MATLAB suite for removing the capacitive leakage effect from EBI Spectroscopic data

By

Muhammad Danish Siddiqui

Suhasini Gopi
MATLAB suite for removing the capacitive leakage effect from EBI Spectroscopic data

Muhammad Danish Siddiqui
d.danish.siddiqui@gmail.com

Suhasini Gopi
shruthihashini@gmail.com

Master thesis

Subject Category: Medical Technology, Electrical Engineering

University of Borås
School of Engineering
SE- 501 90 BORÅS
Phone number: +46 33 435 4640

Examiner: Fernando Seoane Martínez
Supervisor: Rubén Buendía López

Keywords: Electrical Bioimpedance Spectroscopy, Cole Model, Hook effect, Signal Analysis, MATLAB.
Dedicated

To Almighty, our beloved parents and our friends.
**ABSTRACT**

Electrical Bioimpedance (EBI) is the opposition offered by the biological material to the flow of electric current. Nowadays EBI technology is widely used for total body composition and body fluid distribution.

In this project a software suite is developed by using the GUI tool of Matlab, this software is meant to help to remove artefacts from the EBI measurement and to visualize the EBIS measurements and the processing performed on them.

*Hook effect* is one of the major artefacts found in EBI measurements, which creates problems in any analysis. To eliminate the *Hook effect* some methods are followed. The data’s are processed using these methods and they are visualized. For the better understanding, both raw data and the corrected data are plotted in impedance and admittance plots. The corrected data is stored for further use and analysis.
ACKNOWLEDGMENT

Thanks to Almighty

We would like to thank everyone who supported us all the time, all through this project and helping us to make it possible.

We would like to dedicate our entire work to our family and our friends who have been our constant support in all our works and our life. Special thanks to our parents who supported and guided us all our way and their patience for all naughty things we have done from our young age till now. We love you mummy, papa.

We would especially thank, Dr. Fernando Seoane Martinez, for his help and support in this whole program. He has been our major inspiration since we started this program. Working under his guidance we got to learn lots of things which are priceless. And of course our supervisor Ruben Buendia, was a great support and helped us to make this work possible. He was there at all times to help us out when we were stuck. His friendly nature helped us to feel comfortable and learn things without any hurdles. It was really our pleasure working with you Ruben.

Regards

Muhammad Danish Siddiqui

Suhasini Gopi
# CONTENTS

## CHAPTER 1. Introduction

1.1. Introduction ........................................................................................................... 9  
1.2. Motivation .............................................................................................................. 9  
1.3. Goal ......................................................................................................................... 10  
1.4 Work Done ............................................................................................................ 10  
1.5 Structure of The Thesis Report ............................................................................. 10

## CHAPTER 2. Electrical Bioimpedance Background

2.1 Introduction Of EBI ............................................................................................... 11  
2.1.1. Electrical Properties Of Tissue ....................................................................... 11  
2.1.2. Equivalent Circuit Of A Cell ........................................................................... 12  
2.1.3. Frequency Dispersion ..................................................................................... 12  
2.2. Cole Equation ...................................................................................................... 13  
2.2. Hook Effect ......................................................................................................... 14

## CHAPTER 3. METHODS & MATERIALS

3.1. Matlab And Matlab Gui ....................................................................................... 16  
3.1.2. Creating And Displaying A Graphical User Interface .................................. 19  
3.1.3. User Interface Controls .................................................................................. 19  
3.2. Measurements Used .......................................................................................... 20  
3.2.1. Impedimed Bioimp ....................................................................................... 20  
3.3. Td Compensation ............................................................................................... 22  
3.4. Cpar Correction Approach ............................................................................... 22  
3.4.1. Estimation Of The Stray Capacitance ......................................................... 22

## CHAPTER 4. Electrical Bioimpedance analysis toolbox

4.1. Work Flow ........................................................................................................... 24  
4.2 Measurements Loading ....................................................................................... 25  
4.3 Data Processing .................................................................................................. 26  
4.3.1 Hook Effect Detection And Estimation Of The Stray Capacitance ........... 26  
4.3.2 Correchor Factor ............................................................................................. 27  
4.3.3 Correction Function ....................................................................................... 28  
4.3.4 Implementing The Correction Function ....................................................... 30  
4.4 Visualization ....................................................................................................... 31  
4.5. Storage ............................................................................................................... 35

## CHAPTER 5. Discussion

...................................................................................................................................... 37
5.1 Discussions ...........................................................................................................................................37

CHAPTER 6. Conclusions & future work ...................................................................................................38

6.1 Conclusions .........................................................................................................................................38

6.2 Future Work ........................................................................................................................................38

References .................................................................................................................................................39
LIST OF ACRONYMS

EBI  - Electrical Bioimpedance
EBIS - Electrical Bioimpedance Spectroscopy
TBC  - Total Body Composition
GUI  - Graphic User Interface
CHAPTER 1.

INTRODUCTION

1.1 Introduction

Electrical Bioimpedance (EBI) is the study of the opposition offered by biological material, such as human tissue, meat or wood, to the flow of electric charges.

EBI measurements are useful for several clinical applications, most of them for disease diagnosis, as an example, skin cancer detection, known as electronic biopsy [1] can be mentioned.

Into the field of Electrical Bioimpedance Spectroscopy (EBIS) measurements the most remarkable application is Total Body Composition (TBC) assessment, enabling the estimation of extracellular and intracellular water, as well as fat free mass.

One of the most usual artefacts caused by parasitic elements present on the measurement scenario on EBI measurements is the so-called Hook effect, which presents a problem when analyzing the measured EBI spectrum or when performing a Cole fitting.

To avoid producing a wrong fitting or obtaining an erroneous spectral analysis of the measurement, the Hook effect must be corrected if possible or at least compensated prior any kind of analysis is performed on the EBI data.

1.2 Motivation

Nowadays EBI technology is constantly upgrading and widely spreading technology used for the estimation of body composition, and provides detailed structure of fluid distribution of the body.

By using EBIS analysis the parameters of the Cole Equation can be calculated, as well as the components of the electrical equivalent circuit.

By using the Cole Parameters, the body features and, by applying the mixture theory equations, the TBC can be estimated.

MATLAB is probably the most spread and powerful signal processing software tool and EBIS analysis technique should have proper signal analysis software developed into MATLAB. This led us to develop a software suite using a correction method for the capacitive leakage artefact.
1.3 Goal

The main goal of this thesis is to develop a software suite by implementing a MATLAB tool that in a visual, easy and intuitive way, eliminate the stray capacitance estimated, compensate the Hook effect and show the results of the Cole fitting and the admittance features before and after the correction.

1.4 Work Done

A software tool is developed using GUI tool of MATLAB. With this software suite, the files of .mfu extensions are loaded into MATLAB. And they are processed in such a way the stray capacitance and the Hook effect of the raw data are corrected. The result appears in the GUI in such a way that the user can compare the raw data and the corrected data simultaneously. The processed data are again saved back to .mfu extension which can be used by Bioimp software.

1.5 Structure of the Thesis Report

The whole thesis report is organized in five chapters and the references at the last. Chapter 1 contains the introduction to the thesis report and Chapter 2 contains the Electrical Bioimpedance background, focusing on EBI and Hook effect. Chapter 3 describes the EBI analysis suite explaining the needs of EBI applications and explaining the tools used in MATLAB and graphical user interface (GUI). Chapter 4 describes about the analysis suite and the functions used in programming the software. Chapter 5 contains conclusions and discussion part.
2.1 Introduction of EBI

Electrical Bioimpedance measures the electrical properties of the body composition and its tissue characteristics which are reacting to the applied external stimulus. EBI applications are widely spread and increasing, but also under constant development. The application areas are widest in medical technology e.g. in psychophysiology, cardiology, neurology, surgery, cognitive sciences, physiotherapy, and sport medicines [1].

In the last two decades, many emerging techniques have been developed, like e.g. imaging method for tissue or bioimpedance tomography, based on series of electrodes attached in a single plane to the chest of the patient. The image of the tissue is formed with the help of impedance information which was retrieved from the tissue [2].

2.1.1. Electrical Properties of Tissue

The electrical properties of body tissues and cell suspension have been used in analyzing various diseases from a very long time and it is carried on till date. There are lots of research work done until now and is still undergoing in this field which results in producing the EBI applications.

To analyze the response to the applied electric current to the tissues, microscopic study of the cell is very essential. The cell consists of a cell membrane, a double layer of phospholipids
(lipid bilayer), enclosing the cytoplasm, consisting of the cytosol within which the nucleus and all the other cell organelles reside. A general animal cell structure is shown in the Fig.2.1 [2].

2.1.2. Equivalent Circuit of a Cell

The electrical behaviour of the single cell can be easily studied with the equivalent circuit model. This model represents the cell and tissue as a parallel connection of elements, representing the intracellular and the extracellular spaces as well as the plasma membrane. Electrical properties of tissue are given by its constituents and the electrical behaviour can be represented with a model, which is given in figure 2.2 [6].

![Equivalent Circuit of a Cell](image)

**Figure 2.2:** Electrical Equivalent of cell. The circuit (a) is the one relying on cell membrane and the intra- and extra- environment of it. The circuit (b) is the equivalent of circuit (a) and the circuit (c) is the simplification of the circuit (b) [7]

![Conductivity and Permittivity](image)

**Fig 2.3** Frequency dependence of the conductivity and permittivity of the brain grey[8]

2.1.3. Frequency Dispersion

The electrical property of the biological tissue represents some frequency behaviour change due to frequency dependence on the permittivity and the conductivity. The frequency dispersion of the biological tissue is not constant and gives four different dispersion windows.
The four major classes of dispersion are classified into $\alpha$ dispersion, $\beta$ dispersion, $\gamma$ dispersion and $\delta$ dispersion by H.P. Schwan in 1957 [3].

$\alpha$-dispersion:

The $\alpha$-dispersion appears at low frequency in the range of 10 Hz to 10 kHz. The main mechanism of $\alpha$-dispersion are the active cell membrane and gate channels, counter ion effects near the membrane surfaces, dielectric losses, intracellular structures and ionic diffusion.

$\beta$ dispersion:

$\beta$ dispersion generally ranges from 10 kHz to 1 MHz [4], and this dispersion is caused mainly due to low conductivity and capacitive properties of the membrane. This dispersion is also caused due to other internal membrane and their interactions with the intra and extracellular electrolytes.

$\gamma$ dispersion:

$\gamma$ dispersion is mainly due to excess amount of water in the cell and tissue. The water in the tissue is identical to the normal water and it relaxes at 20 GHz, expected for the proteins and amino acids which is present in the cell. Tissue water displays a wide range of dispersion from hundreds of MHz to some GHz.

$\delta$-dispersion:

The minor relaxation between $\beta$ and $\gamma$ is termed to be named as $\delta$-dispersion which is caused by rotation of amino acids, partial rotation of charge side groups of proteins and relaxation of protein-bound water that occurs between 300 and 2000 MHz [3].

2.2. Cole Equation

The Cole equation is a mathematical model that fits the measured EBI data to a suppressed semi-circle in the $Z$ plane, where the two intersections between the circle and the $x$-axis define the resistance at zero frequency, $R_0$ and at infinite frequency, $R_\infty$ (Figure 2.4) (Cole 1940). The Cole equation is as follows:

$$Z(\omega) = R_\infty + \frac{R_0 - R_\infty}{1 + (j\omega \tau)^{\alpha}}$$

Equation (1)
2.3. Hook Effect

In EBI spectroscopy measurements, the *Hook Effect* is one of the most commonly founded measurement artefacts which is caused by stray capacitances. *Hook Effect* are mostly caused by capacitances at high frequencies and gives a hook like deviation on the EBI data produced on the impedance plot, reactance and phase [5].

The impedance plot of EBI measurement gives a hook like deviation in the lower values of impedance which is termed as *Hook Effect*. In the figure below it is observed how at high frequency the reactance is higher than expected, and it is also increasing despite of decreasing which is also expected in EBI systems.

![Figure 2.5 The Hook effect to reactance (a) and phase (b) [5]](image)

![Figure 2.6 Module of the Reactance Vs frequency showing the hook-alike deviation [5]](image)
The deviation produced in the data affect the EBI spectrum at all AC frequencies and it deviates both real and imaginary parts of the EBI system, that is why both, module and phase of the EBI spectra are also deviated [5].
CHAPTER 3.  

METHODS & MATERIALS

3.1. Matlab and Matlab Gui

MATLAB is probably the most spread and powerful signal processing tool, for this reason it was chosen to implement this graphic interface.

The software tool is developed using GUI tool of MATLAB. We selected this method to make use of the most advantages of the MATLAB software and to make our software user friendly.

The software screen consists on push buttons, plotting axes, slider tool and text boxes. These properties used in the software are very helpful to provide a user friendly application. In order to create the GUI application the following elements has been used.

Components:

Pushbuttons, text boxes, slider tool, static text and plotting axes are the components which are used in this software. GUI gives the possibility of drag and drop options for designing the software using these components. It is then interfaced by `uicontrol`, `uimenu` and `uicontextmenu` functions. The graphical plotting is done using the axes function which is integrated to the plotting axes.

Figures:

Figure is the basic structure of the GUI panel which can hold all the components of any combination. Figure is also the function which gives the blank template structure in GUI.

Call-backs:

This function plays a vital role in GUI for interfacing the components and the functions which can be called on a single mouse click. The MATLAB should respond to the special function which is being executed. The code of a specific function is being executed on every call to that function.

The implementation of GUI needs two files, one is .m file and a graphical fig. The .m file is being first developed, in which all functions are programmed and executed on specific event. The graphic fig. is where all the components are placed and labelled. The sub routines are needed to link both these .m files and graphical fig.
function varargout = MainGuide(varargin)

% MAINGUIDE M-file for MainGuide.fig

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

% Copyright 2002-2003 The MathWorks, Inc.

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

% Last Modified by GUIDE v2.5 19-Nov-2010 13:54:16

% Begin initialization code - DO NOT EDIT
    'gui_Singleton = 1;
    'gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @MainGuide_OpeningFcn, ...
    'gui_OutputFcn', @MainGuide_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);

if nargin && ischar(varargin{1})
% End initialization code - DO NOT EDIT
%------------------------------------------------------------------------

Subroutine

function varargout = pushbutton1_Callback(hObject, eventdata, handles)
% Stub for Callback of the uicontrol handles.pushbutton1
fprintf('pushbutton 1 Callback not implemented yet')

%------------------------------------------------------------------------

Subroutine

function varargout = pushbutton2_Callback(hObject, eventdata, handles)
% Stub for Callback of the uicontrol handles .pushbutton2
fprintf('pushbutton 2 Callback not implemented yet')

%------------------------------------------------------------------------

Subroutine Edit Text Button

function varargout = edit1_Callback(hObject, eventdata, handles)
% Stub for Callback of the uicontrol handles.pushbutton1

Code box 3.1 .m file in a guide with two pushbuttons and one edit text button
3.1.2. Creating and Displaying a Graphical User Interface

GUIDE is the tool used to create GUI in MATLAB. GUIDE creates the layout of GUI selecting and using the components in a drag and drop method. Each component has its own properties, live name, colour, text display etc. When GUIDE is executed it creates a large layout area which is grey in colour, where GUI can be laid out. The components of GUI are listed in the left side of the layout. A toolbar which allows modifying the properties of components, adding menus and controlling other features of the GUI is placed on the top of the window.

3.1.3. User Interface Controls

This section explains the components which help to build GUI in MATLAB. Here we discuss about some of the components which are used to build our application suite.

Pushbuttons: pushbutton is a component which is to be clicked to execute a special event. For every mouse click on a pushbutton a call-back function is executed.

Slider tool: Slider tool only accepts the numerical data as input with a specific range of data by moving a bar in a mouse click or mouse move. The slider value is always between the minimum and the maximum value depending on the input data.

Edit text: Edit text allows user to insert and print the text string as input. We are able to write numerical values and text strings in the text box. When the user wants to insert a numerical value, str2double a MATLAB function is used to convert a string to numerical value.

Static text: Static text prints text, numbers, and figures on a static manner. It executes the static text in the form of a label in the layout.
**Toggle Button:** toggle button is an “on and off” button on every mouse click. Check boxes and radio buttons are under the category of the toggle buttons. Check boxes display the options whether they are active or inactive. Radio buttons are used to select the data among several options.

**Axes:** creates axes for graphic plots. It helps us to create a graphic object with the specified property value of the axes. The axes draw image, line, patch, rectangular, surface and other text graphic objects. The basic purpose of the axes are to provide coordinate system for plotted data, axes properties provide considerable control over the way MATLAB displays data.

### 3.2. Measurements Used

The EBI measurements used for testing the GUI made in this project are brain measurements, from healthy adult subjects, taken with SFB7 spectrometer within the range of 3KHZ to 1 MHZ.

Each of the 360 measurements is composed of 256 frequencies measurement points within the frequency range 3KHZ to 1 MHZ. Each obtained spectrum was saved in an .mfu file, which contains the measurement of resistance, reactance and frequency. The measurements are exponentially distributed from lower frequencies to higher frequencies.

#### 3.2.1. Impedimed Bioimp

ImpediMed BioImp SFB7 is a Bioimpedance spectrometer manufactured by Bioimp, which scans 256 different frequencies of signal with the range of 3 kHz and 1000 kHz to measure EBI.

The software analyses and visualizes the multi frequency Bioimpedance data focusing on the analysis based on Cole modelling. ImpediMed Imp SFB7 is a low battery powered equipment and can be used for one shot measurement and for long time patient monitoring.

The Bioimp software targets data’s from Imp SFB7 and also supports files from different sources like ASCII text files containing multi-frequency impedance data.

Bioimp software is mostly used as a total body composition analyser application. The EBI data are visualized in the graphic form where the resistance and the reactance of the data are plotted against the frequency. Cole plot plays a major role of this software and it is placed in the centre of the software panel.

There are different functions which are used for data analyzing. The functions are not only for discarding the frequencies but the time delay compensation is also used. The data might be fitted to the extended Cole Model with time delay compensation. Function allows the data to be tuned, change the rejection and frequencies limits of the data.
Fig. 3.2. SFB7 Bioimpedance meter manufactured by Bioimp [7]

Fig.3.3 Example of data represented by Biolmp [7]
3.3. TD Compensation

TD compensation is the most widely used correction method for Hook effect. It is obtained by multiplying the spectral EBI data by the complex exponential in the form of exp [jωTd].

\[ Z_{corr} (\omega) = Z_{meas} (\omega) \times e^{-j\omega Td} \quad \text{Equation (2)} \]

Where \( Z_{corr} (\omega) \) is the corrected impedance, \( Z_{meas} (\omega) \) is the measured impedance and \( e^{-j\omega Td} \) is the exponential factor which uses the scalar Time Delay \( Td \).

This approach to correct the Hook Effect have 2 big limitations, the first one is that it only have an effect on the phase, while the module is also affected by the Hook Effect, the second one is that the Hook Effect can only correct the measurement at one single frequency while the Hook Effect is affecting to every frequency [5].

3.4. Cpar Correction Approach

TD Compensation is a widely spread technique currently in use to remove Hook effect. But the usage of TD compensation method was not helpful in correcting the Hook effect accurately. The capacitive leakage correction based in Cpar estimation is proposed to remove the Hook effect completely from the EBI data. Cpar estimation method deals with measurement in both real and imaginary parts.

3.4.1. Estimation of the Stray Capacitance

As discussed in the previous topics, the parasitic stray capacitance should be estimated and eliminated to remove the hook effect completely from the EBI measurement. Mostly the EBI data is worked on impedance plane, defined with resistance and reactance. But the admittance contain the real part and the imaginary part same like the impedance. The real part is given by the conductance and denoted by \( G(\omega) \) and the imaginary part is given by the Susceptance and is denoted by \( S(\omega) \). Thus the admittance is denoted by the mathematical expression:

\[ Y_{meas}(\omega)= G_{meas}(\omega)+ j S_{meas}(\omega) \quad \text{........ (5)} \]

Where \( S_{meas} \) the imaginary part of admittance is given by the equation 6

\[ S_{meas}(\omega) = S_{tissue}(\omega) + j\omega C_{stray} \quad \text{........(6)} \]
The Susceptance of the capacitance increases while the frequency decreases. But in the figure 3.6 we can observe that $S_{tissue}(\omega)$ go toward zero at high frequency, the portion of the measured Susceptance due to the tissue can be non-significant respecting the portion of the measured Susceptance due to the stray capacitance and so, the slope of the Susceptance can be considered linear with the value of $C_{stray}$. 

![Figure 3.5 Tissue Electrical susceptance given by the following Cole parameters: $R_\alpha=449.6$, $R^\infty=449.6$, $\alpha=0.7186$, $f_c=30.2$ kHz, at high frequency, between 1MHz and 1GHz [5]](image)

![Figure 3.6 Tissue Electrical susceptance given by the following Cole parameters: $R_\alpha=449.6$, $R^\infty=449.6$, $\alpha=0.7186$, $f_c=30.2$ kHz, at high frequency, between 1MHz and 1GHz [5]](image)
4.1 Work Flow

This section gives the structure of the work flow of the analysis tool in the way it was designed. The flow analysis is explained in each stage. And here follows the flow chart of the software to explain the work more clearly and to show how the work has been carried out.
4.2 Measurements Loading

The data browsing, format conversion and measurement loading functions are termed beneath one single function named `Read_Improved`.

The original data are in .mfu format, as soon as the Load pushbutton is executed, the `Read_Improved` function is called and executes. The pop up window pops out asking to select Cole parameter or the .mfu file. When the .mfu option is selected you are offered to browse for the data, this is done by using the MATLAB function `uigetdir`.

The software proceeds in extracting the data from the .mfu files, converting them to MATLAB format and preparing it to store in a 3-dimensional matrix array. The data are read, converted and the workspace is saved and thereafter loaded.

```matlab
case 'mfu-file',
    FileContent = dir(fullfile(FolderName, '*.mfu'));
    NumberOfFiles= length(FileContent);
    ZDataALL = zeros (256,3,NumberOfFiles);
    Array = zeros (256,3,NumberOfFiles);
    format long
    %WD = cd;
    cd (FolderName)
    char FileName
    for i=1:NumberFiles
        FileName = FileContent(i).name
        r = csvread(FileName,13,0);
        Freq= r(:,1)* 1000;
        Res= r(:,2);
        Reac= r(:,3);
        Ztusc1 = Res + Reac* j;
        ZData = [Freq,Res,Reac];
        ZDataALL(:,:,i)= ZData;
    FileN = cell(1,i);
    end
    for i=1:NumberFiles
        FileName = FileContent(i).name
        FileN(i)=cellstr(FileName);
    end

    Person_1=ZDataALL;
    Mean_Person_1= mean(ZDataALL,3)
    uisave('Person_1','Mean_Person_1','NumberFiles','FileN','MATLAB.mat');

end
```

Code box 4.1. block of codes for loading and reading data.
4.3 Data Processing

EBI data should be free from artefacts for the appropriate analysis of data. There may be artefacts caused by the measuring procedure and other external factors. Hook effect is the commonly found artefact in the EBI data. This is caused due to high frequencies with increasing reactance in Cole plot.

4.3.1 Hook Effect Detection and Estimation of The Stray Capacitance

The estimation of the stray capacitance is detected with the Suspectance of the measurements are at high frequencies as explained at chapter 3, point 3.4. A threshold limit is fixed to check whether the measurement is being affected by capacitive leakage. When the measurements are below the threshold level the measurement is considered free from the capacitive leakage effect. The measurement which crosses the threshold level is then subjected to the corrector function.

```
function [Cpar,d]=Cestimation_def(ZMatrix,F)

w=2*pi*F;
SMatrix=imag(1./ZMatrix);
limm=SMatrix./w;

d1=diff(SMatrix);
d=d1(255);
t=w(end-2:end);
% rel = [629 537 460 375 334 286 249 227] ;
rel=limm(end-2:end);

fh = @(x,p) p(1) + p(2)*exp(-x./p(3));
errfh = @(p,x,y) sum((y(:)-fh(x(:),p)).^2);
p0 = [mean(rel) (max(rel)-min(rel)) (max(t) - min(t))/2];

% search for solution
P = fminsearch(errfh,p0,[],t,rel);
% figure(3)

% plot the result
% figure(4)
% plot(t,rel,'bo',t,fh(t,P),'r-')
% t=6e6:100:5e7;%hold on

%Cpar=fh(1.5e7,P);
Cpartemp=fh(1.5e7,P);
Cpar=Cpartemp;

End
```

**Code box 4.2.** block of codes which explains estimation of Cpar
The above section of codes intends to give an idea of the estimation of the capacitive leakage. The way in which the threshold is selected and being used to differentiate the normal measurements and the measurements being affected by the capacitive leakage.

4.3.2 Corrector Factor

In order to eliminate the hook effect which is present in the EBI data, Td compensate is the most widely used method. But the Td approach compensation method is done by multiplying the measured EBI data by a complex exponential factor in the form of $\exp [j\omega Td]$. Thus Td approach compensation is considered to be scalar. This method is called as Td compensation method which is followed for correcting the hook effect in EBI measurement.

The scalar Td compensation will change only the phase of $Z_{meas}(\omega)$, but since the exponent is imaginary the module is left unchanged.

Next figures show the effect of the corrector factor over an ideal measurement in parallel with a 300 pF stray capacitance.

![Figure 4.1](image.png)

**Figure 4.1** Module comparison multiplying the measured impedance by a scalar $Td$, note that $Z_{meas}(\omega)$ is overlapped [5]
Thus Td compensation should be complex in order to correct the data in both, magnitude and phase. So, the real part will modify the phase and the imaginary part will modify module. In order to eliminate the hook effect completely the Td compensation should be complex and frequency dependant. As the researches continued it was found that a mathematical expression would eliminate the impedance estimation error caused by the parasitic capacitance. So the hook effect will be completely removed.

4.3.3 Correction Function

As discussed in the previous section, the mathematical expressions proved that by eliminating the stray capacitance will completely remove the hook effect from the EBI measurement data. The $F_{corr}(\omega)$ in equation (3) is a logarithmic complex function depending on the natural frequency ($\omega$), $C_{p AR}$ and $Z_{meas}(\omega)$, this when substituted with the Td complex exponential $\exp(j\omega TD)$ and by multiplying the complex EBI spectra $Z_{meas}(\omega)$ produces the final expression for correcting the hook effect which is shown in the equation (4)

$$F_{corr}(\omega) = \frac{\log[1-j\omega C_{stray}Z_{meas}(\omega)]}{j\omega} \tag{3}$$

$$Z_{TUS}(\omega) = Z_{meas}(\omega) \cdot e^{\left(\log(1-Z_{meas}(\omega)) + j\omega C_{p AR}\right)} \tag{4}$$
Next figures show the effect of the correction function over an ideal measurement in parallel with a 300 pF stray capacitance.

**Figure 4.3** Resistance of $Z_{\text{tissue}}(\omega)$ gotten as a result of the product of the measured impedance and the correction function [5]

**Figure 4.4** Reactance of $Z_{\text{tissue}}(\omega)$ gotten as a result of the product of the measured impedance and the correction function [5]
The above figures 4.7 shows that the reactance spectrum for the original $Z_{\text{tissue}}(\omega)$ and the $Z_{\text{tissue}}(\omega)$ is also obtained from using the equation (4). Thus in the figure 4.3.5 it is possible to see that both the spectra are corrected at all frequencies. Thus we come to know that the hook effect can be removed with the help of the correction function.

### 4.3.4 Implementing the Correction Function

As discussed in the earlier topics we have found that the best way to correct the hook effect from the measured EBI data is the correction function. So we have used this method in our software suite to compensate the hook effect.

```matlab
function [S]=Corrector_def()
load('MATLAB.mat')
Person_1(:,3,:)=-Person_1(:,3,:);
S=size(Person_1);

Person_1_Corr=zeros(S);
Person_1_Corr(:,1,:)=Person_1(:,1,:);
NM=S(3);

CparVector=zeros(NM,1);
Data=Person_1(:,1,:);
ZMatrix=(Data(:,2) + 1i*Data(:,3));
ZMatrix_real=(Data(:,2));
ZMatrix_imag=(Data(:,3));
F=Data(:,1);
[Cpar,d]=Cestimation_def(ZMatrix,F);

d
if(d>0)
CparVector(1)=Cpar;
ZMatrixCorr=CorrFunction_def(ZMatrix,Cpar,F);
Person_1_Corr(:,2,1)=real(ZMatrixCorr);
Person_1_Corr(:,3,1)=imag(ZMatrixCorr);
else
disp('Correction do not apply')
end
end
```

**Code 4.3.** block of code explains the corrector function
This block of code is used for correcting the EBI raw data. We also have two more functions which are used in the main function for Cpar estimation and as a corrector function. The below codes gives the structure in a way it was followed.

```matlab
function ZMatrixCorr=CorrFunction_def(ZMatrix,Cpar,F)

    w=F*2*pi;
    Fcorr=-log(1-li*w.*Cpar.*ZMatrix);
    ZMatrixCorr=ZMatrix.*exp(Fcorr);
```

### 4.4 Visualization

Both, raw and the processed data are plotted against each other. This gives a better understanding to the user about the EBI data.

By default the impedance features Resistance, Reactance and Cole Plot are visualized, but the software offer the option of visualizing the features of the admittance plane that is Conductance, Suspectance and Admittance.

The measurement visualized can be chosen using a slider and the measurement file name and the Cpar value estimated are also visualized.
Fig 4.4.1. Impedance plots of both raw and corrected data.

Fig 4.4.2. Admittance plots of both raw and corrected data.
ztemp = Person_1(:,2,:) + Person_1(:,3,:)*(sqrt(-1));
Z1 = cat(2, Person_1(:,1,:), ztemp);
R1 = Person_1(:,1:2,:);
X1 = Person_1(:,1:2:3,:);
ztemp = 1/Z1;
ztemp = ztemp(:,2,:);
Y1 = cat(2, Person_1(:,1,:), ztemp);
ztemp = Y1(:,2,:);
ztemp = real(ztemp);
G1 = cat(2, Person_1(:,1,:), ztemp);
ztemp = Y1(:,2,:);
ztemp = imag(ztemp);
B1 = cat(2, Person_1(:,1,:), ztemp);

%Resistance
axes(handles.Resistance_axes)
semilogx(R1(:,1,FileNum), R1(:,2,FileNum), 'LineWidth', 3)
title('Resistance')
xlabel('F(KHz)')
ylabel('R(\Omega)')
xlim([5e3, 6e5]);
grid on
hold on

%Reactance
axes(handles.axes2)
semilogx(X1(:,1,FileNum), X1(:,2,FileNum), 'LineWidth', 3)
title('Reactance')
xlim([5e3, 6e5]);
xlabel('F(KHz)')
ylabel('-X(\Omega)')
hold on
grid on
Cole
axes(handles.axes3)
plot(R1(:,2,FileNum),X1(:,2,FileNum),'LineWidth',3)
xlabel('R(\Omega)')
ylabel('-X(\Omega)')
title('Cole')
hold on
grid on

% Conductance
% figure (4)
axes(handles.Resistance_axes)
cla
cla reset
cla(handles.Resistance_axes)
%cla(ax,'reset')
semilogx(G1(:,1,FileNum),G1(:,2,FileNum),'LineWidth',3)
title('Conductance')
xlabel('F(KHz)')
ylabel('G(\Omega)')
xlim([5e3,6e5]);
hold on
grid on

% Susceptance
% figure (5)
axes(handles.axes2)
cla
cla reset
cla(handles.axes2)
%cla(ax,'reset')
semilogx(B1(:,1,FileNum),B1(:,2,FileNum),'LineWidth',3)
title('Susceptance')
xlabel('F(KHz)')
ylabel('B(\Omega)')
xlim([5e3,6e5]);
hold on
grid on

% Admissance
% figure (6)
axes(handles.axes3)
cla
cla reset
cla(handles.axes3)
%cla(ax,'reset')
plot(G1(:,2,FileNum),B1(:,2,FileNum),'LineWidth',3)
title('Admissance')
xlabel('G(S)')
ylabel('B(S)')
hold on
grid on

Code box 4.4. block of codes explains the visualization.
This block of coding is given for plotting the raw data in six different graphic plots and placing the graphic plots in their specific axes.

There are six plots in the first frame of the software suite. Resistance, Reactance and Cole plot.

Resistance, the first plot of the main panel is plotted against frequency. The resistance plot of the raw data and the corrected data are plotted against each other. The next is the reactance plot which is plotted against the Frequency. And the last plot is the Cole plot which is an impedance plot that is Reactance Vs Resistance.

The next sets of graphical plots are executed when the admittance pushbutton in the main panel of the suite is executed on a single mouse click. Here the admittance, Conductance and Susceptance are plotted. The raw data of admittance is plotted against the corrected data of the admittance and the same happens with the conductance and Susceptance. These graphical plots play a major role in explaining the results of the corrected data obtained by the software application.

4.5. Storage

The processed data in .mat file extension is converted back to .mfu file extension in such a way that make sure the file can be used back in BIOIMP software in further analysis. The data in third dimensional matix is converted back to .mfu file back to its original format.

```
function ConvertMatToMfu_Callback(hObject, eventdata, handles)
% hObject    handle to ConvertMatToMfu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles   structure

FolderName = uigetdir('C:\Users\Vineel Kumar\Desktop\latestversion123\latestver123\september');
% cd(w1)
% Person_1= read_mfu(FolderName);
% Mean_Person_1= mean(Person_1,3);
% Counter1=1;
% salida=1;

FileContent = dir(fullfile(FolderName,'*.mfu'));
NumberFiles= length(FileContent)
ZDataALL = zeros (256,3,NumberFiles);
Array = zeros (256,3,NumberFiles);
format long %To get all kind of detail in data.
%WD = cd;
% WD = cd ('C:\Users\Vineel Kumar\Desktop\latestversion123\latestver123')

p=load('Z_person_corr.mat')
Z=p.ZPerson_1_corr;

nm=360;
ZWritten=zeros(256,3,nm);
for n=1:nm
ZWritten=Z;
ZWritten(:,1,n)=Z(:,1,n);
```
ZWriten(:,2,n)=(real(Z(:,n)));  
ZWriten(:,3,n)=(imag(Z(:,n)));  
end

n_counter=0;

for n_counter=1:nm

cd ('C:\Users\Vineel Kumar\Desktop\latestversion123\latestver123\september')
fid = fopen(FileContent(n_counter).name);

%tline = fgets(fid);
str = num2str(n_counter)
fileNameMfu= [strcat('Correct_',str,'.mfu')];

for j = 1:13
tline = fgets(fid);
dlmwrite(fileNameMfu,tline ,-’append’,’delimiter’,’’,’precision’,’%.6f’,’newline’,’pc’)
end
fclose(fid);
A=ZWriten(:,3,n_counter);
A(:,3)=-A(:,3);
A(:,1)=A(:,1)/1000;
dlmwrite(fileNameMfu,A,’-append’,’delimiter’,’’,’precision’,’%.3f’,’newline’,’pc’)
end


Code 4.5. .block of codes explains the conversion of .mfu files to .mat file and storage.

This block of codes is used to convert the data back into .mfu extension from .mat extension.
5.1 Discussions

In this work a software system which is used for correcting the capacitive leakage of EBI spectroscopic measurements and visualization of the data has been implemented, on this way a specific software for Bioimpedance spectroscopy analysis is implemented into MATLAB environment, the most spread and powerful signal processing tool.

The Hook effect is considered one of the major artefacts found in the EBI data. Td compensation is the traditional method in which the EBI data is treated to eliminate the hook effect. But only the phase part of the EBI data is corrected using the Td method. Whereas the hook effect influences both phase and module part of the EBI measurement.

Full cancellation of the hook effect requires Td method and the correction factor Fcorr to be complex. So that it corrects both phase and the module part of the EBI data. The stray capacitance of the hook effect is estimated and Fcorr(w) is implemented, Fcorr(w) will compensate the hook effect at all frequencies.

These methods are implemented with the help of the software tool MATLAB. Since the suite might have required specific mathematical processes and functions, the mathematical power of MATLAB was an ensuring option. The software suite loads the raw data, estimates the stray capacitance and corrects the hook effect, thereafter it visualizes the results of the Cole fittings and the impedance and admittance features before and after correction.
6.1 Conclusions

The aim of this thesis was to create a software tool into Matlab environment of functions used for correcting the EBI measurements in such a way the measurements are free from artefacts and thereafter visualize them.

The aim of this thesis is achieved by implementing some functions. The corrector function and the corrector factor are two main functions which were implemented in this software. The visualization handles to visualize both the raw and the corrected data.

The impedance and admittance features of the measurement are plotted. The Cole fitting is plotted and the corrected data is also converted back to the .mfu format, which can be used by the software provided by impedimed for further analysis.

6.2 Future Work

This section includes the possible improvements that could be implemented in order to make the tool more powerful and more complete.

All the measurements are corrected together while executing the Raw/Corr function at a mouse click. For each and every execution the data’s are corrected and saved again, an improvement would be to correct and store all the data the first time and each time the user choose a measurement showing the stored results, or correcting at once only the measurement chosen by the user.

There are also some file path extensions which needs to be changed in every platform the software is executed. The system should be more flexible to these path changes and the visualization part should be speeder. This work of making this software more user friendly and more powerful can be carried out at the future work.
REFERENCES


