking. Treatment at central nervous system level to facilitate learning of myoelectric control using tDCS has been shown to improve motor learning in healthy humans[5][6]. This study seeks to systematically explore the effects of tDCS treatment on rehaibilation of stroke patients.
Though tDCS has been shown to induce neuroplasticity [10] and improve motor learning, tDCS-facilitated motor learning in lower limbs has not been explored systematically. Tanaka et al [7] found that anodal tDCS of the primary motor cortex representation of the tibialis anterior (TA) muscle (M1) had no significant effects on reaction time, but transiently enhanced maximal leg pinch force. Madhavan et al [8] found that M1 anodal tDCS of the primary motor representation of TA muscle applied to the lesioned motor cortex of moderate to well recovered stroke patients enhanced voluntary control of the paretic ankle. Dutta et al [9] showed that 2mA anodal tDCS for over 10 minutes over the cortical representation of TA muscle induced statistically significant increase in MEP (Motor evoked potentials) based measure of cortico spinal excitability and increase in cortico muscular coherence of TA muscle. Moreover it was showed that the cortico-muscular coherence was correlated with the MEP-measure of cortico-spinal excitability following anodal tDCS. In another study, Dutta and colleagues [11] showed that Cerebellar anodal tDCS increased the delay in initiation of TA contraction and decreased the delay in termination of TA contraction while M1 anodal tDCS decreased and increased the same respectively when compared to sham tDCS.
The aim of this project is to systematically study the effects of tDCS treatment at M1 and cerebellum on the control and coordination of the tibialis anterior muscle.

2.2 Methodology

Initially the maximum voluntary contraction (MVC) of the subject is measured. Then the subject is presented with a visuomotor task. The subject has to contract the TA muscle isometrically as fast as possible in response to a visual cue where the TARGET jumps to a randomized value between 40% and 80% of their MVC. Myoelectric visual biofeedback was presented with proportional system dynamics, where the subject modulates the EMG activity to match the TARGET level. The moving average of the rectified EMG from the TA muscle is provided as visual feedback along with the TARGET signal. The average rectified EMG

during three seconds of MVC is used for normalization. The visuo-motor task's continues for 5 s and is preceded by RELAX time (10 s) and READY time (3-5 s) respectively.
For each subject four sessions are conducted, a) Anodal stimulation of the motor cortex b) Anodal stimulation of the cerebellum c) Anodal stimulation of both motor cortex and cerebellum d) Sham stimulation. Each session is divided into three parts, a) Pre stimulation (10 min) b) Stimulation period (15 min) c) Post stimulation (10 min). The subject performs the visuomotor task through all three parts and EEG and EMG readings are continuously recorded. During the stimulation period, a 2 mA direct current is used to simulate the required region of the brain, either M1 or cerebellum. TMS (Magstim, UK) is used to find that area of the motor cortex representing TA muscle called hotspot by finding the area that elicited maximal MEP in resting tibialis anterior muscle.

2.3 Data collection and analysis

Transcranial current stimulation and EEG monitoring is done by Starstim using NIC v1.2 (Neuroelectrics Instrument Controller) software.
Surface EMG is collected from the TA muscle, amplified and band-pass filtered (anti-aliasing, frequency band =10- 500 Hz) before being sampled at 2000 Hz by a 12-bit data acquisition system (NI USB-6009, National Instruments, USA) in a PC. Data-processing and graphical (GUI) display are performed with Matlab R2010a (The MathWorks, Inc., USA) using the Psychtoolbox.

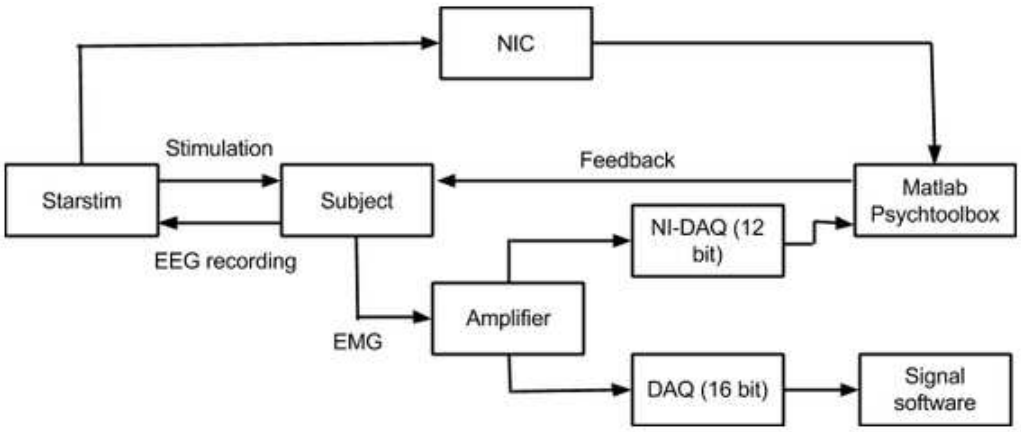


Figure 2.1: Block diagram of the experimental setup

For the post processing, the raw EMG sampled during each task block of the experiment was digitally zero phase band-pass filtered (5th order Butterworth, 3 dB bandwidth = 10-500 Hz), then full-wave rectified, and then zero-phase low-pass filtered (5th order Butterworth, 3 dB frequency cutoff = 25 Hz) to generate its linear EMG-envelope (LE). EEG is cleaned up from artifacts and band pass filtered Three parameters are studied for each subject: a) Reaction time b) Learning rate c) EEG-EMG coherence. The reaction time is defined as the time when the processed EMG signal magnitude crosses the target value. A matlab script was written for finding the same (See Appendix). The learning rate is found by plotting the error (i.e different between the EMG value of the subject and target value) for each trial. EEG-EMG coherence can be found using *mscohere* in matlab. (See Appendix)
It is hypothesized that anodal stimulation of the motor cortex reduces reaction time and stimulation of the cerebellum improves learning rate.

Chapter 3

Data Analysis using Hilbert Transform

3.1 Introduction

EEG signals can be decomposed by the use of PCA, ICA, ARMA, etc. into derivative state variables, the limitation being these assume linearity and stationarity in the dynamics of their sources. Prominent among these transforms is the fast Fourier transform (FFT), which decomposes each raw signal into a family of state variables having fixed frequencies and amplitudes. Another technique is the wavelet transform that is used for linear decomposition of signals varying in amplitude. The temporal resolution of wavelets and the FFT is bounded by the Nyquist criterion: the digitizing rate must be at least twice and preferably three times the highest component frequency. The duration of segments for decomposition must exceed at least one cycle of the lowest component frequency.

However brain dynamics revealed by state variables derived from EEG is noisy, non-stationary, nonlinear, and rife with temporal discontinuities from state transitions [12] . Though the Hilbert transform (HT) like the FFT is a linear operator, it is useful for analyzing non-stationary signals by expressing frequency as a rate of change in phase, so that the frequency can vary with time. Typically multiple time-varying frequencies coexist in raw recordings. Empirical mode decomposition (EMD), the Hilbert-Huang transform [13] , gives high spectral resolution of arbitrary frequencies. More useful for EEG is 'clinical mode decomposition' (CMD) by band pass filtering to decompose raw signals into components corresponding to the divisions of the clinical spectrum. The commonly accepted passbands of clinical EEG are delta: 1-3 Hz, theta: 3-7 Hz, alpha: 7-12 Hz, beta: 12-30 Hz, gamma: 30-80 Hz and epsilon: 80-250 Hz.

Comparison to FFT

The HT and FFT give the same results when both transforms are applied to signals having the relatively long durations needed for the FFT and wavelets [14] [15]. Otherwise they are complementary. The FFT and EMD give high frequency resolution. The HT and CMD give high temporal resolution of rapid changes in analytic state variables for frequency, phase and amplitude. Another advantage of the HT is the sensitive access it gives to modulation patterns of analytic amplitude that are correlated with intentional behaviors [16]. The greatest difficulty in using the HT is to distinguish physiological state transitions from spurious discontinuities in the analytic phase known as phase slip [17]. Phase slip occurs by interference between multiple overlapping signals with differing frequencies and AM patterns. EEG signals commonly have power-law distributions (see 1/f noise) in power spectral densities (PSD) that are not well handled by the FFT. Application of the HT to unfiltered EEG gives analytic phase values resembling a random walk. Band pass filtering using the FFT is necessary to get time series from the HT that is interpretable as state variables.

3.2 Methodolgy

The Hilbert transform of two signals CSD and Hbt can be denoted as,

$$H_{CSD,i} = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{CSD_i(\tau)}{t-\tau} d\tau$$

$$H_{Hbt,i} = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{Hbt_i(\tau)}{t-\tau} d\tau$$

where P is the Cauchy principal value. Then the analytic signal is defined as,

$$Z_{CSD,i} = CSD_i(t) + iH_{CSD,i}(t)$$

$$Z_{Hbt,i} = Hbt_i(t) + iH_{Hbt,i}(t)$$

The matlab function *hilbert* is used to find the analytic signal. The instantaneous amplitudes for the analytic signals can be determined as,

$$A_{CSD,i}(t) = [CSD_i^2(t) + H_{CSD,i}^2(t)]^{\frac{1}{2}}$$

$$A_{Hbt,i}(t) = [Hbt_i^2(t) + H_{Hbt,i}^2(t)]^{\frac{1}{2}}$$

The instantaneous phases for the analytic signals can be determined as,

$$\theta_{CSD,i}(t) = \arctan \frac{H_{CSD,i}(t)}{CSD_i(t)}$$

$$\theta_{Hbt,i}(t) = \arctan \frac{H_{Hbt,i}(t)}{Hbt_i(t)}$$

The instantaneous frequencies for the analytic signals can be determined as,

$$f_{CSD,i}(t) = \frac{1}{2\pi} \frac{d\theta_{CSD,i}(t)}{dt}$$

$$f_{Hbt,i}(t) = \frac{1}{2\pi} \frac{d\theta_{Hbt,i}(t)}{dt}$$

Average instantaneous frequencies are found from the instantaneous frequencies using a sliding window (interval=250 ms). To find the cross spectrum from CSD to Hbt at time t, for the average instantaneous frequency f at t of CSD, the corresponding windows of instantaneous amplitudes and phases are chosen from CSD and Hbt. Let m^{th} window in CSD and n^{th} window in Hbt give the frequency f. Then the cross spectrum is calculated as,

$$C_{f_j}(CSD, Hbt) = A_{CSD,m}(t) A_{Hbt,n}(t) e^{i[\theta_{CSD,m}(t) - \theta_{Hbt,n}(t)]}$$

Similarly, the cross spectrum from Hbt to CSD is calculated as,

$$C_{f_j}(Hbt, CSD) = A_{Hbt,n}(t) A_{CSD,m}(t) e^{i[\theta_{Hbt,n}(t) - \theta_{CSD,m}(t)]}$$

The coherence is calculated as,

$$Coh_{CSD \to Hbt, f_j} = \frac{\langle C_{f_j}(CSD, Hbt)^2 \rangle}{\langle [A_{CSD,m}(t) e^{i\theta_{CSD,m}(t)}]^2 \rangle \langle [A_{Hbt,n}(t) e^{i\theta_{Hbt,n}(t)}]^2 \rangle}$$

$$Coh_{Hbt \to CSD, f_j} = \frac{\langle C_{f_j}(Hbt, CSD)^2 \rangle}{\langle [A_{Hbt,n}(t) e^{i\theta_{Hbt,n}(t)}]^2 \rangle \langle [A_{CSD,m}(t) e^{i\theta_{CSD,m}(t)}]^2 \rangle}$$

where $\langle\rangle$ denotes averaging over multiple paired windows for the given frequency f_j.

3.3 Testing

To test the code a surrogate data set is created.

Case I: Coherent data with phase difference and added global noise

```
Gain_noise=2;
t=0:0.1:100;
rand1=rand(1,length(t));
rand2=rand(1,length(t));
sig1=1*cos(2*pi*90/360+2*pi/21*t)+4*cos(+2*pi*00/360-2*pi/10*t);
sig2=1*cos(2*pi*60/360+2*pi/21*t)+4*cos(-2*pi*70/360-2*pi/10*t);
eeg=sig1+Gain_noise*(-1+2*rand1);
emg=sig2+Gain_noise*(-1+2*rand2);
```

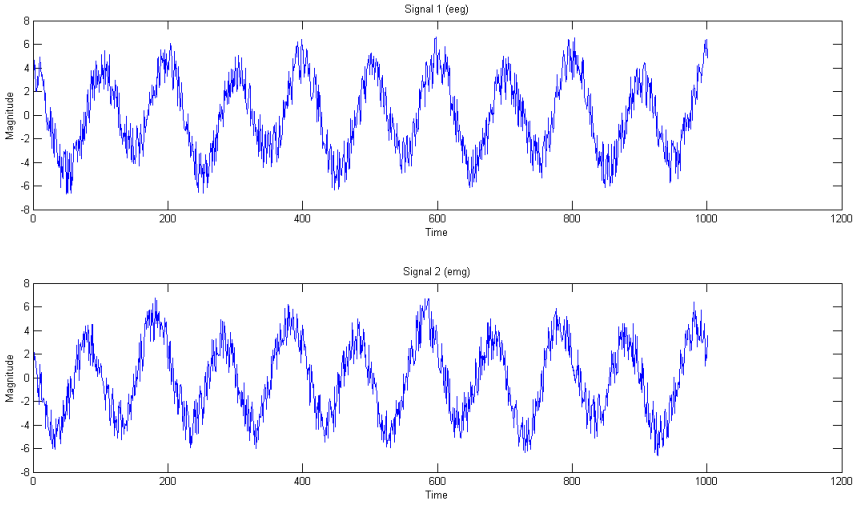


Figure 3.1: Sample data set-Case I (Gain_ noise=2)

Case II: Data set with noise added to the coherent frequency

```
Gain_noise=2;
t=0:0.1:100;
omega11=2*pi/21+Gain_noise*rand(1);
omega21=2*pi/21+Gain_noise*rand(1);
omega12=2*pi/10+Gain_noise*rand(1);
omega22=2*pi/10+Gain_noise*rand(1);
eeg=1*cos(2*pi*90/360+omega11*t)+4*cos(+2*pi*00/360-omega12*t);
emg=1*cos(2*pi*60/360+omega21*t)+4*cos(-2*pi*70/360-omega22*t);
```

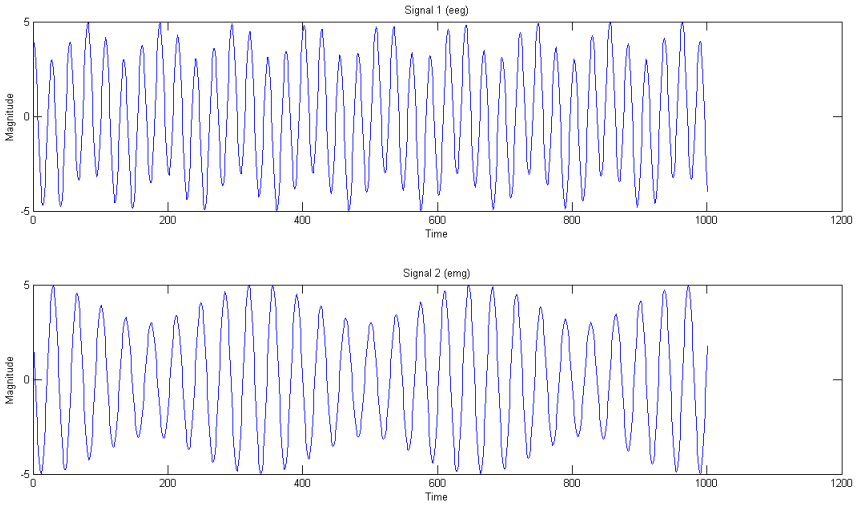


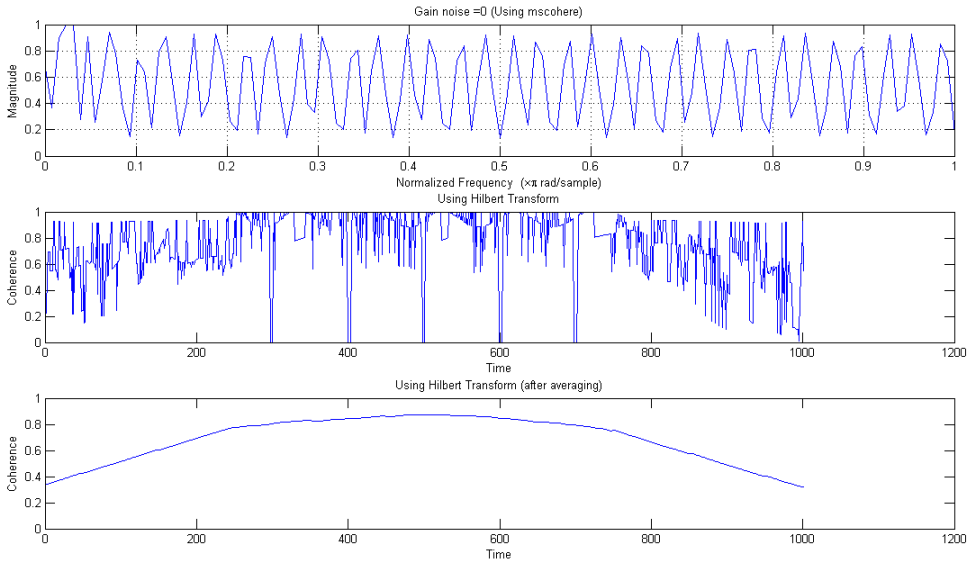
Figure 3.2: Sample data set-Case II (Gain_noise=2)

The two signals are named eeg and emg for identification. The data is analyzed for coherence using both *mscohere* in matlab as well as Hilbert transform method as explained in the previous section. Four different levels of noise are added to test the performance.

3.4 Results

Case I: Coherent data with phase difference and added global noise

Four different gains are applied to the noise, Gain_noise=0,2,5,10. The coherence values after averaging from Hilbert transform method are also plotted. The results are as follows:



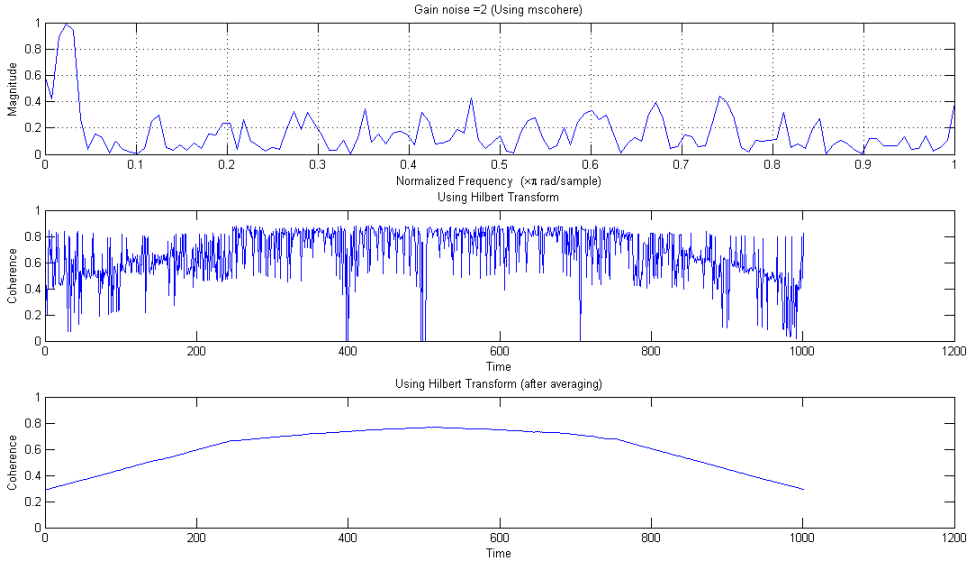


Figure 3.3: Gain_noise = 2

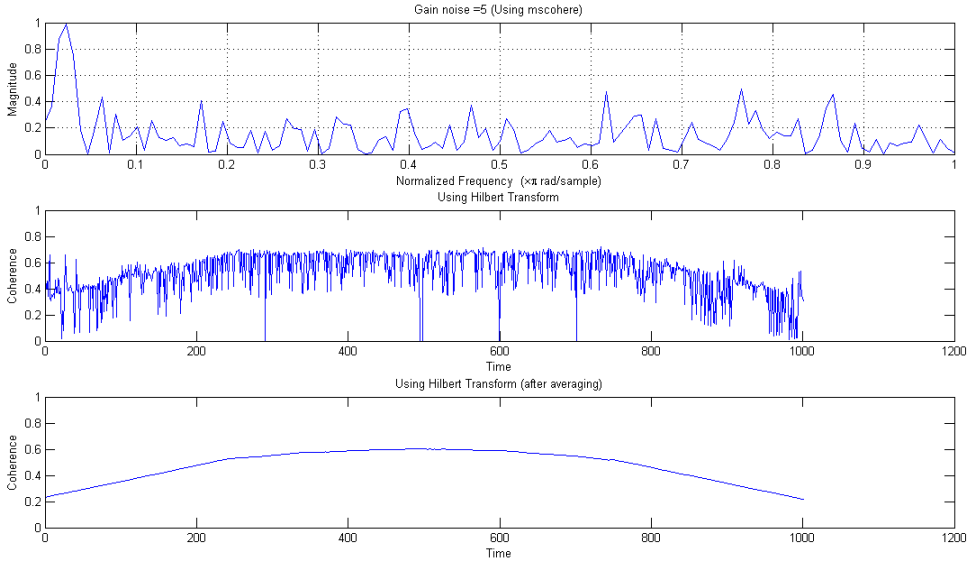


Figure 3.4: Gain_noise = 5

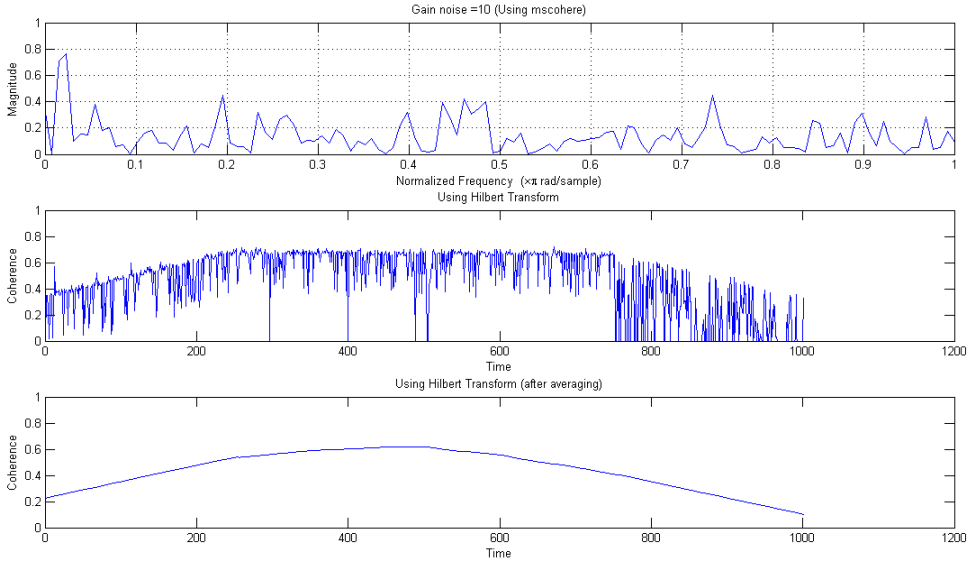


Figure 3.5: Gain_ noise=10

Case II: Data set with noise added to the coherent frequency
Four different gain noise values are used: Gain_noise= 0, 0.1, 2, 5. The results are as follows:

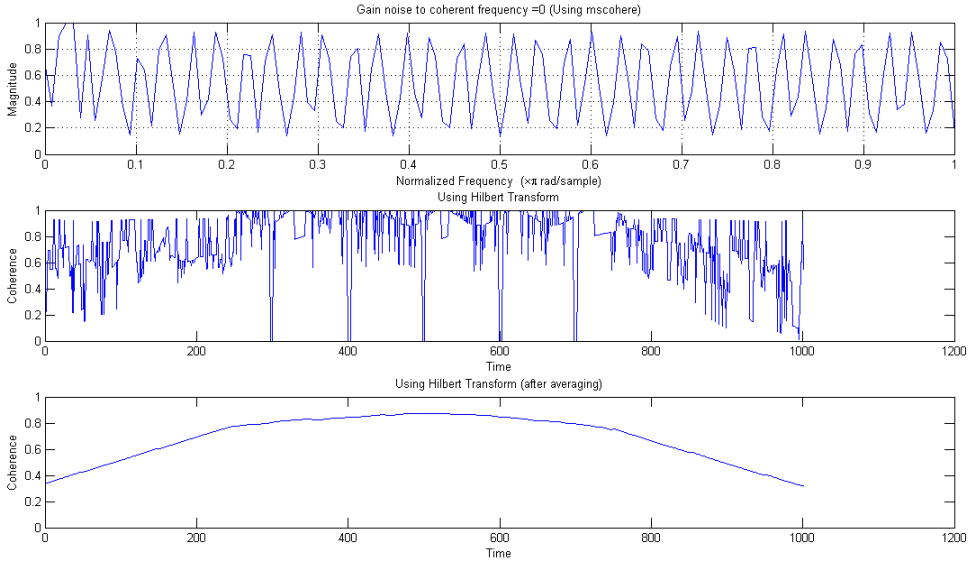


Figure 3.6: Gain_noise = 0

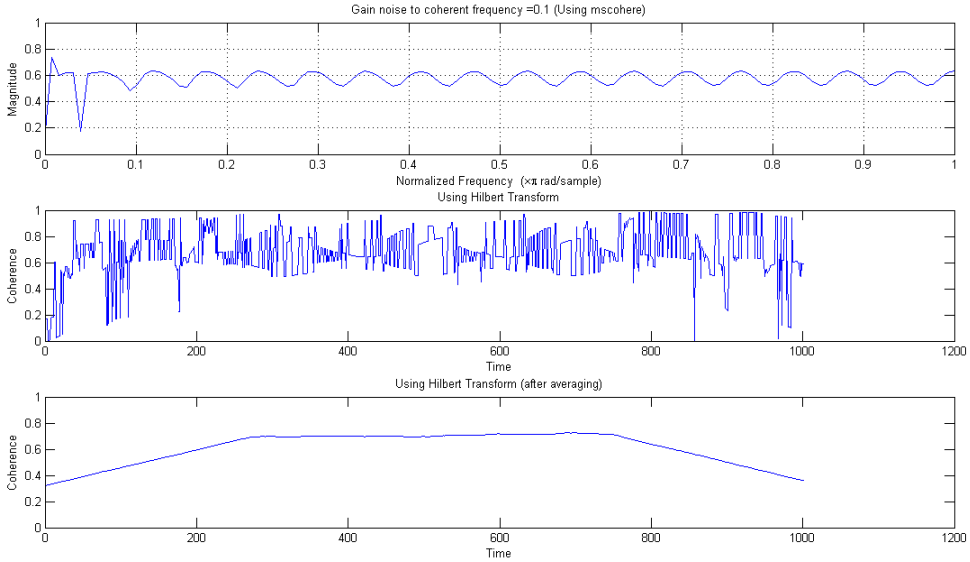


Figure 3.7: Gain_noise = 0.1

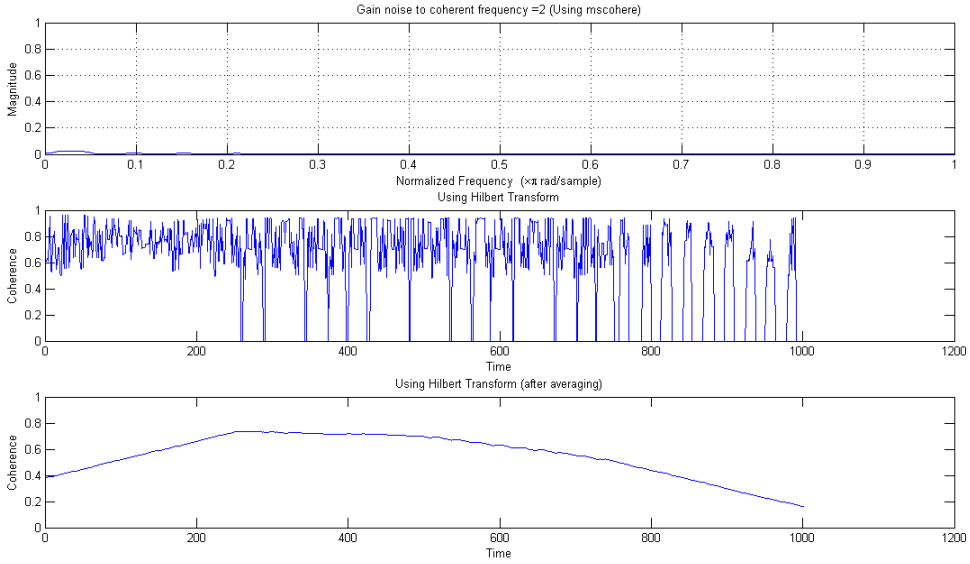


Figure 3.8: Gain_noise = 2

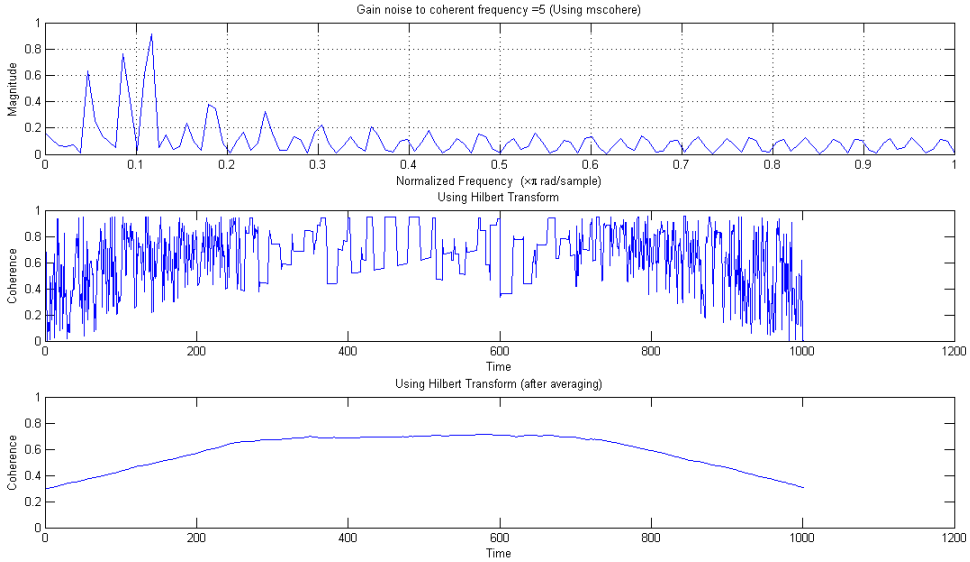


Figure 3.9: Gain_noise = 5

3.5 Conclusion

Hilbert transform method appears to successfully predict coherence of the two signals in the surrogate data set created. Though the magnitude of the coherence values are not similar to those obtained from *mscohere*, the coherence changes predictably according to the amount of noise added to the signal. After reaching a certain level of noise, the method seems to be robust to increasing noise.

Chapter 4

Appendix

4.1 MATLAB script to find reaction times from EMG

```
%This code takes Pre-EMG, Stimulation-EMG and Post-EMG .mat files as inputs
%and calculates reaction times.
%Reaction times are given by:
%Pre- react_time_pre
%Stim- react_time_stim
%Post- react_time_post

clc
clear all

%file names containing pre, stim and post data
load 20140708T160649_testathiraPre.mat;
data_pre=data;
load 20140708T162347_testathiraStm.mat;
data_stim=data;
load 20140708T163146_testathiraPos.mat;
data_post=data;
wind=0.25;
Fs=2000;

%pre processing of signals
data_pre.EMGs(1,:)=[];
data_pre.time(1,:)=[];
data_pre.task(1)=[];
data_pre.targ(1)=[];
data_pre.curs(1)=[];
data_pre.trial(1)=[];
temp=data_pre.EMGs';
pre_emg=temp(:);
temp=data_pre.time';
pre_time=temp(:);

pretask_len=max(size(data_pre.task));
f=2;pre_task=1;
```

```
while f<=pretask_len
    if data_pre.task(f-1)~=data_pre.task(f)
         pre_task=[pre_task data_pre.task(f)];
    else
         pre_task=[pre_task 0];
    end
    f=f+1;
end

pre_task=transpose(pre_task);
pretask_filled=ones(size(data_pre.time))*0;
pretask_filled(:,1)=pre_task;
temp=transpose(pretask_filled);
pretask_filled=temp(:);


data_stim.EMGs(1,:)=[];
data_stim.time(1,:)=[];
data_stim.task(1)=[];
data_stim.targ(1)=[];
data_stim.curs(1)=[];
data_stim.trial(1)=[];
temp=data_stim.EMGs';
stim_emg=temp(:);
temp=data_stim.time';
stim_time=temp(:);

stimtask_len=max(size(data_stim.task));
f=2;stim_task=1;
while f<=stimtask_len
    if data_stim.task(f-1)~=data_stim.task(f)
         stim_task=[stim_task data_stim.task(f)];
    else
         stim_task=[stim_task 0];
    end
    f=f+1;
end

stim_task=transpose(stim_task);
stimtask_filled=ones(size(data_stim.time))*0;
stimtask_filled(:,1)=stim_task;
temp=transpose(stimtask_filled);
stimtask_filled=temp(:);


data_post.EMGs(1,:)=[];
data_post.time(1,:)=[];
data_post.task(1)=[];
data_post.targ(1)=[];
data_post.curs(1)=[];
data_post.trial(1)=[];
```

```
temp=data_post.EMGs';
post_emg=temp(:);
temp=data_post.time';
post_time=temp(:);

posttask_len=max(size(data_post.task));
f=2;post_task=1;
while f<=posttask_len
    if data_post.task(f-1)~=data_post.task(f)
         post_task=[post_task data_post.task(f)];
    else
         post_task=[post_task 0];
    end
    f=f+1;
end

post_task=transpose(post_task);
posttask_filled=ones(size(data_post.time))*0;
posttask_filled(:,1)=post_task;
temp=transpose(posttask_filled);
posttask_filled=temp(:);

f=0;
react_t=0;
n_pre=length(data_pre.task);
n_stim=length(data_stim.task);
n_post=length(data_post.task);


%Pre processing of emg

norm_freq = 1000;
n = 5;
Wn_band = [10, 500]/norm_freq;
Wn_low = 25/norm_freq;
[b1, a1] = butter(n,Wn_band,'bandpass');
[b2, a2] = butter(n,Wn_low,'low');

%pre
y1 = filter(b1,a1,pre_emg);  %Bandpass filter
y2 = abs(y1); %Full wave rectified
pre_emg = filter(b2,a2,y2); %low pass filter
%stim
y1 = filter(b1,a1,stim_emg);  %Bandpass filter
y2 = abs(y1); %Full wave rectified
stim_emg = filter(b2,a2,y2); %low pass filter
%stim
y1 = filter(b1,a1,post_emg);  %Bandpass filter
y2 = abs(y1); %Full wave rectified
post_emg = filter(b2,a2,y2); %low pass filter
```

```
%%Finding reaction times

%pre
pre_ind=find(pretask_filled==3);
react_time_pre=0;
for i=1:max(size(pre_ind))
    t1=pre_time(pre_ind(i));
    targ=data_pre.targ(floor(pre_ind(i)/500));
    j=pre_ind(i)+1;f=0;
    while pre_emg(j)<targ
        j=j+1;
        if pretask_filled(j)==1
            f=-1;
            break;
        end
    end

    if f==-1
        react_time_pre=[react_time_pre NaN];
    else
        t2=pre_time(j);
        react_time_pre=[react_time_pre t2-t1];
    end
end

react_time_pre(1)=[];

%stim
stim_ind=find(stimtask_filled==3);
react_time_stim=0;
for i=1:max(size(stim_ind))
    t1=stim_time(stim_ind(i));
    targ=data_stim.targ(floor(stim_ind(i)/500));
    j=stim_ind(i)+1;f=0;
    while stim_emg(j)<targ
        j=j+1;
        if stimtask_filled(j)==1
            f=-1;
            break;
        end
    end
    if f==-1
        react_time_stim=[react_time_stim NaN];
    else
        t2=stim_time(j);
        react_time_stim=[react_time_stim t2-t1];
    end
end
react_time_stim(1)=[];
```

```
%post
post_ind=find(posttask_filled==3);
react_time_post=0;
for i=1:max(size(post_ind))
    t1=post_time(post_ind(i));
    targ=data_post.targ(floor(post_ind(i)/500));
    j=post_ind(i)+1;f=0;
    while post_emg(j)<targ
        j=j+1;
        if posttask_filled(j)==1
            f=-1;
            break;
        end
    end
    if f==-1
        react_time_post=[react_time_post NaN];
    else
        t2=post_time(j);
        react_time_post=[react_time_post t2-t1];
    end
end
react_time_post(1)=[];


%plotting the reaction times

figure();

react_t=[react_time_pre react_time_stim react_time_post];
plot(react_t);ylabel('Reaction times');xlabel('Trials');
hold on;
trials_pre=max(data_pre.trial);
trials_stim=max(data_stim.trial);
trials_post=max(data_post.trial);
line([trials_pre trials_pre],[0 1],'Color','k','LineWidth',2);
line([trials_pre+trials_stim trials_pre+trials_stim],[0 1],'Color','k','LineWidth',2);
line([trials_pre+trials_stim+trials_post trials_pre+trials_stim+trials_post],[0 1],'Color','k');
```

4.2 MATLAB script to find EEG-EMG coherence using *mscohere*

```
%This code reads EEG data from the output of CleanData.m (artifact free EEG data)
%and EMG data from a mat file.
%EMGdata is bandpass filtered, full wave rectified and low pass filtered.
%Coherence is found using mscohere

clear all
clc

%EEG
eeg_mat=load('20140708162438_PostEEG_testClean.mat');
```

```
cl=['r-';
    'r:';
    'g-';
    'b-';
    'y-';
    'm-';
    'k-';
    'k:'];
eeg=eeg_mat.EEGclean;
eeg_markers=eeg_mat.events;
eeg_time=eeg_mat.time;
Fs=eeg_mat.Fs;
chan=6; %choose channel of eeg

%EMG
load('20140708T163146_testathiraPos.mat');
Fs=2000;

%pre process emg
data.EMGs(1,:)=[];
data.time(1,:)=[];
data.task(1,:)=[];
data.targ(1,:)=[];
data.curs(1,:)=[];
data.trial(1,:)=[];

%% plot eeg data
figure();hold on;
for i=1:8
plot(eeg_time,eeg(:,i),cl(i));
end
plot(eeg_time,eeg_markers*10^8,'k-','LineWidth',2);
hold off

%% plotting raw emg data and markers
h1=figure(1);
temp=transpose(data.EMGs);
emg=temp(:);
plot(emg);
hold on
yl=ylim;
emgtask_len=max(size(data.task));
f=2;emg_task=1;
while f<=emgtask_len
    if data.task(f-1)~=data.task(f)
         emg_task=[emg_task data.task(f)];
    else
         emg_task=[emg_task 0];
    end
    f=f+1;
end
```

```
 emg_task=emg_task';
emgtask_filled=ones(size(data.time))*0;
emgtask_filled(:,1)=emg_task;
temp=transpose(emgtask_filled);
emgtask_filled=temp(:);
temp=transpose(data.time);
emg_time=temp(:);

title('Choose task window to calculate coherence');
plot(emgtask_filled,'k-','LineWidth',2);
ylabel('raw emg reading (with markers)');xlabel('time');

%pick EMG data segment to process
[xemg,yemg] = ginput(1);
line([xemg, yemg],[yl(1), yl(2)],'Color','r','LineWidth',2)
hold off
close(h1)

%% process emg data

norm_freq = 1000;
n = 5;
Wn_band = [10, 500]/norm_freq;
Wn_low = 25/norm_freq;
[b1, a1] = butter(n,Wn_band,'bandpass');
[b2, a2] = butter(n,Wn_low,'low');
y1 = filter(b1,a1,emg);  %Bandpass filter
y2 = abs(y1); %Full wave rectified
emg = filter(b2,a2,y2); %low pass filter

%% obtain windows to calculate coherence

trial_no=data.trial(floor(xemg/500));
[emg_ind temp]=find(emgtask_filled==3);
emg_x1=emg_ind(trial_no);
emg_wind=emg(emg_x1);
emg_x2=emg_x1;
f=0;
while f==0
    emg_x2=emg_x2+1;
    emg_wind=[emg_wind emg(emg_x2)];
    f=emgtask_filled(emg_x2);
end

[eeg_ind temp]=find(eeg_markers==3);
eeg_x1=eeg_ind(trial_no);

temp=eeg(eeg_x1,1);
f=0;
eeg_x2=eeg_x1;
while f==0
```

```
   eeg_x2=eeg_x2+1;
   temp=[temp eeg(eeg_x2,1)];
   f=eeg_markers(eeg_x2);
end
for i=1:8
    eeg_wind(:,i)=eeg(eeg_x1:eeg_x2-1,i);
end


%% EEG power spectrum eeg with ~0.25sec window and 50% overlap
ovl=50;
wind=0.25*Fs;
Hs=spectrum.welch('Hamming',wind,ovl);
figure();hold on;
for i=1:8
    hpsd=psd(Hs,eeg(eeg_x1:eeg_x2,i),'Fs',Fs);
    Pw(:,i)=hpsd.data;
    Ff(:,i)=hpsd.frequencies;
    % powerspectrum 0-50Hz
    plot(Ff(1:ceil(wind*50/Fs),i),Pw(1:ceil(wind*50/Fs),i),cl(i,:),'LineWidth',2);
    title('Power spectrum of eeg');
end
%% EMG power spectrum emg with ~0.25sec window and 50% overlap ??
ovl=50;
wind=0.25*Fs;
Hs=spectrum.welch('Hamming',wind,ovl);
    hpsd=psd(Hs,emg(emg_x1:emg_x2),'Fs',Fs);
    Pw=hpsd.data;
    Ff=hpsd.frequencies;
    hps=figure(1);
    % powerspectrum 5-200Hz
    plot(Ff(5:ceil(wind*200/Fs)),Pw(5:ceil(wind*200/Fs)),cl(i,:),'LineWidth',2);
    title('power spectrum of emg');

%% Downsample emg in case of different frequencies

temp=emg_wind;
emg_wind=ones(max(size(eeg_wind)),1);
k=1;
for i=1:4:2513
emg_wind(k)=temp(i);
k=k+1;
end

%% Coherence
figure();
mscohere(eeg_wind(:,chan),emg_wind,[],[],[],Fs);
title(['Coherence estimate via Welch between filtered eeg', ' and ', 'rectified emg'])
```

4.3 MATLAB script to find EEG-EMG coherence using hilbert transform

```
%This program finds the coherence between two simulated signals using the
%hilbert transform

clear all
clc

%% generating signals
%type 1
t=0:0.1:100;
rand1=rand(1,length(t));
rand2=rand(1,length(t));
sig1=1*cos(2*pi*90/360+2*pi/21*t)+4*cos(+2*pi*00/360-2*pi/10*t);
sig2=1*cos(2*pi*60/360+2*pi/21*t)+4*cos(-2*pi*70/360-2*pi/10*t);
%sig1=1*cos(2*pi*90/360+2*pi/21*t);
%sig2=1*cos(2*pi*60/360+2*pi/21*t);
Gain_noise=10;
eeg=sig1+Gain_noise*(-1+2*rand1);
emg=sig2+Gain_noise*(-1+2*rand2);
len=length(eeg);
teeg=1:len;
temg=1:len;

%% generating signals
% type 2: add noise to the coherent frequency

Gain_noise=2;
t=0:0.1:100;
omega11=2*pi/21+Gain_noise*rand(1);
omega21=2*pi/21+Gain_noise*rand(1);
omega12=2*pi/10+Gain_noise*rand(1);
omega22=2*pi/10+Gain_noise*rand(1);
sig1=1*cos(2*pi*90/360+omega11*t)+4*cos(+2*pi*00/360-omega12*t);
sig2=1*cos(2*pi*60/360+omega21*t)+4*cos(-2*pi*70/360-omega22*t);
eeg=sig1;
emg=sig2;
len=length(eeg);
teeg=1:len;
temg=1:len;

%%

%eeg=abs(detrend(eeg));
%emg=abs(detrend(emg));
%emg=detrend(emg);

%parameters
eegTgap=0.0005;  %time gap in eeg signal
emgTgap=0.0005;  %time gap in emg signal
```

```
win=0.25;   %window size
Fs=2000;    %sampling frequency
Flow=-30;     %lower limit of frequency range
Fhigh=30;   %upper limit of frequency range

%hilbert transform
Heeg=hilbert(eeg);
Hemg=hilbert(emg);
%amplitudes
Aeeg=sqrt(real(Heeg).^2+imag(Heeg).^2);
Aemg=sqrt(real(Hemg).^2+imag(Hemg).^2);
%phases
Phieeg=atan2(imag(Heeg),real(Heeg));
Phiemg=atan2(imag(Hemg),real(Hemg));
%frequencies
feeg=gradient(Phieeg,eegTgap)/2*pi;
femg=gradient(Phiemg,emgTgap)/2*pi;

%moving averagefor frequencies
mask=ones(win*Fs,1)/(win*Fs);
avgfeeg=conv(feeg,mask,'same');
avgfemg=conv(femg,mask,'same');

%finding cross spectrum and coherence

%eeg->emg
for t=1:max(teeg)
    freq=avgfeeg(t);
    if freq>=Flow && freq<=Fhigh
        [mintemp indemg]=min(abs((avgfemg-freq)));
        %find the average frequency in emg closest to the desired frequency
        indeeg=t;
        %window in eeg amd emg corresponding to the average frequency
        for k=1:Fs*win
             windemg(k)=indemg+(Fs*win/2)+1-k;
        end
        for k=1:Fs*win
             windeeg(k)=indeeg+(Fs*win/2)+1-k;
        end
        %cross spectrum
        windeeg=sort(windeeg);
        windemg=sort(windemg);

        for k=1:Fs*win
            if windeeg(k)<1 || windeeg(k)>max(size(eeg))
                Awindeeg(k)=0;
            else
                Awindeeg(k)=Aeeg(windeeg(k));
            end

            if windemg(k)<1|| windemg(k)>max(size(emg))
```

```
                Awindemg(k)=0;
            else
                Awindemg(k)=Aemg(windemg(k));
            end
        end


        CrSpecEeg2Emg=Awindeeg.*Awindemg;

        %cohEeg2Emg(t)=mean(CrSpecEeg2Emg.^2)/(mean(Awindeeg.^2))*sqrt(mean(Awindemg.^2));
        cohEeg2Emg(t)=dot(Awindeeg,Awindemg)^2/(dot(Awindeeg,Awindeeg)*dot(Awindemg,Awindemg));

    else
        cohEeg2Emg(t)=0;
    end
end

m=ones(Fs*win,1)/(Fs*win);
c=conv(cohEeg2Emg,m,'same');
subplot(3,1,1);mscohere(eeg,emg);
title(strcat('Gain noise to coherent frequency = ',num2str(Gain_noise),' (Using mscohere)'));ylim([0 1]
subplot(3,1,2);plot(cohEeg2Emg);ylabel('Coherence');
title('Using Hilbert Transform');ylim([0 1]);xlabel('Time');
subplot(3,1,3);plot(c);title('Using Hilbert Transform (after averaging)');
ylabel('Coherence');ylim([0 1]);xlabel('Time');
```

Bibliography

[1] Feilden, Tom (26 January 2012). "'Human enhancement' comes a step closer". BBC. Retrieved July 21, 2014

[2] V. L. Roger, A. S. Go, D. M. Lloyd-Jones et al., et al., "Heart Disease and Stroke Statistics2012 Update: A Report From the American Heart Association," Circulation, vol. 125, no. 1, pp. e2-e220, 2012.

[3] M. A. Nitsche, and W. Paulus, Excitability changes induced in the human motor cortex by weak transcranial direct current stimulation, J Physiol vol.527(3), pp. 633-639, September 2000.

[4] M. A. Nitsche, A. Schauenburg, N. Lang, D. Liebetanz, C. Exner, W. Paulus, and F. Tergau, Facilitation of implicit motor learning by weak transcranial direct current stimulation of the primary motor cortex in the human, J Cogn Neurosci, vol. 15(4), pp. 619-626, May 2003.

[5] Nitsche MA, Schauenburg A, Lang N, Liebetanz D, Exner C, Paulus W,Tergau F: Facilitation of implicit motor learning by weak transcranial direct current stimulation of the primary motor cortex in the human. J Cogn Neurosci 2003, 15(4):619626.

[6] Antal A, Varga ET, Nitsche MA, Chadaide Z, Paulus W, Kovcs G, Vidnynszky Z: Direct current stimulation over MT+/V5 modulates motion aftereffect in humans. Neuroreport 2004, 15(16):24912494

[7] Tanaka S, Hanakawa T, Honda M, Watanabe K: Enhancement of pinchforce in the lower leg by anodal transcranial direct current stimulation.Exp Brain Res 2009, 196(3):459465

[8] Madhavan S, Weber KA, Stinear JW: Non-invasive brain stimulation enhances fine motor control of the hemiparetic ankle: implications for rehabilitation. Exp Brain Res 2011, 209(1):917

[9] A. Dutta, S. Chugh, Effect of transcranial direct current stimulation on cortico-muscular coherence and standing postural steadiness, The 2nd IASTED International Conference on Assistive Technologies 2012.

[10] Nitsche MA, Roth A, Kuo MF, Fischer AK, Liebetanz D, Lang N, Tergau F, Paulus W: Timing-dependent modulation of associative plasticity by general network excitability in the human motor cortex. J Neurosci 2007, 27(14):38073812

[11] A. Dutta, A. Banerjee, M.A. Nitsche, "Facilitating myoelectric-control with transcranial direct current stimulation", Converging Clinical and Engineering Research on Neurorehabilitation Biosystems & Biorobotics,1, pp. 847-851, 2013

[12] Walter J. Freeman (2007) Hilbert transform for brain waves. Scholarpedia, 2(1):1338

[13] Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, Yen N-C, Tung CC, Liu HH. (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc. R. Soc. Lond. 454: 903-995.

[14] Le Van Quyen M, Foucher J, Lachaux J-P, Rodriguez E, Lutz A, Martinerie J, Varela F. (2001) Comparison of Hilbert transform and wavelet methods for the analysis of neuronal synchrony. J. Neurosci. Meth. 111: 83-98.

[15] Quiroga RQ, Kraskov A, Kreuz T, Grassberger P. (2002) Performance of different synchronization measures in real data: A case study on electroencephalographic signals. Physical Rev E, 6504:U645-U6 58 - art. no. 041903.

[16] Freeman WJ. (2005) Origin, structure, and role of background EEG activity. Part 3. Neural frame classification. Clin. Neurophysiol. 116 (5): 1118-1129

[17] Pikovsky A, Rosenblum M, Kurths J. (2001) Synchronization A Universal Concept in Non-linear Sciences. Cambridge UK: Cambridge U.P.