Real-time acquisition and analysis of Electro-oculography signals
Real-time acquisition and analysis of Electro-oculography signals

Examensarbete utfört i Medicinsk Teknik vid Tekniska högskolan i Linköping av

Kousik Sarathy Sridharan

LiTH-IMT/MASTER-EX--12/011--SE

Handledare: Christer Ahlström
Statens väg- och transportforskningsinstitut (VTI), Linköping, Sweden

Examinator: Göran Salreud
Professor, IMT, Linköpings universitet, Linköping, Sweden

Linköping, 27 January, 2012
# Abstract


Ex-situ testen utfördes med två volontärer medan in-situ testen utfördes på en individ. Data från in-situ testen visade "goda signalkvalitet" i en 'störtad' miljö, vilket överensstämde med designspecifikationerna. För att motivera vikten av kalibrering, användes två kalibreringsparadigmer under ex-situ tests, där en paradigm registrerar normala blinks medan den andra paradigm registrerar långa blinks och resultatet visade skillnad i inspektion och felhastigheter. OBS: Resultaten från prestandatest vid olika nivåer gav "satisfayerande resultat" och bekräftade användbarheten av systemet för experimentella syften i-situ.
Abstract

Electro-oculography signals are corneo-retinal potentials that carry information pertaining to eye movements. This information can be used to estimate drowsiness level of the subject which could provide interesting insights into research of accident prevention. Of all features present, blink duration has been proved to be an effective measure of drowsiness. The aim of this thesis work is to build a portable system to acquire and analyze electro-oculographic (EOG) signals in real-time. The system contains two sub-systems; a hardware sub-system that consists of the filters, amplifiers, data acquisition card and isolation and the software sub-system that contains the program to acquire and analyze the signal and present the results to the observer. The filters were designed starting with simulation, implementation on the prototype board, culminating in the design of a printed circuit board (PCB) and packaging. The complete software was written in Python™ using several relevant libraries for data processing. A text-based user interface was created to enable easy user interaction. The results are graphically displayed in real-time.

Ex-situ tests were done with two volunteers while in-situ test was done on one subject. The data from the in-situ tests showed "good signal quality" in a "noisy" environment concurring with the design specifications. To motivate the importance of calibration, two calibration paradigms were used during ex-situ tests, where one paradigm records only normal blinks while the other records long blinks and the results showed differences in detection and error rates. The observations made from performance tests at various levels gave "satisfactory results" and proved the usefulness of the system for experimental purposes in-situ.
Acknowledgments

I would like to open the acknowledgements by thanking Prof. Göran Salerud, the examiner for this thesis work, a guide and an inspiration to me all through my stay in Sweden. His frank views, motivational comments through my study and thesis work held me in good stead, especially when the sailing was not all too smooth. I would like to acknowledge the confidence Christer Ahlström, my supervisor, had in me and his candour during our meetings which really put things in perspective. I would also like to thank who was Carina Fors was really helpful, particularly when Christer was not available.

The two personalities without whom I am sure this project would not have seen the light of the day are Malcolm Lattore, Bengt Ragnemalm at IMT, LiU to whom I owe my gratitude. Bengt always had words of wisdom when it came to various aspects of electronics design and Malcolm always had an idea up his sleeve whether it came to design, packaging or safety. I especially enjoyed the work on the NI 6008 board to suit the isolator, where we did a lot of "re-engineering". One lesson I learnt from them is the art of discussion and team-work which I am sure is an invaluable lesson.

The environment at Linköping University has been nothing short of stimulating, feeding my vigour to achieve. I wish to thank the various people whom I met on the course of my work who gave their views on various subjects. I would like to thank Jan Andersson and VTI for giving me the opportunity to work on a thesis that would certainly help them in the near future, for the calm environment and for the exciting test drive!

High-fives to my friends, for spending time listening to me, as I incessantly spoke about the difficulties I was facing and for cheering me up. Also, my heartfelt thanks to those who ‘volunteered’ for the tests settling for nothing more than a carton of juice and some chips! Special thanks to my friend Kamalaker who helped me get around issues in \LaTeX.

Last but not the least, I express my eternal indebtedness to my parents who have always encouraged me to pursue my dreams.
To my parents S R Renganayaki and P Sridharan,
for everything
Contents

1 Introduction 1
  1.1 Motivation ................................................. 1
  1.2 Problem statement ....................................... 2
  1.3 Structure of the work ..................................... 2
  1.4 Organization of the thesis ................................. 2
  1.5 Summary .................................................. 2

2 Background 5
  2.1 Road safety ................................................. 5
  2.2 Sleep stages ............................................... 5
  2.3 Measurement of sleep stages ............................... 6
  2.4 Eye movements, blink behaviour and drowsiness ............ 7
  2.5 Blink features in Electro-oculography (EOG) signal .......... 10
  2.6 Electrode position ......................................... 11
  2.7 Summary .................................................. 11

3 Overview of the system 13
  3.1 Preliminaries ................................................. 13
  3.2 Summary .................................................. 14

4 The hardware sub-system 15
  4.1 Preliminaries ................................................. 15
  4.2 Hardware sub-system overview ............................... 16
  4.3 Filter design ............................................... 16
    4.3.1 Filter parameters .................................... 17
    4.3.2 Filter simulation .................................... 17
    4.3.3 Filter simulation results ............................. 19
    4.3.4 Filter result comparison with Vitaport® ............... 19
    4.3.5 Comparison and modifications ......................... 20
  4.4 Right-leg drive circuit ..................................... 22
  4.5 Power to circuit ............................................ 25
  4.6 Interface card and isolation ................................ 26
  4.7 Amplification and range .................................... 28
  4.8 Selection of Components .................................. 28
  4.9 PCB design, layout and Auto-routing ....................... 28
### 5 The Software sub-system

5.1 Preliminaries .................................................. 33
5.2 Requirements of the programming environment .......... 33
5.3 Libraries required ............................................. 34
5.4 LAAS algorithm ............................................... 35
5.5 Software architecture ......................................... 36
  5.5.1 Overall architecture ..................................... 37
  5.5.2 Plotting routine 1 ....................................... 38
  5.5.3 The `timerEvent` function ............................. 38
  5.5.4 Plotting routine 2 ....................................... 39
  5.5.5 Calibration program ..................................... 39
  5.5.6 Analysis program ....................................... 40
  5.5.7 Saving data ............................................. 40
  5.5.8 Text-based user interface .............................. 41
5.6 Screenshots .................................................. 41
5.7 Summary ..................................................... 43

### 6 Results

6.1 Preliminaries .................................................. 45
6.2 Hardware sub-system tests ..................................... 45
  6.2.1 Channel amplification .................................. 45
  6.2.2 Qualitative analysis of channel cross-talk .......... 45
  6.2.3 Test on isolator ....................................... 46
  6.2.4 Test on NI-6008-DAQ card ............................ 46
  6.2.5 Comparison of hardware sub-system performance in ex-situ and in-situ tests ................... 46
6.3 Software sub-system level tests ............................ 47
  6.3.1 Conformity between MATLAB and Python code outputs .. 47
6.4 Performance tests at system level ........................... 49
  6.4.1 Ex-situ performance test ............................. 49
  6.4.2 In-situ performance test ............................. 50
6.5 Summary ..................................................... 50

### 7 Discussion

7.1 Preliminaries .................................................. 51
7.2 Hardware sub-system ......................................... 51
7.3 Challenges in Python ......................................... 52
7.4 Software sub-system ......................................... 53
7.5 System level .................................................. 54
7.6 Future work ................................................... 56

Bibliography ....................................................... 57
This page is intentionally left blank
Chapter 1

Introduction

1.1 Motivation

Vehicle accidents are one of the major problems plaguing the world and a major cause for death leading to increased risk in road transportation. Data shows that it is the nineteenth major cause of mortality and there is a definite need to address this problem [1]. Elaborate understandings of the pre-accident and post-accident scenarios are necessary for design of counter-measures [2, 3]. In the pre-accident scenario, among causes like alcohol or drug intake, lack of skill, aggression, driver sleepiness is also marked to be an important cause for fatal crashes. A recent study even went on to show that driver sleepiness is even a bigger cause for road accidents than alcohol and a "dramatic" interconnection between the two [2].

A review of the data in 1999 on accidents showed that nearly 30% of the accidents are related to drowsiness in United Kingdom [4]. The problem though is to measure and detect driver drowsiness and to quantize it. In order to measure the sleepiness of the driver, among many methods, blink behaviour has been shown to be sensitive to sleepiness [5]. Eye blink duration, especially has been shown to have a high correlation with the alertness of the subject [5, 6].

Electro-oculography (EOG) signals are corneo-retinal potentials that carry information pertaining to eye movements. Although researchers have made foray into the analysis of EOG signals for instance, in relation to sleepiness levels, most of the research is still offline [7] i.e., collection of data such as EOG and video is done and analysis is performed afterwards. This has given valuable insights into the use of EOG signals to rate driver sleepiness levels albeit not yet a viable system for real-time analysis, which would be of good use in research pertaining to driver drowsiness. There is a pertinent need for a system for real-time analysis of the electro-oculogram (EOG) signals, especially the parameter of blink duration, for a better understanding of the relationship and for use as a tool during simulations. Measurement of the electro-oculography (EOG) signals will be explained in detail in section 2.4.
1.2 Problem statement

The aim of the thesis work is to develop a transparent open-ended platform for detection of electro-oculogram (EOG) signals and to detect alertness levels. The software is to be programmed with Python\textsuperscript{TM}\cite{8} and shall be able to acquire, process, analyze signals and visualize acquired data and the results in "real time". The system will have to a portable, work in a "noisy" environment and shall have an easy-to-use interface.

The first part of the work would involve building a sensor system and signal conditioning unit to obtain the EOG signal, filter the data from noise and convert it into digital format to be read by the software. The second part would comprise the software, to process the data and analyze the signals using a suitable algorithm to assimilate blink characteristics such as blink duration and velocity. A graphical user interface (GUI) should be built to make the use of the equipment easy and intuitive; with focus on visualization of data and results of the analysis. The hardware will have to be tested first for safety norms to enable use on human subjects. It will then have to be deployed in simulation environment for testing its performance.

1.3 Structure of the work

An incremental top-down approach with sub-system architecture is used. The initial plan is to start with the hardware sub-system and work incrementally towards building the complete hardware with all the desired functionalities. Then, the software sub-system will be designed which can process and analyze the acquired signals on a real-time basis.

1.4 Organization of the thesis

The thesis report has been organized as follows: Chapter 2 introduces and builds the background to the thesis work. Chapter 3 formulates the architecture and presents the overview of the system. Chapter 4 describes in detail the hardware sub-system design and results obtained during simulations and sub-system tests. Chapter 5 presents the software sub-system design, programming of the software, environment used and also the algorithm that is used in the analysis. Chapter 6 presents the evaluation of the complete system along with discussion of results. Chapter 7 presents the conclusions derived from the thesis work and scope for future work.

1.5 Summary

One of the major problems affecting the world is road accidents and a major cause for accidents is driver drowsiness. The problem though is in detecting driver drowsiness. One of the main indicators of drowsiness is blink duration which can
be monitored using electro-oculography (EOG) signals. Hence, an equipment that could measure, analyze and display blink parameters would be valuable in research.
This page is intentionally left blank
Chapter 2

Background

2.1 Road safety

One of the most predominant methods of transport is roadway. Safety of roadway users has always been an avenue of research. The data from World Health Organization (WHO) shows that road traffic injuries account to 25% of all deaths due to injuries. With roadways being the primary mode of transportation such high fatalities are not desirable. But to avert such fatalities an elaborate understanding of the reasons for such accidents is necessary[2].

The National Transportation and Safety Board in USA pointed out in its report in 1999 that sleepiness is one of the dominating factors contributing to road accidents[9]. On studying the post-crash investigations, which contained in-depth interviews, researchers were able to establish that night driving lessened alertness levels and reduced sleep time prior to driving are important "predictors" of an increased accident risk[3].

With the above data it is clear that sleepiness is a major cause and an important predictor to road crashes, many of which are "self-caused"[2]. These arguments shift the focus to the methods to monitor and quantify sleepiness level of an individual. There are various ways in which the sleepiness of an individual can be estimated.

2.2 Sleep stages

Sleep is generally analyzed using an ensemble of instruments called polysomnography (PSG). The ensemble consists of electro-encephalography (EEG) to measure the brain activity, electro-oculography (EOG) to measure and monitor eye movements and an electro-myography (EMG) to measure muscle tone of the subject under supervision to determine the sleep stage[2].

Sleep is broadly divided into REM (rapid eye movement) and NREM (non rapid eye movement). The NREM sleep is divided into four levels with level 1 starting with the onset of sleep to higher levels of sleepiness. The features appear
in blocks. NREM and REM blocks are around 9 minutes long and they repeat depending on the stage of sleep[10, 11].

The definition of sleep stages are made traditionally with respect to the EEG patterns and the important features found in the EEG signal of each sleep stage is given below. The stages of interest here are ‘Stage-W’ and ‘Stage-N1’ since this is the stage where the subject goes from wakefulness to sleep, meaning drowsy state[12]. It must be observed that these classifications are for those stages of sleep. The sleepiness or drowsiness is found between ‘Stage-W’ and ‘Stage-N1’.

<table>
<thead>
<tr>
<th>Sleep stage</th>
<th>Description</th>
<th>Observed EEG pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage W</td>
<td>Wakefullness</td>
<td>Alpha waves (8-13 Hz)</td>
</tr>
<tr>
<td>Stage N1</td>
<td>NREM 1 sleep</td>
<td>Theta waves (4-7 Hz)</td>
</tr>
<tr>
<td>Stage N2</td>
<td>NREM 2 sleep</td>
<td>Sleep spindles (11 to 16 Hz) and K-complexes</td>
</tr>
<tr>
<td>Stage N3</td>
<td>NREM 3 sleep</td>
<td>Slow wave sleep- delta waves (typically 0.5-2 Hz)</td>
</tr>
<tr>
<td>Stage R</td>
<td>REM sleep</td>
<td>Rapid eye movements and low level EEG voltages</td>
</tr>
</tbody>
</table>

Table 2.1: Sleep stages classification and observed features[12]

Each of the stages of sleep is defined by features that might be found in EEG recording. For instance stage one is signalled by the rising of theta activity and the fading of alpha activity. Stage two is characterized by the theta activity and the presence of sleep spindles, which occur due to the interaction between cortex and the thalamus. Stages 3 and 4 contain slow waves and delta waves. The difference between stages is determined by the feature composition of an epoch[2, 10, 13].

The definitions of sleep stages are based upon the EEG data obtained. The data is visually scored by dividing the data into epochs of specific time duration. The visual scoring has some disadvantages. First of which is the accuracy of the score since small variations in frequency cannot be easily spotted along with the fact of it being tedious[2].

The easier approach is to do a spectral analysis on the data using methods like Short-time Fourier transform (STFT) which give a good idea of time, frequency and power of the data in one plot. There are several other spectral analysis tools that might be suitable for analysis.

2.3 Measurement of sleep stages

The definitions of sleep stages have been clearly addressed. But in the context of the current studies the emphasis is on the transition from stage W to N1 where the subject moves from wakefulness to sleep. The general standard testing methodologies are multiple sleep latency test and maintenance of wakefulness test. In multiple sleep latency test (MSLT) the latency between the start of the sleeping period to the initial signs of sleep is measured and the latency is used as a diagnostic value.
2.4 Eye movements, blink behaviour and drowsiness

The maintenance of wakefulness test is similar to MSLT except that here a subject’s ability to stay awake is measured, rather than a subject’s tendency to fall asleep[2]. Once again, this test is conducted using the ensemble of instruments called a polysomnograph (PSG). The problem with the above tests is the discrete nature of the results. Discreteness is contributed by the fact that the scoring is done in time intervals and hence the continuity in data is lost.

There are other scales of analyzing PSG data to classify sleep stages such as Karolinska Drowsiness Scale(KDS). There are also subjective scales of analysis of sleep stages such as Stanford Sleepiness Scale (SSS), Visual-Analog Scale (VAS) that use a scale of explained parameters where the measurement is generally introspective[14].

There are other ways to measure the onset of sleep such as the standard deviation of lateral position of the vehicle during driving and an ensemble of data such as speed variability, steering position along with lateral position are used[14]. These parameters can be easily measured in simulator tests but can be difficult to measure in road driving due to limitations imposed by sensor capabilities.

One other credible indicator of sleepiness is blink characteristic, especially blink duration and blink velocity. The eye blink duration especially has shown to have a very good correlation with the sleepiness level[5, 14]. There is sufficient data that indicates that blink duration, especially, to be a good indicator of sleepiness, can show a direct link between Karolinska Sleepiness Scale(KSS) and blink duration[5, 14]. Thus blink characteristics when studied in relation with other known indicators of sleep would lead to efficient markers for sleep stage classification.

One method to measure eye movements is by using electrodes placed at specific positions near the eye to detect the EOG signal directly. The placement of electrodes are explained in the section 2.6.

The other method to detect eye movements is using a camera-based technique, where the eye is tracked using the images in the camera and the eye movements, percentage of eye closure and other information is obtained. Some of the prevalent systems that are available in the market are TOBII™, SmartEye™, Seeing machines™etc. This method, though non-contact, could have some complications due to changes in the facial features and movement of face away from cameras. Although there are counter-measures to some of the complications, there is still a need for improvement.

2.4 Eye movements, blink behaviour and drowsiness

Blink characteristics can be measured by monitoring the eye movements. Any system that is capable of monitoring eye movements is called electro-oculograph(EOG). It could be non-contact, which is a camera-based technique or the method in which electrodes are used to measure the corneo-retinal potential directly. EOG signals arise due to the corneo-retinal potential i.e., the potential difference between the cornea and the retina. The cornea rests at a higher potential than the retina which gives rise to an electric dipole[15].
A simple representation of the EOG signal is given in figure 2.1. Firstly, the eye ball colouring pattern should be observed. The red colour denotes the higher potential and the blue colour denotes the lower potential. This illustrates the electric dipole that is present in the eye ball. In this specific case, as pointed out in the figure caption, this is a pictorial representation of a horizontal EOG, meaning the system used to track the horizontal movement of the eye[15, 16].

The electrodes are generally placed on the side of the right and left eyes, called outer canthi. When the eye moves to the right, as shown in figure 2.1, there is a dipole that moves and a differential potential is created between the two electrodes. This signal when processed will lead to the signal shown on the top. Similarly, the signal creates a differential when the eye moves to the left, but is inverted this time and hence is visible through the negative shift in the signal[15]. A vertical EOG is recorded using electrodes above and below the eye and show the movement of the eye in the vertical axis.

The other interesting observation that can be made with the above setup is that of the angles and the amplitudes. The amplitude of the top illustration seems higher than the absolute value of the amplitude on the lower illustration since the eye moves right by 30° in the first case and moves left by 15° to the left in the second[15].

Typically signal magnitude to angular displacement ranges from 5-20 µV/°. Thus the difference in amplitudes can be understood. The achievable accuracy is 2° (absolute) and a maximum rotation of 70° is possible in the human eye. With the above data, 0.4-5 mV is an approximate range of the signal which is used for most practical purposes[15, 16].

There are principally four types of features namely, saccades, fixations, smooth pursuit and blinks. Saccades are quick movements that enable the eye to focus on various regions of interest in the viewing field. The angular movement is generally approximately 20° and the duration is around 10-100 ms, where the time is subject to the angular movement. The second type of movement is called fixation. As the name denotes, these are temporary states at which the eye remains for a specific amount of time. Essentially, these appear as level shifts on an EOG depending
upon the filter characteristics. They last from 0.2 - 1 s on an average. Generally fixations and saccades occur together, i.e., saccades are movements between fixation points. The third type of eye movements is blinks, which are further classified into three types[16, 17]. Smooth pursuit is the feature when the eye smoothly (non-discretely) pursues an object in motion or to compensate for head movements.

The characteristic in the EOG signal, that is pertinent to this particular project are blinks. Blinks are classified mainly of three types according to the need for the blink.

- **Blink reflex**- as the name suggests, is a reflex action which is to protect the eye from any harmful stimulus
- **Involuntary blink** - which is to nourish the cornea and to moisturize the eye continuously.
- **Voluntary blink** - that is induced in case of a conscious decision to blink, generally occurs in case of drowsiness and this might vary in duration and frequency[16, 17].

Of the types of blinks, involuntary and voluntary blinks contain information about sleepiness. The average blink rate for a normal subject is around 12-19 blinks per minute[16] though from a purely physiological perspective only 4-5 blinks are needed[17] and lasts between 100-400 ms[16]. The blink frequency drops when the subject performs a task that demands higher attention. But in case of this thesis work, the longer blinks which occur in case of drowsiness are taken to be more than one second for filter design specifications.

Figure 2.2 gives the relationship between blink amplitude and velocity for both phases computed from *orbicularis oculi* muscle activity (OO-EMG)(18). The principal finding suggests that the closing phase is faster than the opening phase. Also, the maximum velocity during each phase has a linear relationship with the amplitude[18].

To obtain a practical insight using the above information some of the following cases might be considered. When a driver in a car looks at the dashboard, say to check the speed, there is a downward shift on the vertical EOG.

Similarly, when the driver checks the centre rear-view mirror, the
driver will have to look to the top right, which would mean a positive amplitude in the vertical EOG since the driver is looking up and a positive amplitude in the horizontal EOG since the driver looks to the left.

This technique has some advantages and some disadvantages from the perspective of this work. Corneo-retinal potentials are not fixed and it might vary depending upon fatigue, incidence of light upon the eyes and various other such factors, both internal and external. In normal cases where EOG is obtained for diagnostics, this might create problems and might necessitate multiple calibrations during recording. But, since this work aims to evaluate the alertness level of a subject, this quality which is generally considered a drawback will be advantageous[15].

One of the disadvantages of this type of recording is the interference from the facial muscular potentials. Also, there might be some discomfort due to the electrodes on the face which might also cause some signal distortions. There is also a problem of DC drift on the signal which might be caused due to the voluntary movement of facial muscles[15]. This particular problem will be handled in the subsequent sections.

2.5 Blink features in Electro-oculography (EOG) signal

Blink is the feature of interest in the present scenario as explained in section 2.4. Hence the principal aim of the feature detection program is to detect the blinks. The LAAS algorithm [7, 19] is used here in this system to detect blinks.

The LAAS algorithm efficiently uses the velocity or the time derivative of the EOG signal. Shown in figure 2.3 is a representation of the blink characteristics and detection mechanism.

- **Upper and lower threshold** - thresholds set on the basis of velocity or differential of the EOG signal
- **Baseline** - the part of the signal before the feature starts where the signal and its differential are relatively flat; determined by the upper and lower thresholds
- **Closing phase** - initial phase of eye blink when the eyelid closes; start of which is defined by the EOG derivative crossing above the upper threshold ("debh") and end of which is defined by EOG derivative going below upper threshold ("finh")
- **Opening phase** - succeeding phase of eye blink when the eyelid opens; start of which is defined by the EOG derivative crossing below the lower threshold ("debb") and end of which is defined by EOG derivative going above lower threshold ("finb")
- **Closing time** - time taken for the closing phase; defined by the time elapsed between "debh" and "finh"
2.6 Electrode position

Figure 2.3 shows the electrode positions in a pictorial form. The red lines with diamond ends indicate the electrodes that pick up horizontal eye movements and are placed about 1 cm from the outer canthi to left of the left eye and right of the right eye. The vertical EOG is picked up by using electrodes which are placed directly above and below the eye. Only one of the vertical channels is indicated here. Generally two are used for each eye. This is shown using the black line with rounded end. The neutral electrode or the electrode that carries the right leg drive signal (explained in section 4.4) which feeds the inverted noise signal back into the subject[20, 21], is placed in the centre of the forehead. This is denoted by a blue arrow with the arrow tip marking the position of the electrode.

2.7 Summary

With the established fact that driver drowsiness being major cause for accidents, sleep stage classification is necessary. The transition between 'stage of wakefulness' to 'stage N1' is of high importance. Eye blink duration is a key indicator of
drowsiness and hence literature was surveyed to define its features. The direct contact EOG has been used in this work to obtain the signals for processing.
Chapter 3

Overview of the system

Chapter 3 will present the overview of the system with emphasis on the sub-system architecture used to organize this thesis work.

3.1 Preliminaries

A data acquisition and processing system basically consists of a hardware sub-system and a software sub-system. Show in figure 3.1, is the basic overview of the system. There are two types of signals in the system viz., power signal and data signal. The power signal is needed for the hardware to work and the data signal contains the data that is to be acquired and processed.

For better understanding, the system has been divided into a hardware sub-system and a software sub-system. The hardware sub-system consists of the electrodes, filters and amplifiers, data acquisition card and the isolation unit. In turn, the software sub-system consists of the software and computer on which the software runs. The importance and significance of each of the blocks is explained below.

The data signal is acquired through a pair of electrodes and then fed into the filters and amplifiers where the signal is conditioned as per the requirements. The conditioned signal is given to the data acquisition card, which in this case is NI 6008®[22], a USB-powered interface card from National Instruments™, capable of collecting up to eight analog inputs at the same time.

The purpose of the data acquisition card is to enable the analog to digital conversion. Another important use of the data acquisition card is to derive power to drive the filters and amplifiers. It is powered through the Universal Serial Bus (USB) cable and also has an on-board internal reference which is useful to drive the filters and amplifiers.

Isolation is necessary since the instrument is designed to have direct contact with a human subject and to satisfy the safety requirements there is a need for an isolation to prevent passage for excess current. The hardware is connected to a computer which runs the software for acquisition, processing and displaying the signal.
3.2 Summary

The whole system has been divided into the hardware and software subsystem. Hardware sub-system contains the electronic circuit for signal acquisition and conversion. Software sub-system contains the code for acquisition of data, processing, analysis and visualization of results.
Chapter 4

The hardware sub-system

4.1 Preliminaries

This chapter will present the process of the design and realization of the hardware sub-system. Augmenting the last chapter on the overview of the complete system, this chapter will discuss the hardware sub-system, its components, materials used and design methods. The hardware sub-system consists of amplifiers and filters. The function of the hardware sub-system is to take the EOG signal as input, condition it and convert the analog to digital signal and transfer it to the software for processing and display.

Figure 4.1: Block diagram of the hardware sub-system showing the data flow

![Block diagram of the hardware sub-system](image)

Figure 4.1 shows the data signal flow through the desired hardware sub-system from the electrodes to the computer. 'Fc' denotes to the cut-off frequency. The power signal is not considered here.

The desired hardware sub-system consists of the electrodes that pick-up EOG signal. The difference amplifier reduces common-mode noise in the signal. High-pass filter reduces the voltage drift and a low-pass filter band-limits the output signal. The result of these operations results in a conditioned signal which is fed into the data acquisition card. The signal is taken into the computer through an electrical isolation box via USB cable.
4.2 Hardware sub-system overview

A picture of the design is essential to understand the various parameters involved in the design and to show the interconnections and inter-operability between the different parts of the circuit. In figure 4.2, the emphasis is on the power connections.

The inputs are fed into the instrumentation amplifier and are processed using the filters designed. The details of filter design are explained in section 4.3. The common-mode input is taken from the instrumentation amplifier and fed through some filters to obtain the right-leg drive signal and is fed back into the body through the reference electrode which is explained in section 4.4.

The circuit as said before is powered from the interface card and all the power connections are drawn directly from the interface card. The +2.5 V signal is buffered and fed into the circuit. Specifics regarding the design are explained in the relevant sub-sections.

4.3 Filter design

Filters are an integral part of any data acquisition system. They are used to condition the signal with the features to be recorded in mind. In the case of EOG signals, as explained before the main features that are eye blinks, eye movements, saccades and fixation. But, for the current scenario, the features of interest are
4.3 Filter design

eye blinks and eye movements. This section relates to the design of the high-pass, low-pass and amplifier block in figure 4.2.

4.3.1 Filter parameters

Reviewing the literature, the estimate of the expected input voltage at the electrodes ranges between 50 - 3500 µV, with approximately 5-20 µV change per degree of eyeball movement. The frequency range was decided to be in the range 0.1 to 30 Hz, since the blinks last from 100 ms to more than 1 second and beyond that the eye movements are to be preserved.

Vitaport®, a modular, ambulatory medical recorder system[23] is currently being used at The Swedish National Road and Transport Research Institute (VTI), Linköping, Sweden. The output of the Vitaport® system was also used to qualitatively tune the filters once the initial design was made. Generally the sampling frequency is kept above 512 Hz and the signal is low-pass filtered at a cut-off of 30 Hz. As a thumb rule, the sampling frequency is taken to be at least 10 times the highest expected frequency. In the case of this work, sampling frequency is taken to be more than the prescribed limit so as to ensure good signal acquisition and avoid aliasing.

The base design used was the amplifier board design on The ModularEEG web page[24] that presents an amplifier design for collection of EEG signals. But since the requirements of an EOG amplifier are different there was a need to make changes in the filter design.

4.3.2 Filter simulation

The filters were first simulated and evaluated using a Simulation Program with Integrated Circuit Emphasis (SPICE) tool called 5Spice Analysis® (Version 1.64.0)[25] running on a non-commercial evaluation version. The tool comes with some standard libraries of commonly used Integrated circuits (IC).

A requirement on the operational amplifier was single-supply operation, meaning the operational amplifier was only expected to work on a single supply ranging from 2 V to 12 V on the higher voltage level also generally referred to as +Vcc, in this case was +5 V. The lower level is maintained at ground or 0V which is referred to as -Vcc in literature. The reference voltage is the operating reference level, generally mid-way between +Vcc and -Vcc set at +2.5 V.

![Figure 4.3: Filter-amplifier circuit](image)
The filter design as given in the ModularEEG schematic is shown in figure 4.3. The filter and amplifier circuit can be divided into five blocks, where blocks 1 and 3 are the same and are high-pass filter components. They are both single pole high pass filters with a cut-off of 0.16 Hz\[^{[24]}\]. The purpose of high pass filter is to attenuate any DC drift. DC drifts or DC offsets are a regular occurrence in most of bio-potential recordings due to movement of the measurement electrodes. Even touching some types of electrodes during measurement could cause DC drifts and a high pass filter solves this problem.

This is followed by block 2 which is a simple non-inverting amplifier. The gain of the amplifier is shown in the equation 4.1 and substitutions shown in equation 4.2. Currently the gain is set at approximately 6 V/V.

\[
\text{Gain of block 2} = \frac{R_6 + R_7}{R_6 + R_7} + \frac{R_8}{R_6 + R_7} \quad (4.1)
\]

\[
\text{Gain of block 2} = \frac{[20K + 1K] + 100K}{20K + 1K} = 5.76 \approx 6 \quad (4.2)
\]

The block 4 consists of a third low pass filter with a non-unity gain. The filter has a variable gain \textit{Sallen-key topology} and is of second order. The cut-off chosen was 30 Hz. Block 5 is the feedback path of the filter and is designed to provide the desired gain which is similar to the gain calculation in block 2 and is approximately 13.5 V/V and is of similar design as block 2 which again was a simple non-inverting feedback circuit\[^{[24]}\].

The above circuit was fed in and simulated using 5Spice Analysis\(^\text{R}^\text{®}\) tool. Figure 4.4 shows the simulation design that was used in 5Spice analysis\(^\text{R}^\text{®}\) tool. The input was replaced with an oscillator ('Vs1') whose frequency and voltage can be varied. The operational amplifier used in this case is TLC277 (SUB5) which is a commonly used "Single supply" dual operational amplifier. The total amplification of these filter blocks was 96 V/V and the instrumentation amplifier was designed to have an amplification factor of 12.2 V/V but this is not a part of the simulation but was used to build the electronics to test the designed filters.

![Figure 4.4: 5Spice analysis\(^\text{R}^\text{®}\) - simulation diagram](image-url)
4.3 Filter design

All components including IC’s, supplies resistors and capacitors were assumed to be ideal. The circuit was powered through two batteries, one for the 5 V level and one for the reference +2.5 V level. Two test points 'TPv1' and 'TPv2' were placed at the end of the first and second stage to analyze the filter profiles separately. The 5Spice analysis® tool has an option to perform an AC sweep analysis.

4.3.3 Filter simulation results

The parameters set for the analysis results in figure 4.5 and 4.6 were the following:
- Frequency: 0.00001 to 10000 Hz; Scale: LOG; Steps/decade: 10; Source(s): Vs1: 1.0 V; Phase: 0

The profiles when inspected up-close will reveal the following. The 0 dB cross-over frequency for the block 1 and 2 is at 0.0165 Hz which is effect of the high pass filter. The 0 dB cross-over frequency of the low-pass filter is at 236 Hz with the roll-off starting at around 59 Hz which is conformity with the cut-off frequency used for design.

The filter was built with discrete components on a bread board and the output from the filter was recorded and compared with the Vitaport® system. Only one vertical EOG channel from each of the circuit was used for comparison. A pair of electrodes was placed above and below the right eye and was connected to the Vitaport® system. A pair of electrodes was placed above and below the left eye and was connected to the electronics built with discrete components on a bread-board.

Since the behaviour of both eyes are almost similar and most of the movements are common to both eyes in case of any subject without any serious eye problems this method was used to cross check the outputs of the two systems.

4.3.4 Filter result comparison with Vitaport®

Shown below are a set of data collected simultaneously from the Vitaport® system and the electronics shown
The hardware sub-system

in the circuit above. The reason for comparing the outputs of Vitaport® and the filters designed was to ensure a good output and to make sure most of the features of the EOG signal are preserved. The former of the two graphs is the output from the electronics designed (Figure 4.7) and the latter is from the Vitaport® system (Figure 4.8).

The sampling frequency set on the NI-6008® card used for data acquisition was 1000 samples/second. The raw data was then conditioned using a digital 2nd order low-pass Butterworth filter with a cut-off of 20 Hz and a digital 1st order high-pass Butterworth filter with the cut-off of 0.01 Hz implemented in MATLAB® version 7.7.0 (R2008b).

The settings on the Vitaport® were the standard settings that are generally used for all recordings. The Vitaport® provides customization of filters, ranging of input, amplification, sampling rate and storage rate. The custom montage of settings is given below.

Montage file for Vitaport® system

General:
Name: EOG vertical
Data format: WORD
State: On
Sampling rate: 512 Hz
Storage rate: 512 Hz
Preprocess: Mode: No processing
No parameter
Filters: Lowpass(frequency): 70.1Hz
Highpass(Time constant): 5.000 sec
Nulling of DC-signal: n.a.
Range: Display unit: µV
Amplification: 1000.4
Full scale: 2.4990
Max./Min.: 1249.1 µV / -1249.1 µV
Full scale / Offset: 1498.9 µV / 1249.7 µV

4.3.5 Comparison and modifications

The regions of interest are shown in the dotted ellipses drawn upon the figures 4.7 and 4.8. It can be clearly seen that due to the high pass filter action the eye-movement feature is quickly brought back to baseline. This signal when computed upon by blink detection algorithms could cause considerable false positives for blinks. So there was a need to shift the high pass filter’s cut-off to a lower value. But this is a trade-off. If reducing the cut-off of the high pass filter, it will result in more drift which might cause problems. But action was definitely needed since the output was not preserving features.

So a small change was made to the high pass filters in the circuit. 1 µF capacitors were both replaced by 4.7 µF capacitors and it was first simulated to see how the profile has changed. Figure 4.9 shows the new filter response. On keen observation there is a key change in the response curve. The 0 dB crossover
of the high-pass filter block has moved from 0.0165 Hz to 0.0038 Hz, which might seem to be a marginal change, but led to visible changes in the overall output.

On inspection of the figures 4.10 and 4.11, which are the plots of the data acquired simultaneously from the Vitaport® and the electronics built on breadboard similar to the method explained earlier reveals some improvements as well as some pitfalls. First the improvements, which are shown with dotted boxes. The mixing of movement and blink features could be seen and there seems to be conformity between the outputs.

But on close inspection of the valleys of the output from the electronics seems too smoothed and hence the valleys seem to have disappeared and two contiguous blinks seem to form a double blink. These were tackled by setting higher cut-off frequency for low-pass filter used to process the output before plotting. Also, an attempt was made to see if the high-pass filter cut-off could be further reduced to allow more DC signal.

The next target was to attempt at reducing the high-pass cut-off further and also the low-pass cut-off was increased to 30 Hz from 20 Hz to lessen the smoothing effect of the low-pass filter. To decrease the high-pass filter cut-off the capacitor in series was increased from 4.7 $\mu$F to 10 $\mu$F which resulted in the decrease of the 0 dB crossover frequency to 0.00179 Hz. The profile of the AC sweep analysis response is shown in figure 4.12.

With the data set in figure 4.13 and 4.14 and many more that were acquired to check the output of the filter design it was then decided that the outputs were satisfactory and to carry on with the design of the Printed Circuit Board (PCB)
4.4 Right-leg drive circuit

The right-leg drive circuit is an integral part of most of physiological monitoring or data acquisition instruments such as ECG, EEG, EOG, EMG and EEG recorders. The name has a historical background\cite{21} to it, rather than a functional one. It was primarily used in ECG recordings to feed back the common-mode noise signal back into the subject’s body to suppress the common-mode noise arising from the subject\cite{20, 26}. This relates to the "Filter to obtain right leg drive signal" block in figure 4.2.

Although the governing principle remains the same, wherein the common-mode signal is received and then high-pass filtered so that the common-mode signal is inverted and fed back to the subject, except that the feedback is not into the right leg but an area which can be used as reference point. In case of an EOG signal it is the centre-top part of the forehead generally as explained in section
2.6. The design of the right-leg drive has not been altered from the ModularEEG design[24]. The circuit is show below.

The figure 4.15, shows two blocks 1 and 2. Block 1 represents a voltage follower that is attached to the output of every common-mode output from every instrumentation amplifier to buffer the Right leg drive (RLD) signal. The input to the buffer amplifier is shown in 4.14 in the box drawn which shows the common-mode signal being extracted from the input.
Block 2 represents the filter that is designed to ensure the filtering of any DC signal and preventing it from being amplified. Figure 4.17 can be used to understand the function of the circuit better. The response curve shows the maximum magnitude to be around -6 V/V, meaning the system filters the signal and also de-amplifies the signal and as such the amplitude of the signal fed back into the body is very low. The right leg drive is also called the active ground since it actively changes the reference level as the signal noise differs.

For an understanding of the right-leg drive circuit, figure 4.18 could be a good tool. The channels have the same settings. The top channel shows the output \( RLD_1 \) (in figure 4.14) and the bottom shows the \( RLD\_signal \) (refer figure 4.13). Seen here is the noise due to powerline interference which is the noise that is always encountered when designing electronics to obtain physiological signals.

It can be seen (figure 4.16) that the right leg drive signal is of the same shape as that of the input but is inverted. This means that as the shape of the noise changes, say, and inverted signal will be fed back into the skin which can provide a lot of noise suppression which in most cases is the difference between a good signal and a noise-corrupted signal. Thus a right leg drive signal is an integral part of the
4.5 Power to circuit

The power to the circuit comes from the NI 6008® interface card. The three power connections made from the interface card are ground, +5 V and +2.5 V. The maximum current that the +5 V and +2.5 V terminal can source is 200 mA and 1 mA respectively [22]. Since the +2.5 V terminal output is not sufficient to drive the electronics designed, the +2.5 V output is fed into a buffer that is powered with +5 V and ground signals, which means the +2.5 V is also actually sourced from the +5 V terminal. The snapshot shown below (figure 4.17) is the buffer amplifier that is used to feed in the +2.5 V signal from the interface card and buffer it.

The +5 V and ground signals are used directly though some conditioning is done on the board to reduce any noise picked up between components. The figure 4.20 shows a few bypass capacitors between power signals such as +5 V and +2.5 V and ground. Such setups are created for each of the integrated chip so that each of the chip has a separate by-passed power supply.
line in order to reduce the interference between different components.

The labels ‘U3’ and ‘U11’ denote the integrated chips for which the capacitors are placed and they are both by-passed by 100 nF capacitors. Such setups are placed near every integrated as explained before to ensure lesser interference between the components and also to suppress the power-line noise.

4.6 Interface card and isolation

With the filters ready there were two inter-related issues to be sorted out. The isolation, which is an integral part of any medical equipment and the interface card which is integral to the data conversion and acquisition operation.

The isolation selected for use was Single port USB isolator of the UH401 series from B&B electronics[27]. This isolation is built with ADuM4160-Full/Low speed 5 kV USB digital isolator and ADuM6400-Quad channel isolators with integrated DC-DC converters. On inspection of the data sheets it can be found that the Analog devices advised circuitry that is used in this isolation, can only take up a load up till 100mA which is the limit to what it can source through the USB.

When the NI 6008®, which is a 12-bit, 10 kilo samples per second low-cost multifunction DAQ[22] was connected to the computer through the isolator the interface card did not work. On inspection of the specifications of the isolation and the interface card it was clear that it could be due to the ability of the isolation to source the interface card. The idea that was to be used to circumvent this problem was by using two isolation boxes in parallel so that they can both supply a higher current.

But it could be a case where the NI 6008® might consume more than, say, 200 mA at start-up at least. This meant that the current consumption of the interface card had to be measured when it is connected via the isolation and directly.

For this purpose, interface card box was opened up and modified. A fuse near
the USB connector was removed and a 1 Ω resistance was placed in its position so that the current through it can be measured. A general way of measuring the current flow through would be to measure the voltage drop across the resistor and divide it by the resistance and the general way to measure the voltage is to use an oscilloscope.

But a veiled problem in this approach, which is not apparent, is that neither of the ends that we are measuring is held at ground (0 V) which means using a single oscilloscope probe to measure the voltage drop would end up short circuiting the supply and the grounds since one of the terminals of the oscilloscope probe is always held at ground.

The way to circumvent this problem is to use two of the channels; measure the voltage levels at both ends of the resistor and then subtract them to find the voltage drop. In this case since the resistance is 1 Ω, the voltage drop gives the current flow through the resistance directly. The interface card was found to consume more than 215 mA at start-up, at least.

The only alternative was to check the components that were said to be consuming the power and ‘kill’ them. The components on the NI 6008® were all surface mount devices (SMD), which meant the component name, cannot be read directly off the top of it, but has to be looked up in SMD codes to find the actual component serial number.

It so happened that upon inspection, there were already a few probables turning up. Step-up DC-DC converter with 350 mA peak current, DMOS 250 mA low-dropout regulator was one. Also parts of the circuit were the Analog to Digital Converter (ADC), Digital to Analog Converters (DAC) and some operational amplifiers. The NI 6008® board was then inspected closely to understand the circuitry involved, so that the power lines to the integrated circuits (IC’s) mentioned above, can be cut-off so that the current consumption of the NI 6008® be reduced.

The components that were powered-down were DC-DC converter and its induction coil, DC regulator and digital to analog converters. With this the power consumption was reduced to far less than its previous level. It was found during a close inspection that the +5 V power output present on the board is directly from the USB cable from the computer after some conditioning through some diodes which meant the voltage on the +5 V pin was around 4.8 V.

Also, the datasheet of NI 6008® specified that the maximum current that can be sunk from the +2.5 V pin was 1mA which was because this was the onboard reference from the Analog to digital converter which was fed directly into the +2.5 V pin on the board. A simple methodology was adopted to solve this issue. A voltage follower was used to buffer the +2.5 V pin which is powered by +5 V pin which can source up to 100 mA. The circuit to circumvent this problem was explained in section 4.5.
4.7 Amplification and range

With the filters designed and the knowledge about the input signal range and the voltage levels that are to be used, the important issue was to set the amplification of the total channel output including that of the filters. The filters totally at the two stages lead to an amplification of 6 V/V and 16 V/V each totalling up to 96 V/V.

The range of operation was decided to be 5 V (0 V to 5 V) and 2.5V being the reference since the power was to be drawn from the NI 6008® board. Since from the literature[15] and initial testing it was found that the maximum possible input signal along with the all-pervasive 50Hz noise was around 2-2.5 mV the amplification required from the channel was determined to be approximately 1000 V/V. Since the filters already give an amplification of 96 V/V the amplification to be set on the instrumentation amplifiers were left unaltered from the ModularEEG circuit to be at 12 V/V[24].

4.8 Selection of Components

With the basic circuit design at hand, components had to be selected. Open for selection were instrumentation amplifier, dual operational-amplifier and a single operational amplifier for buffering +2.5 V supply.

The instrumentation amplifier chosen was INA128 which is a Precision, low power instrumentation amplifier with low offset voltage, drift and input bias current. It also has a high common mode rejection ratio(CMRR) and input protection for voltages up to 40 V. This is a commonly used instrumentation amplifier with tested capabilities and hence was not a tough choice[28].

In the case of selection of dual operational amplifiers, the choice was limited by some conditions such as the output voltage swing or operational range, supply voltage, feasibility of single supply operation and the other usual considerations such as common mode rejection ratio. The choice was OPA2340 which is a Single-supply and rail-to-rail operational amplifier of the trademarked MicroAmplifier™ series from Texas Instruments (TI). These series of amplifiers are known for their small size and also relatively low power consumption[29].

This particular amplifier has a good output voltage swing which ranges to nearly 0.5 V from the upper rail which means, on a supply voltage of 0-5 V, the output can swing from nearly 0 V to 4.5 V. When tested the swing was actually near 4.75 V which is a very good operating range for the desired design[29]. As far as the single supply operation was concerned this integrated chip (IC) satisfied that condition too. The size of the integrated chip (IC) is quite small and was helpful for modular and small designs which also made it a good choice.

4.9 PCB design, layout and Auto-routing

The printed circuit board (PCB) design, layout and auto-routing were done using Computer-Aided design (CAD) software called EDWinXp®-Electronic Engineer-
The software comes with a number of libraries of different components from the discrete electronic components, integrated circuits from various manufacturers, surface mount device (SMD) components to connection pins, contact pads etc. Also available are customized user-libraries designed by engineers who regularly use the software for design purposes and one such user-library was used for this design. The software also has features to assist in design of such libraries and the component contact pads. Since most of the components that were used for this design were standard electronic component packages, there was no use to design any new components for usage.

The board construction methodology can be minimally described in three steps, namely, preparation of the schematic, arrangement of components on the board and routing. Schematic is nothing but a pictorial representation of the desired circuit. This helps in understanding of connections when the connections are made. NETs denote a specific signal that is carried in a wire or many wires. NETs can be configured in a schematic. For instance, a NET called ‘GND’ was used to denote the ground signal all over the circuit since the ground is common to all parts.

Thus a schematic is used to setup the ground work and to make sure the circuit is correctly designed. The desired circuit was to have three channels and hence most of the circuits were mere repetitions but some parts were unique. Also a schematic serves a good purpose once the layout is done to check the layout for any faults.

The schematic shown below is that of one channel of the amplifier. To understand the features of the schematic design the figure 4.22 is used. The figure has various boxes that are numbered and each of them will be described below. The boxes 1 and 2 show some of the NET names that were used. Box 1 shows the name ‘Ch1_in_p’ which means the input given into channel one that feeds into the positive terminal on the instrumentation amplifier. Similarly ‘Ch1_in_n’ means the input to the channel one that feeds into the negative terminal of the instrumentation amplifier on that channel.

Box 3 shows the by-pass capacitors as explained before, are associated to the power supply lines of each of integrated circuits. The assignment of power lines to an integrated chip is done as shown in box 4. Each of the integrated chips has such a construct that shows the power connections for a particular integrated chip. As shown here the operational amplifier ‘U2’ is powered up by connecting the ‘+5V’ NET to pin 8 and by connecting the NET ‘GND’ to pin 4. Thus a schematic serves the purpose of a visual representation of the circuit that is desired to be designed.

Next step is to ‘Pack’ the components to the ‘Board Layout’, which means the components are approved to be a part of the printed circuit board. Once all the components are imported to the ‘board layout’ window, the components can be moved around using the tool called ‘Ratsnest’ which shows the connections using the information from the schematic. With this as a reference the components are moved around so that they the connections can be made in the easiest possible manner. Once this is done, the ‘routing’, which means drawing the wires to connect the components, can be done manually.
But in this case, a tool called ‘Freeroute’ version 1.2.44 [31] which is downloadable (http://freerouting.net) was used. The software uses resources online through the available internet connection. Once the components are arranged, the design in EDWinXP® is exported to a ‘.dsn’ file which can be opened in this software. There were two layers devoted exclusively to power connections such as +5V and Ground. The other connections including +2.5 V power signals were routed on the other layers.

The routing parameters for the software were left at default values except for the parameters called 'Pull right region: 800'. The auto-router output was fed back into the EDWinXP® software and then again re-touched so as to move some wires around to increase the distance between the wires, which was a qualitative assessment. The above information presents a very minimal description about the design of the electronic circuit.

![Figure 4.22: Schematic for one channel - Shown are instrumentation amplifier, lowpass and highpass filters](image)

### 4.10 Packaging

The printed circuit board (PCB) and the data acquisition card NI 6008® were all encased in a hard plastic box that was slightly modified to fit the boards as shown in figure 4.23. Holes were drilled to mount the electrode sockets using the "Tunn plat borren" or the thin plate drill which is designed to cut through hard but thin surfaces. Seven sockets were mounted onto the box. One hole for the USB connection to the NI 6008® board was cut and another small hole was drilled to make the LED on the NI 6008® board visible outside. The complete system along with the software running is shown in figure 4.24.

### 4.11 Summary

The hardware sub-system development started with the design of filters with Vitaport® as a benchmark system. Electrical isolation to the circuit is provided through a single port USB isolator. The data acquisition board was modified to fit the power source available from the isolation unit. The PCB was designed and
suitable components were selected and soldered to the board. The PCB, data acquisition board and the electrode adapters were encased in a hard plastic box.
This page is intentionally left blank
Chapter 5

The Software sub-system

5.1 Preliminaries

This chapter is intended to describe the software sub-system that is the program that enables data acquisition, data processing, analysis and visualization of results. The software was programmed in Python, which is a general purpose high level programming language. Several other add-on packages were used to program the software which will be explained in detail. The algorithm used for the analysis is also explained. The core ideas of the software are blink detection and visualization of signal and blink parameters.

5.2 Requirements of the programming environment

The idea was to build the software on a platform that does not create licensing issues, that ensures easy portability across platforms and it should be open-ended for further developments. While the choice list may extend more than expected ranging from programming languages like Ruby to frameworks like .NET to numerical computation packages like Scilab and Octave.

Not much can be ruled against the numerical packages except for the real-time data acquisition and plotting functionality requirements. Also another major necessity was the ability to use multiple functionalities from multiple platforms wrapped into one code. To explain, the real-time plotting could be available in a framework such as QT framework but it might lack the data acquisition functions which might be available somewhere else.

So these necessities led to decision of using Python which has many ‘wrappers’ that enable cross-platform implementations. ‘Wrappers’ are codes written to enable use of functionalities from other frameworks. Python has a huge number of wrappers for importing lots of libraries. Python became the choice after some scouting. Also, part of the initial suggestion was to start with Python which turned out to be useful and correct.
5.3 Libraries required

Libraries are sets of previously written and sometimes compiled pieces of codes that enable a particular function. To develop software, not all codes can be written anew and thus these libraries can be useful. The libraries necessary for executing this software program are given below.

- **NI DAQmx Base drivers** - a set of drivers that are used to integrate the NI-DAQ cards to programming interfaces like LabVIEW

- **pydaqtools** - a library of functions that can enable control of National Instruments data acquisition devices. It also has an option for buffered signal acquisition

- Numerical Python (NumPy) and Scientific Python (SciPy) - a library of functions mainly to optimize matrix operations and carry out various other basic functions such as creating a series, obtaining the dimensions of an array

- **PyQt4** - contains sets of Python bindings that enables use of Nokia’s QT framework, necessary for setting some of the plotting function parameters such as cursors, colour of plot and timer events

- **PyQwt** - contains Python bindings for using Qwt, which is a C++ based library with emphasis on fast-plotting

- **csv**- meaning comma-separated values- a library of functions useful to save data into a text file as comma-separated values

The above mentioned libraries are those that were used to code the software. It must be noted that some of these libraries interact with each other in the software. For instance, in case of plotting, although the actual plotting is done using PyQwt, the attributes and timing control is done using PyQt4. Some of the important functions used from the above mentioned wrappers are listed with short descriptions below.

- **pydaqtools**
  - **daqfind** - used to find all data acquisition devices connected to the computer
  - **analog_input** - used to acquire analog samples at a given sampling rate from the channels required

- **PyQwt**
  - **Qwt.QwtPlot** - used to configure plots
  - **Qwt.QwtSymbol** - to set symbols with which data is to be plotted

- **PyQt**
  - **QBrush, QSize, QPen** - to configure plotting attributes
5.4 LAAS algorithm

The **LAAS algorithm** is central to the software and forms the basis for the analysis of the acquired and processed data. It is basically a two-step algorithm for blink localization. Initially a calibration routine is run and the average amplitude and velocity of the blinks detected are calculated. A calibration paradigm can be appropriately designed. With the values obtained from the calibration routine the same routine is again applied for analysis, except that instead of using arbitrary values to locate blinks, the values obtained from the calibration routine are used[7, 19].

A simple explanation of the **LAAS algorithm**[19] is provided below.

1. Filter the vertical EOG signal removing frequency content higher than 10Hz to ensure a smooth time derivative

2. Compute the derivative of the EOG signal

3. Apply thresholds to find the 'debh', 'finh', 'debb', 'finb'

4. Validation of normal blink

5. Test if there are any contiguous blinks; if any, combine them

Validation of normal blink is again done based some arbitrary limitations. While the normal blink can be easily detected with the set limitations, it could miss some of the long blinks of higher amplitude. To tackle this, a set of new limitations are imposed in order to detect the longer blinks.

1. If the opening amplitude is less than 0.7*threshold amplitude and if the closing amplitude is less than threshold
   
   (a) It is not a blink

2. If opening amplitude is less than 1.4*threshold amplitude or closing amplitude is less than twice the amplitude threshold and the duration between 50% closure and 50% opening is greater 0.2 seconds
   
   (a) It is a blink
(b) Copy parameters into the blink parameters matrix

3. Else-If opening amplitude is less than a third of the threshold amplitude or closing amplitude is less than a third of the amplitude threshold and the duration between 50% closure and 50% opening is greater 0.5 seconds

   (a) It is a blink
   (b) Copy parameters into the blink parameters matrix

4. Else-if - the detected feature obeys condition in line 1 and if its duration between 50% closure and 50% opening is 4 seconds

   (a) It is a blink
   (b) Copy parameters into the blink parameters matrix

Some of the implications of this algorithm are that all features that obey the arbitrary limitations and are up to 4 seconds long are classified as blinks. This could lead to some false positives which will be discussed later in detail. Merging of contiguous blinks is done using the following rule.

1. If the two blinks are less than 400 ms apart

   (a) If the average of amplitudes at the end of first and start of the second blink amplitudes are close enough and significantly higher than the average of amplitudes of the end of the start and end of the whole blink and the difference is less than 250 µV, they are both starting and ending on a similar level then they must be the same blink
   i. Combine both blinks into one

However there are some arbitrary limitations that have been imposed, which are mostly empirical. Some of those caveats are listed below:

- The closing phase is limited to one-third of the maximum blink duration
- The opening phase is limited to one-fourth of the maximum blink duration
- For a feature to be considered as a normal blink
  - Amplitude of the closing and opening phases are higher than the threshold and lower than the maximum value
  - Duration of blink is lower than 0.2 seconds

5.5 Software architecture

The software has many functions with sub-functions. For an easy understanding, the flowcharts are organized in top-down manner with the first flowchart depicting the overall architecture and the subsequent flowcharts explaining the functions called from the main program.
5.5 Software architecture

Figure 5.1: Flowchart for the flow through complete software

5.5.1 Overall architecture

The overall architecture of the software sub-system is shown in figure 5.1. Once the program is run, the text-based user interface asks the user to enter basic details about the subject and also prompts the user to enter comments, if any.

The program then presents the user with three options:

1. **Calibration program** - choosing this option will run a calibration routine to obtain standard values to be used in the analysis program.

2. **Analysis program** - choosing this option will run the routine which will perform online analysis of signals using the calibration values.

3. **Exit** - choosing this option will terminate the program.

Before calling the respective functions, the system requests inputs regarding acquisition of samples. After these details are entered, the plots and the NI DAQ card channels are initialized. The corresponding functions are then called indirectly by calling the plotting function. The calibration and analysis functions will be explained in the subsequent sections.
5.5.2 Plotting routine 1

The flowchart shown in figure 5.2 is the routine that controls most of the operations in the software. The reason for using the plotting routine as the centre-piece is that it contains a timer event generator which is a functionality that raises an event after a given time has elapsed. The function that is called when a timer event is generated is also a part of this routine. So, these can be used to generate events inside which data acquisition and analysis routines can be run. On perusal of the above flowchart it can be seen that there is no explicit mention about the calls to either calibration or analysis program. These are performed in the 'timerEvent' function which is called every time the time count elapses. The 'timerEvent' function flowchart is given and explained below. The PyQwt and PyQt libraries are used here to initialize the plots.

Figure 5.2: Flowchart for plotting routine 1

5.5.3 The 'timerEvent' function

The content of the 'timerEvent' function is shown in figure 5.3. The 'timerEvent' function is an in-built function call whenever an event is raised from the 'timerEvent' function. In other words, every time an event is raised, whose elapse rate can be manually set, 'timerEvent' function is called. This property is used effectively to run the calibration and the analysis routines.

Immediately it is checked whether the calibration or analysis program is in progress. If it is found that the calibration program is in progress and if it is found that the calibration time has not expired, then the data is acquired and plotted on real-time basis. Once the calibration time expires, the timer is stopped preventing more calls are made to the 'timerEvent' function. Data is collated into an array and sent to the calibration program to obtain values for analysis. The data from the calibration program is also plotted to verify the calibration procedure.

When selecting 'analysis', then online analysis has to be performed. A moving window records a total data of 10 seconds with an overlap of 9.9 seconds. That is, 100 ms data is acquired, appended with the past 9.9 seconds of data and sent to the analysis program. So, in this case, the delay is approximately 100 ms. The
5.5 Software architecture

results are passed back to the plotting routine so that the features can be marked in real-time and parameters of the features can be plotted separately.

5.5.4 Plotting routine 2

In case of online analysis, there is a need to plot some parameters of the blink, other than the signal itself, overlaid with blink start and end points. The second plot is a separate plot, with its own timer and ‘timerEvent’ function as defined before. The flowchart shown in figure 5.4 shows the routine to initialize the plot.

5.5.5 Calibration program

The LAAS algorithm, as explained before, is initiated with the calibration program. Here there are two principal arbitrary values used.

Velocity of normal blink = 5000 µV/sample
Amplitude of normal blink = 400 µV

Using these numbers the LAAS algorithm is used to detect blinks. In the software, the calibration is done after all the data for calibration is collected for
5.5.6 Analysis program

As stated before, the analysis program is the same as the calibration routine except for the thresholds. Thresholds are set using these arbitrary values obtained from observations. The threshold of velocity is set at 20% of the normal blink value. Amplitude threshold is set at 25% of the normal blink amplitude value. In the software, a moving window is used to collect data and this data window is analyzed for blink occurrence.

5.5.7 Saving data

Data is saved at the end of the analysis using ‘csv’ comma-separated values, a library of functions that enable saving of data into a text file with values separated by commas. The specific commands used have been listed in section 5.3. Three EOG signals, namely horizontal EOG, EOG-vertical right and EOG-vertical left are collected. Each of these is collected on a separate channel. The data from each channel is stored in separate files.

The software has been hard-coded to save the text file in a specific location. There are a total of three files that are saved. The first file contains the comments that have been entered, the second containing the data obtained during calibration on the EOG vertical channel used and the calibration values obtained from the routine and the third containing the data obtained during analysis on all three channels. The files containing information entered and calibration data are named ‘<name_of_subject>_info’ and ‘<name_of_subject>_calib_data’ respectively. The analysis data from the three channels are stored in ‘<name_of_subject>_analysis_data_1’, ‘<name_of_subject>_analysis_data_2’ and ‘<name_of_subject>_analysis_data_3’ respectively. Also, as will be discussed later, another file named <name_of_subject>_exception_time contains the time stamps when exceptions were raised and data was lost.
5.5.8 Text-based user interface

A text-based user interface has been designed to enable the user to interact with the software. The output of the software, the calibration and online analysis are both displayed graphically. The interface first requests for the name of the subject, which in most cases is coded to mask the identity. Later, it will be used to save the data.

Once a decision is made as what is desired to be done, calibration or analysis, the necessary inputs are requested from the user. These inputs are screened to check the limit violations and then are passed on to the corresponding functions to run.

5.6 Screenshots

The Python code can be run from MS-DOS prompt using the keyword ‘python’ in front of the Python file. Figure 5.5 shows the command initially entered and response.

Once the program is started up, it prompts for the name or code assigned to the subject and also any comments that need to be stored. Then a choice can be made as to what mode the program is to be run in as shown in figure 5.6. Generally, the procedure is to first calibrate then with the calibration values run the analysis program. The calibration choice when selected, prompts for the sampling frequency and the time for which it is desired to run the calibration program.

Figures 5.7 show the program response when the ‘Calibrate’ function is chosen. Figure 5.8 shows the plot that appears when the calibration function is running.

In figure 5.8 all three channels show same data for ease of understanding. In reality, top two curves will show right and left vertical EOG and bottom curve will show the horizontal EOG.

Figure 5.9 allows the user to visually inspect the result after the calibration. Red circles denote start of a detected feature and black inverted triangles denote end of a detected feature.
The Software sub-system

Figure 5.7: Calibration function prompts for input

Figure 5.8: Plot showing the data to be used for calibration

Figure 5.9: Plot that appears after the calibration is done

The average velocity and amplitude are displayed for notice as shown in figure 5.10. Status of writing of data to file is displayed. It is worthwhile to note that unit of average velocity is \( mV/sample \) and average amplitude is \( mV \) and compare these figures with section 5.5.5. The software then prompts for the next course of action.

In figure 5.12 all three channels show same data for ease of understanding. In reality, top two curves will show right and left vertical EOG and bottom curve will show the horizontal EOG. Red stars denote start of a feature that is detected and red circles denote end of the detected feature.
5.7 Summary

This chapter presented the basics of **Python**, the advantage of using this language and the libraries necessary to run this software. The architecture of the software
The Software sub-system was explained using flowcharts and the algorithm used to analyze signals was also presented. The various other important features and functions of the software were described. Screenshots were used to show the working of the software.
Chapter 6

Results

6.1 Preliminaries

The product development, as is the norm, is accompanied by testing at various levels and various configurations as detailed in the respective sections. Some tests were also done at the sub-system level and are relevant when testing the system on the whole. This chapter will list some of those results and the testing conditions imposed.

6.2 Hardware sub-system tests

The hardware sub-system was designed and constructed from scratch with testing done at certain levels. Once the PCB was ready and connected to the NI 6008®, testing was done on the channel properties and channel-channel interactions.

6.2.1 Channel amplification

The channel amplification was evaluated using small signal amplitudes of 1 mV, 2 mV and 3 mV from an oscillator, de-amplified using a voltage divider and the output of the channel was checked on the oscilloscope. The signal was a sine wave of given amplitude at 15 Hz. The channels other than the one under inspection were left open. The table of results is shown below in table 6.1.

6.2.2 Qualitative analysis of channel cross-talk

The analysis for assessment for cross-talk was done by feeding the channel under inspection with a sine wave of 15 Hz and amplitude of 1 mV. The other two channels were fed with square wave signals at 5 Hz and amplitude of 3 mV. The output of the channel under inspection was checked for traces of square wave oscillations. The study was repeated twice to ensure fair trial. This was a qualitative analysis and the results are shown in table 6.2.
Table 6.1: Amplification characteristic of the channels

<table>
<thead>
<tr>
<th>S.No</th>
<th>Input mV</th>
<th>Channel #1</th>
<th>Channel #2</th>
<th>Channel #3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gain1 V/V</td>
<td>Gain2 V/V</td>
<td>Gain3 V/V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dB</td>
<td>dB</td>
<td>dB</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1.15</td>
<td>1.15</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1150</td>
<td>1150</td>
<td>1100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>61.2</td>
<td>61.2</td>
<td>60.8</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2.00</td>
<td>1.95</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>975</td>
<td>1016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>59.8</td>
<td>60.1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3.00</td>
<td>3.05</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>1016</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>60.1</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 6.2: Qualitative assessment of cross-talk between channels

<table>
<thead>
<tr>
<th>S.No</th>
<th>Channel #1</th>
<th>Channel #2</th>
<th>Channel #3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No discernable effect seen in output</td>
<td>No discernable effect seen in output</td>
<td>No discernable effect seen in output</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.2.3 Test on isolator

Isolation, as explained earlier, is an integral part of any medical equipment. Since it was a modular unit which was already tested to provide up to 4 kV isolation, there was no need to test the performance of the isolation again. The isolation has been pre-tested and its specifications and certifications have been presented in Appendix C.

6.2.4 Test on NI-6008-DAQ card

The NI 6008® is used both to power the amplifier circuit and for data acquisition. The voltage levels on the power connections on the NI 6008®, viz., +5 V, +2.5 V and ground, were checked for consistency. The voltage levels on the power connections were of satisfactory levels. The acquisition function needed no separate testing.

6.2.5 Comparison of hardware sub-system performance in ex-situ and in-situ tests

The system was evaluated in-situ, i.e., in the simulator which is the application of this work. Visual inspection of the figures 6.1 and 6.2 confirm the equivalence in performance of the hardware sub-system in-situ compared to ex-situ.
6.3 Software sub-system level tests

The software was developed using Python. The MATLAB code of the same algorithm was used as a standard to test the functioning of the Python code. Similar to the hardware sub-system, the software was tested at various levels while being programmed and some of the relevant test results are presented below.

6.3.1 Conformity between MATLAB and Python code outputs

The LAAS algorithm was programmed in Python and was first tested for conformity in offline mode. The conformity was not tested statistically, rather qualitatively. Three sets of calibration and test data sets were obtained using the already tested hardware. The data was first fed into the MATLAB program and then to the Python-offline code. The times taken to run the codes have also been noted for comparison as it is an important factor in software sub-system design. The list of parameters that were checked and the results are given in a table 6.3.

There are some differences in the working of MATLAB and Python, reflected in the table below, one being the difference being speed of operation. One could allude this to the various function calls which in case of MATLAB is different from that of Python. The difference in the numbers is also an issue, which could be attributed to changes made while programming the LAAS algorithm to suit the functions available in Python that might offset the feature detection by a few
<table>
<thead>
<tr>
<th>S.No</th>
<th>No.of data points</th>
<th>Time consumed for program to run</th>
<th>No.of features detected</th>
<th>Start points of features: first 5 data points</th>
<th>Calibration values</th>
<th>Time consumed for program to run</th>
<th>No.of features detected</th>
<th>Start points of features: first 5 data points</th>
<th>Calibration values</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>3.9278, 5.7950, 2.3100, 1.6390</td>
<td>0.2791</td>
<td>0.5827</td>
<td>3.2590, 4.2530, 4.2510, 5.2900, 6.2500</td>
<td>0.9677</td>
<td>0.7194</td>
<td>5.3417</td>
<td>3.1690, 4.2510, 5.2900, 6.2500, 7.2500</td>
<td>0.5827</td>
</tr>
<tr>
<td>#2</td>
<td>4.1120, 2.3210, 1.6390</td>
<td>0.4479</td>
<td>0.74</td>
<td>2.3220, 3.2210, 4.2180, 5.2600, 6.2540</td>
<td>0.9677</td>
<td>0.7194</td>
<td>5.3417</td>
<td>3.1690, 4.2510, 5.2900, 6.2500, 7.2500</td>
<td>0.5827</td>
</tr>
<tr>
<td>#3</td>
<td>4.1120, 2.3210, 1.6390</td>
<td>0.4479</td>
<td>0.74</td>
<td>2.3220, 3.2210, 4.2180, 5.2600, 6.2540</td>
<td>0.9677</td>
<td>0.7194</td>
<td>5.3417</td>
<td>3.1690, 4.2510, 5.2900, 6.2500, 7.2500</td>
<td>0.5827</td>
</tr>
</tbody>
</table>

Table 6.3: Evaluation of conformity between MATLAB and Python codes
6.4 Performance tests at system level

The sub-systems once integrated had to be tested at the system level. The performance tests were done on two different subjects in day time when the subjects were awake. The necessary criteria for good performance are good sensitivity and low error rate. Sensitivity is described by the ratio of true positive or number of detected blinks which are blinks and sum of number of true positives and false negatives or number of missed blinks. Error rate is defined by the ratio of wrong detections to number of true positives. Sensitivity is a measure of how good the system is in detecting blinks. Error rate is a measure of how much the system mistakes a feature for a blink.

6.4.1 Ex-situ performance test

The subjects were told to continue their daily schedule as usual. Once they arrived for tests they were told the procedure in which the test was to be conducted. The test contained two legs. Each leg contained two parts. The first being the calibration which was run for nearly one minute followed by an analysis routine which was run for approximately 12 minutes. The results of the test is shown in table 6.4.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Calibration paradigm</th>
<th>Calibration values</th>
<th>Blinks detected</th>
<th>Blinks missed</th>
<th>Wrong detections</th>
<th>Sensitivity</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average Velocity mV/sample</td>
<td>Average amplitude mV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td>#1</td>
<td>2.3875</td>
<td>0.2372</td>
<td>150</td>
<td>1</td>
<td>61</td>
<td>99.34%</td>
</tr>
<tr>
<td>#1</td>
<td>#2</td>
<td>2.4851</td>
<td>0.3452</td>
<td>146</td>
<td>6</td>
<td>28</td>
<td>96.05%</td>
</tr>
<tr>
<td>Subject</td>
<td>#1</td>
<td>1.4393</td>
<td>0.3352</td>
<td>210</td>
<td>3</td>
<td>72</td>
<td>98.59%</td>
</tr>
<tr>
<td>#2</td>
<td>#2</td>
<td>4.8482</td>
<td>0.5688</td>
<td>208</td>
<td>12</td>
<td>37</td>
<td>94.54%</td>
</tr>
</tbody>
</table>

Table 6.4: Table containing results of ex-situ performance tests

There were essentially two calibration paradigms used. In the first, the candidates were told to look at a specific area denoted by a square of approximately 5 in x 5 in at the eye level and relax. This was to avoid eye movements and to ensure only blinks are registered in the calibration. The second was when the subjects were asked to close and open eyes when asked to, at a quick rate. This was to feed the calibration program with slow eye blinks of higher amplitude. The errors
Results

seem to be systematic since there is a perceptible difference in the errors in case of each of the calibration procedure and hence could be attributed to errors in data handling.

6.4.2 In-situ performance test

Since principal the objective of the work is to build an instrument that could work in the simulation setting, in-situ test was done to evaluate the system performance. The test was done on one subject and consisted of four trials. One trial was done using the first calibration paradigm and three using the second calibration paradigm. Table 6.5 shows the results of the in-situ tests. These results are in congruence with the results and analysis presented in ex-situ performance test.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Calibration paradigm</th>
<th>Average Velocity mV/sample</th>
<th>Average amplitude mV</th>
<th>Blinks detected</th>
<th>Blinks missed</th>
<th>Wrong detections</th>
<th>Sensitivity</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>#1</td>
<td>7.4010</td>
<td>0.6534</td>
<td>182</td>
<td>3</td>
<td>32</td>
<td>98.38%</td>
<td>17.58%</td>
</tr>
<tr>
<td>2.</td>
<td>#2</td>
<td>4.8885</td>
<td>0.8710</td>
<td>30</td>
<td>1</td>
<td>6</td>
<td>96.77%</td>
<td>20%</td>
</tr>
<tr>
<td>3.</td>
<td>#2</td>
<td>3.2565</td>
<td>1.1294</td>
<td>116</td>
<td>8</td>
<td>26</td>
<td>93.55%</td>
<td>22.41%</td>
</tr>
<tr>
<td>4.</td>
<td>#2</td>
<td>3.2241</td>
<td>1.0605</td>
<td>67</td>
<td>4</td>
<td>8</td>
<td>94.37%</td>
<td>11.94%</td>
</tr>
</tbody>
</table>

Table 6.5: Table containing results of in-situ performance tests

6.5 Summary

This chapter summarizes the data acquired through tests from experiments to evaluate the performance of the system. The data acquired are from various levels viz., sub-system and system level. The data presented here will be dissertated and discussed in the next chapter.
Chapter 7

Discussion

7.1 Preliminaries

The initial goals were to develop an open-ended platform which can acquire, process, analyze and visualize EOG signals on a real-time basis and work in a simulator environment. The development process met with several complications and difficulties. The hardware sub-system designed was able to obtain EOG signals and condition them. The software sub-system then acquired the conditioned signal, processed it and displayed results to the user on a real-time basis.

7.2 Hardware sub-system

Selection of the electrodes to be used, which are at the 'frontier' of the equipment, are of prime importance. Silver-silver chloride (Ag-AgCl) are the prescribed electrodes to be used with this system. When electrodes were repeatedly used for testing purposes, it was observed through visual inspection of signals obtained with and without the right leg drive signal, that there was a decrease in signal quality. This is due to the drying up of the gel leading to a change in capacitance. The right-leg drive electrode, which is the neutral had a drastic effect on the signal quality and hence proves the performance and design integrity. But it has to be noted that phase matching is very important in case of the right leg drive signal, which will have an impact on the quality of data.

The incremental approach followed in the design of the hardware sub-system was a good plan since it was easy to track errors and problems. Also, since the design was result-driven with an existing and currently-in-use product as a benchmark, the end-result was good and did not need any bulky changes to be made. Filter could be re-designed, but it would be hard to improve the performance of these filters.

The use of a commercial isolator was useful since it saved work and time, since acquiring certifications for medical use would have been time-consuming, although necessitated alterations to be done to the NI DAQ board. The NI DAQ board
has been stripped down to the minimum power consumption keeping in mind the functions that are to be carried out by the NI DAQ board.

The hardware sub-system was tested as a whole after the fabrication of the PCB. As shown in table 6.1, the amplifier design and experimental values are in close conformity. The actual design estimate of amplification was 1152 and the results from tests show close values. As expected there is also no cross talk between channels as this was also an issue addressed at the design stage. It was also observed that it takes nearly one minute for the filters to settle down once the electrodes are connected.

The signal to noise ratio (SNR) could have been checked at the design stage or at least before packaging which could have been apt pointer to the robustness of the hardware sub-system. But this was realized after the packaging was completed and hence was not done. Visual comparison between signals obtained in-situ and ex-situ was done and the performance was nearly matched, furthering the conformity with the design specifications.

7.3 Challenges in Python

Python being an easy-to-use programming language, is used by many programmers and researchers. Several teams around the world develop libraries using Python and post it on the internet in specific sites under re-use licenses. Thus a huge repository of libraries is available. The problem however is the redundancy and dilemma of choice. For instance, plotting the data in real-time, has at least a few useful alternatives.

Some of the libraries are Matplotlib, wxPython and PyQwt. Each of them have their own pros and cons. Matplotlib is a plotting tool that is designed to be simple to use and very similar to MATLAB plotting, hence the name. The rendered graphic is of good quality and several small tweaks can be made to represent a lot of information in one graphic. The problem though is the time it takes to render a plot. This makes it incapable of being used for plotting in real-time. Qwt is a plotting library with emphasis on fast plotting written in C++. PyQwt is a Python wrapper that enables use of C++ in Python. The plotting is fast enough for use in real-time, but the coding is more complex when compared to Matplotlib.

Documentation could be a problem in case of certain libraries and packages, although it might be easy to find code fragments that might be of use. Installation of packages after a choice is made is also sometimes difficult. One advantage overriding the lack of documentation, in some cases, is the availability of support in appropriate forums and from the authors of the libraries or wrappers. Most packages have their source files and are custom-made for installation to systems running on Linux operating system requiring command line installation.

Another aspect is the co-ordination with other libraries used. For instance, PyQwt takes care of plotting, but other features and attributes are borrowed from another wrapper PyQt4. This is a classic example of libraries interacting with each other in a hassle-free manner. Again, not much documentation can
be found on making them interact but sample programs are always available for a start.

Sample programs are good starting points although might not directly suit the needs. For instance, the available sample plotting routine can only update a single data point every time it is called on only one plot. The need was two plots that can update different number of samples, as programmed. So, the sample program has to be understood and modified to suit the use. Ironically, the most difficult part of the plotting routine was to stop it. Since, the sample program uses a single timer function, the problem at hand entailed two, which needed hands-on manoeuvring.

The MATLAB code for *LAAS algorithm* was already developed by others, but the conversion of the code into Python was not easy. There are several subtle differences between similar MATLAB and Python functions. Some of the exceptions are handled in case of MATLAB, while Python does not have much exception handling which necessitates in some minor alterations in the code to make it executable in Python. The basic features of an object oriented programming language such as looping structures, object handling were similar to other famous contemporaries such as C++.

An issue was the computation time since the analysis and plotting had to be online. *NumPy*, which is a package that enables scientific and numerical operation in Python, handles arrays and large matrices in an efficient manner. This fact is proven with the implementation of online analysis.

Finding the correct function for a particular computation was also important. The MATLAB equivalents of certain functions might have a few alternatives and the function that is the most optimized was chosen. In case of the 'find' function MATLAB, there are a few functions in various libraries that could be useful. But the 'where' function in the Matplotlib library was found to be optimal and easy to use. But finding the optimal function needs delving deeper into the functions and conducting trials.

There are also compatibility issues when writing the code into Python script. One of the problems is handling of 'global' variables. Global variables are handled slowly in Python and hence global variables were avoided to the extent possible. Most of the arrays were passed between functions of the *LAAS algorithm* to avoid use of global variables.

### 7.4 Software sub-system

Python, as a development platform, has abundance of information and to plough through the information is a challenge in itself. There was a need for special libraries for various operations and the main criterion for selection was speed of operation, since the objective was real-time analysis. A certain feature of the *LAAS algorithm* that manages to predict the start and end of the blink feature compensating for features super-imposed on eye movements has not been implemented. This could be a valuable addition though does not affect the feature detection per se.

The objective to write a Python code similar to that of its MATLAB equiva-
lent was not straight forward. Several syntactic bugs were to be fixed before the algorithm worked. The initial evaluation, as shown in table 6.3 was done to ensure conformity between the outputs of the two codes. It is worthwhile to compare the time taken for the MATLAB and Python codes to run the calibration and analysis routine as this was instrumental in the real-time software sub-system design. Although the Python code takes a longer time to run, the speed is sufficient for real-time use.

There is also good conformity between the average velocity and amplitude data between MATLAB and Python codes, further emphasizing the match in the logical performance. It is important to note here that most of functions are common for the calibration and analysis routines in the LAAS algorithm, as explained earlier. So, a match between calibration values will certainly mean a match between the numbers from analysis results. For a better picture the first five start points of features detected have also been added, which adds to the previous conclusions. This was necessary because this algorithm is the backbone of the analysis. The result was a good conformity between the data from the two sets of codes which paved the way for moving to the next level of development.

There are still some glitches in the software which might not seem apparent, but would lead to better working if corrected. The blink plot might sometimes plot the partial time of blinks when the data sent to the algorithm contains a partial blink. This might not cause misinterpretation of results since the actual duration of the blink is plotted immediately afterwards.

An exception handling procedure has been placed to deal with buffer overflow problems. The time of acquisition is used to gauge the buffer level and if the acquisition takes very less time, it is interpreted as buffer nearing full capacity and hence the buffer is reset avoiding the abrupt stoppage of the program. This might cause a data loss of around 100-200 ms and its time of occurrence is noted in a separate text file.

Another critical aspect that might be important is the data saving methodology. It must be noted that the data is saved only after all the testing is complete. In case of premature closure of the program all the data will be lost.

### 7.5 System level

The performance of the hardware in terms of signal quality and working of the LAAS algorithm programmed in Python have been established. The problem however is that the LAAS algorithm detects all features, including eye-movements that are not of interest. This creates a necessity for an appropriate calibration paradigm which is central to the performance of the algorithm. In a bid to find a good calibration paradigm, a few options were tested.

Upon the perusal of results of these tests, two were found to be appropriate to show the impact of the calibration paradigm on analysis. These are the two paradigms that were used for performance tests as shown in table 6.4. The sensitivity of the algorithm as such, is very high and hence detects most of the blinks. But this also leads to a lot of false positives where the software mistakenly inter-
interprets it as blinks, though they are not. Most of these false positives were found to be eye movements such as moving the eyes to look up or down and sometimes muscle movements created due to talking or laughing.

The blink characteristics such as blink velocity and amplitude have high variability among subjects. Therefore, this affects the analysis of data drastically and small changes in thresholds could be made to accommodate concerns. This would be at the risk of worsening the performance, as moving thresholds would mean manoeuvring around the trade-off between false positives and false negatives. There is a firm need for a design of an appropriate calibration paradigm in this regard, in order to reduce the error rate. The performance of both when evaluated, seemed to point to a trade-off. But sufficient data is not available to definitively stake a claim. There are essentially two ends from which the improvements can be made. One from the design of an appropriate calibration procedure and the other involves tweaking the LAAS algorithm, especially threshold determination procedures to eliminate false positives.

Performance tests were performed in the simulator and the signal quality was similar to that of other tests which were made in less noisy environments, the results of which are shown in table 6.5 This reaffirms the claims made in the hardware design.

It can be seen that the Python program would just be able to cater to the needs and thus needed no revision in this case. The LAAS algorithm programmed into Python is not optimized for fast computations and could be imperative if there is a desire to quicken the update rate of the plot. The slow processing can be attributed to the presence of looping over the whole set of data which might not be ideal when used for real-time applications and hence needs some changes.

The plotting, although solves the objectives, it might not be the optimal way to address issues. A good alternative is to plot the data on a real-time basis. When a certain number of samples are obtained for plotting, this bulk of samples can be sent for analysis. This could increase the update rate on the data, but was not attempted due to its complexity.

Plotting is a trade-off between time and quality, where a comparison between Matplotlib and PyQwt was made. Naturally, due to the condition of real-time analysis, PyQwt was chosen. Future developments in such libraries could be used to the advantage, although might not be guaranteed.

There could always be a call for more interaction with the software, since a text-based user interface always leaves a lot to be desired. Although a certain level of testing has been done to ensure usability of the system and testing has been done to evaluate its performance in the simulator, it would be beneficial to perform some more tests in experimental situations to check for any lapses in design or add-ons that could better the usability of the system, either at the hardware or the software sub-system level. Although the system works well, there are certain improvements that would make it more viable and useful. These include algorithm for blink detection and some changes in hardware to make the system more portable.
7.6 Future work

The system currently needs to be connected to a computer which runs the software, both for processing and power. There is a possibility to convert the device into a wearable one, by using on-board processors. However this would definitely require lot of work in building the firmware for the on-board processor. Once this is available, this would create a need for on-board non-volatile storage to make data available for further analysis.

There is also scope for using a computer to do the processing, as in the current scenario, but use wireless data transmission instead of using USB cables. This will certainly improve the portability of the equipment and also would be the best possible isolation. There is sufficient space left in the case to add a few more boards and even batteries to power circuits. The power consumption is rather low at present which could be an advantage.

The LAAS algorithm although efficient in its function, is inefficient in computational terms. In case of real-time analysis, there is a need to optimize the algorithm. A good approach would be to implement the algorithm in C++ and use a Python wrapper to run it in a Python platform. This will drastically improve the performance. There are several exceptions that could have been handled to make the software more robust. Also, a methodology to save data in real-time or after a certain time elapse could be really beneficial considering the cost of each dataset.

The interaction could be made easier using a graphical interface rather than a text-based interface. PyQt4 implemented through Qt Designer could be a very good tool to start with for programming a more interactive user interface. More interaction would certainly ensure dynamic changes to plots and will give a more intuitive feel.
Bibliography


Appendix A

Python Libraries

A.1 Necessary libraries

The software uses a lot of Python libraries and is necessary for the software to run. The library list that is compulsory, its location and size are given below.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Name of library</th>
<th>URL</th>
<th>Purpose</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>NIDAQmx-9.2</td>
<td><a href="http://joule.ni.com/nidu/.../en">http://joule.ni.com/nidu/.../en</a></td>
<td>Drivers for NI 6008 DAQ board</td>
<td>1175 MB</td>
</tr>
<tr>
<td>2.</td>
<td>Python(x,y)- bundle</td>
<td><a href="http://code.google.com/p/pythonxy/">http://code.google.com/p/pythonxy/</a></td>
<td>Provides a plethora of libraries for use in Python</td>
<td>421 MB</td>
</tr>
<tr>
<td>3.</td>
<td>pyDAQtools</td>
<td><a href="http://www.pydaqtools.org/pyDAQTools">http://www.pydaqtools.org/pyDAQTools</a></td>
<td>Provides libraries for controlling NI 6008</td>
<td>1.2 MB</td>
</tr>
</tbody>
</table>

Table A.1: Table of libraries necessary for software operation
A.2 Additional libraries

Additional libraries that could be useful in further development and improvements are given below. These libraries are not necessary, but beneficial in use.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Name of library</th>
<th>URL</th>
<th>Purpose</th>
</tr>
</thead>
</table>

Table A.2: Table of other useful libraries
Appendix B

Connecting the wires

The setup consists of two USB cables with a type A and type B one either end of both cables shown in figure B.2. The figure B.2, shows the images of the two types.

Figure B.1: Parts of the setup

Figure B.2: USB connector types
Source: http://en.wikipedia.org/wiki/USB
1. Case containing amplifier and NI 6008 board. Has a female type B USB adapter
2. Type B connector of wire 1, inserted into the USB adapter in the casing
3. Type A at the other end of wire 1 inserted into the adapter in isolation box
4. Isolation box
5. Type B connector of wire 2 connected to the isolation box
6. Wire to computer USB port

The connectors are designed in such a way that the wires can be connected in only one way. So, there is no possibility of wrong connections.
Appendix C

Specifications of Isolator

C.1  UH401- 1 Port USB to USB Isolator

The isolator is based on the optical isolator integrated circuits (IC’s) from Analog Devices. The device has been tested and certified for medical use needing no further testing of the system for compliance with standards for medical devices. Some of the features of the isolator from the datasheet available at (http://www.bb-europe.com/bb-euro/literature/UH401Series-0310ds.pdf) are listed below.

C.2  Technology specifications

Standard:  USB 1.1, 12 Mbps
Power
Input voltage:  5V DC
Downstream power:  Up to 100 mA with full power upstream connection
Isolation:  4 KV

C.3  Environmental specifications

Operating temperature: -40 to 80°C
MTBF:  1,049,851 hours
MTBF calculation:  Parts count reliability prediction

C.4  Mechanical specifications

Enclosure:  IP30 plastic case
Dimensions:  43.2 x 50.8 x 20.3 mm
C.5 Approvals and certifications

*Emissions:*
FCC Class A, CISPR Class A (EN55022)

*CE:*
EN61000-6-2:2005 (Heavy Industry)
EN61000-4-2:2008 (ESD) +/-8kV Contact, +/-15kV Air
EN61000-4-3:2006 (RI) 10V/m, 80-1000MHz; 3V/m, 1.3 to 2.7 GHz
EN61000-4-4:2004 (EFT Burst) +/-2kV DC ports; +/-1kV signal ports
EN61000-4-5:2005 (Surge) +/-500V DC ports; +/-1kV signal ports
EN61000-4-6:2005 (CI) 10Vrms, 0.15 to 80 MHz
EN61000-4-8:2001 (Magnetic) 10A/m, 50Hz & 60Hz
Medical (non life-support) IEC60601-1-1:2000 Medical electrical equipment, Safety requirement (UH401 only) IEC60601-1-2:2007 Medical electrical equipment, Electromagnetic compatibility (EMC)