Assignment4
Source localization

Vangjush Komini
r0612470
vangjush.komini@uzleuven.be

August 28, 2016

Biomedical Data Processing, Part II Course

icon1.png

1 Introduction

Brain is a network of 10^{10} neurons. Each neuron could generate a small electrical activity which alone could not be picked up from the electrodes. This is due to the interference of other surrounding activities. In case of a big number of neurons simultaneously active, the outcome signal is sufficiently high to be picked from the scalp electrodes. This electrical activity could be modeled as a dipole. Brain electrical activity has been investigated via surface electrodes since 1930. Widely known as electroencephalogram (EEG) is the potential difference of electrodes in a time course. This readings has facilitated the diagnoses of many neurological diseases.

In simulation setup, dipole modeling could enable the generation of electrode potential via the solution of Poisson equation and Neuman boundary conditions. This is also known as the forward problem. Additionally, potential given the electrode, the underlying dipole localization and direction could also be estimated via different optimization setups. This is also known to be the inverse problem where the final outcome is source localization [?].

2 Forward problem

2.1 Theory

2.1.1 Poisson's equation

Poisson's equations computes the potentials distribution in a volume conductor given the applied current source. The current density $J(x, y, z)\{\frac{A}{m^2}\}$ is a vector field whereas its divergence defines the current source density $I_m = \nabla J\{\frac{A}{m^3}\}$. In case of a enclosed volume $r_1(x_1, y_1, z_1)$ by current sink of positively charge represent the removal of positive ions. This yield a singularity current source density $-I\delta(r - r_1)$. A small volume around the current source $r_2(x_2, y_2, z_2)$ is also constructed. Injection of positively charge ions into this volume is also another singular current source density $I\delta(r - r_2)$. Superposition of this cases outcome:

$$\nabla J = I\delta(r - r_2) - I\delta(r - r_1) \tag{2.1.1}$$

As regard to the current density J and the electrical field E Ohm's law gives the relation via:

$$J = E\sigma \tag{2.1.2}$$

where $\sigma(r)$ is rank-1 tensor representation of the anisotropic conductivity which varies with the direction. Under the Faraday's law in quasi-static conditions ($\nabla x E = 0$) a relation between the electrical field and scalar potential field is given via gradient operation:

$$E = -\nabla V \tag{2.1.3}$$

The direction of ∇V represents the most rapid increase of the potential V at a give point. It sign minus is based on the orientation flow from high to low potential. Poisson equation is obtained via combination of equations **??**, **??** and **??**.

$$\nabla(\sigma\nabla(V)) = -I_m = I\delta(r - r_2) - I\delta(r - r_1) \tag{2.1.4}$$

2.1.2 Boundary conditions

Inability to diminish charge particles at the compartments interface is seen as the current leaving one region is equal to the current entering another region. Additionally the very low conductivity of the human head does not permit any charge to penetrate the head. These two conditions presented by Neuman are equationed below:

$$J_1 e_n = J_2 e_n \tag{2.1.5}$$

$$J_1 e_n = 0 \tag{2.1.6}$$

2.1.3 The current dipole

The current source and sink remove the same amount of charges I consequently dipole modeling is the best approach. Its momentum d is defined via $d = I * p * e_d$ where $||d|| = ||I * p * e_d||$ and e_d the orientation. Orthogonal projection of these dipole onto Cartesian axes is $d = d_x e_x + d_y e_y + d_z e_z$. Due to the linearity of Poisson equation it could be decomposed into:

$$V(r, r_{dip}, d) = d_x V(r, r_{dip}, e_x) + d_y V(r, r_{dip}, e_y) + d_z V(r, r_{dip}, e_z) \tag{2.1.7}$$

2.1.4 Solution of the forward problem

Solution of the equation **??** with gives the scalp potential at an electrode due to a singular dipole with momentum d. A current dipole with momentum $d = de_d$ positioned at r_{dip} in an infinite conductor with conductivity σ_i produces a potential field at distance r as:

$$V(r, r_{dip}, d) = \frac{d(r - r_{dip})}{2\pi\sigma||r - r_{dip}||^3} \tag{2.1.8}$$

which is also outlined in figure ?? The easiest model to solve the Poisson equation for a head model is a the spherical head model which is fast and easy to be implemented. More sophisticated approaches are Boundary Element Method (BEM), Finite Element Method (FEM) and Finite Difference Method (FDM) which are generally employed for realistic shape instead of spherical which is not realistic. Due to the difference in conductivity of skull from scalp and brain a three shell concentric spherical head has to been proposed for spherical head model. Semi-analytic solution for Poisson equation utilizing the proposed geometrical modelling is:

$$V = \frac{1}{4\pi S R^2} \sum_{i=1}^{\infty} \frac{X(2i+1)^3}{g_i(i+1)i} b^{i-1} \{ i d_r P_i(cos\theta) + i d_r P_i(cos\theta) \} \tag{2.1.9}$$

given that:

$$g_i = \{(i+1)X + i\}\{\frac{iX}{i+1} + 1\} + (1-X)\{(i+1)X + i\}(f_1^{i_1} - f_2^{i_1}) - i(1-X)^2\{\frac{f_1}{f_2}\}^2 \tag{2.1.10}$$

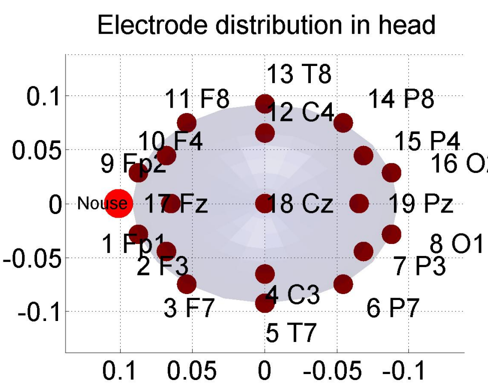
where:

- d_r is the radial component
- d_t is the tangential component
- R is the radius of the outer shell
- S is the conductivity of the scalp and brain tissue
- X is the ratio between the skull and soft tissue conductivity
- b is the relative distance of the dipole from the center
- ϕ is the polar angle of the surface point
- $P_i(.)$ is the Legendre polynomial
- $P_i^1(.)$ is the associated Legendre polynomial
- i is an index
- $i_1 = 2 * i + 1$
- r_1 is the radius of the inner shell
- r_2 is the radius of the middle shell
- $f_1 = \frac{r_1}{R}$
- $f_2 = \frac{r_2}{R}$

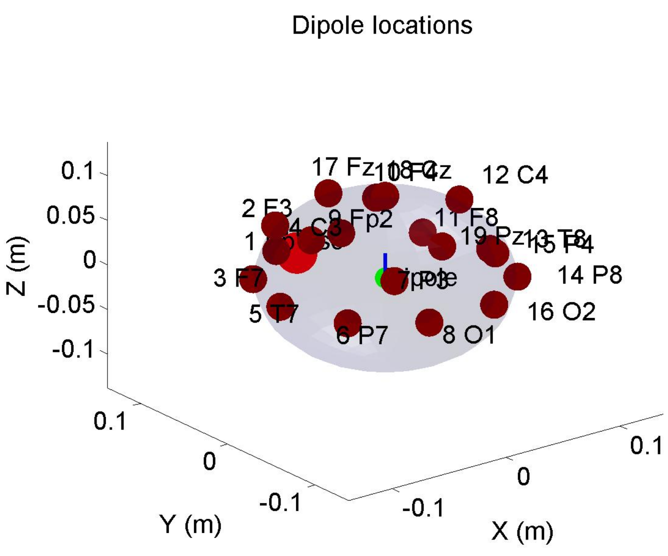
This is the equation for scalp potentials produced from an dipole located in the z axis. In case of an of an arbitrary dipole, Euler angles are employed for the rotation of the coordinate system accordingly.

2.2 EEG montage

Electrode placement is of great importance to display the EEG adequately. There is a big diversity of EEG montage which in same cases it should be adopted for specific applications. The most accepted model is 10/20 for classical EEG recording. There are symmetric placement between Nasion-Inion marked as Frontal pole (F_p, Central (C), Parietal (P), occipital (O) and Temporal T. The electrodes named with zero (Z) are the ones corresponding at the central line. The odd numbers are used for points on the left hemisphere whereas the even number for points on the right of the hemisphere. There is a symmetry of the node placement respectively to the Nassion-Ion line ??.



(a) Epical view



(b) 3D view

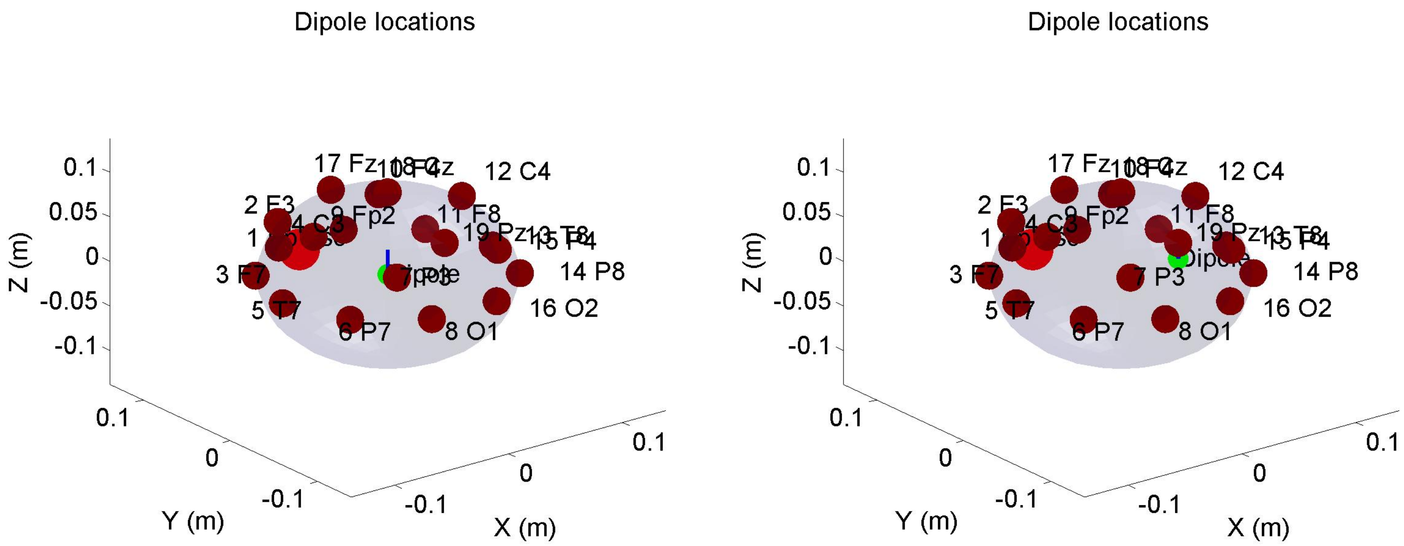
Figure 2.2.1: EEG montage for the simulation model

In figure ?? is the spherical model which will be used as well as the electrode placement as the most standard configuration. The spherical model we are using consist o a three shell model figure ?? with respective radius:

- $R_{skull} = 0.08[m]$
- $R_{scalp} = 0.086[m]$
- $R_{brain} = 0.092[m]$

2.3 Dipole location

The modeled dipole has two main parameters position and direction which are encoded into Cartesian coordinates. In figure **??** a simple dipole location at the center which is oriented towards z axis. This is acquired via command line $Show-Dipole-Placement([0,0,0,0,0,1],hm,4)$ where $Show-Dipole-Placement([x_p,y_p,z_p,Ori_x,Ori_y,Ori_z],Head-Model,Nr_{Figure})$. In case it would be necessary to place an dipole half way from the center towards the right ear oriented towards z axis figure **??** that would be possible via command $Show-Dipole-Placement([0.05,0,0,0,0,1],hm,5)$.



(a) Dipole modeled in the center

(b) Dipole located halfway from center to the right ear.

Figure 2.3.1: Dipole locations

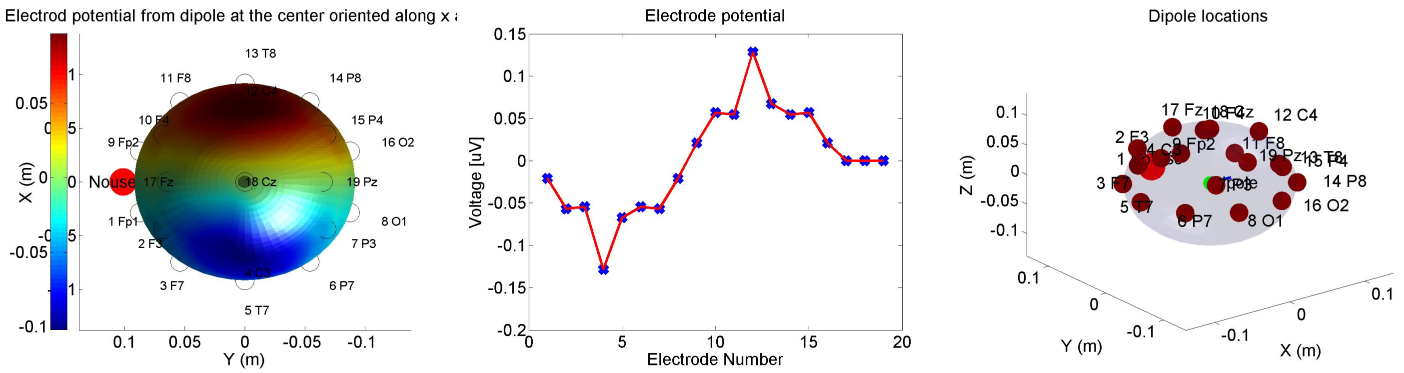
2.3.1 Potential computation

Spherical head model has been utilized to solve the Poisson equations and provides the potential distribution at particular coordinate due a dipole at particular direction and position. In case there will be a dipole located in the center oriented towards the x position the potential field distribution will be as in the figure **??**. Since the dipole is not positioned along the z axis from [?] are all the necessary transformation needed to for the coordinates rotation which are also outlined in Appendix **??**. The generalization of equation **??** for any dipole location can be written as:

$$V_x = T_x D_x \quad (2.3.1) \qquad V_y = T_y D_y \quad (2.3.2) \qquad V_z = T_z D_z \quad (2.3.3)$$

Figure 2.3.2: Voltage computation

whereas the potential V at R_2 due to dipole D located at R_1 is: $V = V_x + V_y + V_z$. Regarding the electrode potential values the plot in figure **??** outline individual values. The dipole location and orientation is plotted in figure **??**. The potential field is therefore mostly oriented towards the right ear in the figure **??** which coincides with the orientation of the dipole inside the head.

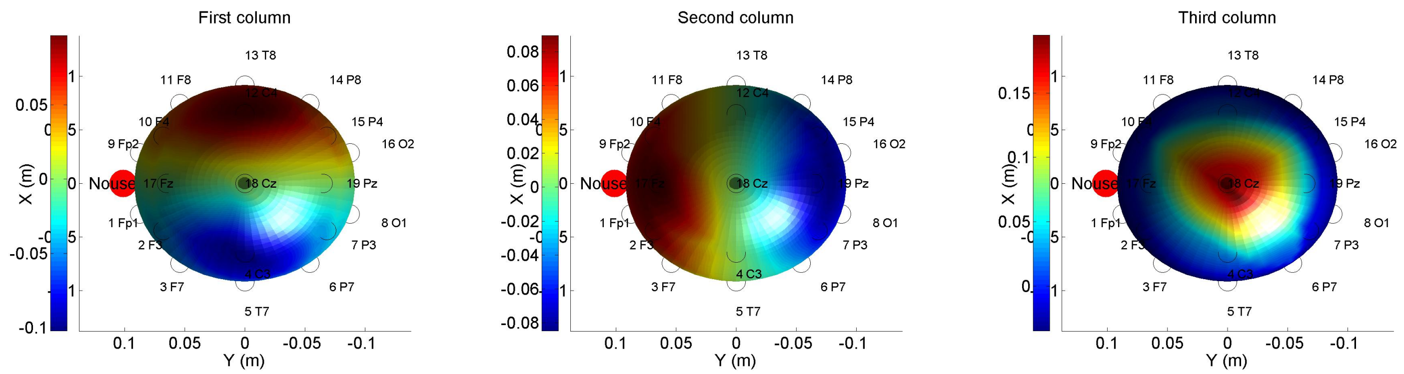


(a) Potential distribution. (b) Potential at each electrode. (c) Dipole location.

Figure 2.3.3: Potential distribution

2.4 Lead Field Matrix

The lead field matrix contains the potential intensity computed from the spherical model, oriented along x, y and z axis. The final potential distribution corresponding to the a particular dipole orientation is the superposition of each lead field matrix along each axis, whereby their contribution is proportional to the orientation of the dipole. In figure **??** are the potential distribution of each component where it is clearly noted maximum along X,Y and Z axis.

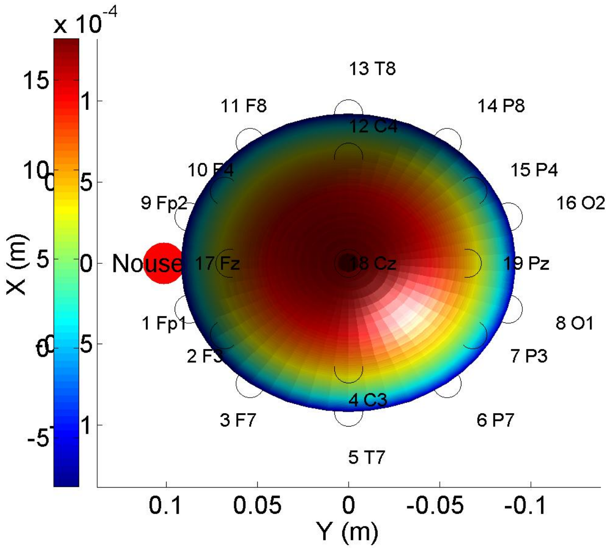


(a) Potential generated along x (b) Potential generated along y (c) Potential generated along z

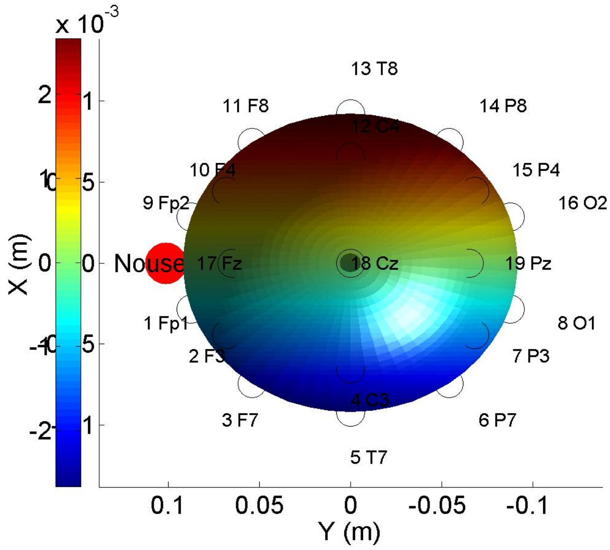
Figure 2.4.1: Lead field matrix

2.5 Linearity of Maxwell equation

The numerical verion of the Maxwel equation is linear therefore the principle of superposition is applied. Therefore in this case, the potential produced from the dipole D1 and D2 located at the center where the first one oriented along x **??** and the second one along z in figure **??** produces the potentials in figure **??** and figure **??**. Due to the linearity the final potential from these two dipole will be as in figure **??**. Differently this potential would be excactly the same as the potential distribution outcomed from a single dipole where the orientation is the vectorial sum of dipole D1 and D2 figure **??**. This equality is also testified from dipole distributions in figure **??**.

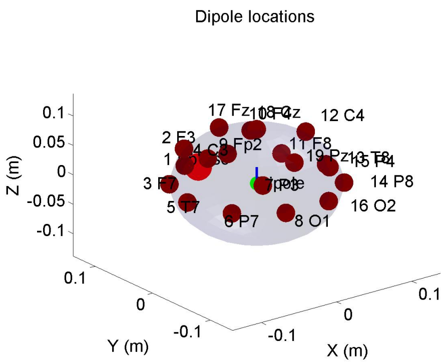


(a) Voltage from dipole oriented along x

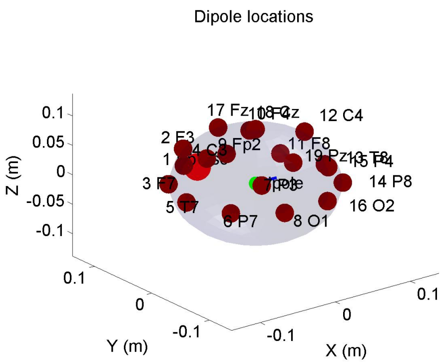


(b) Voltage from dipole oriented along y

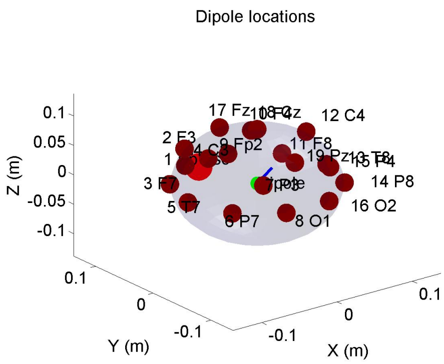
Figure 2.5.1: Voltage distribution



(a) Dipole along x

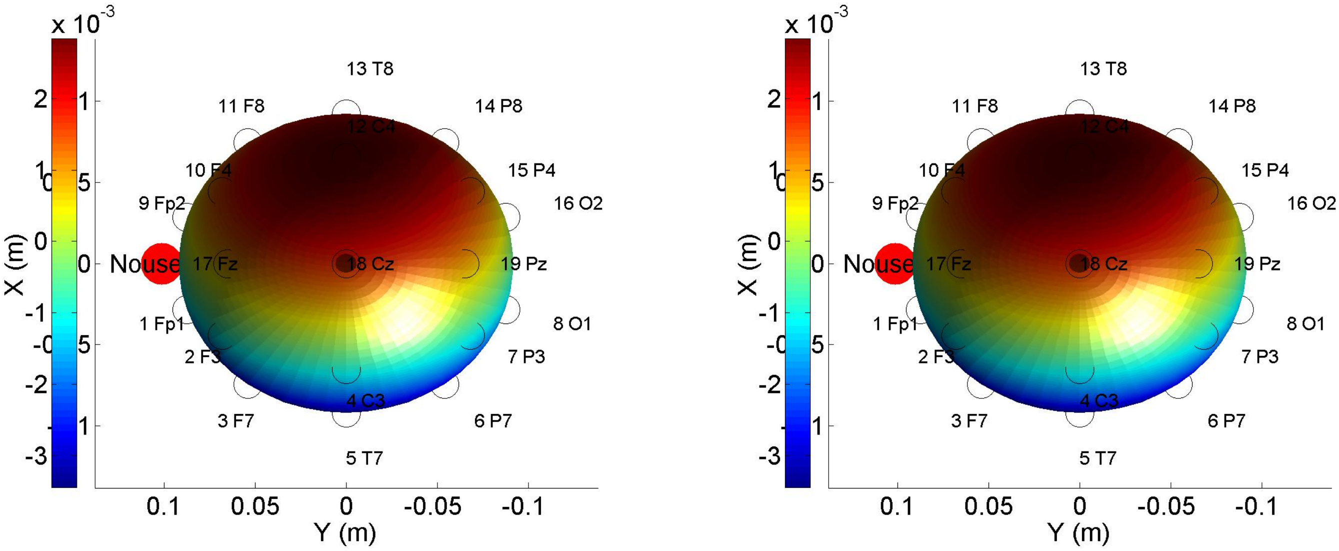


(b) Dipole along z



(c) Dipole with orientation along xz

Figure 2.5.2: Dipole location

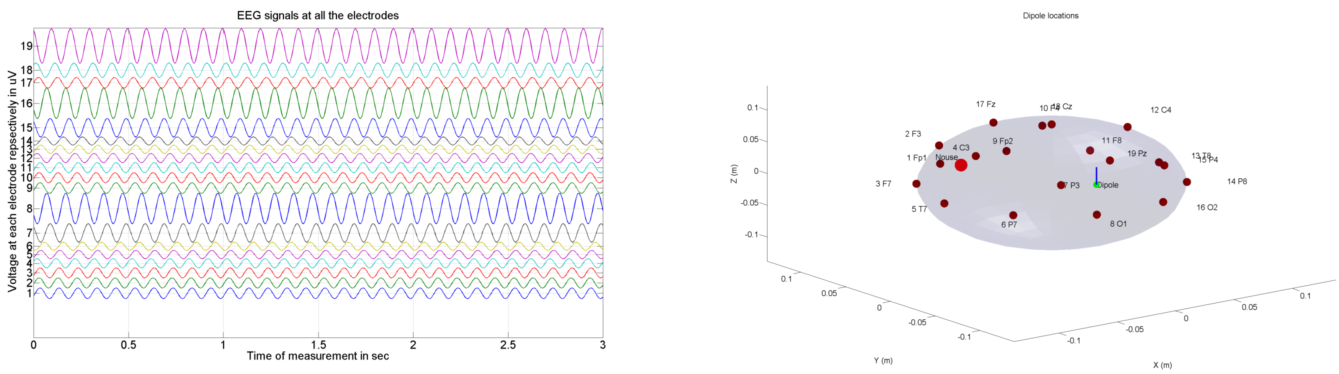


(a) Voltage superposition (b) Voltage from the dipole along xy

Figure 2.5.3: Potential distribution

2.6 EEG time series generation

Hereby EEG time series are being simulated by rotating along the xy plane the dipole located in the coordinates (0,-0.05,0.02) figure **??**. The sampling frequency is 200 Hz which is much higher than the dipole rotation frequency which is 10 Hz in our case. After 3 sec EEG simulation from the time series values plotted in figure **??** what can be observed is that the electrodes near the dipole has a much higher amplitude comparing to the ones sitting further from the dipole.



(a) EEG plot (b) Dipole

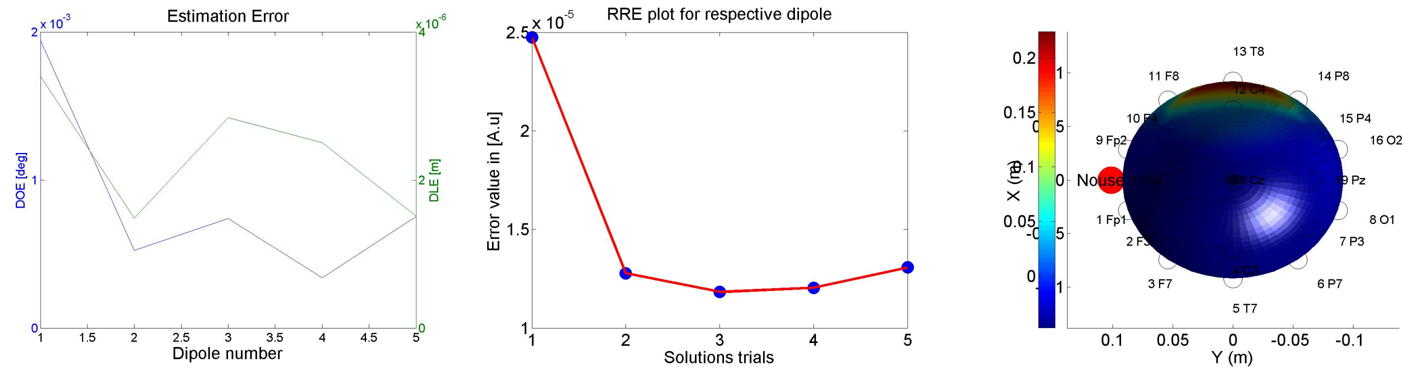
Figure 2.6.1: EEG simulation

3 Inverse problem

Inverse problem in EEG tends to estimate the source by its location and orientation given an instance measurement. This is an ill-posed problem due to its non-unique outcome and highly sensitive to the minor changes (unstable solution) [?]. Two set of methods in solving inverse problem in EEG exist including here, parametric and non-parametric. In this work parametric method is being utilized where it makes an exhaustive search upon a set of initial points for the best fit. The optimization is done iteratively until an arbitrary threshold is achieved.

3.1 Estimation of a dipole using a single time instance

Inverse method has been applied on the on potential distribution outcomed from the dipole located in the coordinates (0.06,0,0.01) oriented along x. The dipole orientation error (DOR) equation **??** and dipole location error (DLE) equation **??** for this particular measurement are outlined in figure **??** whereas the RRE plot in figure **??**. From these two figures a consistency of the errors is noted in both these scales for all the five trials. Whereas the potential to be used for this inverse problem is plotted in figure **??**.



(a) Localisation errors (b) RRE estimation (c) Voltage distribution to be used.

Figure 3.1.1: Inverse problem for a dipole located at (0.06,0,0) oriented along x

3.2 Estimation of a dipole using the simulated EEG

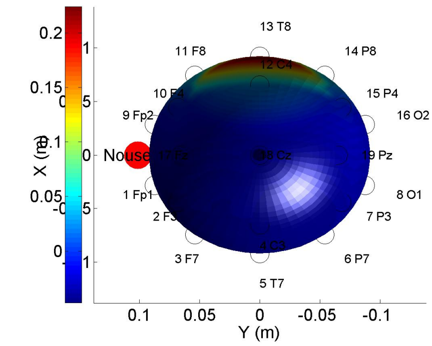
Hereby a certain potential is read from the simulated EEG in figure **??** and the via inverse model the dipole is located from this particular time instance of the EEG. In the table **??** are the dipole locations data for this particular implementation.

Table 1: Inverse problem with electrode position altered

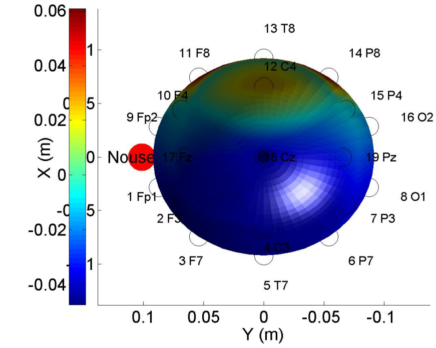
	X	Y	Z	$OriX$	$OriY$	$OriZ$	RRE
height$D1$	$4.74e-07$	$-5.00e-02$	$2.00e-02$	$6.03e-06$	$1.00e+00$	$1.79e-05$	$1.41e-05$
$D2$	$4.23e-02$	$-1.89e-02$	$-7.85e-02$	$3.65e-01$	$8.89e-01$	$-2.78e-01$	$4.77e-01$
$D3$	$3.50e-07$	$-5.00e-02$	$2.00e-02$	$4.45e-06$	$1.00e+00$	$-3.92e-05$	$2.64e-05$
$D4$	$7.14e-07$	$-5.00e-02$	$2.00e-02$	$9.08e-06$	$1.00e+00$	$2.79e-06$	$1.56e-05$
$D5$	$2.65e-05$	$-5.00e-02$	$2.00e-02$	$3.38e-04$	$1.00e+00$	$9.32e-05$	$3.92e-04$

3.3 Error due to not incorporating a malfunctioning electrode

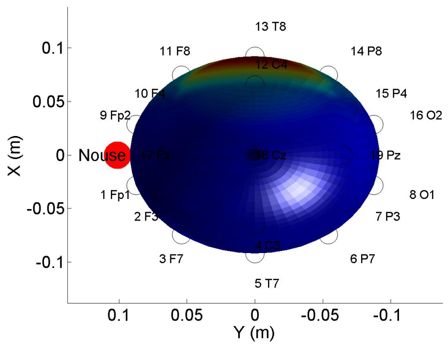
Hereby the importance of electrode functioning is verified. Using a dipole located at (0.06,0,0.01) oriented along x is used for this study. The potential distribution when all the electrodes are working is in figure ??, when the electrode P8 is off the potential distribution is as in the figure ?? whereas when the electrod F7 is off the potential distribution is as in the figure ??



(a) All the electrodes are working.



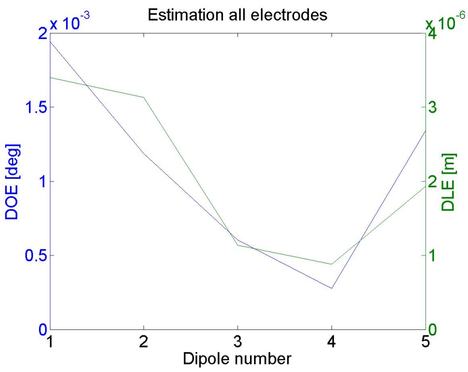
(b) P8 is off.



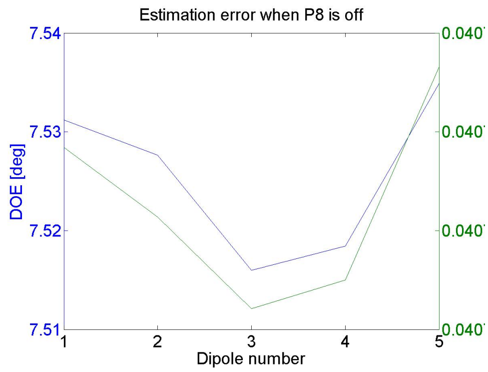
(c) F7 is off.

Figure 3.3.1: Voltage distribution measurement

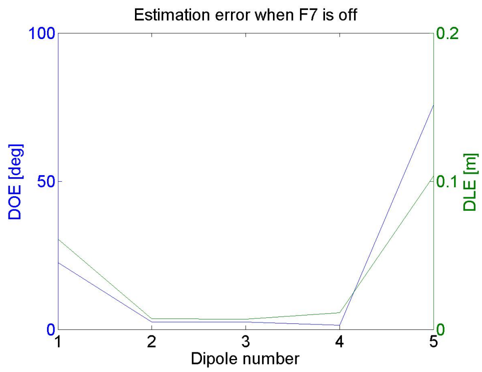
The inverse problem is applied in all these three cases. DOE and DLE plots for respective method are outlined in figure ??. As it is excepted these error are at lower scale for the case of full electrode well functioning ?? whereas for the P8 case these errors are significantly higher campare to the F7 case respectively in figure ?? and ??. At similiar figure is also the RRE plots in figure ?? where is is easitly observed the scale of error for the P8 case as compare to the F7 case. This is as a result that the P8 electrode location. This electrode is right in front of the main stream of the potential consequently being responsible for most flow. This flow is high in this direction due to the dipole orientation. Since the P8 is not functioning the potential measurement is measured dramatically wrong in this case.



(a) All the electrodes are working



(b) P8 electrodes is off



(c) F7 electrodes is off

Figure 3.3.2: Localisation error for the three cases

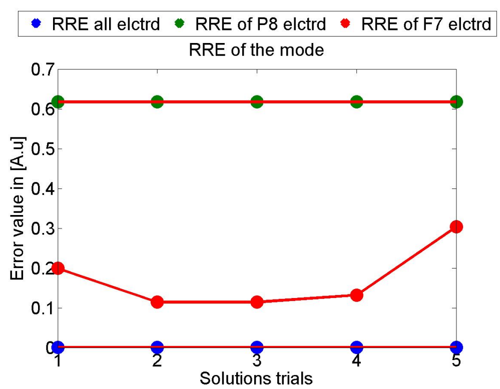


Figure 3.3.3: RRE estimation for the three cases

3.4 Error by different conductivity of the tissues

Skull conductivity is also an important parameter for the inverse model. Hereby an experimental study is being conducted where the inverse problem is being uplied over a head model with different skull conductivity as compare to the head model where the forward problem is being computed. Hereby 10 different dipole are being tested where 5 different trials are tested for each dipole in the inverse problem. In the table ?? the solution are listed where Trial1D1 stands for trial 1 dipole 1. Whereas the DOE and DLE are plotted respectively in table ?? and ?? for each dipole trial over five different estimation for each trial.

Table 2: Inverse problem with skull conductivity altered

	X	Y	Z	$OriX$	$OriY$	$OriZ$	RRE
height$Tria1D1$	$4.69e-02$	$1.66e-02$	$3.00e-02$	$7.07e-01$	$7.07e-01$	$-6.10e-06$	$1.59e-05$
$Tria1D2$	$4.69e-02$	$1.66e-02$	$3.00e-02$	$7.07e-01$	$7.07e-01$	$4.15e-06$	$9.22e-06$
$Tria1D3$	$4.69e-02$	$1.66e-02$	$3.00e-02$	$7.07e-01$	$7.07e-01$	$1.15e-05$	$1.41e-05$
$Tria1D4$	$4.69e-02$	$1.66e-02$	$3.00e-02$	$7.07e-01$	$7.07e-01$	$-1.62e-05$	$2.08e-05$
$Tria1D5$	$4.69e-02$	$1.66e-02$	$3.00e-02$	$7.07e-01$	$7.07e-01$	$2.45e-06$	$1.81e-05$
$Tria2D1$	$2.81e-02$	$4.19e-02$	$2.04e-02$	$7.07e-01$	$7.07e-01$	$7.42e-06$	$1.41e-05$
$Tria2D2$	$2.81e-02$	$4.19e-02$	$2.04e-02$	$7.07e-01$	$7.07e-01$	$-3.32e-05$	$2.12e-05$
$Tria2D3$	$2.81e-02$	$4.19e-02$	$2.04e-02$	$7.07e-01$	$7.07e-01$	$4.10e-06$	$1.55e-05$
$Tria2D4$	$2.82e-02$	$4.19e-02$	$2.04e-02$	$7.07e-01$	$7.07e-01$	$1.53e-05$	$3.80e-05$
$Tria2D5$	$2.81e-02$	$4.19e-02$	$2.04e-02$	$7.07e-01$	$7.07e-01$	$-1.69e-05$	$2.00e-05$
$Tria3D1$	$5.51e-03$	$1.09e-02$	$2.57e-02$	$1.43e-06$	$7.07e-01$	$7.07e-01$	$8.37e-06$
$Tria3D2$	$5.51e-03$	$1.09e-02$	$2.57e-02$	$3.05e-06$	$7.07e-01$	$7.07e-01$	$1.58e-05$
$Tria3D3$	$5.51e-03$	$1.09e-02$	$2.57e-02$	$6.80e-07$	$7.07e-01$	$7.07e-01$	$2.41e-05$
$Tria3D4$	$5.51e-03$	$1.09e-02$	$2.57e-02$	$1.20e-06$	$7.07e-01$	$7.07e-01$	$8.06e-06$
$Tria3D5$	$5.51e-03$	$1.09e-02$	$2.57e-02$	$5.60e-06$	$7.07e-01$	$7.07e-01$	$2.41e-05$
$Tria4D1$	$1.10e-02$	$3.03e-02$	$4.06e-02$	$-2.45e-06$	$-9.83e-06$	$1.00e+00$	$2.06e-05$
$Tria4D2$	$1.10e-02$	$3.03e-02$	$4.06e-02$	$-5.84e-06$	$5.67e-06$	$1.00e+00$	$2.29e-05$
$Tria4D3$	$1.10e-02$	$3.03e-02$	$4.06e-02$	$-7.02e-07$	$-7.14e-06$	$1.00e+00$	$1.27e-05$
$Tria4D4$	$1.10e-02$	$3.03e-02$	$4.06e-02$	$-1.46e-06$	$-1.00e-05$	$1.00e+00$	$1.84e-05$
$Tria4D5$	$1.10e-02$	$3.03e-02$	$4.06e-02$	$-6.65e-06$	$9.30e-07$	$1.00e+00$	$2.24e-05$
$Tria5D1$	$3.73e-02$	$7.99e-02$	$-1.53e-02$	$-1.35e-01$	$-6.77e-01$	$7.24e-01$	$2.93e-01$
$Tria5D2$	$3.16e-02$	$4.56e-02$	$2.72e-02$	$7.07e-01$	$1.69e-06$	$7.07e-01$	$2.15e-05$
$Tria5D3$	$3.16e-02$	$4.56e-02$	$2.72e-02$	$7.07e-01$	$2.04e-06$	$7.07e-01$	$4.44e-06$
$Tria5D4$	$3.16e-02$	$4.56e-02$	$2.72e-02$	$7.07e-01$	$-1.17e-05$	$7.07e-01$	$1.64e-05$
$Tria5D5$	$3.16e-02$	$4.56e-02$	$2.72e-02$	$7.07e-01$	$-6.21e-06$	$7.07e-01$	$1.68e-05$
$Tria6D1$	$1.64e-02$	$3.60e-02$	$6.53e-02$	$1.00e+00$	$-1.37e-05$	$-2.43e-05$	$1.73e-05$
$Tria6D2$	$1.64e-02$	$3.60e-02$	$6.53e-02$	$1.00e+00$	$-2.75e-08$	$1.05e-06$	$6.29e-06$
$Tria6D3$	$1.64e-02$	$3.60e-02$	$6.53e-02$	$1.00e+00$	$2.70e-06$	$9.33e-06$	$1.36e-05$
$Tria6D4$	$1.64e-02$	$3.60e-02$	$6.53e-02$	$1.00e+00$	$5.41e-06$	$8.77e-06$	$8.85e-06$
$Tria6D5$	$1.64e-02$	$3.60e-02$	$6.53e-02$	$1.00e+00$	$3.07e-06$	$1.72e-06$	$1.25e-05$
$Tria7D1$	$5.98e-02$	$6.93e-03$	$5.01e-02$	$5.77e-01$	$5.77e-01$	$5.77e-01$	$5.85e-05$
$Tria7D2$	$8.88e-02$	$-6.78e-03$	$1.20e-02$	$6.94e-01$	$1.68e-01$	$7.00e-01$	$2.55e-01$
$Tria7D3$	$5.99e-02$	$6.93e-03$	$5.01e-02$	$5.77e-01$	$5.77e-01$	$5.77e-01$	$1.73e-05$
$Tria7D4$	$5.99e-02$	$6.93e-03$	$5.01e-02$	$5.77e-01$	$5.77e-01$	$5.77e-01$	$1.38e-05$
$Tria7D5$	$5.99e-02$	$6.93e-03$	$5.01e-02$	$5.77e-01$	$5.77e-01$	$5.77e-01$	$1.38e-05$
$Tria8D1$	$1.67e-02$	$3.58e-02$	$5.08e-02$	$1.00e+00$	$1.10e-07$	$-5.77e-06$	$1.28e-05$
$Tria8D2$	$1.67e-02$	$3.58e-02$	$5.08e-02$	$1.00e+00$	$3.36e-06$	$8.28e-06$	$9.94e-06$
$Tria8D3$	$1.67e-02$	$3.58e-02$	$5.08e-02$	$1.00e+00$	$-9.13e-06$	$-1.84e-05$	$2.87e-05$
$Tria8D4$	$1.67e-02$	$3.58e-02$	$5.08e-02$	$1.00e+00$	$-5.01e-06$	$-1.73e-05$	$2.50e-05$
$Tria8D5$	$1.67e-02$	$3.58e-02$	$5.08e-02$	$1.00e+00$	$-4.31e-06$	$-7.02e-06$	$1.06e-05$
$Tria9D1$	$6.59e-03$	$5.97e-03$	$3.87e-02$	$-7.46e-07$	$7.07e-01$	$7.07e-01$	$4.04e-05$
$Tria9D2$	$6.59e-03$	$5.97e-03$	$3.87e-02$	$6.57e-07$	$7.07e-01$	$7.07e-01$	$1.04e-05$
$Tria9D3$	$6.59e-03$	$5.97e-03$	$3.87e-02$	$-1.38e-07$	$7.07e-01$	$7.07e-01$	$9.16e-06$
$Tria9D4$	$6.59e-03$	$5.97e-03$	$3.87e-02$	$1.38e-06$	$7.07e-01$	$7.07e-01$	$2.04e-05$
$Tria9D5$	$6.59e-03$	$5.97e-03$	$3.87e-02$	$2.78e-06$	$7.07e-01$	$7.07e-01$	$1.22e-05$
$Tria10D1$	$7.13e-04$	$4.09e-02$	$1.30e-03$	$-1.21e-06$	$1.00e+00$	$-5.74e-07$	$1.09e-05$
$Tria10D2$	$7.12e-04$	$4.09e-02$	$1.30e-03$	$7.88e-06$	$1.00e+00$	$-4.78e-05$	$3.57e-05$
$Tria10D3$	$7.13e-04$	$4.09e-02$	$1.30e-03$	$-3.34e-06$	$1.00e+00$	$-1.23e-05$	$1.19e-05$
$Tria10D4$	$7.14e-04$	$4.09e-02$	$1.30e-03$	$-6.28e-06$	$1.00e+00$	$-1.39e-06$	$1.43e-05$
$Tria10D5$	$7.13e-04$	$4.09e-02$	$1.30e-03$	$-5.90e-06$	$1.00e+00$	$-2.32e-05$	$1.69e-05$

Table 3: DOE for skull

	$D1$	$D2$	$D3$	$D4$	$D5$
height$Trial1$	$3.78e-04$	$3.18e-04$	$6.87e-04$	$9.83e-04$	$5.82e-04$
$Trial2$	$5.65e-04$	$1.91e-03$	$3.02e-04$	$1.15e-03$	$1.12e-03$
$Trial3$	$1.98e-04$	$7.40e-04$	$5.63e-04$	$4.09e-04$	$4.72e-04$
$Trial4$	$5.80e-04$	$4.66e-04$	$4.11e-04$	$5.79e-04$	$3.85e-04$
$Trial5$	$6.54e+01$	$7.27e-04$	$2.30e-04$	$1.02e-03$	$6.47e-04$
$Trial6$	$1.60e-03$	$6.03e-05$	$5.56e-04$	$5.90e-04$	$2.01e-04$
$Trial7$	$3.52e-03$	$2.56e+01$	$3.40e-03$	$5.79e-04$	$8.32e-04$
$Trial8$	$3.31e-04$	$5.12e-04$	$1.18e-03$	$1.03e-03$	$4.72e-04$
$Trial9$	$7.26e-05$	$3.60e-04$	$1.11e-04$	$8.11e-04$	$1.77e-04$
$Trial10$	$7.67e-05$	$2.78e-03$	$7.32e-04$	$3.69e-04$	$1.37e-03$

Table 4: DLE for skull

	$D1$	$D2$	$D3$	$D4$	$D5$
height$Trial1$	$2.16e-06$	$8.69e-07$	$1.05e-06$	$1.63e-06$	$1.53e-06$
$Trial2$	$9.94e-07$	$2.02e-06$	$1.06e-06$	$3.29e-06$	$1.50e-06$
$Trial3$	$6.71e-07$	$1.85e-06$	$2.42e-06$	$9.89e-07$	$2.35e-06$
$Trial4$	$2.24e-06$	$1.53e-06$	$9.52e-07$	$1.72e-06$	$1.54e-06$
$Trial5$	$5.49e-02$	$2.28e-06$	$3.64e-07$	$1.69e-06$	$1.71e-06$
$Trial6$	$2.47e-06$	$9.20e-07$	$1.50e-06$	$5.06e-07$	$9.78e-07$
$Trial7$	$1.54e-05$	$4.97e-02$	$4.18e-06$	$3.60e-06$	$3.19e-06$
$Trial8$	$7.02e-07$	$6.77e-07$	$2.66e-06$	$2.20e-06$	$8.37e-07$
$Trial9$	$3.42e-06$	$7.23e-07$	$6.98e-07$	$2.09e-06$	$1.17e-06$
$Trial10$	$1.61e-06$	$3.05e-06$	$1.06e-06$	$1.84e-06$	$1.89e-06$

In case the skull conductivity is not given the setup in figure **??** could be utilized for a particular measurement.

3.5 Inverse problem with electrode position altered

Similarly to the case when the skull conductivity was changed hereby a altering in the electrode position is performed. In table ?? are the result listed accordingly. Whereas the DOE and DLE are plotted respectively in table ?? and ?? for each dipole trial over five different estimation for each trial. The spherical angle is changed as in equation ?? for a particular displacement.

Table 5: Inverse problem with electrode position altered

	X	Y	Z	$OriX$	$OriY$	$OriZ$	RRE
height$Tria1D1$	$7.56e-03$	$9.50e-03$	$7.13e-02$	$-1.85e-02$	$7.24e-01$	$6.89e-01$	$7.67e-02$
$Tria1D2$	$2.56e-03$	$1.51e-02$	$1.43e-04$	$-5.90e-02$	$3.64e-01$	$9.30e-01$	$6.28e-01$
$Tria1D3$	$7.62e-03$	$9.50e-03$	$7.13e-02$	$-1.86e-02$	$7.24e-01$	$6.90e-01$	$7.67e-02$
$Tria1D4$	$7.60e-03$	$9.51e-03$	$7.13e-02$	$-1.84e-02$	$7.24e-01$	$6.89e-01$	$7.67e-02$
$Tria1D5$	$7.59e-03$	$9.48e-03$	$7.13e-02$	$-1.86e-02$	$7.24e-01$	$6.89e-01$	$7.67e-02$
$Tria2D1$	$2.05e-02$	$2.05e-03$	$5.95e-02$	$-3.05e-02$	$9.99e-01$	$-4.48e-02$	$1.28e-01$
$Tria2D2$	$2.05e-02$	$2.05e-03$	$5.95e-02$	$-3.05e-02$	$9.99e-01$	$-4.48e-02$	$1.28e-01$
$Tria2D3$	$2.05e-02$	$2.03e-03$	$5.95e-02$	$-3.07e-02$	$9.99e-01$	$-4.52e-02$	$1.28e-01$
$Tria2D4$	$2.05e-02$	$2.05e-03$	$5.95e-02$	$-3.06e-02$	$9.99e-01$	$-4.47e-02$	$1.28e-01$
$Tria2D5$	$2.05e-02$	$2.03e-03$	$5.95e-02$	$-3.06e-02$	$9.99e-01$	$-4.51e-02$	$1.28e-01$
$Tria3D1$	$8.00e-02$	$2.63e-02$	$-2.84e-02$	$-5.45e-01$	$2.11e-02$	$8.38e-01$	$1.50e-01$
$Tria3D2$	$7.80e-02$	$1.91e-02$	$1.18e-02$	$6.15e-01$	$5.83e-01$	$5.31e-01$	$5.56e-02$
$Tria3D3$	$7.81e-02$	$1.90e-02$	$1.18e-02$	$6.15e-01$	$5.83e-01$	$5.31e-01$	$5.56e-02$
$Tria3D4$	$7.81e-02$	$1.90e-02$	$1.19e-02$	$6.16e-01$	$5.83e-01$	$5.30e-01$	$5.56e-02$
$Tria3D5$	$7.80e-02$	$1.91e-02$	$1.18e-02$	$6.15e-01$	$5.83e-01$	$5.31e-01$	$5.56e-02$
$Tria4D1$	$1.28e-02$	$3.53e-02$	$6.78e-02$	$5.32e-01$	$5.84e-01$	$6.14e-01$	$9.88e-02$
$Tria4D2$	$1.28e-02$	$3.53e-02$	$6.79e-02$	$5.31e-01$	$5.83e-01$	$6.14e-01$	$9.88e-02$
$Tria4D3$	$1.28e-02$	$3.53e-02$	$6.78e-02$	$5.32e-01$	$5.84e-01$	$6.14e-01$	$9.88e-02$
$Tria4D4$	$1.28e-02$	$3.53e-02$	$6.79e-02$	$5.32e-01$	$5.84e-01$	$6.14e-01$	$9.88e-02$
$Tria4D5$	$1.28e-02$	$3.53e-02$	$6.78e-02$	$5.31e-01$	$5.84e-01$	$6.14e-01$	$9.88e-02$
$Tria5D1$	$8.87e-03$	$2.86e-02$	$1.75e-02$	$1.00e+00$	$7.73e-03$	$-1.38e-02$	$7.10e-02$
$Tria5D2$	$8.90e-03$	$2.86e-02$	$1.74e-02$	$1.00e+00$	$7.64e-03$	$-1.35e-02$	$7.10e-02$
$Tria5D3$	$8.91e-03$	$2.85e-02$	$1.75e-02$	$1.00e+00$	$7.92e-03$	$-1.36e-02$	$7.10e-02$
$Tria5D4$	$8.90e-03$	$2.86e-02$	$1.74e-02$	$1.00e+00$	$7.55e-03$	$-1.34e-02$	$7.10e-02$
$Tria5D5$	$8.92e-03$	$2.86e-02$	$1.74e-02$	$1.00e+00$	$7.66e-03$	$-1.32e-02$	$7.10e-02$
$Tria6D1$	$4.08e-02$	$8.09e-02$	$-1.52e-02$	$4.60e-01$	$8.11e-02$	$8.84e-01$	$3.34e-01$
$Tria6D2$	$4.16e-02$	$3.58e-02$	$3.48e-02$	$6.93e-01$	$7.20e-01$	$-2.01e-02$	$9.23e-02$
$Tria6D3$	$4.16e-02$	$3.57e-02$	$3.48e-02$	$6.93e-01$	$7.20e-01$	$-1.98e-02$	$9.23e-02$
$Tria6D4$	$5.37e-03$	$6.32e-02$	$5.94e-02$	$5.68e-01$	$8.23e-01$	$2.11e-03$	$3.51e-01$
$Tria6D5$	$-9.01e-04$	$6.87e-02$	$5.87e-02$	$3.87e-01$	$8.89e-01$	$2.46e-01$	$3.06e-01$
$Tria7D1$	$-2.19e-03$	$5.94e-02$	$3.50e-02$	$6.89e-01$	$7.68e-04$	$7.25e-01$	$1.58e-01$
$Tria7D2$	$-1.03e-03$	$9.09e-02$	$2.45e-03$	$6.14e-01$	$-5.40e-01$	$5.76e-01$	$2.24e-01$
$Tria7D3$	$-2.17e-03$	$5.95e-02$	$3.50e-02$	$6.89e-01$	$2.26e-04$	$7.25e-01$	$1.58e-01$
$Tria7D4$	$-2.19e-03$	$5.95e-02$	$3.50e-02$	$6.89e-01$	$1.01e-04$	$7.25e-01$	$1.58e-01$
$Tria7D5$	$-2.19e-03$	$5.95e-02$	$3.50e-02$	$6.89e-01$	$3.12e-04$	$7.25e-01$	$1.58e-01$
$Tria8D1$	$4.33e-02$	$7.98e-02$	$-1.58e-02$	$9.15e-02$	$2.12e-01$	$9.73e-01$	$3.37e-01$
$Tria8D2$	$4.63e-02$	$5.08e-02$	$3.50e-02$	$-1.32e-02$	$7.21e-01$	$6.93e-01$	$4.11e-02$
$Tria8D3$	$4.63e-02$	$5.08e-02$	$3.50e-02$	$-1.33e-02$	$7.21e-01$	$6.93e-01$	$4.11e-02$
$Tria8D4$	$7.03e-03$	$9.09e-02$	$1.32e-02$	$5.55e-01$	$-2.65e-01$	$7.88e-01$	$2.55e-01$
$Tria8D5$	$6.48e-02$	$2.29e-02$	$6.02e-02$	$7.69e-03$	$9.15e-01$	$4.03e-01$	$2.94e-01$
$Tria9D1$	$9.54e-03$	$5.58e-02$	$3.10e-02$	$6.54e-01$	$7.52e-01$	$-8.26e-02$	$1.80e-01$
$Tria9D2$	$9.61e-03$	$5.58e-02$	$3.09e-02$	$6.54e-01$	$7.52e-01$	$-8.14e-02$	$1.80e-01$
$Tria9D3$	$9.55e-03$	$5.58e-02$	$3.10e-02$	$6.54e-01$	$7.52e-01$	$-8.26e-02$	$1.80e-01$
$Tria9D4$	$9.55e-03$	$5.58e-02$	$3.10e-02$	$6.54e-01$	$7.52e-01$	$-8.26e-02$	$1.80e-01$
$Tria9D5$	$9.58e-03$	$5.58e-02$	$3.09e-02$	$6.54e-01$	$7.52e-01$	$-8.25e-02$	$1.80e-01$
$Tria10D1$	$5.21e-03$	$1.50e-02$	$3.05e-03$	$7.01e-01$	$2.35e-02$	$7.12e-01$	$1.46e-01$
$Tria10D2$	$5.17e-03$	$1.50e-02$	$3.06e-03$	$7.02e-01$	$2.39e-02$	$7.12e-01$	$1.46e-01$
$Tria10D3$	$5.22e-03$	$1.50e-02$	$3.06e-03$	$7.01e-01$	$2.35e-02$	$7.12e-01$	$1.46e-01$
$Tria10D4$	$5.20e-03$	$1.50e-02$	$3.05e-03$	$7.01e-01$	$2.36e-02$	$7.12e-01$	$1.46e-01$
$Tria10D5$	$5.19e-03$	$1.49e-02$	$3.01e-03$	$7.01e-01$	$2.39e-02$	$7.12e-01$	$1.46e-01$

As it could be noted, the error due to the electrode displacement sits at much higher values as compare to the case when the skull conductivity is changed. This infer a great importance of electrode placement in source localisation via inverse methods.

Table 6: DOE for electrode misplacement

	$D1$	$D2$	$D3$	$D4$	$D5$
height$Trial1$	$1.78e+00$	$2.39e+01$	$1.75e+00$	$1.75e+00$	$1.77e+00$
$Trial2$	$3.11e+00$	$3.11e+00$	$3.13e+00$	$3.10e+00$	$3.12e+00$
$Trial3$	$7.95e+01$	$3.43e+00$	$3.47e+00$	$3.53e+00$	$3.45e+00$
$Trial4$	$3.37e+00$	$3.39e+00$	$3.35e+00$	$3.36e+00$	$3.38e+00$
$Trial5$	$9.09e-01$	$8.88e-01$	$9.03e-01$	$8.83e-01$	$8.77e-01$
$Trial6$	$6.75e+01$	$1.59e+00$	$1.59e+00$	$1.04e+01$	$2.56e+01$
$Trial7$	$1.45e+00$	$3.27e+01$	$1.48e+00$	$1.47e+00$	$1.45e+00$
$Trial8$	$3.31e+01$	$1.34e+00$	$1.34e+00$	$6.83e+01$	$2.13e+01$
$Trial9$	$6.18e+00$	$6.12e+00$	$6.17e+00$	$6.17e+00$	$6.17e+00$
$Trial10$	$1.42e+00$	$1.43e+00$	$1.42e+00$	$1.42e+00$	$1.44e+00$

Table 7: DLE for electrode misplacement

	$D1$	$D2$	$D3$	$D4$	$D5$
height$Trial1$	$1.27e-03$	$7.11e-02$	$1.30e-03$	$1.29e-03$	$1.29e-03$
$Trial2$	$3.68e-03$	$3.69e-03$	$3.71e-03$	$3.68e-03$	$3.66e-03$
$Trial3$	$3.90e-02$	$2.85e-03$	$2.89e-03$	$2.96e-03$	$2.87e-03$
$Trial4$	$6.26e-03$	$6.28e-03$	$6.23e-03$	$6.25e-03$	$6.26e-03$
$Trial5$	$2.17e-03$	$2.15e-03$	$2.20e-03$	$2.16e-03$	$2.15e-03$
$Trial6$	$6.49e-02$	$3.12e-03$	$3.10e-03$	$5.09e-02$	$5.78e-02$
$Trial7$	$5.24e-03$	$4.61e-02$	$5.21e-03$	$5.22e-03$	$5.21e-03$
$Trial8$	$5.59e-02$	$3.44e-03$	$3.45e-03$	$5.71e-02$	$4.51e-02$
$Trial9$	$3.78e-03$	$3.71e-03$	$3.77e-03$	$3.76e-03$	$3.74e-03$
$Trial10$	$4.56e-03$	$4.54e-03$	$4.58e-03$	$4.55e-03$	$4.54e-03$

A Forward problem appendix

A.1 Head models

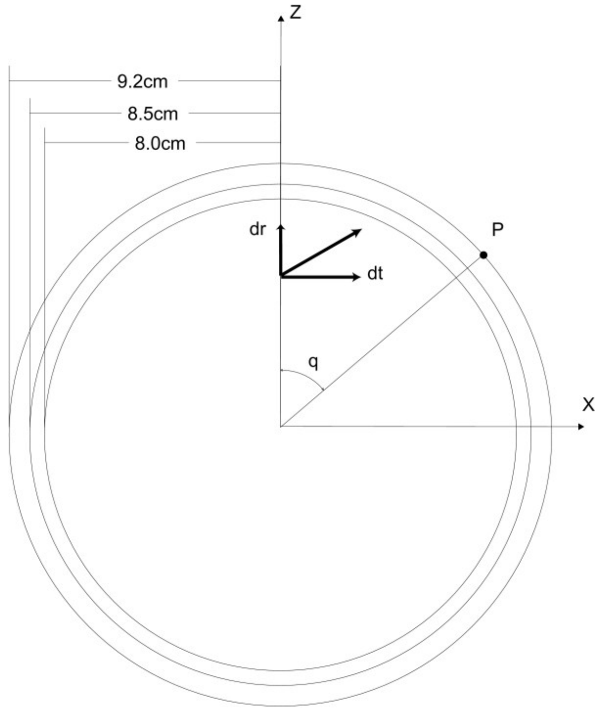


Figure A.1.1: Spherical head model

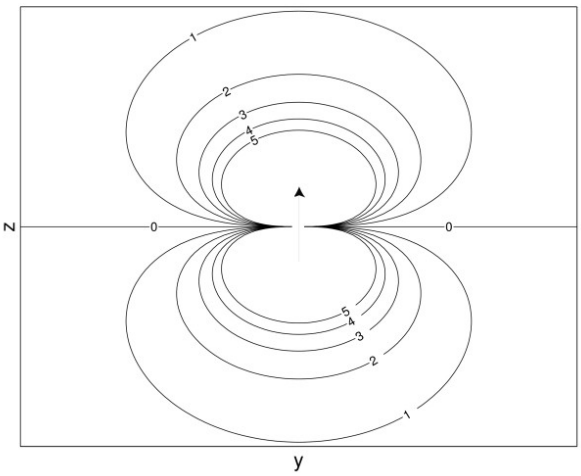


Figure A.1.2: Dipole electrical field

1.png

Figure A.1.3: Dipole modeling

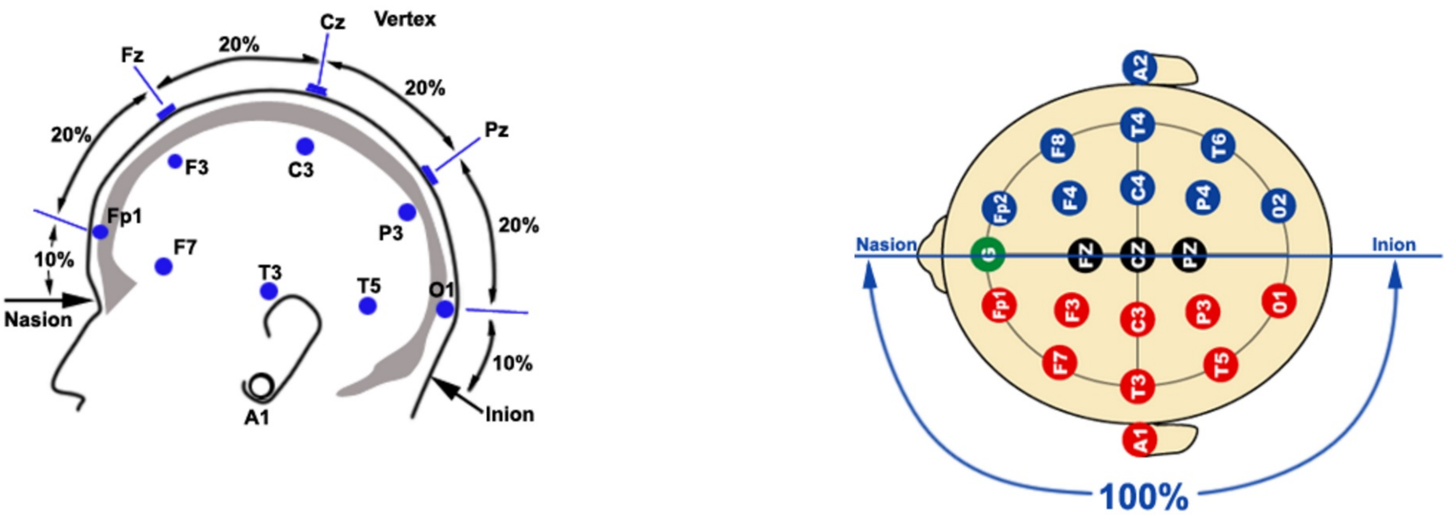


Figure A.1.4: EEG montage

A.2 The Potential of a General Dipole

Consider a dipole with a dipole moment $\vec{D} = (D_x, D_y, D_z)$ located at $\vec{R_1} = (X_1, Y_1, Z_1)$, and a point on the scalp $\vec{R_2} = (X_2, Y_2, Z_2)$. The radius vectors $\vec{R_1}$ and $\vec{R_2}$ define a plane in space, which will be called the major plane. The cross product $\vec{N} = \vec{R_1} \times \vec{R_2}$ is perpendicular to the major plane. The cross product $\vec{t} = \vec{N} \times \vec{R_1}$ is in the major plane, and perpendicular to $\vec{R_1}$. Based on an identity from algebra, $\vec{t} = \vec{R_2}(\vec{R_1} \cdot \vec{R_1}) - \vec{R_1}(\vec{R_1} \cdot \vec{R_2})$, or by components

$$t_x = X_2 \cdot p - X_1 \cdot q \quad (A.2.1) \qquad t_y = Y_2 \cdot p - Y_1 \cdot q \quad (A.2.2) \qquad t_z = Z_2 \cdot p - Z_1 \cdot q \quad (A.2.3)$$

where $p = {X_1}^2 + {Y_1}^2 + {Z_1}^2$ and $q = X_1 \cdot X_2 + Y_1 \cdot Y_2 + Z_1 \cdot Z_2$. $\vec{T} = (T_x, T_y, T_z)$ is the unit vector in the $\vec{t}$ direction, which is the tangential direction to $\vec{R_1}$ in the major plane. Its components are

$$T_x = t_x/|t| \quad (A.2.4) \qquad T_y = t_y/|t| \quad (A.2.5) \qquad T_z = t_z/|t| \quad (A.2.6)$$

where $|t| = ({t_x}^2 + {t_y}^2 + {t_z}^2)^{1/2}$. The unit vector $\overrightarrow{RR} = (R_x, R_y, R_z)$ in the radial direction $\vec{R_1}$ is

$$R_x = X_1/|\vec{R_1}| \quad (A.2.7) \qquad R_y = Y_1/|\vec{R_1}| \quad (A.2.8) \qquad R_z = Z_1/|\vec{R_1}| \quad (A.2.9)$$

where $|\vec{R_1}| = ({X_1}^2 + {Y_1}^2 + {Z_1}^2)^{1/2}$. The cosine of the angle between $\vec{R_1}$ and $\vec{R_2}$ is

$$cos\alpha = q/|\vec{R_1}| \cdot |\vec{R_2}| \tag{A.2.10}$$

where $|\vec{R_2}| = ({X_2}^2 + {Y_2}^2 + {Z_2}^2)^{1/2}$. The potential that the dipole $\vec{D}$ generates at the scalp point $\vec{R_2}$ is the sum of the potentials contributed by D_x, D_y and D_z. When these expressions are substituted in equation **??** the potential at $\vec{R_2}$ due to D_x turns to be

$$V_x = \left\{ \frac{1}{4\pi S R^2} \sum_{i=1}^{\infty} \frac{X(2n+1)^3}{d_n(n+1)n} .b^{n-1} \{ n R_x P_n(cos\alpha) + T_x {P_n}^1(cos\alpha) \} \right\} D_x \tag{A.2.11}$$

and similarly, the potential due to D_y and D_z would be

$$V_y = \left\{ \frac{1}{4\pi S R^2} \sum_{i=1}^{\infty} \frac{X(2n+1)^3}{d_n(n+1)n} .b^{n-1} \{ n R_y P_n(cos\alpha) + T_y {P_n}^1(cos\alpha) \} \right\} D_y \tag{A.2.12}$$

$$V_z = \left\{ \frac{1}{4\pi S R^2} \sum_{i=1}^{\infty} \frac{X(2n+1)^3}{d_n(n+1)n} .b^{n-1} \{ n R_z P_n(cos\alpha) + T_z {P_n}^1(cos\alpha) \} \right\} D_z \tag{A.2.13}$$

where $b = |\vec{R_1}|/R$¡ and $cos\alpha$ is given by equation **??**. The potential V at $\vec{R_2}$ due to $\vec{D}$ at $\vec{R_1}$ is $V = V_x + V_y + V_z$. Assigning a Reference Electrode

1) An arbitrary convenient point on the surface, denoted by $\vec{R_0}$, is declared as the reference. All potentials are measured with respect to the electrode at this point. (If the potentials have already been measured with respect to a different point, such as left earlobe, they have to be adjusted accordingly. The measured potential at $\vec{R_0}$ (with respect to the earlobe) should be subtracted from the measured potential at each point).

2) Equations **??** are corrected for the new reference point, and become

$$V_x' = (T_{x12} - T_{x10}).D_x \quad (A.2.14) \qquad V_y' = (T_{y12} - T_{y10}).D_y \quad (A.2.15) \qquad V_z' = (T_{z12} - T_{z10}).D_z \quad (A.2.16)$$

where T_{xl0}, T_{yl0}, T_{zl0} are the transfer coefficients from a dipole at $\vec{R_1}$ onto surface point $\vec{R_0}$. V', the theoretical potential at $\vec{R_2}$ with respect to $\vec{R_0}$ due to a dipole at $\vec{R_1}$ is given by

$$V' = V_X' + V_Y' + V_Z'. \tag{A.2.17}$$

B Inverse problem appendix

B.1 Formulas

$$DLE = \sqrt{(x_{orig} - x_{est})^2 - (y_{orig} - y_{est})^2 - (z_{orig} - z_{est})^2} \tag{B.1.1}$$

$$DOE = arccos\left\{ \frac{Or_{xorig} * x_{est} + Or_{yorig} * y_{est} + Or_{zorig} * Or_{zest}}{\sqrt{Or_{x_{est}}^2 - Or_{y_{est}}^2 - Or_{z_{est}}^2} * \sqrt{Or_{x_{orig}}^2 - Or_{y_{orig}}^2 - Or_{z_{orig}}^2}} \right\} \tag{B.1.2}$$

Spherical coordinate for the electrode displacement:

$$Angle = \frac{360 * Displacement}{2 * \pi * Radius} \tag{B.1.3}$$

B.2 Skull conductivity measurement

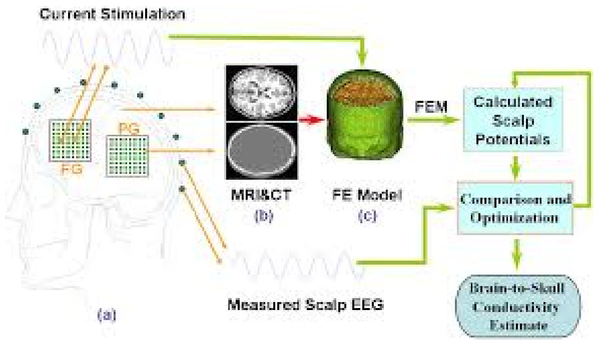
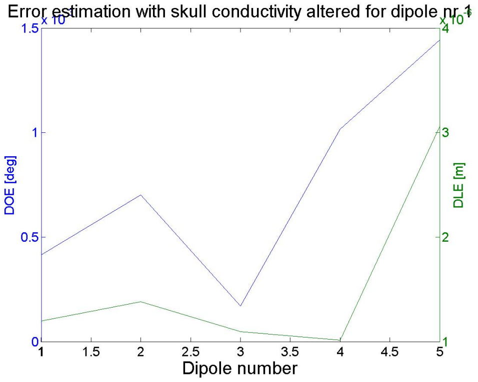
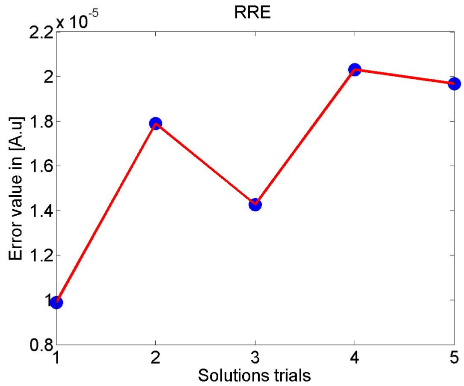
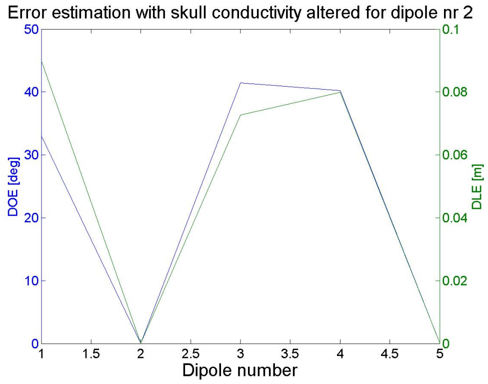


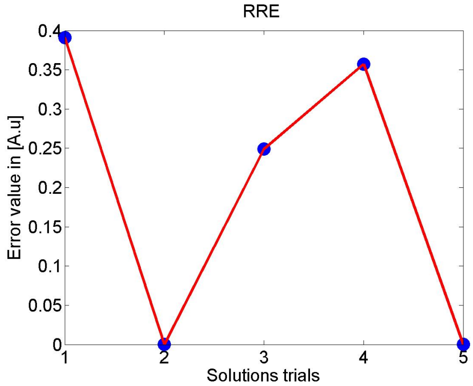
Figure B.2.1: Skull conductivity measurement[?]

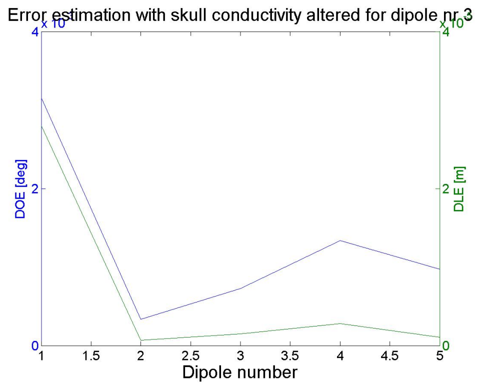
B.3 Skull conductivity error plots

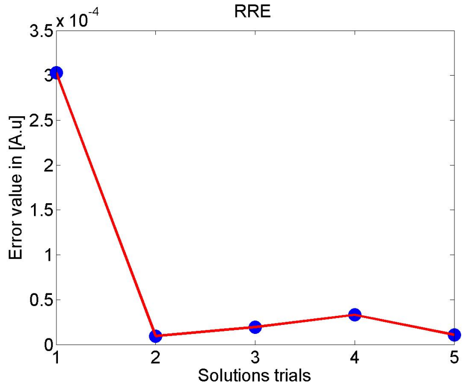

Error estimation with skull conductivity altered for dipole nr 1
DOE [deg]
DLE [m]
Dipole number

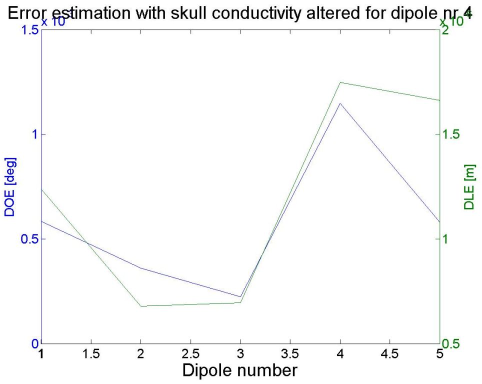

RRE
Error value in [A.u]
Solutions trials

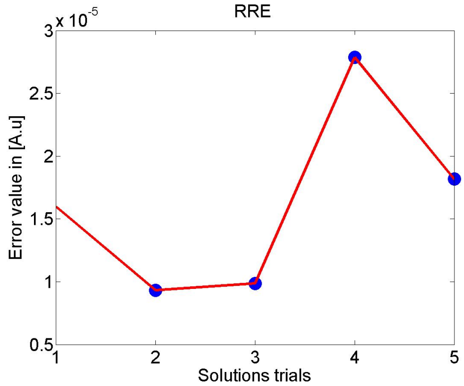

Error estimation with skull conductivity altered for dipole nr 2
DOE [deg]
DLE [m]
Dipole number

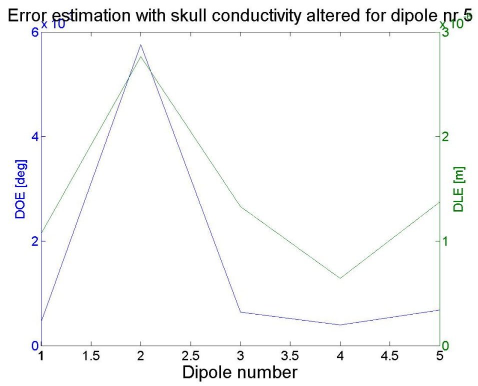

RRE
Error value in [A.u]
Solutions trials

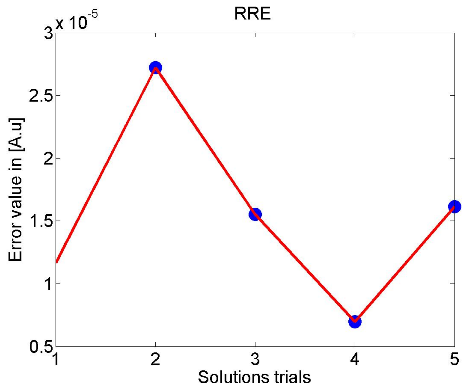

Error estimation with skull conductivity altered for dipole nr 3
DOE [deg]
DLE [m]
Dipole number

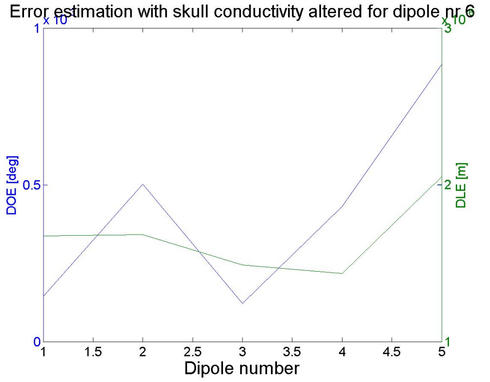

RRE
Error value in [A.u]
Solutions trials

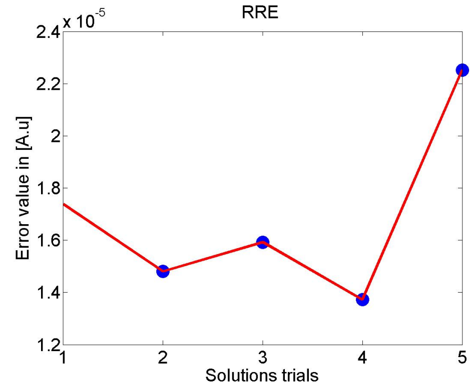

Error estimation with skull conductivity altered for dipole nr 4
DOE [deg]
DLE [m]
Dipole number

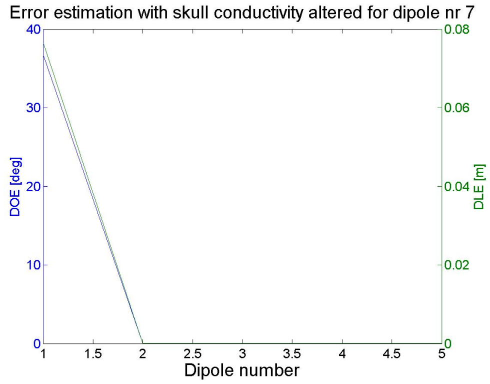

RRE
Error value in [A.u]
Solutions trials

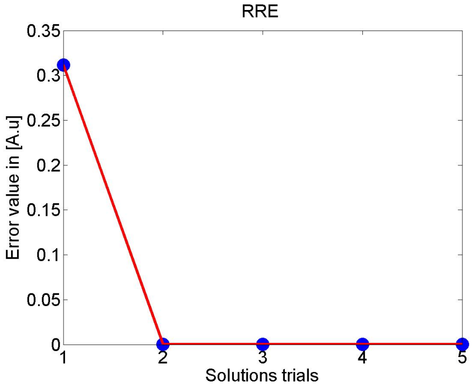

Error estimation with skull conductivity altered for dipole nr 5
DOE [deg]
DLE [m]
Dipole number

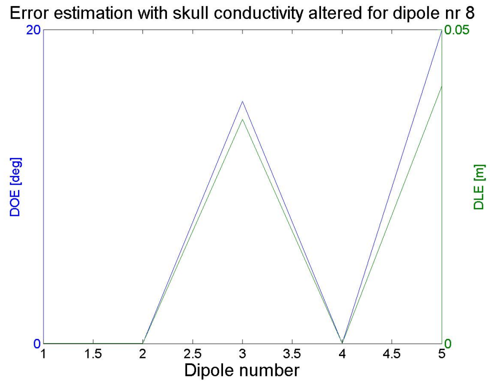

RRE
Error value in [A.u]
Solutions trials

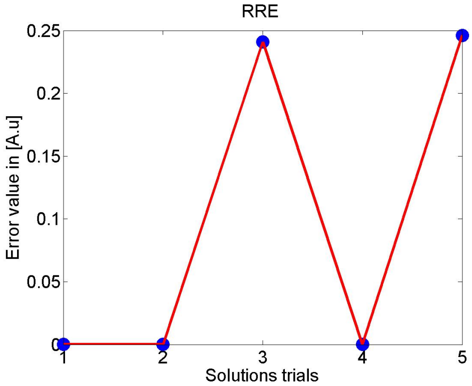

Error estimation with skull conductivity altered for dipole nr 6
DOE [deg]
DLE [m]
Dipole number

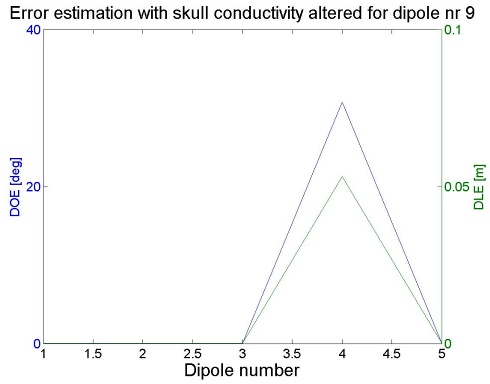

RRE
x 10^-5
Error value in [A.u]
Solutions trials

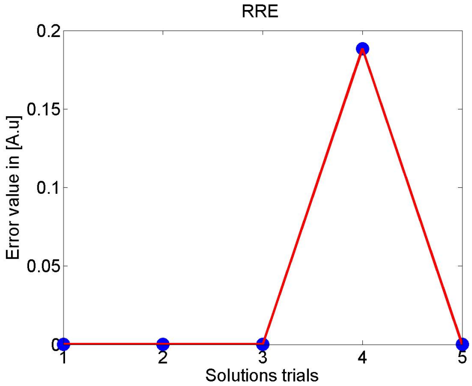

Error estimation with skull conductivity altered for dipole nr 7
DOE [deg]
DLE [m]
Dipole number

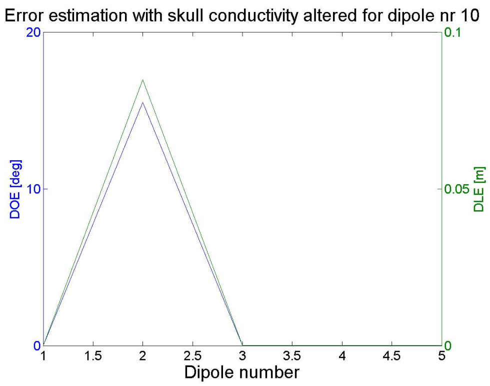

RRE
Error value in [A.u]
Solutions trials

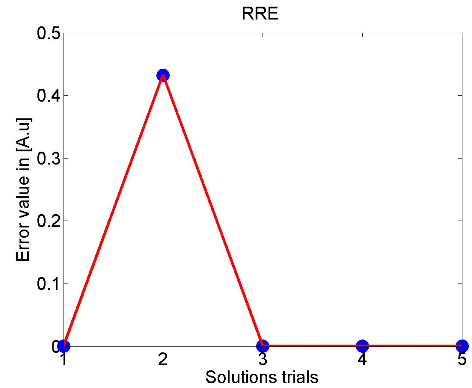

Error estimation with skull conductivity altered for dipole nr 8
DOE [deg]
DLE [m]
Dipole number


RRE
Error value in [A.u]
Solutions trials

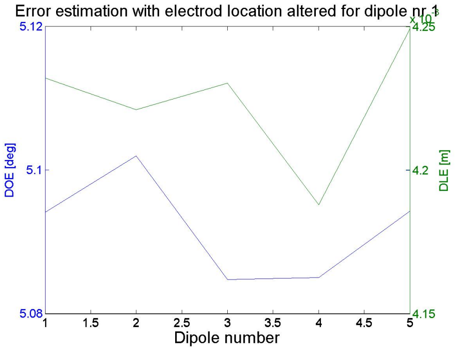

Error estimation with skull conductivity altered for dipole nr 9
DOE [deg]
DLE [m]
Dipole number

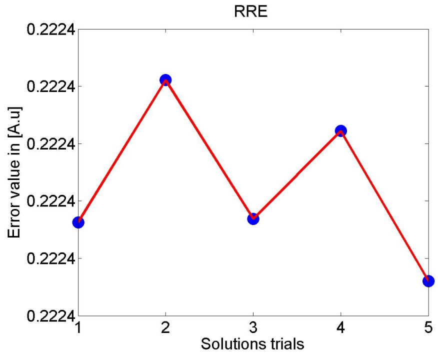

RRE
Error value in [A.u]
Solutions trials

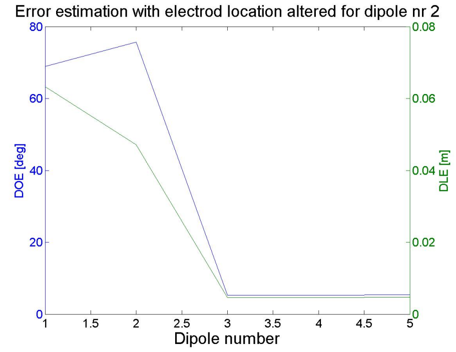

Error estimation with skull conductivity altered for dipole nr 10
DOE [deg]
DLE [m]
Dipole number

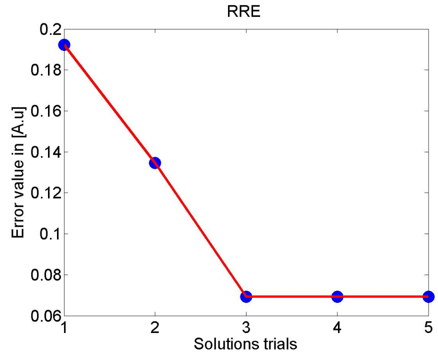

RRE
Error value in [A.u]
Solutions trials

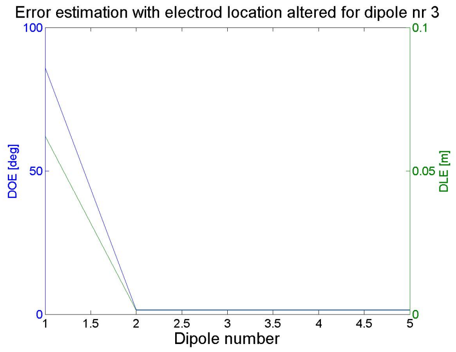
B.4 Electrode misplacement error plots

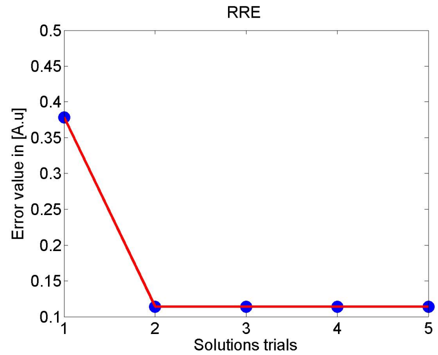


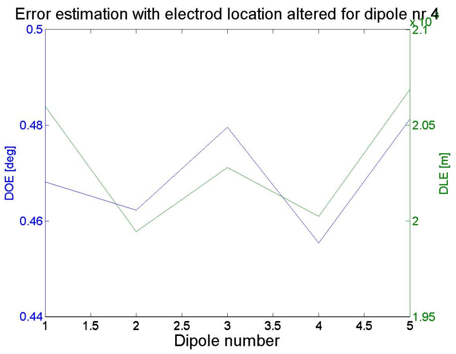


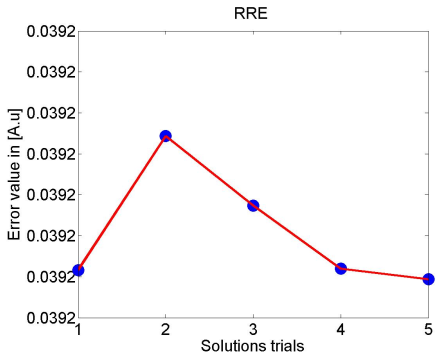


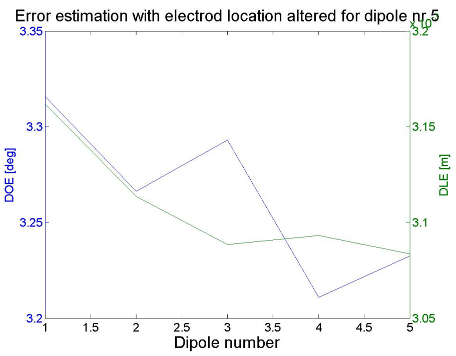


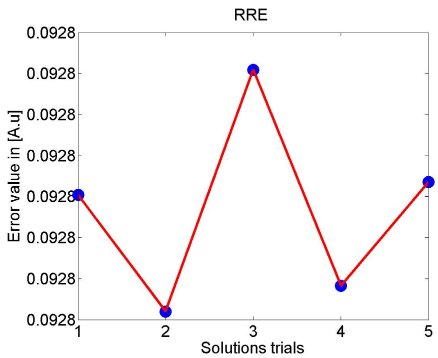


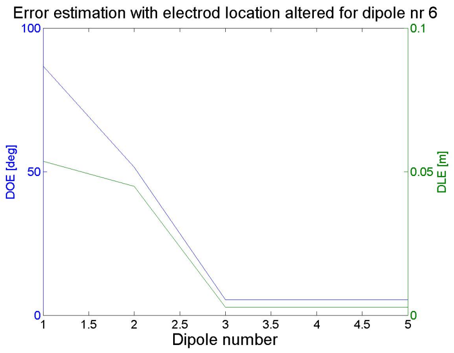


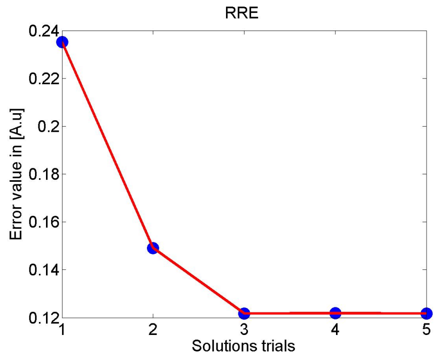


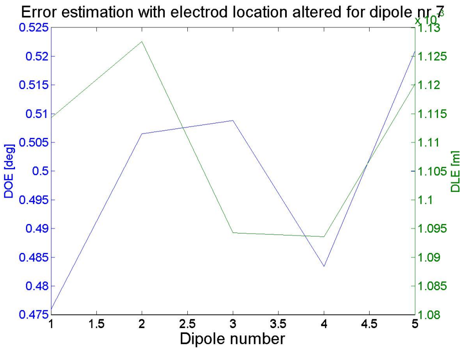


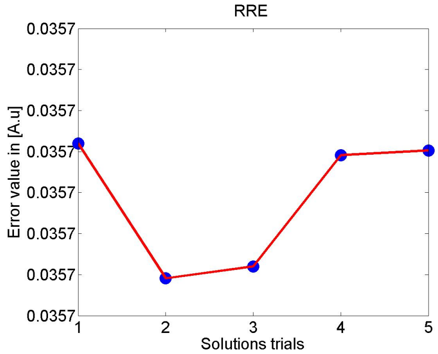


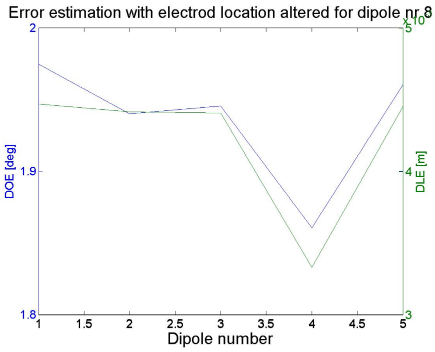


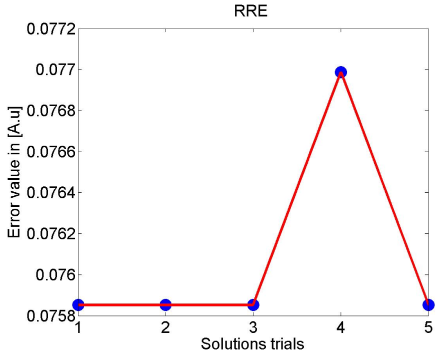

Error estimation with electrod location altered for dipole nr 6
DOE [deg]
DLE [m]
Dipole number

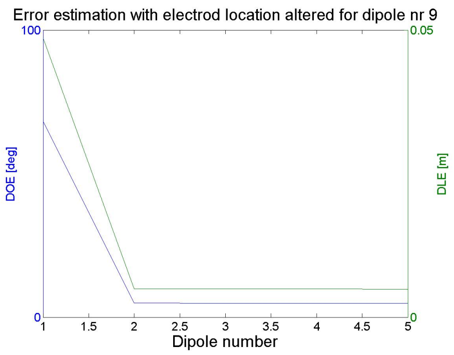

RRE
Error value in [A.u]
Solutions trials

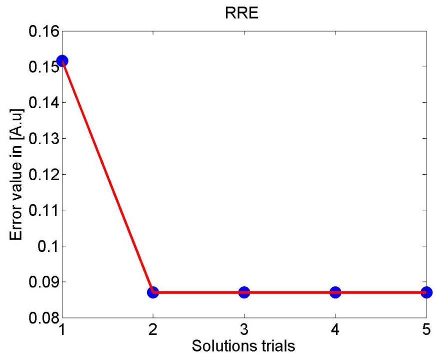

Error estimation with electrod location altered for dipole nr 7
DOE [deg]
DLE [m]
Dipole number

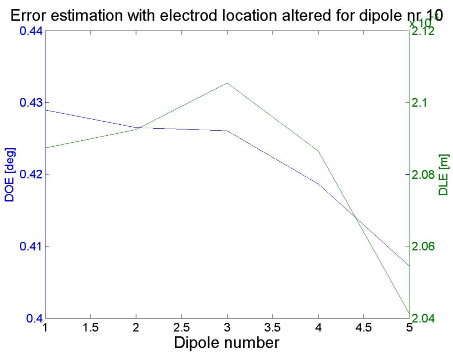

RRE
Error value in [A.u]
Solutions trials

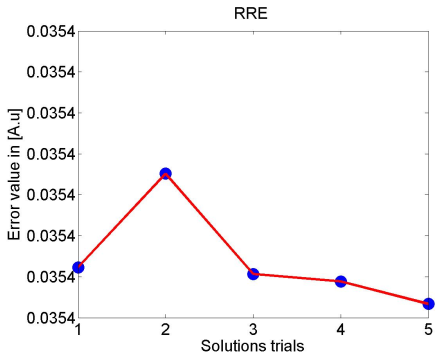

Error estimation with electrod location altered for dipole nr 8
DOE [deg]
DLE [m]
Dipole number


RRE
Error value in [A.u]
Solutions trials


Error estimation with electrod location altered for dipole nr 9
DOE [deg]
DLE [m]
Dipole number


RRE
Error value in [A.u]
Solutions trials


Error estimation with electrod location altered for dipole nr 10
DOE [deg]
DLE [m]
Dipole number


RRE
Error value in [A.u]
Solutions trials

C Main code session

C.1 First part

```

clear all
close all
clc
%%
% 1
load headmodel19                  % Load the head model
%%
index=1;
% 2 Show electrode position
Electrode_Position_V(hm,3)% Plot the location of the electrodes
[index]=Save_Image(index);
% index=2;

%%
% 3
Show_Dipole_Placement([0 0 0 0 0 1],hm,4)    % Show the dipole at the center orientation towards z
[index]=Save_Image(index);
Show_Dipole_Placement([.05 0 0 0 0 1],hm,5)  % Show the dipole at the middle of center towards the right ear orientation towards z
[index]=Save_Image(index);
%%
% 4 dipole at the center orinted along x axis
source.loc=[0 0 0.05];               % Input a source at the center
source.ori=[1 0 0];                  % Input the orientation towards x axis
[V1,L]=Spherical_Head_Model(hm,source);% Comput the voltage at the electrodes from this dipole with the forward solution
Potential_Distribution(V1,hm,6,1)           % Show the potential distribution in the electrodes
title('Electrod potential from dipole at the center oriented along x axis')
[index]=Save_Image(index);
Plot_Electrode_Potential(V1,9)
[index]=Save_Image(index);
Show_Dipole_Placement([0 0 0 1 0 0],hm,8)    % The dipole location and its orientation has been plot
[index]=Save_Image(index);
%%
% 5 study the composition of L
Potential_Distribution(L(:,1),hm,17,1)
title('First column')
[index]=Save_Image(index);
Potential_Distribution(L(:,2),hm,18,1)
title('Second column')
[index]=Save_Image(index);
Potential_Distribution(L(:,3),hm,19,1)
title('Third column')
[index]=Save_Image(index);
%%
% % 6 two potential at the center, where one along the x axis the other along the z axis
source1.loc=[0 0 0];                  % Input a source at the center
source1.ori=[0 0 1];                  % Input the orientation towards z axis

source2.loc=[0 0 0];                  % Input a source at the center
source2.ori=[1 0 0];                  % Input the orientation towards x axis

source3.loc=[0 0 0];                  % Input a source at the center
source3.ori=[1 0 1];                  % Input the orientation towards x axis
%
[V1,L1]=Spherical_Head_Model(hm,source1);  % Comput the voltage at the electrodes from this dipole with the forward solution
[V2,L2]=Spherical_Head_Model(hm,source2);  % Comput the voltage at the electrodes from this dipole with the forward solution
[V3,L2]=Spherical_Head_Model(hm,source3);  % Comput the voltage at the electrodes from this dipole with the forward solution
VT=V1+V2;                             % Superimpose the voltages since the Maxwel equation is a linear relatoin
Potential_Distribution(VT,hm,20,1)           % Show the potential distribution in the electrodes
[index]=Save_Image(index);
Potential_Distribution(V1,hm,20,1)           % Show the potential distribution in the electrodes
[index]=Save_Image(index);
Potential_Distribution(V2,hm,20,1)           % Show the potential distribution in the electrodes
[index]=Save_Image(index);
Potential_Distribution(V3,hm,20,1)           % Show the potential distribution in the electrodes
[index]=Save_Image(index);
Show_Dipole_Placement([0 0 0 0 0 1],hm,8)    % The dipole location and its orientation has been plot
[index]=Save_Image(index);
Show_Dipole_Placement([0 0 0 1 0 0],hm,8)    % The dipole location and its orientation has been plot
```

```
[index]=Save_Image(index);
Show_Dipole_Placement([0 0 0 1 0 1],hm,8)    % The dipole location and its orientation has been plot
[index]=Save_Image(index);
%%
% 7 Simulate EEG
sourceEEG.loc=[0 -0.05 0.02];       % Input a source at the the indicated location
sourceEEG.ori=[0 0 1];              % Input the orientation towards z axis
f=200;                              % Sampling frequency
RotateFreq=10;                      % Rotation frequency in time domain
Duration=3;                         % Duration of the EEG signal
[EEG]=Simulate_EEG(sourceEEG,hm,f,RotateFreq,Duration);
%%
[dim1,dim2]=size(EEG);
t=linspace(0,3,dim1);
figure('units','normalized','outerposition',[0 0 1 1])
Vangjush_PlotEEG(EEG',t)
title('EEG signals at all the electrodes','FontSize',30)
xlabel('Time of measurement in sec','FontSize',30)
ylabel('Voltage at each electrode repsectively in uV','FontSize',30)
set(gca,'FontSize',30);
[index]=Save_Image(index);
Show_Dipole_Placement([0 -0.05 0.02 0 0 1],hm,8)    % The dipole location and its orientation has been plot
[index]=Save_Image(index);
%%
%  Potential_Movie(hm,EEG,f)
```

C.2 Second part

```

clear all
close all
clc
%%
load headmodel19                  % Load the head model
%% 1
sourceEEG1.loc=[0.06 0 0.01];                  % Input a source at the the indicated location
sourceEEG1.ori=[1 0 0];                        % Input the orientation towards x axis
[V_EEG1,L_EEG1]=Spherical_Head_Model(hm,sourceEEG1); % Compute the voltage done by this potential
[Result]=Solve_Inverse_Problem(V_EEG1,hm);% Solve the inverse proble meaning the computation of the dipole parameters from the computed dipole
index=100;
Plot_EE(Result,sourceEEG1,index)
 [index]=Save_Image(index);
PlotRRE(Result(:,7),index)                                  % Plot the set of the RRE values
 [index]=Save_Image(index);
Potential_Distribution(V_EEG1,hm,index,1)                    % Plot the potential at each electrode
[index]=Save_Image(index);

%% 3
sourceEEG2.ori=[1 0 0];                              % Input the orientation towards x axis
sourceEEG2.loc=[0.06 0 0.01];                        % Input a source at the the indicated location

[VEEG2,LEEG2]=Spherical_Head_Model(hm,sourceEEG2);          % Compute the voltage done by this potential
Potential_Distribution(VEEG2,hm,index,1)                           % Show the computed voltages at each electrode
 [index]=Save_Image(index);
VEEG2Error=VEEG2;
VEEG2Error(13)=0;                                    % The P8 electrode is shut down to see the impact
Potential_Distribution(VEEG2Error,hm,index,1)                      % Show the computed voltages at each electrode and the impact of the down electrode
 [index]=Save_Image(index);
VEEG2Error1=VEEG2;
VEEG2Error1(3)=0;                                    % The third electrode is shut down to see the impact
Potential_Distribution(VEEG2Error1,hm,5,index)                     % Show the computed voltages at each electrode and the impact of the down electrode
[index]=Save_Image(index);

result_point_error=Solve_Inverse_Problem(VEEG2,hm);                    % The inverse proble is computed in the case when all electrods are working
result_point_error1=Solve_Inverse_Problem(VEEG2Error,hm);          % The inverse proble is computed in the case when the p8 electrode is down
result_point_error2=Solve_Inverse_Problem(VEEG2Error1,hm);        % The inverse proble is computed in the case when the third electrode is down
%%
PlotRRE([result_point_error(:,7),result_point_error1(:,7),result_point_error2(:,7)],index)
legend('RRE all elctrd','RRE of P8 elctrd','RRE of F7 elctrd','Location','northoutside','Orientation','horizontal')
title('RRE of the mode')
[index]=Save_Image(index);
```

```
%%
Plot_EE(result_point_error ,sourceEEG2,index)
title('Estimation all electrodes','FontSize',20)
[index]=Save_Image(index);
Plot_EE(result_point_error1 ,sourceEEG2,index)
title('Estimation error when P8 is off','FontSize',20)
[index]=Save_Image(index);
Plot_EE(result_point_error2 ,sourceEEG2,index)
title('Estimation error when F7 is off','FontSize',20)
[index]=Save_Image(index);
%%
index=110;
hm1=hm;
hm.condskull=(1/80)*hm.condsoft;                % The conductivity of the skull is altered to see the impact
Nr_Dipoles=10;                                  % The total number of the random dipoles around the head is computed
[Random_Dipoles]=Generate_Random_Dipole(hm,Nr_Dipoles);

for i=1:10
    sourceEEGS.loc(1:3)=Random_Dipoles(i,1:3);              % Location of the sources in inputed
    sourceEEGS.ori(1:3)=Random_Dipoles(i,(1:3)+3);            % Orientation of the sources in inputed
    [A,~]=Spherical_Head_Model(hm1,sourceEEGS);    % Voltage at each electrode for this specific dipole is computed
    Estimation1{i}=Solve_Inverse_Problem(A,hm);              % Dipole parameters for the computed volatage are computed
   [DOE(i,:),DLE(i,:)]=Plot_EE(Estimation1{i},sourceEEGS,index);
    a=sprintf('Error estimation with skull conductivity altered for dipole nr %d',i);
    title(a,'FontSize',20)
    [index]=Save_Image(index);
    close all
    Temp=Estimation1{i};
    PlotRRE(Temp(:,7),index)
    a=sprintf('RRE ');
    title(a);
    [index]=Save_Image(index);
end
%%
str1='Skull';
Vangjush_Parameter_2_Latex_Table(Estimation1,str1)

str1='Skull1';
Vangjush_Parameter_2_Latex_Table1(DOE,str1)

str1='Skull2';
Vangjush_Parameter_2_Latex_Table1(DLE,str1)
%%
clear all
index=200;
load headmodel19
hm1=hm;
Nr_Dipoles=10;                                  % The total number of the random dipoles around the head is computed
[Random_Dipoles]=Generate_Random_Dipole(hm,Nr_Dipoles);
offset=1e-3;
phi=offset*360/2/pi/hm.radius(3);
hm.elpos(1,2)=hm.elpos(1,2)+1;                  % The first electrode is displaced by 1 cm
for i=1:Nr_Dipoles
    sourceEEGS.loc(1:3)=Random_Dipoles(i,1:3);              % Location of the sources in inputed
    sourceEEGS.ori(1:3)=Random_Dipoles(i,(1:3)+3);            % Orientation of the sources in inputed
    [A,~]=Spherical_Head_Model(hm1,sourceEEGS);    % Voltage at each electrode for this specific dipole is computed
    Estimation1{i}=Solve_Inverse_Problem(A,hm);              % Dipole parameters for the computed volatage are computed
    [DOE(i,:),DLE(i,:)]=Plot_EE(Estimation1{i},sourceEEGS,index);
    a=sprintf('Error estimation with electrod location altered for dipole nr %d',i);
    title(a,'FontSize',20)
    [index]=Save_Image(index);
    close all
    Temp=Estimation1{i};
    PlotRRE(Temp(:,7),index)
    a=sprintf('RRE ');
    title(a);
    [index]=Save_Image(index);
end
str1='Electrode';

Vangjush_Parameter_2_Latex_Table(Estimation1,str1)

str1='Electrode1';
```

```
Vangjush_Parameter_2_Latex_Table1(DOE, str1)

str1='Electrode2';
Vangjush_Parameter_2_Latex_Table1(DLE, str1)
%% 2
% Simulate EEG
sourceEEG.loc=[0 -0.05 0.02];        % Input a source at the the indicated location
sourceEEG.ori=[0 0 1];               % Input the orientation towards z axis
f=200;                               % Sampling frequency
RotateFreq=10;                       % Rotation frequency in time domain
Duration=3;                          % Duration of the EEG signal
[EEG]=Simulate_EEG(sourceEEG,hm,f,RotateFreq,Duration);
%%
[dim1,dim2]=size(EEG);
t=linspace(0,3,dim1);
figure('units','normalized','outerposition',[0 0 1 1])
Vangjush_PlotEEG(EEG',t);
title('EEG signals at all the electrodes','FontSize',30)
xlabel('Time of measurement in sec','FontSize',30)
ylabel('Voltage at each electrode repsectively in uV','FontSize',30)
set(gca,'FontSize',30);
[x,y]=ginput(1);
V=EEG';
Temp=V(:,round(x*f));
%%
result_point_error=Solve_Inverse_Problem(Temp,hm);
str1='EEG';
Vangjush_Parameter_2_Latex_Table2(result_point_error,str1)
```

C.3 Aid functions

```
function Bounding(scale,r)
%%
Bounds=[-scale*r scale*r];
xlim(Bounds);
ylim(Bounds);
zlim(Bounds);
end
```

```
function [RRE,V_in]=CalculateRRE(Results,V_EEG,hm)
[dim1,~]=size(Results);              % Extract the dimensions of the results, number of dipoles dim1 and
    data length dim2
for i=1:dim1
    for j=1:3
        source.loc(j)=Results(i,j);      % Input the location of the source from the first 3 data values
        source.ori(j)=Results(i,j+3);    % Input the orientation of the source from the last 3 data values
    end
    [V, ~]=solve_forward(hm,source);     % Calculate the forward problem meaning the computation of the
    voltage
    V_in(i,:)=V;                         % Store this values
    RRE(i)=sum((V-V_EEG).^2)./sum(V.^2); % Compute the RRE value for respective computation
end
end
```

```
function [DLE]=Dipole_Location_Error(Loc1,Loc2)
%%
DLE=sqrt(sum((Loc1-Loc2).^2));
end
```

```
function [DOE]=Dipole_Orientation_Error(Loc1,Loc2)
%%
d1=sqrt(sum(Loc1.^2));
d2=sqrt(sum(Loc2.^2));
DOE=acosd(sum(Loc1.*Loc2)/d1/d2);
end
```

```
function c=Dipole_Position_Inverse(dp,hm,V)
%%
ndip=size(dp,1);
L=zeros(size(V,1),3*ndip);
for i=1:ndip
    source.loc=dp(i,:);
    [~,L]=Spherical_Head_Model(hm,source);
end
c=norm((eye(size(V,1))-L*pinv(L))*V,'fro')/norm(V,'fro');
return;
```

```
function Draw_Nouse(r,Font1)
%%
plot3(r*0,r*1.1,r*-0.2,'r.','MarkerSize',100)
text(r*0,r*1.41,r*-0.2,'Nouse','FontSize',Font1);
end
```

```
function DrawSpehere(Nr_Points,r)
%%
[X,Y,Z]=sphere(Nr_Points);
s1=surf(r*X,r*Y,r*Z);
set(s1,'FaceColor',[0.9 0.9 0.9],'EdgeColor','green');
set(s1,'FaceColor',[0 1 1],'EdgeColor','red');
set(s1, 'FaceAlpha', 0.1)
shading interp
light
end
```

```
function Electrod_Names(Head_Model,r,Font)
%%
Element_Position=Head_Model.elpos;
Electrode_Names=Head_Model.meas;
%%
Electr_Position_Cartesian=Spherical_To_Cartesian_Coordiante([Element_Position, r*ones(length(
    Element_Position),1)]');
Electr_Position_Cartesian_text=Spherical_To_Cartesian_Coordiante([Element_Position, r*(ones(length(
    Element_Position),1)+0.3)]');

for i=1:size(Electr_Position_Cartesian,2)
    text(Electr_Position_Cartesian_text(1,i)',Electr_Position_Cartesian_text(2,i)',
    Electr_Position_Cartesian_text(3,i)'+r*0.011,...
        [num2str(i) ' ' Electrode_Names.el.lbl{i}],'FontSize',Font);
end
end
```

```
function Electrode_Position_V(Head_Model,~)
%%
% figure(nr_fig)
figure('units','normalized','outerposition',[0 0 1 1])
r=Head_Model.radius(end);
Nr_Points=20;
Font1=20;
Font=30;
scale=1.5;
manner='filled';

DrawSpehere(Nr_Points,r)
hold on

Position_Of_Electrodes(Head_Model,r,manner);
Electrod_Names(Head_Model,r,Font)

Draw_Nouse(r,Font1);

Labeling(Font)
title('Electrode distribution in head','FontSize',Font);
Bounding(scale,r);
set(gca,'FontSize',Font);
end
```

```
function [DOE,DLE]=Estimation_Error(Result,sourceEEG1)
%%
for i=1:min(size(Result))
    DOE(i)=Dipole_Orientation_Error(Result(i,4:6),sourceEEG1.ori);
    DLE(i)=Dipole_Location_Error(Result(i,1:3),sourceEEG1.loc);
end
end
```

```
function [Random_Dipoles]=Generate_Random_Dipole(hm,Nr_Dipoles)
%%
Random_Dipoles=zeros(Nr_Dipoles,3);
center=hm.center;
h=waitbar(0,'Generating random dipoles inside head model');
for l=1:Nr_Dipoles
    i=hm.radius(1);
    j=hm.radius(1);
    k=hm.radius(1);
    while(sqrt(sum(([i-center(1),j-center(2),k-center(3)].^2)))>hm.radius(1))
        i=rand(1);
        j=rand(1);
```

```
        k=rand(1);
    end
    Random_Dipoles(l,1:3)=[i j k];
    Random_Dipoles(l,4:6)=[randi([0 1],1,1) randi([0 1],1,1) randi([0 1],1,1)];

    waitbar(l/Nr_Dipoles,h)
end
delete(h)
end
```

```
function [Random_Dipoles]=Generate_Random_Dipoles(hm,Nr_Dipoles)
%%
Random_Dipoles=zeros(Nr_Dipoles,3);
center=hm.center;
Random_Dipoles(1,1:3)=center;
h=waitbar(0,'Generating random dipoles inside head model');
for l=2:Nr_Dipoles
    i=hm.radius(1);
    j=hm.radius(1);
    k=hm.radius(1);
    while(sqrt(sum(([i-center(1),j-center(2),k-center(3)].^2)))>hm.radius(1))
        i=rand(1);
        j=rand(1);
        k=rand(1);
    end
    Random_Dipoles(l,1:3)=[i j k];
    waitbar(l/Nr_Dipoles,h)
end
delete(h)
end
```

```
function [R,d]=GenerateDipole(x,r)
R=zeros(x,6);                            % Initialization of the dipole data consistently to the dimesions
for i=1:x
    [y,d(i)]=RandomInsideSphere(r);  % Generate randomly location coordinate of the dipole within the sphere
    where the radius is the smalles in the Head Modes
    for j=1:3
        R(i,j)=y(j);                     % Input data into the matrix
    end
    for j=1:3
        x=randsample(1*1000,1)/1000;% Generate randomly the orientation of the dipole
        R(i,j+3)=x;                      % Input data into the matrix
    end
end
end
```

```
function [RotateFreqRad_Inter,TimeArray]=GenerateTimeSeries(f,RotateFreq,Duration)
% f=200;
T=1/f;                                     % Comput the period of sampling from the sampling frequency
TimeArray=0:T:Duration;                    % Time array of the sampling
TimeArray(end)=[];       % Removing the last sample in time since we start from 0 sec
%%
T_Retota=1/RotateFreq;                     % Compute the rotation period in sec
RotateAngle=0:T_Retota:20*Duration;         % Compute the array of the rotation in sec
RotateAngle(end)=[];  % Removing the last sample in time since we start from 0 sec
RotateFreqRad=RotateAngle*pi;              % Compute the array of rotation angle in rad
%%
A=length(RotateFreqRad);                   % Length of the rotate angle array
B=length(TimeArray);                       % Length of the time array
% A1=rem(B,A);                               % Division residue between this data
B1=round(B/A);                             % The scale of data size between this two data
RotateFreqRad_Inter=interp(RotateFreqRad,B1);
end
```

```
function Labeling(FontSize)
xlabel('X (m)','FontSize',FontSize);
ylabel('Y (m)','FontSize',FontSize);
zlabel('Z (m)','FontSize',FontSize);
set(gca,'FontSize',FontSize);
end
```

```
function [Control]=Localization_Error(Dipole,HeadModel,Voltage)
%%
Number_Dipoles=size(Dipole,1);
% Lead_Field_Matrix=zeros(size(V,1),3*Number_Dipoles);
%%
for i=1:Number_Dipoles
    source.loc=Dipole(i,:)';
    [~,Lead_Field_Matrix]=solve_forward(HeadModel,source);
```

```
    Control(i)=norm((eye(size(Voltage,1))-Lead_Field_Matrix*pinv(Lead_Field_Matrix))*Voltage,'fro')/norm(
    Voltage,'fro');
end
Control=sum(Control(i));
end
```

```
function [Result]=Minimize_Function(Random_Dipoles,HeadModel,V)
%%
h=waitbar(0,'Locating the sources');
Nr_Dipoles=size(Random_Dipoles,1);
for i=1:Nr_Dipoles
    % Location calculation
    [Result(i,1:3),Voltage]=fminsearch('Localization_Error',Random_Dipoles(i,1:3),optimset('TolX',1.e-4,'
    TolFun',1.e-5,'Display','off','MaxFunEvals',100000,'MaxIter',10000),HeadModel,V);
    % Orientation calculation
    source.loc=Result(i,1:3);
    [~,L]=solve_forward(HeadModel,source);
    Orientation=pinv(L)*V;
    Orientation_Nomalized=Orientation./norm(Orientation);
    Result(i,4:6)=[Orientation_Nomalized(1),Orientation_Nomalized(2),Orientation_Nomalized(3)];
    Result(i,7)=Voltage;
    waitbar(i/Nr_Dipoles,h);
end
delete(h);
end
```

```
function Poly_Leng_L=PLG(Leng, mode, Value)
%%
if (mode < 0 || mode > Leng || abs(Value) > 1.0)
    disp('Bad arguments in routine plgndr')
    Poly_Leng_L=NaN;
    return ;
end
temp4=1;
if (mode > 0)
    somx2=sqrt(1-Value.^2);
    temp=cumsum([1,2*ones(1,mode-1)]);
    temp1=-somx2*(temp);
    temp3=cumprod(temp1);
    temp4=temp3(end);
end
if (Leng== mode)
    Poly_Leng_L=temp4;
    return
else
    temp5=Value*(2*mode+1)*temp4;
    if (Leng == (mode+1))
        Poly_Leng_L=temp5;
        return
    else
        for i=mode+2:Leng
            Poly_Leng_L=(Value*(2*i-1)*temp5-(i+mode-1)*temp4)/(i-mode);
            temp4=temp5;
            temp5=Poly_Leng_L;
        end
    end
end
```

```
function [DOE,DLE]=Plot_EE(Result,sourceEEG1,~)
%%
[DOE,DLE]=Estimation_Error(Result,sourceEEG1);
figure('units','normalized','outerposition',[0 0 1 1])
X=1:length(DOE);
[hAx,~,~]=plotyy(X,DOE,X,DLE);
title('Estimation Error','FontSize',20)
xlabel('Dipole number','FontSize',20)
ylabel(hAx(1),'DOE [deg]','FontSize',15) % left y-axis
ylabel(hAx(2),'DLE [m]','FontSize',15) % right y-axis
set(hAx,'FontSize',15);

end
```

```
function Plot_Electrode_Potential(V,nr)
figure(nr),plot(V,'x','LineWidth',13),hold on,plot(V,'r','LineWidth',3)
xlabel('Electrode Number','FontSize',20)
ylabel('Voltage [uV]','FontSize',20)
title('Electrode potential','FontSize',20)
set(gca,'fontsize', 20);
end
```

```
function PlotEEG(sig,t)
% t = linspace(0,2,1024);
% sig = rand(32,1024);
[dim1,dim2]=size(sig);
% calculate shift
mi = min(sig,[],2);
ma = max(sig,[],2);
shift = cumsum([0; abs(ma(1:end-1))+abs(mi(2:end))]);
shift = repmat(shift,1,dim2);

%plot 'eeg' data
plot(t,sig+shift)

% edit axes
set(gca,'ytick',mean(sig+shift,2),'yticklabel',1:dim1)
grid on
% ylim([mi(1) max(max(shift+sig))])

end
```

```
function PlotPotemtialMap(V_in,hm)
[dim1, ~]=size(V_in);
for i=1:dim1
    a=sprintf('Estiantion %d',i);
    showpotentials(V_in(i,:)',hm)
    title(a)
end
end
```

```
function PlotRRE(RRE,~)
%%
% figure(Nr)
figure('units','normalized','outerposition',[0 0 1 1])
plot(RRE,'x','LineWidth',10)
hold on
plot(RRE,'o','LineWidth',10)
hold on
plot(RRE,'r','LineWidth',3)
xlabel('Solutions trials','FontSize',20);
ylabel('Error value in [A.u]','FontSize',20);
title('RRE plot for respective dipole','FontSize',20);
set(gca,'FontSize',20);
end
```

```
function PlotSIR(SNR,F1)
figure
plot(SNR,F1,'x')
hold on
plot(SNR,F1,'o')
hold on
plot(SNR,F1,'r')
xlabel('SNR trial'),ylabel('SIR')
end
```

```
function Position_Of_Electrodes(hm,r,manner)
%%
Elect_Postion=Spherical_To_Cartesian_Coordiante([hm.elpos(:,1)'; hm.elpos(:,2)'; r*1*ones(size(hm.elpos(:,2)))']);
X_el=Elect_Postion(1,:)';
Y_el=Elect_Postion(2,:)';
Z_el=Elect_Postion(3,:)';
S = 500*ones(size(X_el));
C = 500*ones(size(X_el));
scatter3(X_el,Y_el,Z_el,S,C,manner),view(-90,90)
end
```

```
function Potential_Distribution(V,Head_Model,~,barcolor)
%%
scale=1.5;
Font=20;

% figure(Nr_Fig)
figure('units','normalized','outerposition',[0 0 1 1])
set(axes,'NextPlot','add');
r=Head_Model.radius(end);
[X,Y,Z]=sphere(50);
fvc=surf2patch(r*X,r*Y,r*Z);
manner='ko';

```

```
Position_Of_Electrodes(Head_Model,r,manner)
Electrod_Names(Head_Model,r,Font-5)

Se=Spherical_To_Cartesian_Coordiante([Head_Model.elpos(:,1)'; Head_Model.elpos(:,2)'; r*1.17*ones(size(
    Head_Model.elpos(:,2)))']);
F=TriScatteredInterp(Se(1,:)',Se(2,:)',Se(3,:)',V);
VI=F(fvc.vertices(:,1),fvc.vertices(:,2),fvc.vertices(:,3));
p1=patch(fvc);
set(p1,'FaceColor','interp','EdgeColor','none','FaceVertexCData',VI);


lightangle(-45,30)
h.FaceLighting = 'gouraud';
h.AmbientStrength = 0.3;
h.DiffuseStrength = 0.8;
h.SpecularStrength = 0.9;
h.SpecularExponent = 25;
h.BackFaceLighting = 'unlit';

if barcolor==1
    c=colorbar('westoutside');
    % c=colorbar('AxisLocation','in')
    c.Label.String = 'Potential distribution';
end
Draw_Nouse(r,Font);
Labeling(Font)
Bounding(scale,r);
view(-90,90)
end
```

```
function Potential_Movie(Head_Model,EEG,fs)
%%
a=size(EEG);
for i=1:max(a);
    Potential_Distribution(EEG(i,:)',Head_Model,30,0)
    drawnow;
    pause(1/fs);
end
end
```

```
function [x,y]=RandomInsideSphere(r)
for i=1:3
    x(i)=randsample(r*9000,1)/10000; %Randomply genereate three different value within the range o to the
    radius which has been put as an input
end

while(sqrt(sum(x.^2))>=r)           % If the initial distance from the center [0 0 0] is outside the sphere
    , re initialize another sample.
    for i=1:3                       %Keep on doing this till the condistion is meet
        x(i)=randsample(r*9000,1)/10000;
    end
end
y=sqrt(sum(x.^2));
end
```

```
function [B]=RotateDipole(A,x,y,z)

Rotate_y=[cos(-y) 0 -sin(-y);0 1 0;sin(-y) 0 cos(-y)];      % Rotation matrix along y direction
Rotate_x=[1 0 0;0 cos(x) sin(x);0 -sin(x) cos(x)];           % Rotation matrix along x direction
Rotate_z=[0 cos(z) -sin(z);sin(z) cos(z) 0;0 0  1];          % Rotation matrix along x direction
Total_Rotation_Matrix=Rotate_x;%*Rotate_y;%*Rotate_z;           % Rotation matrix for all directions
B=Total_Rotation_Matrix*A;                                   % Rotated point


end
```

```
function [index]=Save_Image(index)
%%
File=sprintf('.jpg');
Name=strcat('C:\Users\vkomin1\Dropbox\Apps\ShareLaTeX\Katholieke university of Leuven\Bio-Medical Data
    Processing part II\A4\Images4\',num2str(index),File);
% saveas(gcf,Name)
index=index+1;
close all
end
```

```
function Show_Dipole_Placement(dipole,Head_Model,Nr_Fig)
%%
Font=20;
```

```matlab
Nr_Points=20;
scale=1.5;
manner='filled';
% figure(Nr_Fig)
figure('units','normalized','outerposition',[0 0 1 1])
r=Head_Model.radius(end);
set(axes,'NextPlot','add');

DrawSpehere(Nr_Points,r)
Draw_Nouse(r,Font);

Position_Of_Electrodes(Head_Model,r,manner)
Electrod_Names(Head_Model,r,Font)

%plot dipoles
for i=1:size(dipole,1)
    dipole(i,4:6)=0.3*r*dipole(i,4:6)./norm(dipole(i,4:6));
    plot3(dipole(i,1),dipole(i,2),dipole(i,3),'g.','MarkerSize',50);
    plot3([dipole(i,1) dipole(i,1)+dipole(i,4)],...
        [dipole(i,2) dipole(i,2)+dipole(i,5)],...
        [dipole(i,3) dipole(i,3)+dipole(i,6)],'b','LineWidth',3);
    a=8e-4*max([dipole(i,1),dipole(i,2),dipole(i,3)]);
    text(dipole(i,1)+a,dipole(i,2)+a,dipole(i,3)+a,'Dipole','FontSize',Font);
end

Labeling(Font)
title('Dipole locations','FontSize',Font);
Bounding(scale,r);
set(gca,'FontSize',Font);
view(3)
end
```

```matlab
function [EEG]=Simulate_EEG(sourceEEG,hm,f,RotateFreq,Duration)
%%
[RotateFreqRadReplecated,~]=GenerateTimeSeries(f,RotateFreq,Duration); % Compute the array of the rotation angles
for i=1:length(RotateFreqRadReplecated)
    Rotation=[0 sin(RotateFreqRadReplecated(i)) cos(RotateFreqRadReplecated(i))];%RotateDipole(sourceEEG.ori',RotateFreqRadReplecated(i),0,0)      % Rotation the dipole consistently to the rotation angles
    sourceEEG.ori=Rotation;
    [EEG(i,:),~]=Spherical_Head_Model(hm,sourceEEG);                       % Compute the voltage done by this potential
end
end
```

```matlab
function [Result]=Solve_Inverse_Problem(V,Head_Model)
%%
% Generate 5 random dipoles
n=5;
[Random_Dipole]=Generate_Random_Dipoles(Head_Model,n);
for i=1:n
    [Result(i,1:3),RRE]=fminsearch('Dipole_Position_Inverse',Random_Dipole(i,1:3),optimset('TolX',1.e-4,'TolFun',1.e-5,'Display','off','MaxFunEvals',100000,'MaxIter',10000),Head_Model,V);
    source.loc=Result(i,1:3);
%     source.ori=Result(i,4:3);
    [~,L]=Spherical_Head_Model(Head_Model,source);
    Orientation=pinv(L)*V;
    Orientation=Orientation./norm(Orientation);
    Result(i,4:6)=Orientation;
    Result(i,7)=RRE;
end
end
```

```matlab
function Result=Solve_Inverse_Problem_Dipole_Ana(V,hm)
%%
n=5;
[Random_Dipoles]=Generate_Random_Dipoles(hm,n);
[Result]=Minimize_Function(Random_Dipoles,hm,V);
end
```

```matlab
function [V,L]=Spherical_Head_Model(Head_Model,source)
%%
Location=source.loc;
Electrode_position=Head_Model.elpos;
Radiuses=Head_Model.radius;
X=Head_Model.condratio;
S=Head_Model.condsoft;
rhead=Radiuses(3);
r2=rhead;
```

```

f1=Radiuses(1)/Radiuses(3);
f2=Radiuses(2)/Radiuses(3);

K1=1/(4*pi*S*rhead^2);



NOFTERMS=80;

theta=Electrode_position(:,1);
phi=Electrode_position(:,2);
Nr_Of_Electrodes=length(theta);




Tx=zeros(Nr_Of_Electrodes,1);% Tangent component along X
Ty=zeros(Nr_Of_Electrodes,1);% Tangent component along Y
Tz=zeros(Nr_Of_Electrodes,1);% Tangent component along Z

t=zeros(Nr_Of_Electrodes,1);
q=zeros(Nr_Of_Electrodes,1);
R2=zeros(Nr_Of_Electrodes,3);
cosalpha=zeros(Nr_Of_Electrodes,1);

V=zeros(Nr_Of_Electrodes,1);
Vx=zeros(Nr_Of_Electrodes,1);
Vy=zeros(Nr_Of_Electrodes,1);
Vz=zeros(Nr_Of_Electrodes,1);

for i=1:Nr_Of_Electrodes
    R2(i,1)=rhead*sin(theta(i))*cos(phi(i));
    R2(i,2)=rhead*sin(theta(i))*sin(phi(i));
    R2(i,3)=rhead*cos(theta(i));
end


R1=Location;
r1=sqrt( R1(1)^2+R1(2)^2+R1(3)^2 );
if r1~=0
    %% Computation of Tangential and Radial components
    b=r1/rhead;
    p=r1^2;
    for i=1:Nr_Of_Electrodes
        q(i)=R1(1)*R2(i,1) + R1(2)*R2(i,2)+ R1(3)*R2(i,3);
        cosalpha(i)=q(i)./r1./r2;
        if (abs(cosalpha(i))>1.0)
            cosalpha(i)=cosalpha(i)/abs(cosalpha(i));
        end
        Tx(i)=R2(i,1)*p - R1(1)*q(i);
        Ty(i)=R2(i,2)*p - R1(2)*q(i);
        Tz(i)=R2(i,3)*p - R1(3)*q(i);
        t(i)=sqrt(sum([Tx(i)  Ty(i) Tz(i)].^2));
        if (t(i)~=0)
            Tx(i)=Tx(i)/t(i);
            Ty(i)=Ty(i)/t(i);
            Tz(i)=Tz(i)/t(i);
        end
    end
    Rx=R1(1)/r1;% Radial component along X
    Ry=R1(2)/r1;% Radial component along Y
    Rz=R1(3)/r1;% Radial component along Z

    %%
    for i=1:NOFTERMS
        i1=2*i+1;
        gi=((i+1)*X+i)*(i*X/(i+1)+1) + (1-X)*((i+1)*X+i)*(f1^i1-f2^i1) -...
            i*(1-X)*(1-X)*(f1/f2)^i1;
        fact1=X*i1^3*b^(i-1)/gi/(i+1)/i;
        for j=1:Nr_Of_Electrodes
            fact2=i*PLG(i,0,cosalpha(j));
            fact3=-PLG(i,1,cosalpha(j));
           %              fact2=i*legendre(i,cosalpha(j));
           %              fact3=-legendreP(i,cosalpha(j));
            Vx(j)=Vx(j)+fact1*(fact2*Rx+fact3*Tx(j));
            Vy(j)=Vy(j)+fact1*(fact2*Ry+fact3*Ty(j));
            Vz(j)=Vz(j)+fact1*(fact2*Rz+fact3*Tz(j));
        end
```

```
    end
else
    i=1; i1=2*i+1;
    gi=((i+1)*X+i)*(i*X/(i+1)+1) + (1-X)*((i+1)*X+i)*(f1^i1-f2^i1) -...
        i*(1-X)*(1-X)*(f1/f2)^i1;
    fact1=X*i1^3^(i-1)/gi/(i+1)/i;
    for j=1:Nr_Of_Electrodes
        Vx(j)=fact1*PLG(1,0,R2(j,1)/r2);
        Vy(j)=fact1*PLG(1,0,R2(j,2)/r2);
        Vz(j)=fact1*PLG(1,0,R2(j,3)/r2);
        %         Vx(j)=fact1*legendre(1,R2(j,1)/r2);
        %         Vy(j)=fact1*legendre(1,R2(j,2)/r2);
        %         Vz(j)=fact1*legendre(1,R2(j,3)/r2);
    end
end
%%
Voltage_Ref=zeros(Nr_Of_Electrodes,3);
for i=1:Nr_Of_Electrodes
    Voltage_Ref(i,:)=[Vx(i) Vy(i) Vz(i)];
end

% average referencing
Voltage_Ref = Voltage_Ref-ones(Nr_Of_Electrodes,1)*sum(Voltage_Ref)/Nr_Of_Electrodes;
L=K1*Voltage_Ref*1e-3;
if  1 < length(fieldnames(source))
    V=L*source.ori';
else
    V=[];
end
end
```

```
function Cartesian_Coordinates=Spherical_To_Cartesian_Coordiante(Sphere_Coord)
Azimuth=Sphere_Coord(1,:);
Elevation=Sphere_Coord(2,:);
Radius=Sphere_Coord(3,:);
%%
x=Radius.*cos(Elevation).*sin(Azimuth);
y=Radius.*sin(Elevation).*sin(Azimuth);
z=Radius.*cos(Azimuth);
%%
Cartesian_Coordinates(1,:)=x;
Cartesian_Coordinates(2,:)=y;
Cartesian_Coordinates(3,:)=z;
end
```

```
function Vangjush_Parameter_2_Latex_Table(Control,str1)
%% LLE
name_file=strcat('C:\Users\vkomin1\Dropbox\Apps\ShareLaTeX\Katholieke university of Leuven\Bio-Medical Data
    Processing part II\A4\Files\',str1,'.txt');
fileID = fopen(name_file,'w');
fprintf(fileID,'%s',' \hline');
for i=1:10
    Temp=Control{i};
    for j=1:5
        fprintf(fileID,'$Tria%dD%d$&$',i,j);
        for k=1:7
            if k<7
                fprintf(fileID,'%6.2e$&$',Temp(j,k));
            end
            if k==7
                fprintf(fileID,'%6.2e$\\\\',Temp(j,k));
            end
        end
        fprintf(fileID,'%s',' \hline');
    end
end
fprintf(fileID,' \n');
fclose(fileID);
%%
end
```

```
function Vangjush_Parameter_2_Latex_Table1(Control,str1)
%%
name_file=strcat('C:\Users\vkomin1\Dropbox\Apps\ShareLaTeX\Katholieke university of Leuven\Bio-Medical Data
    Processing part II\A4\Files\',str1,'.txt');
fileID = fopen(name_file,'w');
fprintf(fileID,'%s',' \hline');
for j=1:10
    fprintf(fileID,'$Trial%d$&$',j);
```

```
    for k=1:5
        if k<5
            fprintf(fileID,'%6.2e$&$',Control(j,k));
        end
        if k==5
            fprintf(fileID,'%6.2e$\\\\',Control(j,k));
        end
    end
    fprintf(fileID,'%s',' \hline');
end
fprintf(fileID,' \n');
fclose(fileID);
%%
end
```

```
function Vangjush_Parameter_2_Latex_Table2(Control,str1)
%%
name_file=strcat('C:\Users\vkomin1\Dropbox\Apps\ShareLaTeX\Katholieke university of Leuven\Bio-Medical Data Processing part II\A4\Files\',str1,'.txt');
fileID = fopen(name_file,'w');
fprintf(fileID,'%s',' \hline');
for j=1:5
    fprintf(fileID,'$D%d$&$',j);
    for k=1:7
        if k<7
            fprintf(fileID,'%6.2e$&$',Control(j,k));
        end
        if k==7
            fprintf(fileID,'%6.2e$\\\\',Control(j,k));
        end
    end
    fprintf(fileID,'%s',' \hline');
end
fprintf(fileID,' \n');
fclose(fileID);
%%
end
```

```
function Vangjush_PlotEEG(sig,t)
%%
% t=0:size(sig,2)-1;
[dim1,dim2]=size(sig);
mi = min(sig,[],2);
ma = max(sig,[],2);
shift1 = cumsum([0; abs(ma(1:end-1))+abs(mi(2:end))]);
shift = repmat(shift1,1,dim2);
plot(t,sig+shift,'LineWidth',2)
set(gca,'ytick',mean(sig+shift,2),'yticklabel',1:dim1)
grid on
end
```