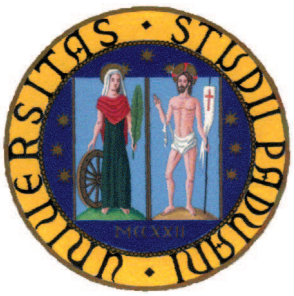
SigMate: A Comprehensive Automated Tool for Processing and Analysis of Extracellular Brain Signals Recorded by Neuronal Probes



A thesis submitted for the degree of

Doctor of Philosophy in Bioengineering

By:

Mufti Mahmud

Department of Information Engineering

University of Padova

Supervisor:

Prof. Stefano Vassanelli

Department of Human Anatomy & Physiology

University of Padova

January 2011 (XXIII Cycle)

To my father who is not among us today to see me becoming a doctorate and to my mother without whose unconditional love and prayers I would have never made it this far...

"*Homo sapiens* only philosophizes in his spare times. He is still too low on the intellectual scale, too utterly dominated by the reflexes of his stomach. Thought in his brain is a bird of passage, an irritating guest who interrupts the endless traffic of interest and greed."

– Dr. Santiago Ramón y Cajal

Acknowledgments

I am grateful to the members of the examination committee. They have helped me immensely over the last three years either in the course or mentally with advices. My gratitude will always be there to Professor Stefano Vassanelli, my supervisor who continuously guided me from all the directions and to Professor Alessandra Bertoldo, who always managed sometime from her busy schedule for me whenever I had to talk to her.

Heartiest thanks also to the NeuroChip laboratory colleagues – Dr. Stefano Girardi who helped me a lot during the experiments, Dr. Marta Maschietto whose intelligent ways of organizing things encourages me to be bit more organized, Mr. Mohammed Mostafizur Rahman who was and is (and probably will be) always there to help me out in any situation, Dr. Marco dal Maschio who taught me many things about data acquisition setups, Dr. Michele Scorzeto, who helped me out in the complicated Italian bureaucracies, Ms. Elisabetta Pasqualotto who helped me out with my published work, Ms. Silvia Lattanzio talking to whom was nice, and my students (supervising them made me learn many things).

Of course to my wife Ms. Tamanna Sharmeen who has been by my side all the time, inspiring me when I was upset and sacrificing her precious needs to let me do my work. My mother who always thinks I am the best, and all my siblings who think and believe I can do well in whatever I do.

Abstract

The ionic gating across the neuron membrane generates neuronal activity in the brain. During the last two decades rapid advances in microelectronics and microelectrode technology have provided scientists with many devices enabling them to record extracellularly the transmembrane potentials near the electrode in the brain. These devices that are implanted invasively without causing too much tissue damage, can record from hundreds of neurons, and also simultaneously from a number of channels generating a huge amount of data. Inferring meaningful conclusions by analyzing this massive amount of data often recorded from noisy experimental conditions is a big challenge for the neuroscience and neuroengineering community and sophisticated signal processing and analysis tools are required. But, relatively little work has been done on development of comprehensive signal processing tools operable on different software platforms and that can be easily diffused to the scientific community.

Though individual tools are available for signal visualization, spike detection and sorting, spike train analysis, yet analysis of local field potentials (LFPs) are still done manually. Most of these tools are developed by laboratories for their own requirements. Moreover, no software tools are available to date integrating all the signal processing steps under a single platform. This thesis aims at developing a comprehensive tool called 'SigMate' for processing and analysis of extracellular potentials; capable of performing operations ranging from signal visualization and basic operations to single sweep analysis and simulation of neuronal activity. The software package is designed to avoid file-type based incompatibility among different acquisition software and works with the neuronal data files in ASCII format. The functionalities of SigMate are described briefly below.

- Signal visualization (2D and 3D) and basic operations: This is the starting and home module of the software package that provides connectivity to other functionalities. With signal visualization it includes basic operations like signal averaging, noise estimation, +/- averaging, mean square and root mean square noise estimation. In every module a visualization pane is provided with zooming, panning and data cursor options.
- Basic file operations: Usually, incompatibility between acquisition and analysis tools poses a barrier in quick analysis of the recorded signals. However, most of the acquisition tools provide a way to convert the recorded files into ASCII format files and most of the analysis tools require specifically formatted files. To meet this need, the module includes operations like file splitting, file concatenating, and file column rearranging.
- Artifact removal: Stimulus artifacts very often obscure the real neuronal response in signals. This module performs artifact removal for both slow and fast stimulus artifacts with an optional baseline correction operation.
- Noise characterization: Invasive neuronal recording setups involve sophisticated electronic devices. Due to the wide variety of neural probes used by different labs a unique method for noise analysis is required. This module measures the quality of the recorded signals through noise estimation using detection of steady states.
- Latency estimation: Very often neuroscientists use latency information to understand the signal propagation in the brain. This module calculates latency and automatically determines cortical layer activation order using LFPs and current source density (CSD) data by applying CSD analysis on the LFPs.
- Spike detection and spike train analysis tool: Neuronal spikes are most widely studied signals. Many tools address spike detection and spike train analysis in the existing literature and this module

adapts 'Wave_Clus', a popular tool among them.

- Single sweep LFP clustering: LFPs represent cumulative response of neuronal populations around the recording electrode and are studied as an average of many single sweeps. Single sweep LFPs contain response of a neuronal population at a particular time instance and shows a range of shapes. As the shape of an LFP is considered as a fingerprint of the underlying neuronal network generating it, a shape based clustering system is presented in this module to facilitate the study of neuronal circuit activation.
- Interface with EEG based robotic system: This module contains an interface with the "Simulink" based EEG acquiring system developed by g.tec medical engineering GmbH. Using this module, it is possible to establish communication with a robotic device for navigation.
- Simulations: Neuronal simulations for optimization of stimulation protocol and simulation of calcium based model for flicking-based short-term plasticity.

Except the spike detection and spike train analysis tool, the rest of the features are in-house developed algorithms which are tested rigorously with datasets recorded using standard micropipette, implantable and planar EOSFETs from anesthetized rats upon different stimulations.

In conclusion, with the growth of neuronal probes, amount of acquired data are increasing and the need of one single software package performing all necessary processing and analysis on the data has become crucial. This thesis is the first step towards meeting that need. As the software has been extensively tested with three possible sources of data, we believe that once it is disseminated to the community (which will happen in the near future), it will serve a good deal in processing and analyzing extracellularly recorded neurophysiological signals.

Sommario

1.1 Motivazioni

I segnali neurali registrati con sonde neurali invasive o non invasive richiedono un'elaborazione e un'analisi rigorosa per arrivare a comprendere l'attivit generata dalla sottostante rete neurale in risposta a degli stimoli. Nel corso degli ultimi due decenni, il rapido sviluppo della microelettronica e della tecnologia del microelettrodo ha permesso agli scienziati di registrare contemporaneamente segnali provenienti da centinaia di neuroni usando numerosi canali. L'ottenimento di risultati significativi attraverso l'elaborazione e analisi di questa enorme quantit di dati registrati in condizioni sperimentali non ottimali rappresenta una grande sfida per le neuroscienze e la comunit della neuroingegneria. Anche se sono gi disponibili singoli software per eseguire l'analisi, ad esempio, di un treno di spike, il sorting e rilevamento del picco dello spike, non sono per ancora stati sviluppati strumenti software che integrino tutti gli step necessari per il processing del segnale EEG, degli spike neurali, e il calcolo dei potenziali di campo (local field otential - LFPs). Pertanto, la comunit della neuroingegneria sente pi che mai necessario lo sviluppo di un unico pacchetto software in grado di eseguire tutto il processing e l'analisi standard dei segnali neurali registrati. Questa tesi presenta come risultato finale un pacchetto software, "SigMate", costruito integrando assieme vari moduli per permettere l'elaborazione e l'analisi di LFP e di segnali EEG per il brain-machine-interface (BMI), la simulazione di un singolo neurone, e la rilevazione, l'ordinamento e l'analisi di un treno di spike.

1.2 Scopi e Obiettivi

Il pacchetto software SigMate sviluppato allo scopo di essere completo, adattabile, robusto e open-source. Per raggiungere questi obiettivi sono stati integrati metodi gi disponibili, presenti nella letteratura scientifica del settore e gi affermati all'interno di essa, con altri metodi che sono stati sviluppati durante lo svolgimento della tesi. Le capacit di analisi di SigMate permettono di elaborare nello stesso ambiente segnali EEG, spikes, e calcolare LFP. In particolare:

- **Algoritmi adattabili e robusti**: gli algoritmi per l'analisi di segnali neurali registrati usando sonde neurali multicanali devono essere: (i) adattabili per tener conto del numero sempre crescente di siti e canali di registrazione, e (ii), robusti ossia capaci di elaborare calcoli su grandi moli di dati, in modo accurato e veloce, quindi evitando lunghe attese al suo utilizzatore.
- **Algoritmi adattabili e robusti**: gli algoritmi per l'analisi di segnali neurali registrati usando sonde neurali multicanali devono essere: (i) adattabili per tener conto del numero sempre crescente di siti e canali di registrazione, e (ii), robusti ossia capaci di elaborare calcoli su grandi moli di dati, in modo accurato e veloce, quindi evitando lunghe attese al suo utilizzatore.
- **Performance**: per verificare la performance, l'accuratezza dei risultati, e la giusta integrazione dei moduli, sono stati usati segnali neurali registrati dalla corteccia di topo (in particolare da quella parte sottile della corteccia somatosensoriale (SI) che corrisponde ad una mappatura uno-a-uno dei baffi del naso del ratto) usando tre metodi diversi: (i) con micropipette standard, (ii) con Electrolyte-Oxide-Semiconductor Field Effect Transistor (EOSFET) messi su chip, e (iii) con EOSFET impiantabili.
- **Open-source**: il pacchetto software sar distribuito come open-source attraverso una GNU-General Public License (GPL) e per questa ragione Matlab stato selezionato come ambiente di sviluppo. L'utilizzatore libero di operare proprie modifiche adattando il software alle proprie esigenze.

1.3 Overview della tesi

La tesi organizzata in 5 capitoli. Il primo capitolo contiene l'introduzione, il secondo fornisce gli elementi di base che servono alla comprensione dei vari problemi affrontati e presenta anche una review della letteratura. Il capitolo 3 descrive i metodi per il setup del sistema e l'acquisizione dei segnali. I capitoli 4 e 5 descrivono la ricerca sviluppata durante lo svolgimento della tesi, mentre il capitolo 6 contiene un sommario e un overview sui possibili sviluppi futuri di questo lavoro.

Contents

List of Publications

Journal Publications

[J1]. **M. Mahmud**, E. Pasqualotto, A. Bertoldo, S. Girardi, M. Maschietto, S. Vassanelli (2010), "**An Automated Method for the Detection of Layer Activation Order in Information Processing Pathways of Rat Barrel Cortex under Mechanical Whisker Stimulation**," *Journal of Neuroscience Methods [in press].*

Conference Proceedings

[CP1]. **M. Mahmud**, A. Bertoldo, S. Girardi, M. Maschietto, E. Pasqualotto, S. Vassanelli (2011), "**SigMate: A Comprehensive Software Package for Extracellular Neuronal Signal Processing and Analysis**," *5th international IEEE EMBS Conference on Neural Engineering (IEEE–NE2011)*, Cancun, Mexico, 27 April–1 May, 2011 [Submitted].

[CP2]. S. Girardi, M. Maschietto, R. Zeitler, **M. Mahmud**, S. Vassanelli (2011), "**High Resolution Cortical Imaging Using Electrolyte–(Metal)–Oxide–Semiconductor Field Effect Transistors**," *5th international IEEE EMBS Conference on Neural Engineering (IEEE–NE2011)*, Cancun, Mexico, 27 April–1 May, 2011 [Submitted].

[CP3]. **M. Mahmud**, D. Travalin, A. Bertoldo, S. Girardi, M. Maschietto, S. Vassanelli (2011), "**An Automated Method for Clustering Single Sweep Local Field Potentials Recorded from Rat Barrel Cortex**," In: *Proceedings of the ISSNIP Biosignals and Biorobotics Conference 2011 (BRC2011)*, Vitoria, Brazil, 6-8 January 2011.

[CP4]. **M. Mahmud**, D. Travalin, A. Bertoldo, S. Girardi, M. Maschietto, S. Vassanelli (2010), "**A Contour Based Automatic Method to Classify Local Field Potentials Recorded from Rat Barrel Cortex**," In:

Proceedings of the 5th Cairo International Conference on Biomedical Engineering (CIBEC2010), Cairo, Egypt, 16-18 December 2010, pp. 163–166.

[CP5]. F. F. Milone, **M. Mahmud**, T. A. Minelli, M. M. Rahman, S. Vassanelli (2010), "**CNS 10 Hz LED 650 nm Stimulation: Measures and Hypotheses on the Possible Mechanisms of Reinforcement of the Alpha Brain Rhythms**," In: *Mind Force 2010: ConVersActions on the Embodied Mind, Centre for the Study of Complex Systems, University of Siena, Italy*, 7-9 October 2010.

[CP6]. **M. Mahmud**, D. Hawellek, A. Bertoldo (2010), "**EEG Based Brain-Machine Interface for Navigation of Robotic Device**," In: *Proceedings of the 3rd IEEE/RAS–EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob2010)*, Tokyo, Japan, 26-29 September 2010, pp. 168-172.

[CP7]. **M. Mahmud**, A. Bertoldo, S. Girardi, M. Maschietto, S. Vassanelli (2010), "**SigMate: A** MATLAB**–based Neuronal Signal Processing Tool**," In: *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC2010)*, Buenos Aires, Argentina, 31 August–4 September 2010, pp. 1352-1355.

[CP8]. **M. Mahmud**, A. Bertoldo, M. Maschietto, S. Girardi, S. Vassanelli (2010), "**Automatic Detection of Layer Activation Order in Information Processing Pathways of Rat Barrel Cortex under Mechanical Whisker Stimulation**," In: *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC2010)*, Buenos Aires, Argentina, 31 August - 4 September 2010, pp. 6095-6098.

[CP9]. **M. Mahmud**, S. Girardi, M. Maschietto, A. Bertoldo, S. Vassanelli (2010), "**Processing of Neuronal Signals Recorded by Brain-Chip Interface from Surface of the S1 Brain Cortex**," In: *Proceedings of the 36th Annual Northeast Bioengineering Conference (NEBEC2010)*, Colombia University, New York, USA, 26-28 March 2010.

[CP10]. **M. Mahmud**, S. Girardi, M. Maschietto, M. M. Rahman, A. Bertoldo, S. Vassanelli (2009), "**Noise Characterization of Electro-**

physiological Signals Recorded from High Resolution Brain–Chip Interface," In: *Proceedings of the International Symposium on Bioelectronics and Bioinformatics (ISBB2009)*, Melbourne, Australia, 9-11 December 2009, pp. 84-87.

[CP11]. **M. Mahmud**, S. Girardi, M. Maschietto, M. M. Rahman, A. Bertoldo, S. Vassanelli (2009), "**Slow Stimulus Artifact Removal through Peak–Valley Detection of Neuronal Signals Recorded from Somatosensory Cortex by High Resolution Brain–Chip Interface**," In: *IFMBE Proceedings of the World Congress 2009 in Medical Physics and Biomedical Engineering (WC2009)*, Munich, Germany, 7-12 September 2009, vol. 25/IV, pp. 2062-2065.

[CP12]. **M. Mahmud**, D. Hawellek, A. Valjamae (2009), "**A Brain-Machine Interface Based on EEG: Extracted Alpha Waves Applied to Mobile Robot**," In: *Proceedings of the 2009 ECSIS Symposium on Advanced Technologies for Enhanced Quality of Life (AT-EQUAL 2009)*, Iasi, Romania, July 22-26, pp. 28-31.

[CP13]. M. Maschietto, **M. Mahmud**, S. Girardi, S. Vassanelli (2009), "**A High Resolution Bi-Directional Communication through a Brain-Chip Interface**," In: *Proceedings of the 2009 ECSIS Symposium on Advanced Technologies for Enhanced Quality of Life (AT-EQUAL 2009)*, Iasi, Romania, July 22-24, pp. 32-35.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

Neuronal signals recorded by means of non–invasive/invasive neuronal probes require rigorous processing and analysis to understand the underlying neuronal network activity caused by stimuli. During the last two decades rapid development in the microelectronics and microelectrode technology have enabled the scientists to record simultaneously from hundreds of neurons and from a number of channels. Inferring meaningful conclusions by proccessing and analyzing this massive amount of data recorded from noisy experimental conditions is a big challenge for the neuroscience and neuroengineering community. Though individual tools are available to perform processing of the spike train analysis, spike detection and sorting, yet no software tools are available to date which integrate all the signal processing steps for EEG, neuronal spikes, and local field potentials (LFPs). Therefore, a comprehensive software package capable of performing standard processing and analysis on extracellularly recorded neuronal signals is required for the community. This thesis presents such a comprehensive software package, "SigMate", bundling together various modules for processing and analyzing the LFPs, an EEG based brain–machine–interface (BMI), a single neuron simulation environment, and a popular tool for spike detection, spike sorting, and spike train analysis.

1.2 Aims and Objectives

Considering the availability of the neuronal signal processing and analysis tools, the SigMate software package is developed with the objectives to be comprehensive, adaptable and simple, robust, and open–source.

- **Comprehensiveness**: The goal of this thesis has been to develop a comprehensive software package capable of performing processing and analysis on neuronal signals extracellularly recorded using neuronal probes. To make this package comprehensive, along with the *in–house* developed methods, standard available tools are adopted from the literature after extensive study and considering popularity of the tool in the community. The various features of the SigMate cover EEG, spikes, and LFPs.

- **Adaptable and simple algorithms**: In developing methods to perform processing and analysis on neuronal signals recorded using multisite and multichannel neuronal probes, there are two main challenges: (i) adaptability of the methods to the growing number of simultaneously recording sites and channels, and (ii) simple, yet robust algorithms capable of performing calculations on the huge data pool in a faster way without letting the user to wait a long time for the results. To meet these two challenges, the methods are made adaptable to the rapidly changing field of neuronal probe technology and the algorithms are kept as simple as possible capable of handling massive signal files quickly and accurately.

- **Performance**: To check the performance, accuracy, and correctness of the modules neuronal signals (evoked field potentials as well as spikes) are recorded from the rat barrel cortex (a tiny part of the rat somatosensory cortex (S1) containing a clear one–to–one mapping to the whiskers present on the rat snout) using three different approaches: (i) standard borosilicate micropipette (with 1 MΩ resistance), (ii) planar Electrolyte–Oxide–Semiconductor Field Effect Transistor (EOSFET) based chips, and (iii) implantable EOSFET based chips.

- **Open–source**: The software package is to be diffused in the community soon through an open–source GNU–General Public License (GPL) and for this reason MATLAB is chosen as the development environment. The users are free to make changes in the source code of the software core modules to adapt the code to their needs.

1.3 Overview of the Thesis

The rest of this thesis is organized into five chapters. Chapters 2 contain background material as of literature review, Chapters 3 mainly describe methods for system setup and signal acquisition, Chapter 4 and 5 mainly describe research specific to this thesis, and Chapter 6 contains a summary and an outlook on future work. The detailed contents are listed in the following:

- In Chapter 2, a general introduction to the field of neuronal signal analysis research is given. Topics reviewed include current advancement in different methods developed for analyzing brain activity. The software tools and / or packages available on the literature are divided into five categories: (i) EEG based tools, (ii) platform framework, (iii) MEA based tools, (iv) spike train analysis tools, and (v) other tools. Each of these categories are discussed with sufficient details to have an idea about the state–of–the–art of the field.

- In Chapter 3, experiments and signal acquisition techniques are discussed. This chapter is logically divided into two parts: (i) human based experiment, and (ii) experiments based on animal model (rat). The human based experiments are used in recording EEG signals for setting up the EEG based BMI. At the beginning of the experiments based on rats, the animal preparation (basically the neurosurgery) is discussed. After that, the experimental setups and signal acquisition methods for three different experimental techniques are elaborated: (i) rat–on–chip experiment, (ii) classical borosilicate micropipette based experiment, and (iii) implantable EOSFET based experiment. Along with the experimental techniques overview of the acquired signals are also provided.

- In Chapter 4, the architecture of the SigMate is described. The SigMate is designed using a three–layer model. The individual modules are outlined with their basic functionalities and interactions among various objects of each module is depicted using communication diagrams. These diagrams clearly indicate the information flow and various execution levels of each module of SigMate.

- In Chapter 5, the core modules of the SigMate are discussed in detail. Six features are highlighted: (i) artifact removal, (ii) noise characterization, (iii) latency estimation and layer activation order detection, (iv) clustering of single sweep LFPs, (v) EEG based brain–machine interfacing, and (vi) neuronal simulation environment. All these features are described independently and contain introduction, method description, and results and discussion of their own.

- In Chapter 6, the contributions of this thesis are summarized and an outlook on possible extensions of the presented work is provided.

Chapter 2

Literature Review

2.1 Background

Advances in the microelectronics and microelectrode technology have enabled the scientists to record from hundreds of neurons and also simultaneously from a number of channels (Buzsaki [2004], Prochazka *et al.* [2001], Wise *et al.* [2004]). These signals recorded from the brain by means of non–invasive/invasive neuronal probes require rigorous processing and analysis to understand the underlying neuronal network activity. These techniques support the analysis of the function as well as the structure of individual electrogenic cells in the context of surrounding neuronal network. Also to quantify brain activity underlying perception and to explain this activity as the outcome of elementary neuronal response is one of the major challenges of sensory systems neuroscience. Inferring meaningful conclusions by analyzing this massive amount of data recorded from noisy experimental conditions is a big challenge for the neuroscience and neuroengineering community (Buzsaki [2004]). Though individual tools are available to perform processing for the spike train analysis, spike detection and sorting, yet very few software tools are available to date which integrate all the signal processing steps (Kwon *et al.* [2009], Quiroga *et al.* [2004]). Also, commercially available tools for the analysis of such data, however, cannot be easily adapted to newly emerging requirements for data analysis and visualization, and cross compatibility between them is limited.

There are a few software packages developed for academic and commercial purposes (Bologna *et al.* [2010], Bonomini *et al.* [2005], Cui *et al.* [2008], Delorme & Makeig [2004], Egert *et al.* [2002], Goldberg *et al.* [2009], Gunay *et al.* [2009], Hazan *et al.* [2006], Herz *et al.* [2008], Huang *et al.* [2008], Kwon *et al.* [2009], Magri *et al.* [2009], Mrup *et al.* [2007], Novellino *et al.* [2009], Quiroga *et al.* [2004], Smith & Mtetwa [2007], Vargas-Irwin & Donoghue [2007], Vato *et al.* [2004], Versace *et al.* [2008], Wagenaar *et al.* [2005]). These software packages mainly deal with data visualization, spike detection and sorting, spike train analysis, processing and analysis of signals recorded using multi–electrode arrays, EEG signal analysis, and cross–software platform. Also, a couple of open platforms are under development to promote sharing of different laboratory–developed tools across the worldwide web (Lidierth [2009], Meier *et al.* [2008]). However, there is no comprehensive standard tool/package to date incorporating analysis on spike trains, LFPs, EEG based interfacing, and basic neuronal simulations. Moreover, data format conversion and using multiple packages on a single dataset to perform different operations are time consuming and cumbersome. Thus an umbrella tool is required to perform these file operations.

2.2 Current Advances

The current advancement in the development of the software tools for processing and analyzing neuronal signals recorded from the brain can be divided in following five categories:

- Electroencephalography (EEG) based tools
- Platform framework
- Multi electrode array (MEA) based tools
- Spike train analysis tools
- Other tools

The following subsections contain detailed information regarding the specific categories.

2.2.1 EEG Based Tools

In the past decades, pioneer researchers have tried to apply to EEG data analysis techniques developed in electrical engineering and information theory, including time/frequency analysis (Bressler & Freeman [1980], Makeig [1993] , Pfurtscheller & Aranibar, Tallon-Baudry *et al.* [1996], Weiss & Rappelsberger [1996]) and Independent Component Analysis (ICA) (Jung *et al.* [2001], Makeig *et al.* [1997, 1999]). These techniques have revealed EEG processes whose dynamic characteristics are also correlated with behavioral changes, though they cannot be seen in the averaged event related potentials (ERPs). For example, short–term changes in spectral properties of the ongoing EEG in specific frequency bands may be correlated with cognitive processes, e.g., expectancy of a target stimulus (Makeig *et al.* [1999]) and with visual awareness (Rodriguez *et al.* [1999]). The sufficiency of studying average ERPs has also been questioned by Makeig *et al.* [2002], who showed that some average ERP peaks may result from partial synchronization of oscillatory EEG processes to time locking events in single data trials.

Currently, most EEG researchers still interpret their data by measuring peaks in event–locked ERP averages. Free availability of more general and easy–to–use signal processing software for EEG data may encourage the wider adoption of more inclusive approaches. The EEGLAB software toolbox for MATLAB (freely available from `http://www.sccn.ucsd.edu/eeglab/`) allows processing of collections of single EEG data epochs using ICA and spectral analysis as well as data averaging techniques. This toolbox combines the advantages of ICA, time–frequency analysis, and multi–trial visualization (Delorme & Makeig [2003]; Delorme *et al.* [2002], Makeig *et al.* [1999, 2002]). In EEGLAB, all these functions are available under a common graphic interface developed in MATLAB. EEGLAB extends the collection of publicly available MATLAB packages for brain imaging including SPM (Friston [1995]) and FRMLAB (Duann *et al.* [2002]) for functional MRI studies and Brainstorm (Baillet *et al.* [1999]) for EEG/MEG source analysis.

Another open source toolbox "ERPWAVELAB" is developed for multi–channel time-frequency analysis of event related activity of EEG and MEG data (Mrup *et al.* [2007]). The toolbox provides tools for data analysis and visualization of the most commonly used measures of time-frequency transformed event related

data as well as data decomposition through non–negative matrix and multiway (tensor) factorization. The decompositions provided can accommodate additional dimensions like subjects, conditions or repeats and as such they are perfected for group analysis. Furthermore, the toolbox enables tracking of phase locked activity from one channel-time-frequency instance to another as well as tools for artifact rejection in the time-frequency domain. ERPWAVELAB can freely be downloaded from `www.erpwavelab.org`, and requires EEGLAB (Delorme & Makeig [2004]) and runs under MATLAB.

2.2.2 Platform Framework

FIND (Finding Information in Neural Data) is a platform–independent, open source framework for the analysis of neuronal activity data based on MATLAB (Meier *et al.* [2008]). The authors have published the structure of the FIND framework describing its functionality, measures of quality control, and the policies for developers and users (Meier *et al.* [2008]). Within FIND they have developed a unified data import from various proprietary formats, simplifying standardized interfacing with tools for analysis and simulation. These analysis tools address various types of neural activity data, including discrete series of spike events, continuous time series and imaging data. The toolbox also provides solutions for the simulation of parallel stochastic point processes to model multi–channel spiking activity.

The other software package named "sigTOOL" runs in the MATLAB and has been designed to promote the sharing of laboratory–developed software across the worldwide web (Lidierth [2009]). sigTOOL has been used to build an analysis application for dealing with neuroscience data with a user–friendly graphical user interface which implements a range of waveform and spike–train analysis functions. The interface allows many commonly used neuroscience data file formats to be loaded (including those of Alpha Omega, Cambridge Electronic Design, Cyberkinetics Inc., Molecular Devices, Nex Technologies and Plexon Instruments). Waveform analysis functions selectable from the interface support waveform averaging (mean and median), auto- and cross–correlation, power spectral analysis, coherence estimation, digital filtering (feedback and feedforward) and resam-

pling. Spike–train analyses include interspike interval distributions, Poincar plots, event auto- and cross–correlations, spike–triggered averaging, stimulus driven and phase–related peri–event time histograms and rasters as well as frequencygrams. As claimed by the author, full sigTOOL functionality will be provided to support the user-developed code, including the ability to record a user action history for batch processing of files and support for exporting the results of analyses to external graphics editing software and spreadsheet–based data processing packages.

To fully harvest the benefits of data access, data storage and data analysis, together with supporting activities for teaching and training possibilities, a set of coordinated activities is required. Through development of tools aiming at neurophysiological data, the German Neuroinformatics Node (G-Node) aims at addressing these aspects as part of the International Neuroinformatics Coordination Facility (INCF) (Herz *et al.* [2008]). Based on its technical and scientific scope, the Node could play a substantial role for cellular and systems neurophysiology as well as for the neuroscience community at large.

Yet another package, named, "KInNeSS" is under development with the ultimate goal of creating a modular framework that will help researchers across different disciplines to effectively collaborate using a modern neural simulation platform (Versace *et al.* [2008]).

2.2.3 MEA Based Tools

The MATLAB based toolbox "MEATools" (`http://www.brainworks.uni-freiburg.de/projects/mea/meatools/overview.htm`) is an open source toolbox distributed under GNU public license, which regulates its free use and redistribution (Egert *et al.* [2002]). The general idea of this toolbox is to provide a basic set of routines and graphical user interfaces (GUIs) that facilitate access to the original data files and to feed these to an expandable set of tools for their visualization and analysis. It also serves as framework to which additional functions can be added. File access is supported by a minimal set of routines that can be adapted or exchanged for different data acquisition systems. To facilitate the adaptation and extension of the MEATools functionality as well as the addition of new tools for data analysis, the source code and extensive documentation are freely available.

"MeaBench" ia a software tool for data acquisition and online analysis of multi-electrode recordings, especially from micro–electrode arrays (Wagenaar *et al.* [2005]). Besides controlling data acquisition hardware, MeaBench includes algorithms for real–time stimulation artifact suppression and spike detection, as well as programs for online display of voltage traces from 60 electrodes and continuously updated spike raster plots. MeaBench features real–time output streaming, allowing easy integration with stimulator systems. We have been able to generate stimulation sequences in response to live neuronal activity with less than 20 ms lag time. MeaBench is open–source software, and is available for free public download at `http://www.its.caltech.edu/~pinelab/wagenaar/meabench.html`.

"DATA–MEAns" is a free open-source software for the classification and management of neural ensemble data (Bonomini *et al.* [2005]). The main goal of this tool is to provide a graphical user interface that links the experimental data to a basic set of routines for analysis, visualization and classification in a consistent framework. To facilitate the adaptation and extension as well as the addition of new routines, tools and algorithms for data analysis, the source code and documentation are freely available.

The "Neural Signal Manager" (NSM), aims at analyzing a huge quantity of data recorded by means of MEAs in a fast and efficient way (Novellino *et al.* [2009]). The NSM offers different approaches for both spike and burst analysis, and integrates state–of–the–art statistical algorithms, such as the interspike interval histogram or the post stimulus time histogram, with some recent ones, such as the burst detection and its related statistics.

The software package, named, "SPYCODE", offers the scientific community a 'smart' tool for multi–channel data processing recorded from MEAs (Bologna *et al.* [2010]). As claimed by the authors, the package contains (i) a rich repertoire of algorithms for extracting information both at a single channel and at the whole network level; (ii) the capability of autonomously repeating the same set of computational operations to 'multiple' recording streams (also from different experiments) and without a manual intervention.

2.2.4 Spike Train Analysis Tools

Among all the neuronal signals, spikes attracted most attention in the neuroscience community in terms of analysis and processing. Therefore, many various ways have been developed for spike detection, sorting, and analyzing the spike trains (Goldberg *et al.* [2009], Hazan *et al.* [2006], Magri *et al.* [2009], Pouzat & Laurent [2002], Quiroga *et al.* [2004], Smith & Mtetwa [2007], Vargas-Irwin & Donoghue [2007], Vato *et al.* [2004]). These different tools perform spike detection, sorting and analysis considering various aspects of the spikes and their performances are also comparable. Amongst all these available tools, SigMate has adapted the Wave_Clus tool developed by Quiroga *et al.* [2004] (Mahmud *et al.* [2010a]).

A platform has also been developed for performing spike detection and analysis for the signals recorded by means of MEAs (Huang *et al.* [2008]).

2.2.5 Other Tools

Apart from the categories discusses above, there are few other software tools developed to perform specialized or specific analysis on neuronal signals.

A MATLAB/C toolbox, "Brain–SMART" (System for Multivariate AutoRegressive Time series, or BSMART), for spectral analysis of continuous neural time series data recorded simultaneously from multiple sensors (Cui *et al.* [2008]). Available functions include time series data importing/exporting, preprocessing (normalization and trend removal), AutoRegressive (AR) modeling (multivariate / bivariate model estimation and validation), spectral quantity estimation (auto power, coherence and Granger causality spectra), network analysis (including coherence and causality networks) and visualization (including data, power, coherence and causality views). The tools for investigating causal network structures in respect of frequency bands are unique functions provided by this toolbox. All functionality has been integrated into a user–friendly graphical user interface (GUI) environment designed for easy accessibility. This toolbox is freely available (`http://www.brain-smart.org`) under the GNU public license for open source development.

A software package called "NeuroQuest"® contains neural signal processing

and data analysis algorithms (Kwon *et al.* [2009]). The software package is designed for analyzing neural recordings. As mentioned by the authors, NeuroQuest® provides a friendly Graphical User Interface (GUI) environment with advanced spike train analysis tools beyond the standard tools in many commercial software packages that enable identifying and visualizing neuronal connectivity across multiple brain regions at a scale not possible with state–of–the–art functional neuroimaging techniques such as fMRI.

Database representation of raw spike data may reduce the storage space and their processing time by several orders of magnitude (Gunay *et al.* [2009]). The database analysis toolbox, "PANDORA", is capable of acquiring data from common recording and simulation platforms and exchanging data with external database engines and other analysis toolboxes, which make analysis simpler and highly interoperable. PANDORA is available to be freely used and modified because it is open–source (`http://software.incf.org/software/pandora/home`).

Chapter 3

Experiments and Signal Acquisition

3.1 Introduction

For testing the EEG based BMI model module of the SigMate, the EEG signals were acquired from a number of subjects. But, for the other modules (except the neuronal simulation), signals were acquired from anesthetized rat brain using three different experimental conditions. The following sections contains detailed description of the experimental conditions and procedures followed to perform them.

3.2 EEG Acquisition Experiments

The EEG signal were recorded using a four channel commercial EEG recording device, g®.MOBIlab, manufactured by the g.tec medical engineering GmbH, Austria (`http://www.gtec.at/`). Out of the four channels three channels were used during the recording. Two of them were used in recording simultaneous signals from 'O1' and 'O2' and the third electrode was used as a reference, placed at 'F' or 'FP' position (figure 3.1 shows the international 10–20 electrode mapping of EEG). The manufacturing body provided a module and a framework (implemented in the MATLAB (`http://www.mathworks.com`) using MATLAB scripting

and Simulink) to acquire and process the EEG signals. The module worked as an interface between the recording device and the computer used for recording the EEG. The framework was reengineered to analyze and process the acquired EEG signals from the occipital region of the subject. After the data were read, they were amplified with a certain gain and sent to the computer for on–the–fly processing. The figure 3.2 shows the schematic of the signal acquisition and processing system. The signals detected by the recording electrodes were fed into a preamplifier, where the different channel signals were separated and then amplified by a predefined gain (shown as right faced triangles). These amplified signals of two channels were then fed into the computer at a sampling rate of 256 Hz through an USB port for further processing.

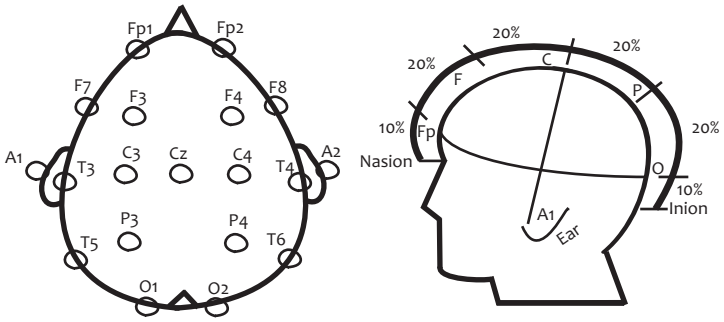


Figure 3.1: The electrode positions on the scalp for EEG recording according to the standard 10–20 international system.

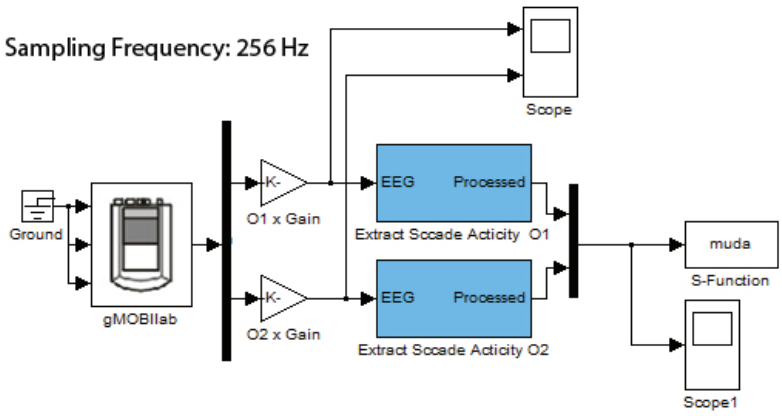


Figure 3.2: Schematic of EEG signal acquisition and conditioning process.

The devices used for the EEG signal acquisition is shown in figure 3.3.

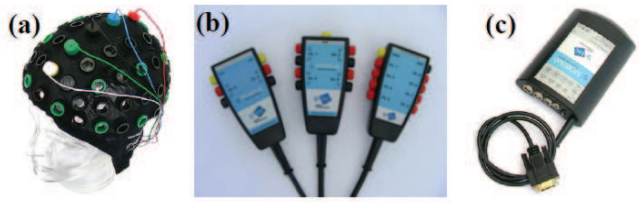


Figure 3.3: EEG communication interface devices. (a). cap and electrodes. (b). connector and/or multiplexer. (c). preamplifier.

3.3 Animal Preparation

Wistar rats were maintained in the Animal Research Facility of the Department of Human Anatomy and Physiology (University of Padova, Italy) under standard environmental conditions.

P30-P40 male rats were anesthetized with an induction mixture of Tiletamine (2 mg/100 g weight) and Xylazine (1.4 g/100 g weight). The anesthesia level was monitored throughout the experiment by testing eye and hind-limb reflexes, respiration and checking the absence of whiskers' spontaneous movements. Whenever necessary, additional doses of Tiletamine (0.5 mg/100 g weight) and Xylazine (0.5 g/100 g weight) were provided.

During the surgery and the recording section, animals were kept on a common stereotaxic apparatus under a stereomicroscope and fixed by teeth and ear bars. The body temperature was constantly monitored with a rectal probe and maintained at about 37° C using a homeothermic heating pad. Heart beat was assessed by standard ECG. To expose the cortical area of interest, anterior-posterior opening in the skin was made along the medial line of the head, starting from the imaginary eyeline and ending at the neck. While the skin was kept apart using halsted-mosquito hemostats forceps, the connective tissue between skin and skull was gently removed by means of a bone scraper. Thus, the skull over the right hemisphere was drilled to open a window in correspondence of the somatosensory cortex, S1 ($-1 \div -4$ AP, $+4 \div +8$ ML) (Swanson [2003]). Meninges were

then carefully cut by means of forceps at coordinates −2.5 AP, +6 LM for the subsequent insertion of the recording micropipette.

Throughout all surgical operations and recordings, the brain was bathed by a standard Krebs solution (in mM: NaCl–120, KCl–1.99, $NaHCO_3$–25.56, KH_2PO_4–136.09, $CaCl_2$–2, $MgSO_4$–1.2, glucose–11), constantly oxygenated and warmed at 37° C.

At the end of the surgery, contralateral whiskers were trimmed at about 10 mm from the mystacial pad.

3.4 Experiments and Setups

3.4.1 Rat–On–Chip Experiment

3.4.1.1 The Experiment

After the surgery, the rat was carefully placed "upside-down" on a custom-made plastic pad featuring a chip chamber, a heated gel–based "mattress" and a fixation system to fix the rat's head. The chip was inserted into the socket and the rat was placed on the pad, being ready for recording. This system allowed subsequent perfect positioning of the selected cortex area in the center of the chip, maintaining the body temperature and reducing movement–generated artifacts. Figure 3.5 depicts the signal recording setup with its various components. Figure 3.4 (a) shows an empty pad with the chip mounted on its socket and ready for an experiment; (b) shows the positioning of the animal for the recording.

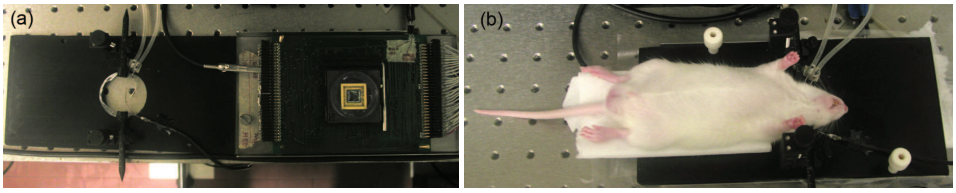


Figure 3.4: (a) Empty pad and socket with inserted chip. (b) Pad carrying the anesthetized rat with its head placed on the chip.

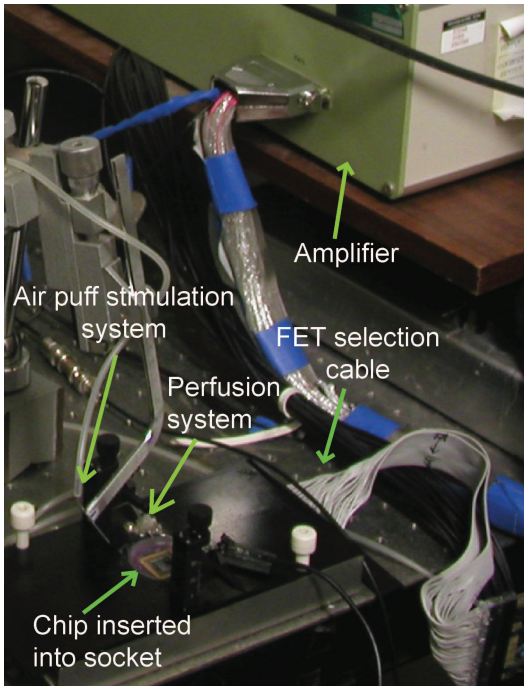


Figure 3.5: Signal recording setup depicting the socket with the chip, the perfusion system, FET selection cable that facilitates the selection of 16 FETs for simultaneous recordings, the custom built amplifier, and the air–puff stimulation system.

3.4.1.2 Chip Description

The chips are based on two linear arrays, each consisting of 31 insulated Electrolyte Oxide Semiconductor Field Effect Transistors (EOSFET and FET are used synonymously in the rest of the text), spaced 30 μm to 40 μm, and 32 Electrolyte–Oxide Semiconductor Capacitors (EOSCs) integrated in between two adjacent transistors. The area covered by each FET is either 3.1 μm $\times$ 7.2 μm or 3.5 μm $\times$ 9.0 μm (Schmidtner & Fromherz [2006]).

Figure 3.6 (a) shows the chip with its chambers, and a magnification of the internal recording structures in (b). These chips were fabricated at the Technical University of Berlin, Berlin, Germany and during the fabrication five masks were used for creating the structures: mask1 for the active areas, mask2 for the p+ inlets, mask3 for the passivation, mask4 for the gate definition, and mask5 for the

metalization. The relative structure of various masks used during the fabrication process can be seen in figure 3.7.

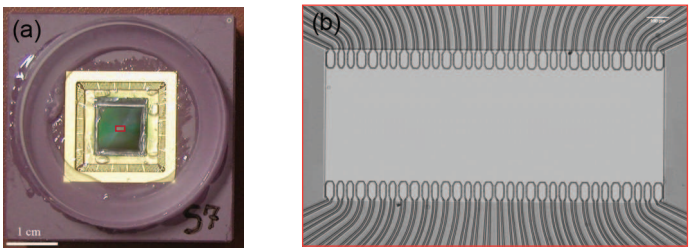


Figure 3.6: (a) A chip used during the experiment and (b) its magnification showing the recording structures, scale bar 100 μm.

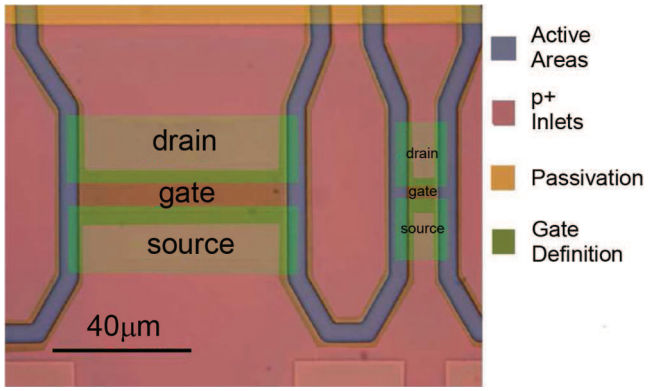


Figure 3.7: Relative structures of various masks used during the various phases of fabrication process.

To evoke neuronal response, 40 psi, 50 ms – lasting air–puffs were provided with a delay of 10 ms to the contra–lateral whiskers by means of a microinjection pneumatic picopump (PV80, WPI Inc., USA). The chip was connected to a computer by a custom–built amplifier and the neuronal signals were recorded by an in–house software called Patch–Panel, developed in LabView (v7.1, `http://www.ni.com/labview/`). Simultaneous recordings from 16 EOSFETs were performed at 20 kHz sampling rate during each session (figure 3.5). Baseline sweeps were recorded at the beginning of each experiment as control signals to be able to assess the presence of stimulus artifacts.

3.4.1.3 The Recorded Signals

This type of experiment was unique of its kind. Using these chips it was possible to perform an electrical imaging of the cortical area under investigation at a higher resolution than the existing techniques. Figure 3.8 shows a 3D plot of simultaneously recorded signals from 13 EOSFETs during an experiment. Signal propagation along the different EOSFETs provide an electrical image of the cortical region in contact with the recording sites.

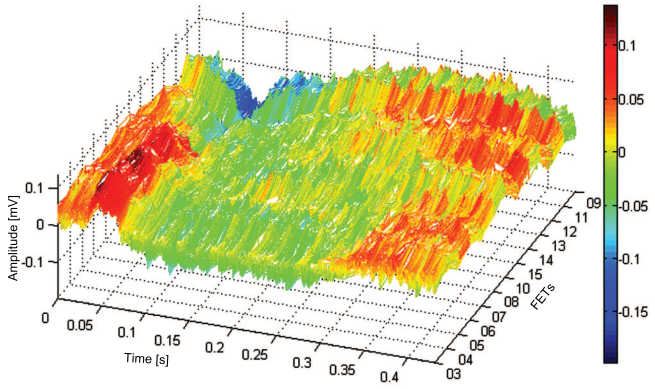


Figure 3.8: 3D plot of signals simultaneously recorded by 13 FETs.

3.4.2 Classical Micropipette Based Experiment

3.4.2.1 The Experiment

The recording of LFPs from S1 was performed by means of borosilicate micropipettes (1 MΩ resistance), filled with Krebs solution. The pipette was fixed to a micromanipulator so that it was 45° tilted with respect to the vertical axis of the manipulator, thus being inserted perpendicularly to S1 cortex surface. The figure 3.9 depicts the experimental setup and the stimulus waveform used in driving the stimulator.

LFPs were evoked by single whiskers mechanical stimulation performed with a custom–made speaker that provides dorsal–ventral movements through a connected tube. The speaker was driven by a waveform generator (Agilent 33250A 80 MHz, Agilent Technologies) providing 1 ms, 10 V square stimuli with 150 ms

delay. Each whisker, starting from the posterior group, was individually inserted into the tube and the corresponding response was checked at -750 μm depth (cortical layer IV), in order to find the most responsive whisker for the selected recording point in the cortex. The so-called "principal whisker" was then chosen for the recording, and the evoked LFPs were recorded from all the cortical layers with a 90 μm recording pitch. For each depth, 100 sweeps with 500 ms duration are recorded at 20 kHz sampling rate. An open source software, 'WinWCP' (Version: 4.1.0) developed by the SIPBS, University of Strathclyde, UK (`http://spider.science.strath.ac.uk/sipbs/software_ses.htm`) was used for recording the signals.

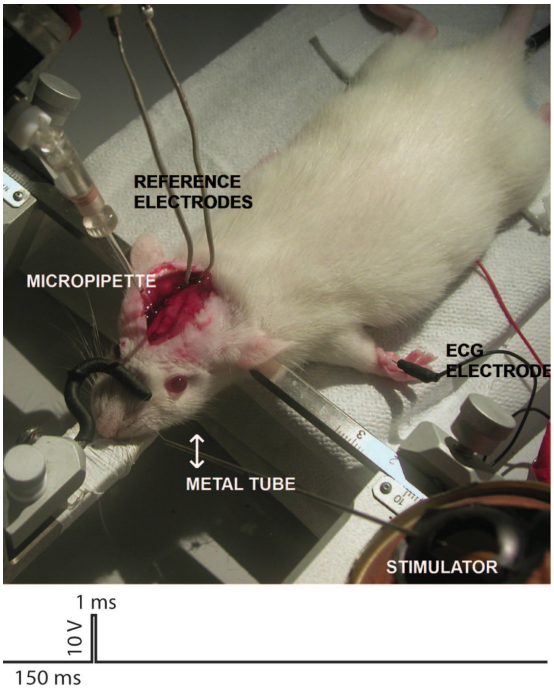


Figure 3.9: Experimental setup depicting its various components. The arrow on the metal tube connected to the stimulator shows the direction of its movement. Bottom is the stimulus waveform used in driving the speaker, causing dorsal–ventral movement of whisker that is inserted in the metal tube.

3.4.2.2 The Recorded Signals

The LFPs recorded from a barrel column of the rat S1 cortex by stimulating the corresponding whisker can be differentiated by their specific characteristics based on the depth or layer they are recorded from. Figure 3.10 shows a representative depth profile of one of our experiments.

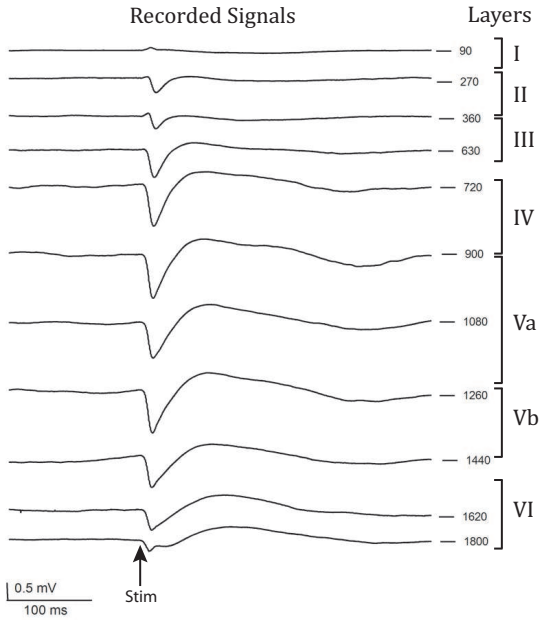


Figure 3.10: Depth profile of local field potentials recorded from the E1 barrel column by stimulating the E1 whisker where the different features of the signals can be easily seen. The full depth profile contained equidistant recordings spaced by 90 μm, but for the ease of visualization only representative signals from each layer are shown.

As illustrated in Ahrens & Kleinfeld [2004] and Kublik [2004], usually in upper cortical layers (I, II) the signals are expected to have a small positive peak, followed by a main negative peak, a positive peak and a slow negative valley that gradually tends to reach the baseline at the end. In the middle layers (III, IV, and V) the signals are expected to have the main negative peak (without the first small positive peak) followed by a slow positive peak and a slow negative valley tending to reach zero at the end. Deep in the cortex (layer VI), the main

negative peak becomes smaller and usually gets divided into two smaller negative peaks, followed by a slow positive peak and then the slow negative valley. These characteristics of the signals can be exploited in automated detection of the layers from the recorded signals.

3.4.3 Implantable EOSFET Based Experiment

The Experiment The chips were fixed to the micromanipulator so that they were 45°–tilted respect to the vertical axis of the manipulator, thus being inserted perpendicularly to S1 cortex surface. Neuronal signals were evoked by single whiskers mechanical stimulation performed with a custom–made speaker providing dorsal–ventral movements through a connected tube. The speaker was driven by a waveform generator providing 1 ms, 10 V square stimuli with 150 ms delay. Each whisker, starting from the posterior group, was individually inserted into the tube and the corresponding response was checked in order to find the most responsive whisker for the selected recording point in the cortex. The "principal whisker" was then chosen for the recording. The chip was left into the brain for about 5 hours, throughout the duration of the experiment.

LFPs evoked by single whiskers stimulation were recorded by 4 FETs separated by a 90 μm distance at different cortical depths. One hundred traces averages are shown. It is evident that transistors can record different signal amplitudes depending on the recording depth, even within the same layer.

Figure 3.11 shows the setup of this type of experiment. In the figure A.: the general view of the setup. Stereomicroscope, heating pad and body temperature controller, stereotaxic apparatus and ECG electrodes are indicated. A PC driven micromanipulator was used to hold and insert the chip into the brain in a careful and spatially controlled way. A speaker was used to drive movements of a selected whisker by means of a connected tube. In B.: Magnification of the rat head: the opening in the skull in correspondence of the S1 cortex and the chip inserted into the cortical area are visible. The chip was carefully inserted through a slit in the meninges and the selected whisker was inserted into a tube mounted on a speaker for stimulation.

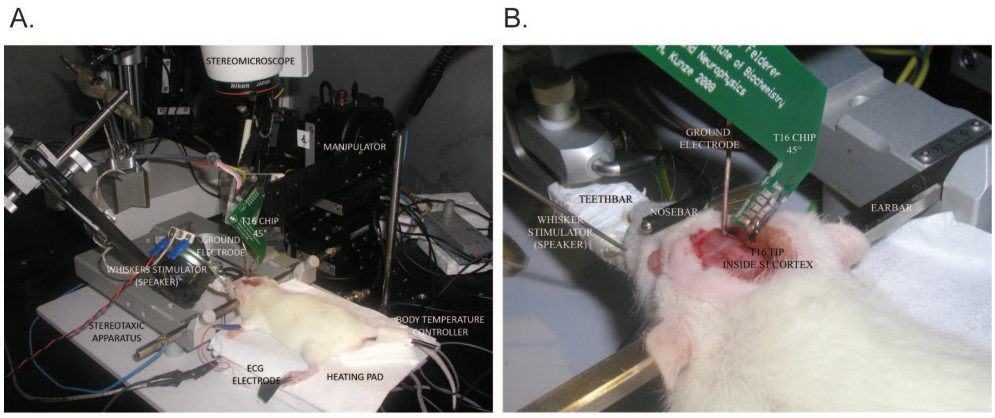


Figure 3.11: Neuronal signal recording setup using 4 FET implantable chip. A. Overview of the setup. B. Magnification of the recording area.

The figure 3.12 (a) shows the close view at the 4 recording sites of the implantable chip. These 4 recording sites were capable of performing simultaneous recording at different depths. The figure 3.12 (b) shows the simultaneous recordings with the deepest FET at 305 μm. It is clearly visible that the different FETs record signals at depths with varied signal morphology. This is also evident in case of the signals in figure 3.12 (c) where the deepest FET is at 845 μm.

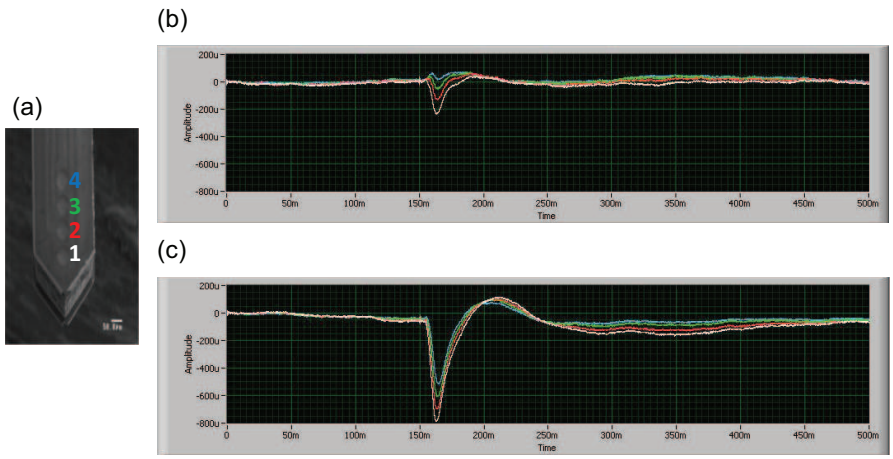


Figure 3.12: (a) Chip tip showing different FETs. (b) Simultaneous recording from 4 FETs with FET1 at 305 μm. (c) Simultaneous recording from 4 FETs with FET1 at 845 μm.

3.4.3.1 The Signals

As described in the section 3.4.3, the 4 FETs are capable of recording signals from different depths simultaneously. By moving the chip up–and–down, depth profiles were recorded at a pitch of 50 μm. The figure 3.13 shows such a depth profile.

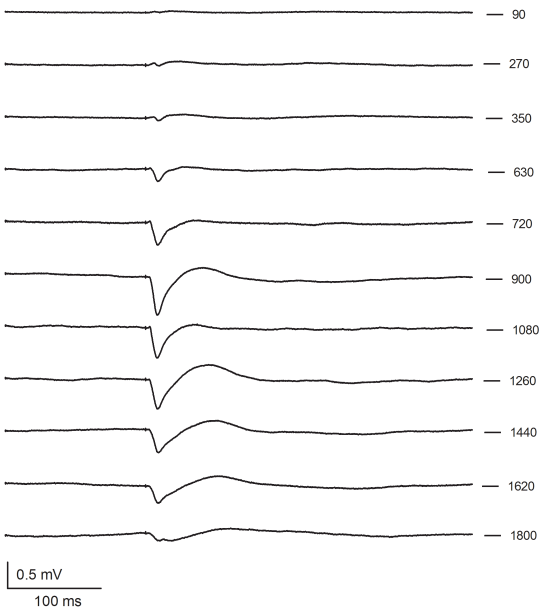


Figure 3.13: Depth profile of LFPs recorded from the a barrel column by stimulating its corresponding whisker where the different features of the signals can be easily seen. The full depth profile contained equidistant recordings spaced by 50 μm, but for the ease of visualization only representative signals from each layer are shown.

Chapter 4

SigMate Architecture

4.1 Overview of SigMate

SigMate is designed using a multi–layered approach with three layers.

- Presentation layer (top layer): This is the topmost level of the application. The presentation layer contains the Graphical User Interface (GUI) of the application. It communicates with the middle layer by requesting the user commands.

- Application layer (business logic, logic layer, data access layer, or middle layer): The logic layer is separated from the presentation layer and, as its own layer, it controls an applications functionality by performing detailed processing.

- Data layer (bottom layer): This layer consists of databases and/or storages. Here information is stored and retrieved. This layer keeps data neutral and independent from applications or business logic. Giving data its own layer also improves scalability and performance.

This type of three–layered architecture is a client-server architecture in which the user interface, functional process logic ("business logic"), computer data storage and data access are developed and maintained as independent modules, may also reside in separate platforms. Apart from the usual advantages of modular

software with well–defined interfaces, the three–layer architecture is intended to allow any of the three layers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the top layer would only affect the user interface code.

The figure 4.1 shows the three–layered architecture of the SigMate software package.

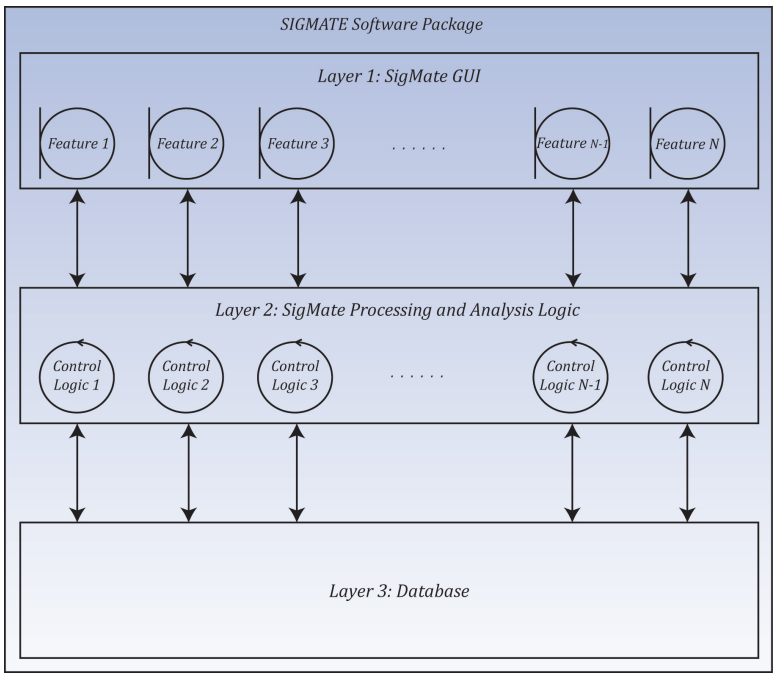


Figure 4.1: Three–layered architecture of the SigMate software package.

4.2 Modules of SigMate

The SigMate software package is designed to perform various processing and analysis on the neuronal signal files. The main functionalities included at present are: data display (2D and 3D) with zooming, panning and data cursor options, slow stimulus artifact removal (including baseline correction), noise characterization with noise estimation and baseline correction, basic file operations (including file splitting, file concatenating and file column rearranging), latency estimation (in LFPs and CSDs) with the possibility to detect the Cortical Layer Activation

Order (CLAO), EEG based robotic device interface, Hodgkin–Huxley based neuronal simulation environment, and the spike train analysis package adopted from Quiroga *et al.* [2004]. Apart from the spike train analysis module, the rest of the features are *in–house* developed algorithms which are tested rigorously with datasets recorded using standard borosilicate micropipette and EOSFETs from anesthetized rats. The following subsections describe briefly the individual modules with diagrams showing communication among objects. Figure 4.2 shows the use case diagram of the software package.

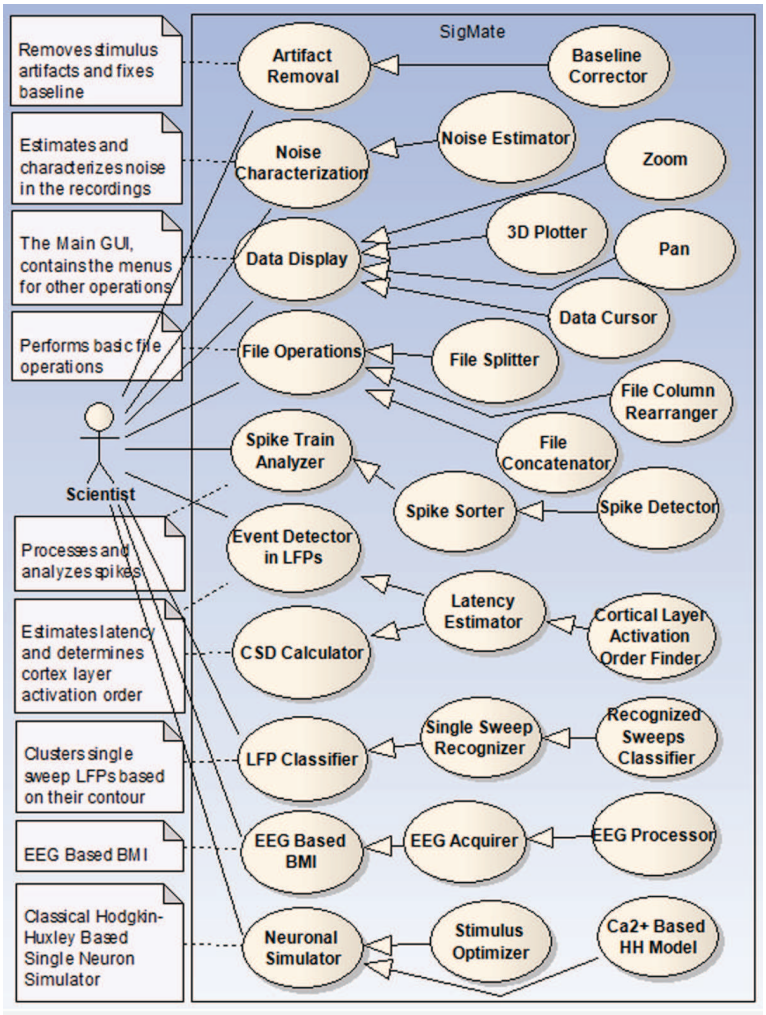


Figure 4.2: Use case model of the SigMate software package.

4.2.1 Signal Visualization and Basic Operations

This module's GUI is opened to the user once the application is initialized. It contains the itemized menubar for connecting to the other modules of the package. This module does not only provide the user with the flexibility of viewing the data in 2D and 3D, but also provides the possibility to perform averaging of single sweeps, estimate the noise, perform ± averaging, and calculate the mean square and root mean square to have an idea about the signal's amplitude. These basic operations are useful to preliminarily understand the quality of the signals. The communication diagram is shown in figure 4.3. From the communication diagram the inter–object communication can easily be viewed. The various operations pertaining to logical bodies are numbered and color coded.

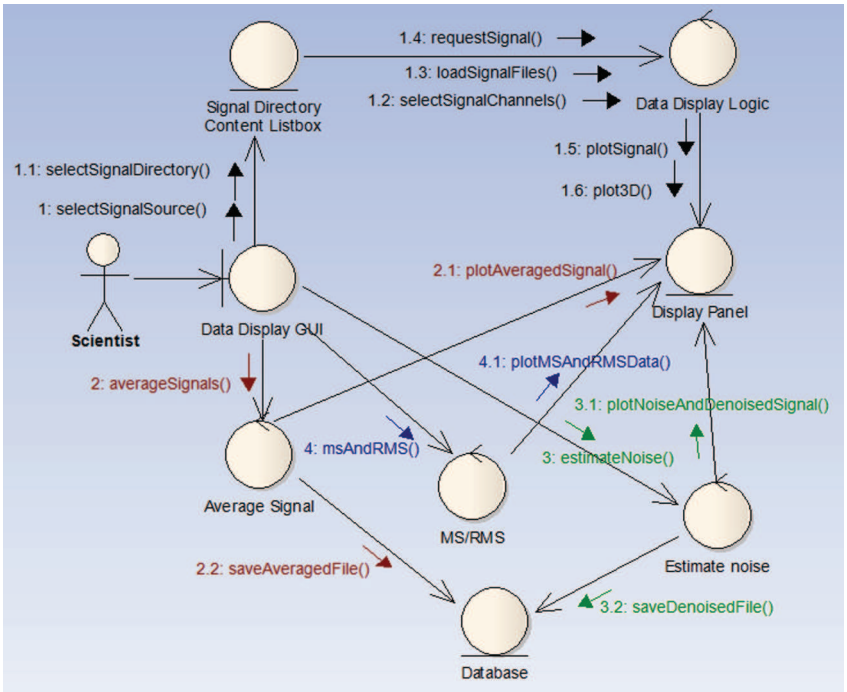


Figure 4.3: Communication diagram of the data display module. The numbers before the function calls denote order of function call. The block–head arrows show the information flow and the open arrows demonstrate the communication between objects.

4.2.2 File Operations

Few basic file operations are being incorporated in the software package that are often time consuming for the scientists who use different software for signal recording and performing signal processing and analysis. These operations include: file splitting (splits multi–sweep file into many single–sweep files based on sampling frequency), file concatenation (concatenates multiple single–sweep files into a multi–sweep file), and file column rearranging (retains only selected channels and eliminates the unselected ones). As the communication diagram demonstrates in figure 4.4, the operations are mutually exclusive. Different operational pathways are followed to perform the various operations offered by the module.

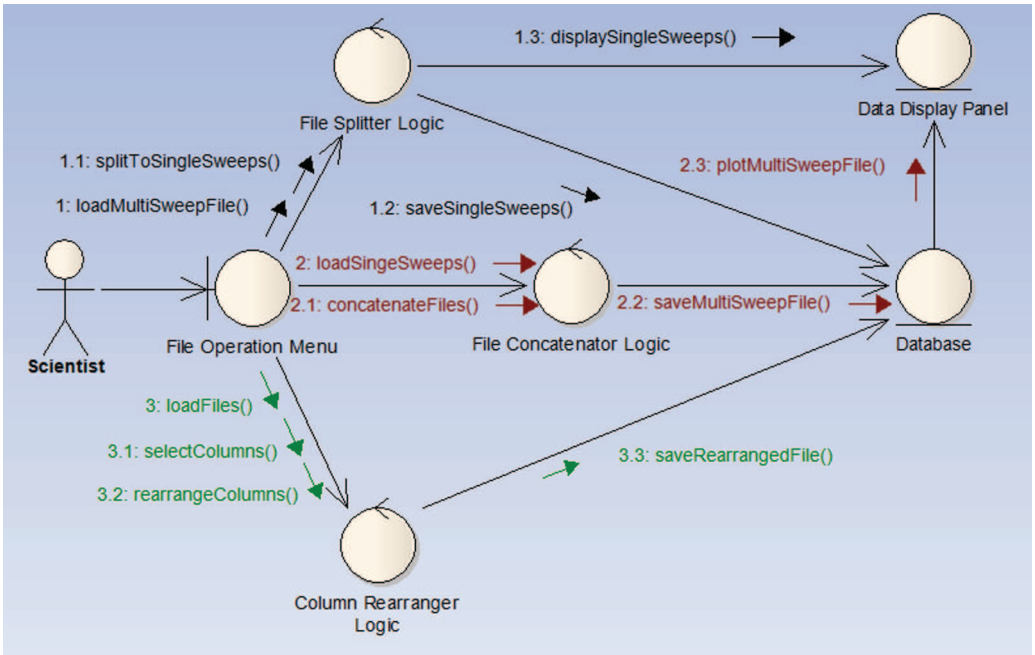


Figure 4.4: Communication diagram of the file operations module.

4.2.3 Artifact Removal

This module performs artifact removal as well as baseline correction. It expects the control signals (recorded without providing stimulation) and signals with evoked response upon stimulation. It utilizes an *in–house* algorithm (listed in

appendix .1) for detection of peak–valley pairs in a signal. Point to note here is, for each peak there is a corresponding valley which constitutes a peak–valley pair. The average of this peak–valley pair gives the estimated point of the signal part in which the pair is detected. Therefore, the averages of these pairs provide an estimation of the signal (Mahmud *et al.* [2009b]). The mean of this estimation is subtracted from a signal for the baseline correction. The estimation of the control signal is subtracted from the evoked signal to remove artifact from the evoked signal. Figure 4.5 shows the communication diagram of the module.

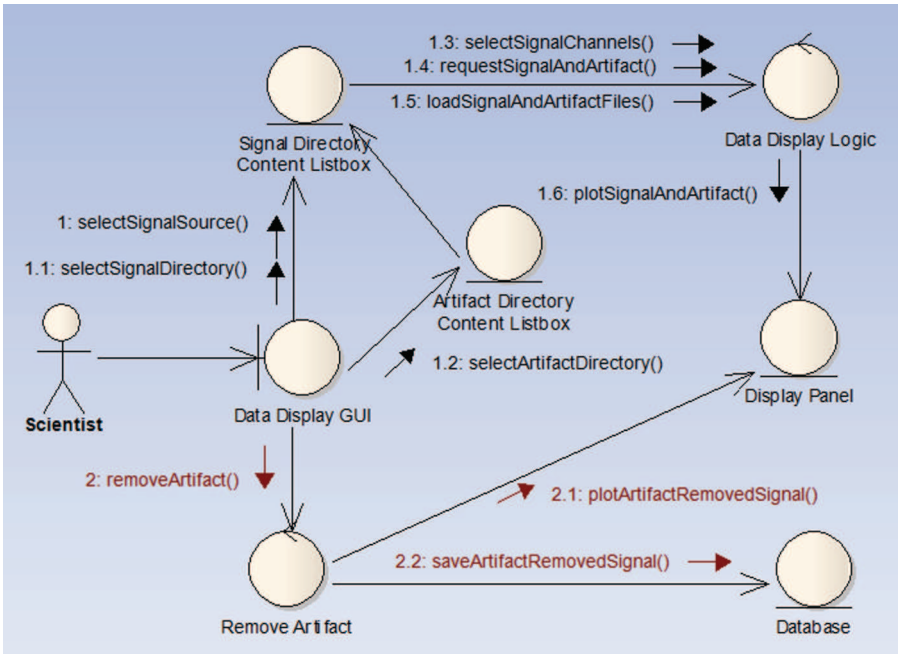


Figure 4.5: Communication diagram of the slow stimulus artifact removal module.

4.2.4 Noise Characterization

The noise characterization module assesses the quality of the recorded signals and quantifies the noise present in the signals. It uses *in–house* algorithms for the detection of the steady–states of a sweep. These steady–states are detected as the prestimulus part (first steady–state, FSS) and the part of the signal from the end of the evoked response until the end of the signal (second steady–state, SSS) through comparison of the standard deviation of the signal. Once the steady–

states are detected, mathematical models are fitted to calculate the measurement errors (MEs) present in the signal (Mahmud *et al.* [2009a]). First order statistical information such as mean and standard deviation of each ME are used to quantify the noise. Also, distribution of the noise and their estimation are shown. The communication diagram of the module is shown in figure 4.6.

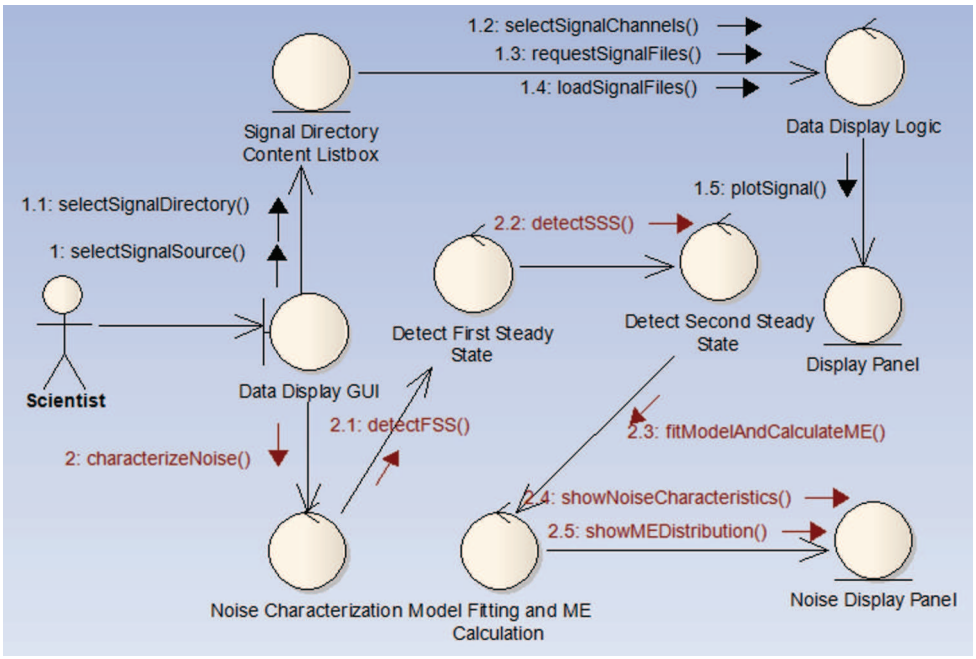


Figure 4.6: Communication diagram of the noise characterization module.

4.2.5 Latency Estimation

This module calculates latencies in the LFPs, CSDs and saves these latencies for determining the CLAO. As the latencies are calculated from two different entities, they are represented as two different modules in the software package. However, the CLAO can be determined using the calculated latencies and thus they are designed together. The communication diagrams of these two modules can be seen in figure 4.7.

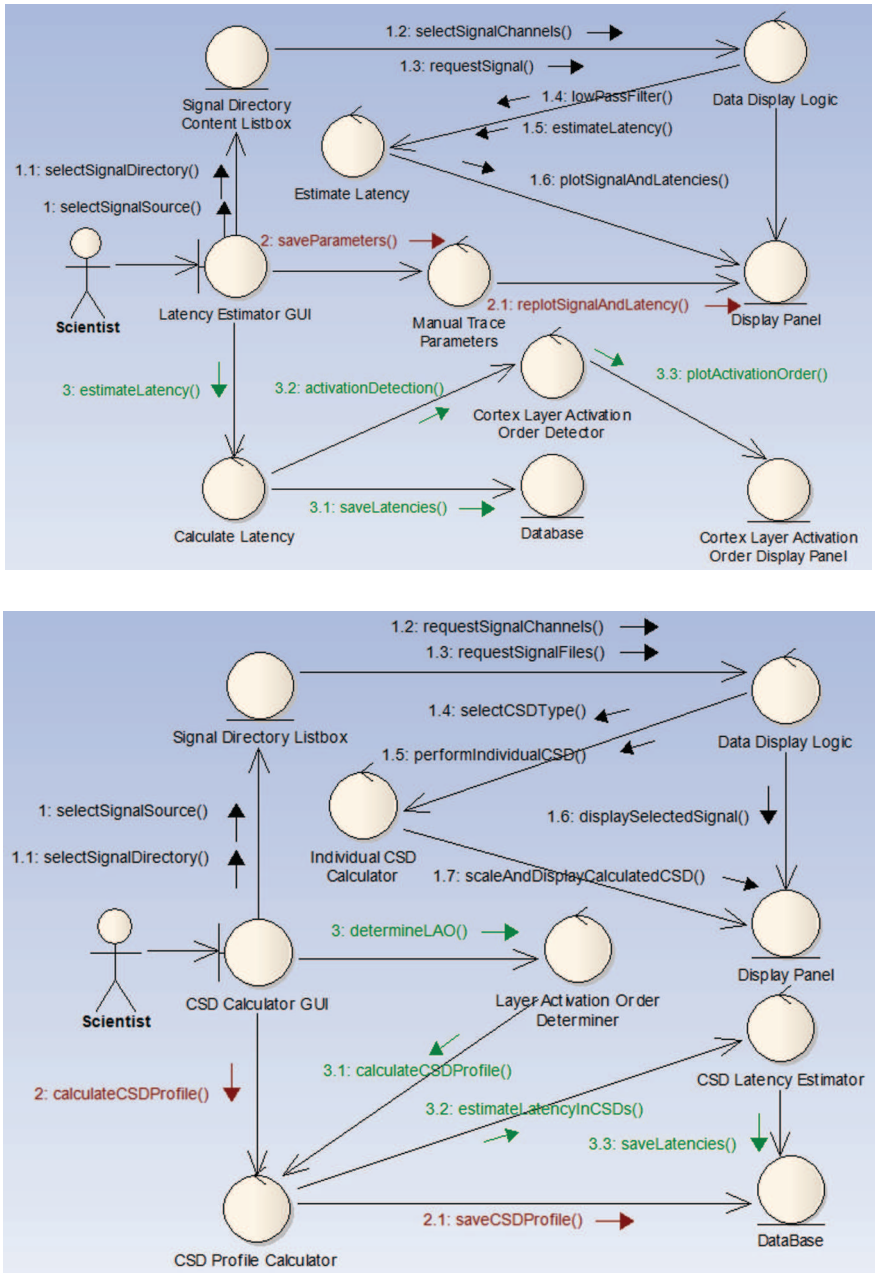


Figure 4.7: Top: Communication diagram of the latency estimation in LFPs. Bottom: Communication diagram of the latency estimation in the CSDs.

4.2.5.1 Event Detection and Latency Computation in LFPs

To detect latencies in the LFPs, it calls a function capable of detecting the events present in an LFP signal (Mahmud *et al.* [2010b]). After detecting the events, latencies are calculated as the differences between detected events and stimulus–onset.

4.2.5.2 Latency Computation in CSDs

From the LFPs, CSDs are calculated using four different methods: standard CSD analysis method, delta–inverse CSD method, step–inverse CSD method, and spline–inverse CSD method (Pettersen *et al.* [2006]). From the CSD profile the latencies are estimated as the difference between the first sink's peak and the stimulus onset (Mahmud *et al.* [2010c]).

4.2.5.3 Determination of Cortical Layer Activation Order (CLAO)

Once the latencies in the LFPs and CSDs are calculated, the CLAO is automatically determined using them. The latencies from both the LFPs and CSDs are layerwise grouped basing on the a priori information about the recording depths. For the LFPs, the CLAO is defined as an ascending ordered list of minimum latencies of the second event in each layer. Similarly, the CLAO from the CSDs are calculated by sorting the minimum layerwise latencies in ascending order (Mahmud *et al.* [2010c]).

4.2.6 Single Sweep LFP Clustering

Clustering or classification of single sweep LFPs plays an important role in understanding the underlying neuronal network that generated the signals. As shape information is very important in LFPs, this module exploits the contour information for performing the classification. The module is designed to perform the operations in three main steps: template generation, single sweep recognition through template matching, and clustering single sweeps using intelligent K–Means clustering (Mahmud *et al.* [2010d]). Once the signals are clustered, local averages of the clusters are computed, events in each local average are detected,

and latencies of the events are calculated. The amplitudes and the latencies of the events are stored for further processing. Figure 4.8 shows the communication diagram where the objects created by different steps and the interaction between them are clearly shown.

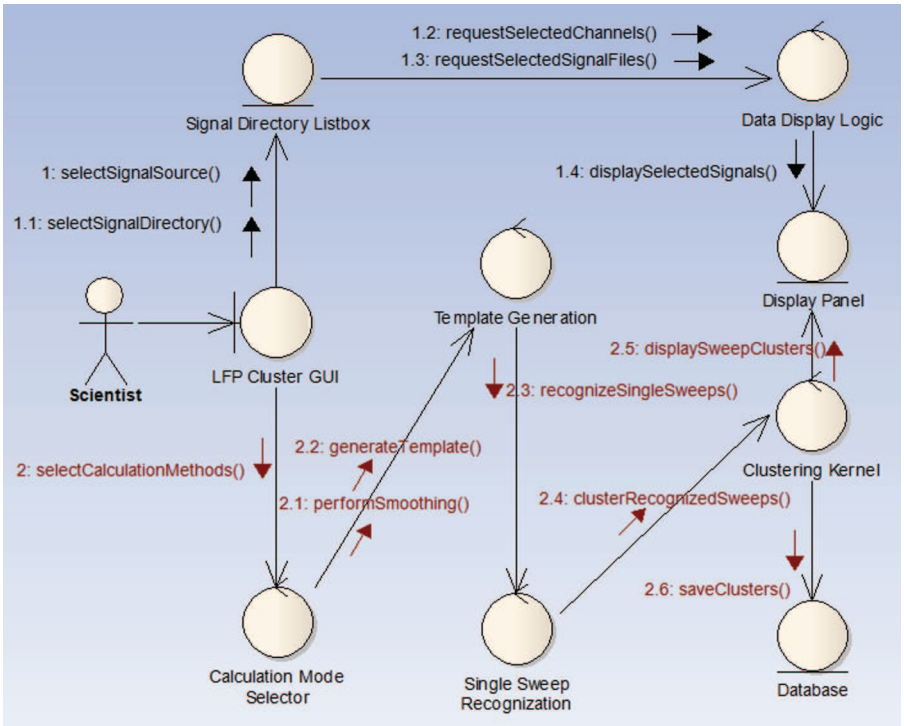


Figure 4.8: Communication diagram of the single sweep LFP clustering.

4.2.7 Neuronal Simulation Environment

The neuronal simulation environment contains basically two separate components for simulating single neuronal behavior. The first one simulates the single neuron behavior based on various stimuli, thus, allows the user to optimize the stimulus protocol. The second one instead simulates the single neuron behavior based on Ca2+ models allowing the user to study single neuron behavior by controlling the amount of glutamate and calcium concentration. Both the models are derived from Hodgkin–Huxley formulation and contain a general waveform generator capable of delivering waveforms in different shapes, frequencies, and amplitudes.

Figure 4.9 shows the communication diagram with indications about the interaction among the objects.

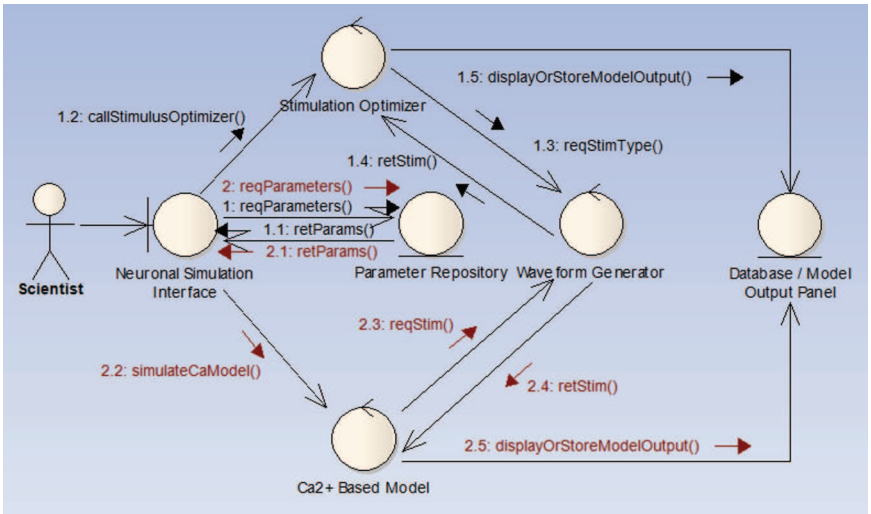


Figure 4.9: Communication diagram of the neuronal simulation environment.

4.2.8 Other Modules

There are two different modules present in SigMate which are not designed explicitly. They are: (i) EEG based interfacing with robotic systems, and (ii) spike detection, spike sorting, and spike train analysis.

The EEG based interfacing system does not contain a GUI of itself rather uses the framework and GUI provided by the device manufacturer which is capable of calling the functions performing signal processing and analysis to generate the control signals. Considering this, the explicit design of the module was not done.

The spike processing and analysis bundle is adapted from Quiroga *et al.* [2004]. SigMate provides an interface to connect to this tool with proper parameters. Therefore, no design of this module is performed.

Chapter 5

SigMate Modules: Methods, Results and Discussions

5.1 Artifact Removal

5.1.1 Method

This artifact removal using peak–valley detection technique is straight-forward, computationally efficient, and simple to implement. Figure 5.1 shows the GUI of the artifact removal process. The module is implemented in MATLAB (`http://www.mathworks.com`) platform.

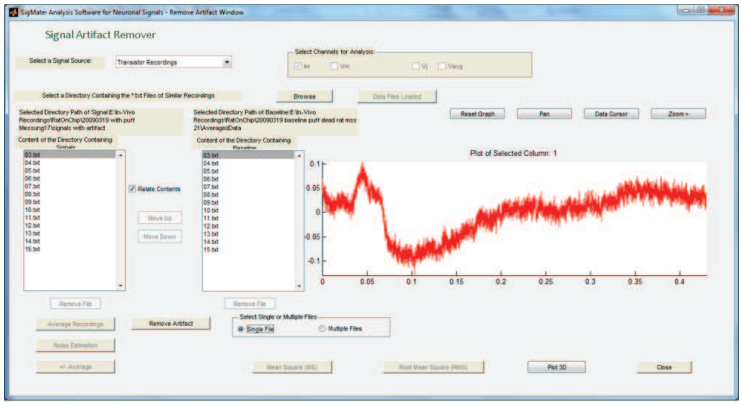


Figure 5.1: GUI of the artifact removal module. It offers the possibility to perform artifact removal on single signal file or batch processing of multiple files.

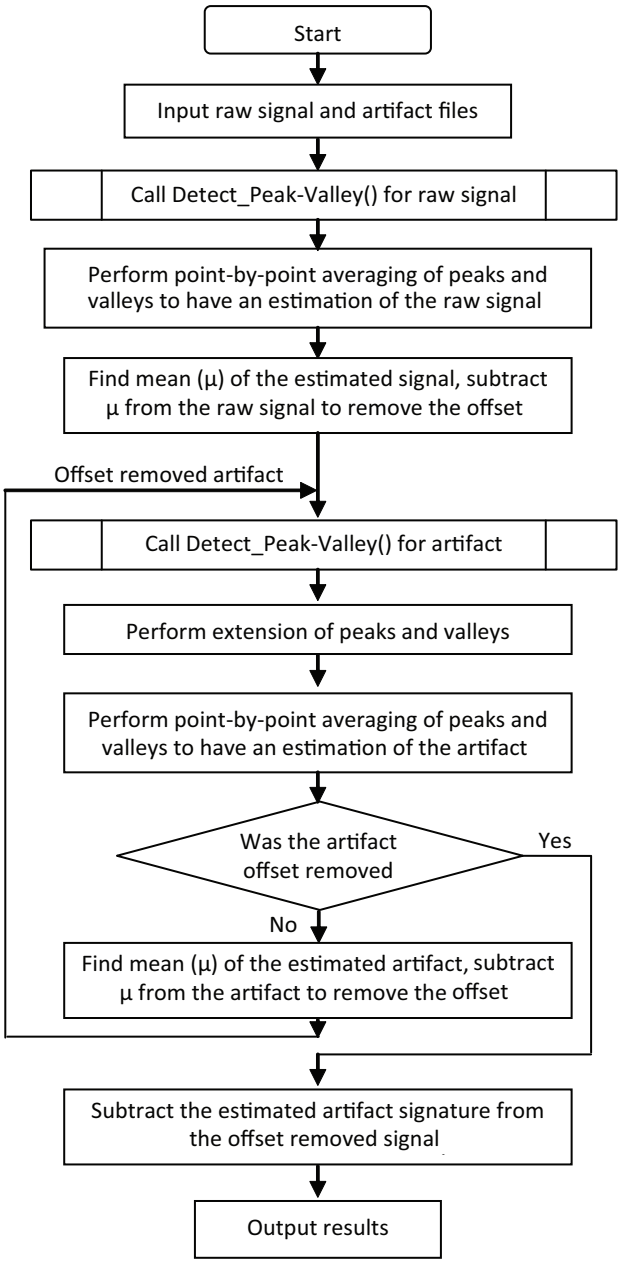


Figure 5.2: Flowchart of the artifact removal method.

This module performs artifact removal as well as baseline correction. It expects the control signals (A_{Orig}, recorded without providing stimulation) and signals with evoked response (S_{Orig}) upon stimulation. It utilizes an in–house algorithm (listed in appendix .1) for detection of peak–valley pairs in a signal. Point

to note here is, for each peak there is a corresponding valley which constitutes a peak–valley pair. The average of this peak–valley pair gives the estimated point of the signal part in which the pair is detected. Therefore, the averages of these pairs provide an estimation of the signal (Mahmud *et al.* [2009b]). The mean ($\mu_{EstimatedS}$) of this estimation is subtracted from a signal for the baseline correction ($S_{OffRem} = S_{Orig} - \mu_{EstimatedS}$). The same technique is applied for correcting the baseline of the control signal ($A_{OffRem} = A_{Orig} - \mu_{EstimatedA}$). The estimation of the baseline corrected control signal ($A_{OffRemEstimated}$) is subtracted from the baseline corrected evoked signal to remove artifact from the evoked signal ($S_{Clean} = S_{OffRem} - A_{OffRemEstimated}$). Figure 5.2 shows flowchart of the artifact removal method.

5.1.2 Results and Discussion

The figure 5.3 shows the estimation of the artifact calculated by the Detect_Peak-Valley() algorithm as described in appendix .1. The offset of this signal is corrected using the mean of such estimation with the raw artifact, as described in section 5.1.1.

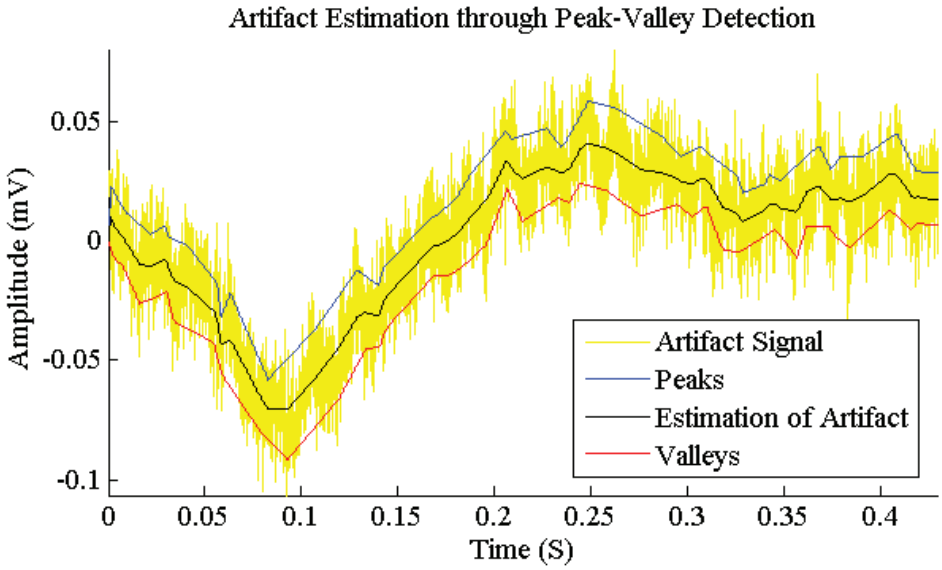


Figure 5.3: Artifact and its estimation through peak–valley detection using the signal's standard deviation as threshold.

Figure 5.4 shows the traces before and after stimulus artifact removal. The first trace from the top is the artifact. The second trace shows the artifact contaminated signal, and the third trace the signal after artifact removal. At the bottom the air puff stimulation duration is shown.

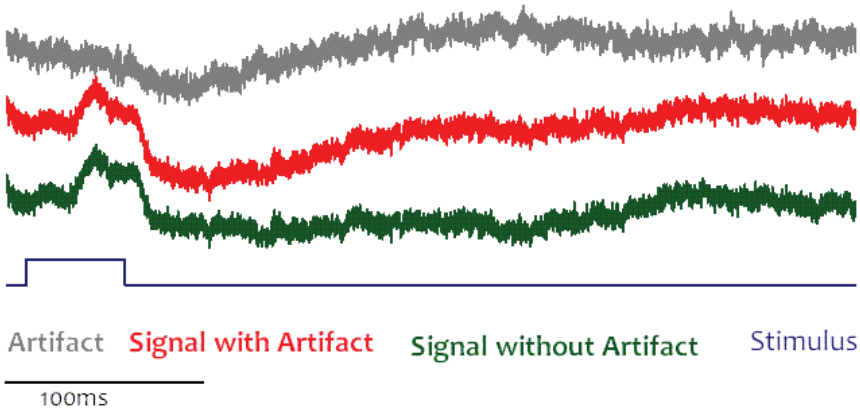


Figure 5.4: Traces of artifact (gray), evoked potential with artifact (red), artifact removed evoked potential (green), and the stimulus.

This method is also applied to perform batch processing of signals to remove stimulus artifacts. The figure 5.5 (a) shows the signals recorded from 13 EOSFETs (each EOSFET's signal is digitized and stored in a separate file) with the stimulus artifact (the two arrows point the artifact region) on which the stimulus artifact removal technique is applied. The figure 5.5 (b) shows the signals without stimulus artifacts.

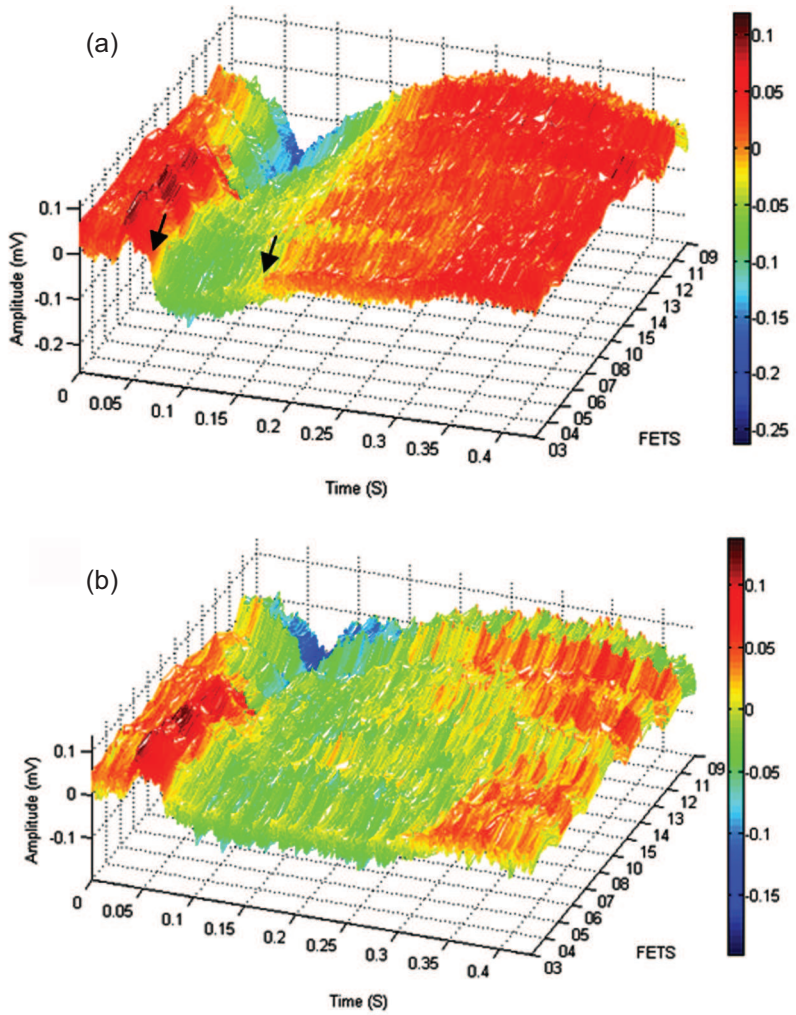


Figure 5.5: Signals before and after stimulus artifact removal. The color–bars show the amplitude intensity of the signals. (a) Raw signals recorded from 13 EOSFETs. The two arrows show the stimulus artifact region. (b) Signals without stimulus artifact as a result of batch processing of the stimulus artifact removal method proposed in this work.

5.2 Noise Characterization

During the last two decades rapid advances in microelectronics and microelectrode technology have provided scientists with many devices enabling them to

record extracellularly from the brain. The wide diversity of these devices let the scientists to acquire high precision recordings from one or a few single neurons, low spatial resolution recordings with a sparse sampling within the networks; also, silicon based high spatial resolution recordings to have unprecedented information about the underlying neuronal network activity (Prochazka *et al.* [2001]; Wise *et al.* [2004]; Maschietto *et al.* [2009]). This has facilitated the scientists' understanding of the brain activity and gradually the underlying neuronal network. But, with the growing availability of the high resolution tools, sophisticated signal processing is required to assess the quality of the signals in terms of noise contaminations.

Many previous studies have been done to reduce noises from physiological signals. Some important studies include: adaptive noise canceling (Widrow *et al.* [1975]), template subtraction (Woo *et al.* [2006]), independent component analysis (Snellings *et al.* [2006]), time–lagged recurrent neural network (Kale & Dudul [2009]), virtual referencing among recording electrodes and inter–electrode correlation (Paralikar *et al.* [2009]), average noise cycle subtraction (Jiruska *et al.* [2009]), algorithms basing on spatial filtering (De Cheveigne & Simon [2008]), discrete wavelet transform (Oweiss & Anderson [2001]), wavelet filtering (Zanchettin & Ludermir [2007]), and subspace methods based on wavelet and PCA (Cutmore & Celka [2008]). Many recording devices have been developed as well (Uranga *et al.* [2004]; Sahin [2005]; Harrison & Charles [2003]; Perelman & Ginosar [2007] and Saha *et al.* [2010]) for low–noise and low–power recording. Also, experimental techniques have been adapted with increasing number of reference electrodes (Shaw *et al.* [2003]) and design of amplifier system for *in–vivo* neuronal recording (Dagtekin [2006]).

However, it is impossible to denoise completely recorded signals from the intrinsic noise of a signal acquisition system. Especially high resolution cortical signal acquisition systems often employ many complex and high precision electronic devices, causing these recordings to be contaminated with intrinsic noise of different sources. The noises generated from some of the sources come with their own signatures, so, are easy to identify and remove from the recordings. However, noises generated from heating of resistors in the measurement chain, connectivity of instruments with mismatched impedance near the recording site,

and the light source do not have a predefined signature, as a consequence, the bare eye noise assessment of the recorded signals cannot be performed. Therefore, noise analysis and characterization is required to understand and assess the quality of the recording.

Recently, spike sorting and noise reduction algorithms are developed to perform noise characterization and modeling for *in–vivo* neural recordings (Yang *et al.* [2009]) under the assumption that multiple sources produce aggregated noise that deviates from the conventional white Gaussian noise. So, the signals are interpreted with a noise of $1/f^x$ family ($x = 1.50 \pm 0.50$). Also, another study showed a $1/f^2$ power law in the spectrum of local field potentials (LFPs) (Milstein *et al.* [2009]). However, no work till date has been done in assessing the quality of signals recorded from cortical surface by high–resolution chips.

This work presents an automated method for noise characterization of the cortical signals recorded by a high resolution brain–chip interface. This method uses the SD of each signal sweep to detect the steady–state parts (without any brain activity) of the signal. Firstly, the method detects two parts in each signal: the FSS (first part of the signal before the evoked response) and the SSS (starting from end of the evoked response until the end of the signal). It then calculates and characterizes the ME by assuming it to be normally distributed. For this method, the steady–states' detection is very crucial, as they represent the noisy portions of the signal without any useful information about the brain activity. Therefore this method works basing on an accurate and easy–to–implement algorithm capable of detecting the two steady–states which provide the exact noise information about the signals.

5.2.1 Theory

5.2.1.1 Measurement Error Model

Measurement error is the difference between a measured value of quantity and its true value (Dodge [2003]). Mathematically, one can state that the measured values, *z(t)*, is obtained by the sum of the true values and the ME, *e(t)*, also called noise, having Gaussian distribution with zero mean. Assuming that the model $y(t, \mathbf{p})$ is the mathematical representation of the true values, the measured

values can be represented by the function given in equation 5.1.

$$z(t) = y(t, \mathbf{p}) + e(t) \tag{5.1}$$

where the functional dependence of y on time t and the parameter vector $\mathbf{p} = [p_1, \ldots, p_n, p_{n+1}]^T$ has been made explicit. We consider that time is the only independent variable and that measurements are made precisely at the known times $t_i, i = 1, \ldots, N$.

5.2.1.2 Model Fitting

There are a variety of methods for regression analysis (Landlaw & DiStefano Third [1984]). The weighted least squares (WLS) estimator method generates estimate of parameter $\mathbf{p}$ that minimizes the weighted sum of squared difference between the observed $z(t_i)$ and the model predictions $y(t_i, \mathbf{p})$. This is also called the weighted residual sum of squares (WRSS) and is calculated using the equation 5.2.

$$WRSS_j(\mathbf{p}) = \sum_{N}^{i=1} w_{ji}[z(t_i) - y_j(t_i, \mathbf{p})]^2 \tag{5.2}$$

with $j = 1, \ldots, m$; where m is the number of candidate models for predicting $y(t, \mathbf{p})$. The weight functions are calculated using the variance σ_{ji}^2, with $w_{ji} = 1/\sigma_{ji}^2$ (Landlaw & DiStefano Third [1984]).

The recorded signal is assumed to be represented by an n–order polynomial model, where the coefficients of the polynomial of degree n, organized in parameter vector p, are calculated in a least squares sense that fits the data $z(t)$. This parameter vector $\mathbf{p}$ is of length $n+1$, and contains the polynomial coefficients in descending powers as in right hand side of equation 5.3 which is obtained through least squares estimator method using QR decomposition of the Vandermonde matrix constructed from t.

$$y(t, \mathbf{p}) = p_1 t^n + p_2 t^{n-1} + \ldots + p_n t + p_{n+1} \tag{5.3}$$

when $y(t, p)$ is the prediction of data at time t.

For each of the m candidate models of order n, the parameter vector $\mathbf{p}$, the predicted data $y(t, \mathbf{p})$, and the WRSS are calculated. The Akaike information criterion (AIC) is then applied (Akaike [1974]) using equation 5.4 to select the optimal model from the set of possible candidates in terms of polynomial order that best fits the data.

$$AIC = Nln(WRSS) + 2n \tag{5.4}$$

The minimal AIC value represents the optimal model with order n and parameter $\mathbf{p}$, and implies that the model with this set of parameters will provide best results in calculating the ME using equation 5.1.

5.2.2 Method

The method is implemented in MATLAB (v7.9.0529, R2009b; `http://www.mathworks.com/`) with an easy to use graphical user interface (GUI) for facilitation of its use among non–programming background users (shown in figure 5.6). This method is a part of the SigMate software package (Mahmud *et al.* [2010a]) which will be disseminated to the neuroscience community shortly under the GNU GPL (`http://www.gnu.org/licenses/gpl.html`).

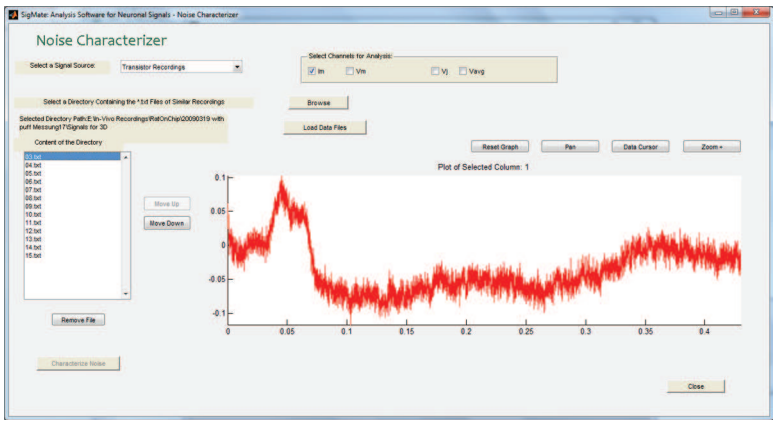


Figure 5.6: Graphical user interface of the noise characterization method.

Before starting the steady–state detection, the signals are scanned for presence of any artifacts caused by the stimuli. In some of the signals a slow stimulus arti-

fact is found and removed from the signals using a method proposed by Mahmud *et al.* [2009b] without causing distortion to the signal shape.

The signal is assumed to be described by model in equation 5.1, i.e., the ME is additive and Gaussian. In characterizing the noise for every sweep, detection of the exact starting point of the evoked response is very important due to the latency variance of evoked response among different sweeps. The following subsections describe the method in detail.

5.2.2.1 Detection of First–Steady–State

At first, the FSS is initialized with the prestimulus part (10 ms) of the signal. Subsequently, the signal is divided into parts of 3 ms interval, the first part is appended to the FSS (called cumulated FSS, cFSS), and the SD of the cFSS is calculated. Then iteratively the SD of the cFSS is compared to that of each signal part. If the SD of the current part is less than that of the cFSS, this part is appended to it; and its SD is determined for the next run of the iteration. But, if the SD of the current part is greater than that of the cFSS, the current part is fitted with a straight line. If this line exceeds the SD of the whole signal, the cumulated FSS is considered to be the final FSS; otherwise, the current part is appended to the cFSS. In this way the iteration is continued until the final FSS is obtained. The detailed algorithm of FSS detection is given in .2.2.

5.2.2.2 Detection of Second–Steady–State

The SSS provides a large data pool to better understand the signal quality. For its detection, the FSS is taken as a template. In the beginning of the detection process, the whole signal (excluding the FSS) is flipped horizontally and divided into smaller segments (called SSS–segment) according to the size of the FSS. Then, one–by–one the SD of the individual SSS–segment is compared to that of the FSS; the SSS–segments with smaller SDs are joined together to form the initial SSS. This process continues until a SSS–segment with greater SD is encountered. At this point, a second comparison is done between the SD of the next SSS–segment (referred as check–segment) and that of the FSS to avoid false spike (if any) caused by unavoidable sources. If the SD of the check–segment is less than

that of the FSS, the check–segment along with the previous segment is appended to the SSS and the iteration continues. Otherwise, the iteration is stopped and the SSS is determined as the previously formed one. The detailed SSS detection algorithm is shown in .2.3.

5.2.2.3 Characterization of Noise

Single Sweep Approach Having detected the FSS and the SSS, mathematical models (described in equation 5.3) are fitted to both the steady–states. The goodness of the model fitting implied by its optimal order is found by the minimum Akaike's Information Criterion (AIC) value calculated using the formula in equation 5.2 and equation 5.4. After finding the optimal fit, the MEs of the steady–states are calculated using the equation 5.1. Then the first order statistical information for the steady–states and their respective MEs are derived by individual calculation of the mean, and the SD. The statistical distributions of MEs for both FSSs and SSSs are computed and plotted using histograms, and estimation of distribution is done using a normal kernel function for better visualization purpose (Bowman & Azzalini [1997]).

Furthermore, this method also evaluates the quality of signals using two other approaches: 1. by averaging the detected steady–states, and 2. by averaging the calculated MEs. The following subsection describes these approaches. The step–by–step description of the algorithm is listed in the first part of the .2.1.

Averaged Steady–State and ME Approach Once the analysis of the single sweeps is done and the steady–states are available, they are averaged across different sweeps. Usually, the detected steady–states of various sweeps are of different lengths and this imposes a problem in averaging them. To overcome this, the longest steady–state is found for both the steady–states, and the shorter ones are zero–padded to have same length steady–states. Then the zero–padded steady–states are averaged across different sweeps to obtain averaged steady–states (called averaged FSS and averaged SSS). From the averaged steady–states first order statistical information (μ and σ) are calculated and mathematical model defined in equation 5.3 is fitted. MEs are calculated for both the averaged steady–states using the equation 5.1. The first order statistical information (μ

and σ) and distribution are calculated for the respective MEs.

Again, from the single sweep results, the MEs are calculated for each steady–states (both FSS and SSS). As the calculated MEs of various sweeps have different lengths, they are zero–padded as before. These zero–padded ME–FSSs and ME-SSSs are averaged across all the sweeps and first order statistical information (μ and σ) and the distribution are calculated and reported to the user.

5.2.3 Results and Discussion

The method is tested on different datasets for a range of candidate models with m = 6 and model order n = 2 to 7. Except for a few signals it provides a successful noise characterization. The cases where it fails are the highly oscillatory signals with high SD. In a pool of 65 datasets, the algorithm failed to detect the exact evoked response for 5% of the signals. As the signals from 16 FETs are simultaneously recorded, it is assumed that the experimental conditions for all the FETs are simular. Thus, to assess the quality of the recordings from an experiment, steady–states from all the simultaneously recorded signals are not required; therefore, the failed cases are negligible and don't harm the assessment performed by the method.

The noise characterization results upon application of this method from a few representative experiments (with simultaneous recordings from 13 to 16 FETs) are presented here. We show figures of one FET recording for both the FSS and SSS to demonstrate the method's workability pictorially, but we tabulate the statistical information for all the 16 FETs.

Figure 5.7 shows a representative result of the detection of the FSS (left). It shows the raw trace (in black) with detected FSS (in green), and the ME is calculated by equation 5.1 (in purple). It is noticeable that the ME is boxed by the data bounds around $\pm 2\sigma$ from the mean. This data bounding is imposed to reduce the deviation of mean from zero considering the less number of data–points in the FSS. The histogram of the ME distribution and its density function is also seen (right). From the histogram and the density function it can be seen that the ME is having a *quasi*–Gaussian distribution satisfying the hypothesis made at the beginning of the analysis.

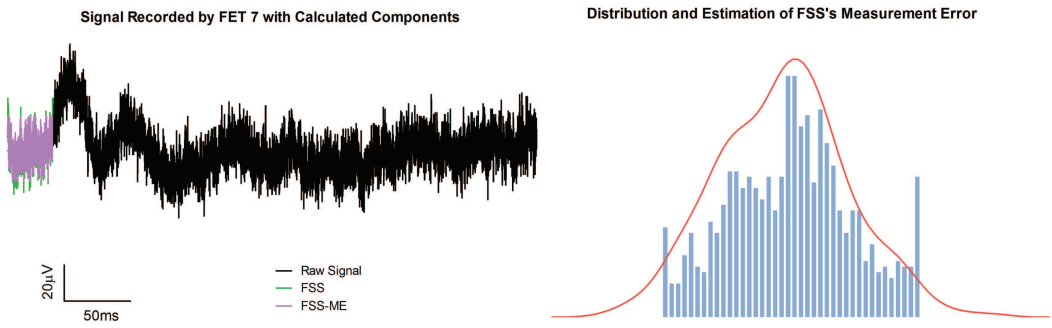


Figure 5.7: Left: figure showing the raw trace (in black); the detected FSS (in green); and the ME of the FSS, FSS–ME (in purple). Right: histogram of statistical distribution of FSS–ME and its estimated density function (*quasi*–Gaussian).

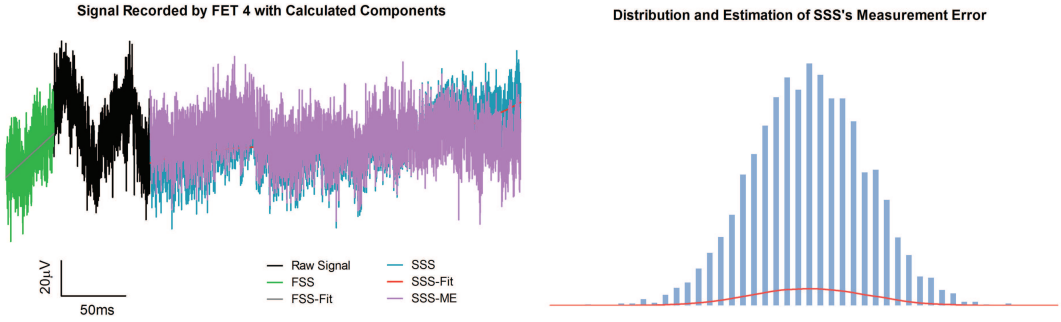


Figure 5.8: Left: figure showing the raw trace (in black); the detected FSS (in green); the SSS (in blue); fitted mathematical model, SSS–Fit (in red); and the SSS's ME, SSS–ME (in purple). Right: histogram of statistical distribution of SSS's ME and its estimated density function (Gaussian).

Figure 5.8 shows representative result of first and SSS's detection (left). It depicts the raw trace (in black) with detected FSS (in green), the fitted straight line for the FSS to facilitate the detection of the response–onset (in grey), the SSS (in light blue), the model generated signal using equation 5.3 for the SSS (in red), and the ME of the SSS (in purple). The FSS is fitted with the straight line to determine the starting point of the evoked response. The SSS is detected using FSS as a template and SD based comparison. The model fitting is seen as an oscillating wave (in red) due to the fact that the portion of the signal after

the evoked response usually contains spontaneous oscillatory brain activity uncorrelated to the given stimuli. The ME calculated in the region of the SSS has a Gaussian distribution demonstrating that the model in equation 5.3 can represent the spontaneous oscillatory brain activity. As the SSS is having relatively large number of data points compared to the FSS, the information provided by the SSS (ME and distribution) more profoundly exhibits the signal quality than that of the FSS.

In table 5.2.3 the means and SDs of the FSSs and the SSSs and their respective MEs are tabulated. These values are calculated using the single sweep approach. Analyzing these results we can notice that the means of the MEs for all the signals (μ_{me-fss} and μ_{me-sss}) are close to zero and the SDs of these MEs (σ_{me-fss} and σ_{me-sss}) are consistent around 0.01.

Table 5.1: Mean and SD of FSS, SSS with their MEs

FET	μ_{fss}	σ_{fss}	$\mu_{\text{me-fss}}$	$\sigma_{\text{me-fss}}$	μ_{sss}	σ_{sss}	$\mu_{\text{me-sss}}$	$\sigma_{\text{me-sss}}$
FET01	0.0067	0.0123	0.0831×10^{-3}	0.0118	0.0016	0.0153	9.397×10^{-15}	0.0108
FET02	0.0038	0.0114	0.0618×10^{-3}	0.0109	0.0038	0.0118	1.091×10^{-12}	0.0110
FET03	0.0013	0.0105	0.0750×10^{-3}	0.0101	0.0068	0.0101	1.513×10^{-13}	0.0096
FET04	0.0098	0.0144	0.0036×10^{-3}	0.0139	0.0014	0.0144	1.612×10^{-14}	0.0119
FET05	0.0044	0.0133	0.0045×10^{-3}	0.0128	0.0038	0.0140	9.183×10^{-15}	0.0120
FET06	0.0056	0.0126	0.1397×10^{-3}	0.0121	0.0012	0.0117	7.995×10^{-9}	0.0111
FET07	0.0012	0.0089	0.0394×10^{-3}	0.0086	0.0004	0.0085	2.580×10^{-10}	0.0082
FET08	0.0038	0.0098	0.0935×10^{-3}	0.0095	0.0019	0.0104	1.268×10^{-13}	0.0093
FET09	0.0062	0.0118	0.0671×10^{-3}	0.0113	0.0027	0.0110	2.213×10^{-15}	0.0092
FET10	0.0015	0.0104	0.0404×10^{-3}	0.0100	0.0023	0.0107	2.356×10^{-15}	0.0101
FET11	0.0134	0.0133	0.0068×10^{-3}	0.0128	0.0007	0.0146	8.215×10^{-15}	0.0116
FET12	0.0004	0.0109	0.0183×10^{-3}	0.0107	0.0028	0.0139	5.100×10^{-13}	0.0119
FET13	0.0029	0.0108	0.0545×10^{-3}	0.0104	0.0010	0.0122	6.045×10^{-15}	0.0108
FET14	0.0060	0.0113	0.0293×10^{-3}	0.0108	0.0007	0.0101	9.123×10^{-9}	0.0101
FET15	0.0079	0.0102	0.1083×10^{-3}	0.0098	0.0063	0.0009	6.182×10^{-8}	0.0110
FET16	0.0083	0.0118	0.1523×10^{-3}	0.0112	0.0088	0.0108	5.653×10^{-16}	0.0099

Legend: FET: Field Effect Transistor, μ_{fss}: mean of the FSS, σ_{fss}: SD of the FSS, $\mu_{\text{me-fss}}$: mean of the FSSs ME, $\sigma_{\text{me-fss}}$: SD of the FSSs ME, μ_{sss}: mean of the SSS, σ_{sss}: SD of the SSS, $\mu_{\text{me-sss}}$: mean of the SSSs ME, $\sigma_{\text{me-sss}}$: SD of the SSSs ME.

Figure 5.9 shows bar graphs comparing the means of FSS and SSS with their respective MEs for signals recorded during another experiment. The results are similar to table 5.2.3, i.e., the means of the MEs are significantly smaller compared to those of the steady–states (the y–axis scale is log_{10} based). Also, the standard deviations of the steady–states and the MEs are stable in the range of 0.01 to 0.015 (seen in figure 5.10).

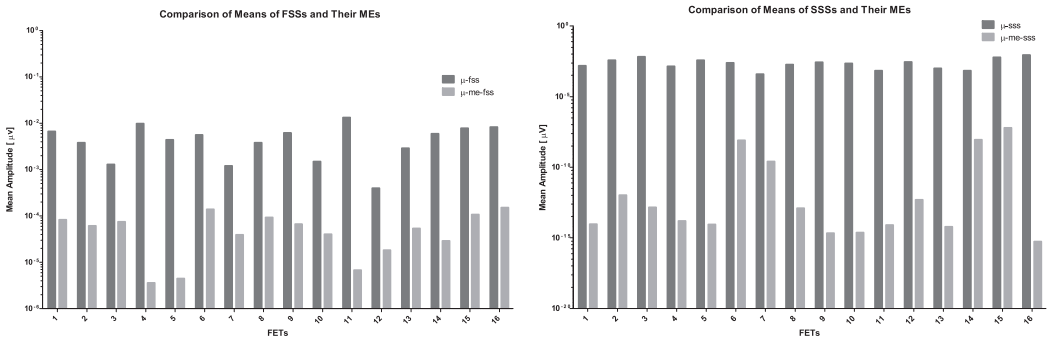


Figure 5.9: Graphs showing means of FSSs and their MEs (left); SSSs and their MEs (right). The y–axis scale is log_{10} based.

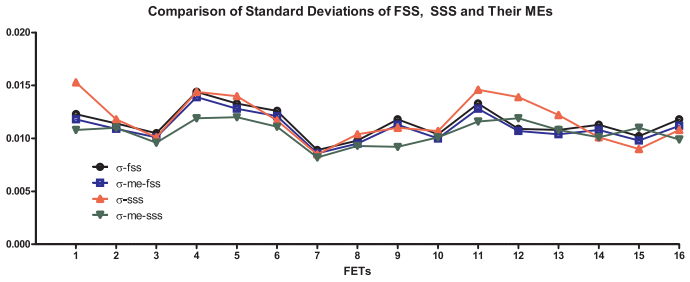


Figure 5.10: Standard deviations of FSS, ME–FSS, SSS, and ME–SSS.

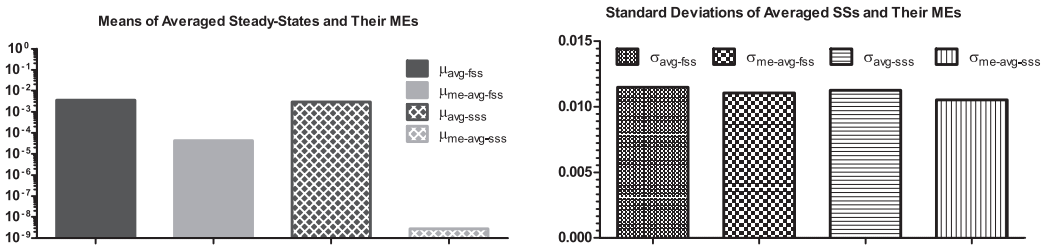


Figure 5.11: Left: means of averaged FSS and SSS with their respective MEs. Right: Standard deviations of Averaged FSS, SSS, and their respective MEs.

After analyzing the single sweeps, the averaged steady–states and averaged MEs are analyzed. Figure 5.11 (left) shows the means of the steady–states obtained by averaging the FSSs and the SSSs in various sweeps and their MEs

calculated by fitting models to them. It is noticeable that the means of the MEs are relatively smaller and the SDs of them are steady around 0.012 as obtained using the single sweep approach (seen in figure 5.11, right).

The results presented above show that the means of the MEs of the steady–states are very close to zero and the signals satisfy the assumption of Gaussianity which imply good quality of the recorded signals. Therefore, the noise characterization performed by the method helps the user to determine the quality of the recorded signals quickly. Also, the presented analyses above show the method's reliable workability for the cortical surface signals recorded by high resolution brain–chip interface. This method is able to provide first order statistical information and distribution of the noise present in the recorded signals to the user helping them to decide whether more detailed and specific noise analysis is required or not.

5.3 Latency Estimation and Layer Activation Order Detection

To explain brain activity underlying perception as the outcome of elementary neuronal responses is one of the major challenges of sensory systems neuroscience. Through "whisking", rodents make extremely fine discriminations of the environment, e.g., object localization, basing on shapes and textures of the objects (Ahissar & Knutsen [2008]). The mammal cortex shows a high degree of areal and laminar differentiation and also a representation of sensory surfaces. Especially for the rodents there is a precise topological map of the mystacial pad in the S1 cortex, in which for each whisker there is a so called "barrel" that receives the tactile information (Diamond *et al.* [2008]). Barrels play a very important role in segregation, integration and transmission of sensory information as sensory innervations at each whisker follicle are numerous (larger follicles receives terminations from approximately 200 trigeminal ganglion cells and the smaller follicles about 50) (Fox [2008]). During transmission of the information different layers of the barrel cortex are activated at different times. Studies have shown that intra- and transcolumnar microcircuits in the barrel cortex segregate and in-

tegrate information during this activation (Schubert *et al.* [2007]) and that these microcircuits have specific understanding of 'what', 'where' and 'when' aspects of the tactile information acquired by the whiskers.

To have precise knowledge about this information processing pathway by means of extracellular recording and offline signal analysis, an automated, reliable, and quick method is required. Indeed, scientists commonly perform this work manually spending lot of time especially when signals are recorded using neural probes with multiple recording sites.

In this work, we present an automated, simple to implement and computationally efficient method (computational complexity $O(n^2)$) capable of detecting various events (E1–E4) that characterize the LFPs recorded from different layers of the barrel cortex upon mechanical whisker stimulation. Latencies of the different events from the stimulus–onset are determined and the activation order of the cortical layers is calculated using the latency of E2 (i.e. the highest negative peak).

Generally, however, scientists determine the cortical layer activation order based on the current flow through the cortical layers by calculating the current source density corresponding to the LFPs. Therefore, this analysis is also implemented in the program, thus allowing for automated calculation of the layer activation order from CSDs obtained using the δ-source inverse current source density (δ-Source iCSD) method (Pettersen *et al.* [2006]). The program first calculates the latency of the first sink's peak from the stimulus–onset of each CSD, then it groups the recordings layerwise and stores the minimum latencies corresponding to each layer in an increasingly ordered list. The layer activation order is determined automatically by taking the minimum latency of each layer. The program was tested on LFPs measured from the rat barrel cortex under whisker stimulation. Resulting CSDs and layer activation order were comparable with previously recorded data (Jellema *et al.* [2004]) and compatible with the intracortical network architecture of the barrel cortex (Fox [2008]). We found that the activation order estimated using the LFPs and CSDs are similar. Also, automated results on layer activation order using LFPs were supported by an in-depth manual analysis of the same data samples.

5.3.1 Method

5.3.1.1 Determining Cortical Layer Activation Order Directly from LFPs

This method is implemented using the MATLAB (`http://www.mathworks.com`) scripting with an easy to use Graphical User Interface (GUI). The figure 5.12 shows the GUI that encapsulates the implementation for the ease of use of the non-programming background users. The figure 5.13 shows the flowchart of its basic operational steps (Mahmud *et al.* [2010b]).

The method takes the signal files recorded from the rat barrel cortex upon whisker stimulation as input. For each file it calls a module (the flowchart of the module is shown in figure 5.14) capable of detecting the events present in that signal and calculating the latencies from the starting of the evoked response.

The layer of recording is determined based on the signal characteristics and amplitude of the events. Finally the activation order of different cortical layers in the barrel column is determined by sorting the layerwise minimum latencies of the second event (E2).

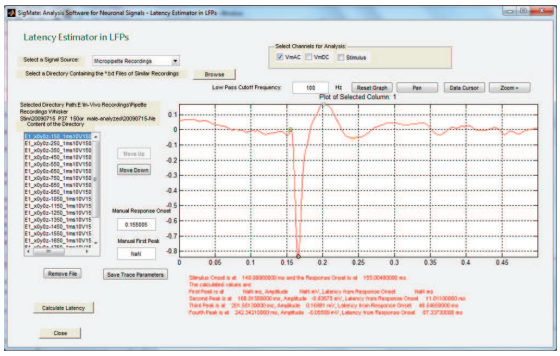


Figure 5.12: GUI of the layer activation order calculation method using LFPs. This GUI provides an easy way for the non-programming background users to use the method in analyzing their data obtained from experiments.

The flowchart of the module used in detection of events and calculation of latencies is shown in figure 5.14. In this module, firstly the signal is low-pass filtered with 250 Hz cutoff frequency and is translated by setting the signal amplitude at

the stimulus-onset to zero. This translation helps in avoiding the slow deviation of signal that might obscure the real amplitude of the events. The calculation of the latencies is based on the detection of various signal events (see Sec. 3.4.2.2) by calculating signal derivatives. A major change in the derivatives gave us the opportunity to detect an event.

The event detection starts with the detection of the response–onset, which is considered as the starting point of the evoked response. To detect the exact response–onset, the standard deviation of the signal's steady-state (the signal before the stimulus-onset) is calculated. The signal from the stimulus-onset to the next 10 ms is divided into very small parts (0.5 ms duration), and derivatives of these parts are calculated. The response–onset is the time instance of the signal when a small part's derivative is found to exceed ± standard deviation of the steady–state.

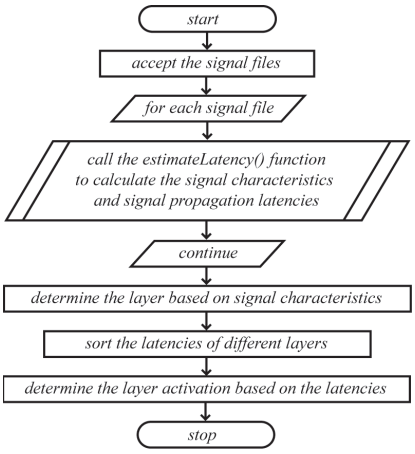


Figure 5.13: Flowchart showing the operational steps of the layer activation detection method using LFPs.

The events are time locked, which means that a change in the signal derivative in a particular time window (either from up to down or from down to up) denotes a particular event. Thus, dividing the signal (from the response–onset till the end of the signal) into smaller parts and then scanning for change of derivative is used to detect event occurrences.

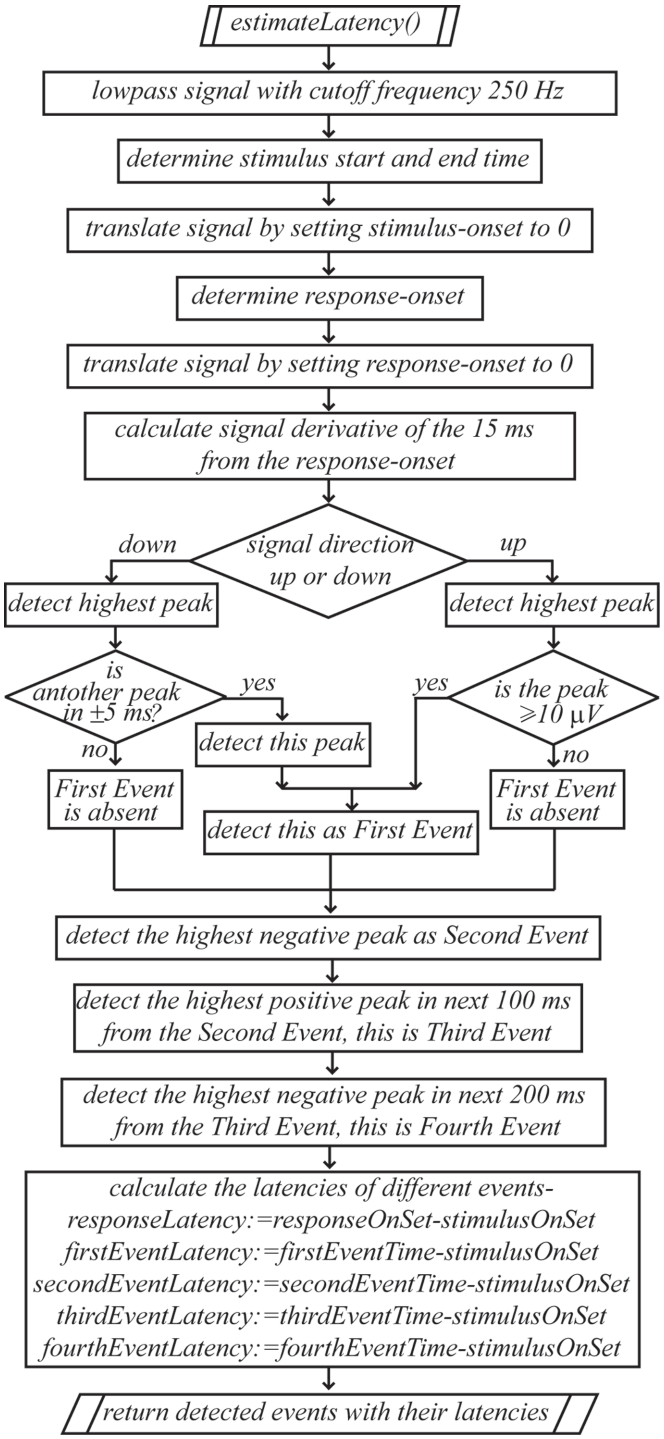


Figure 5.14: Flowchart of the event detection and latency calculation module.

Special care is taken in case of the first event, which may or may not be present in a signal and if present, may have either positive or negative direction. In case of the positive first event, a threshold of 10 μV is set to make sure that it indeed is an event and not just background spontaneous brain activity. If the signal is found to be going down, then the maximum negative peak is found and from this peak the signal ranging $\pm$ 5 ms is scanned for occurrence of yet another negative peak. If this second negative peak is found, the first event is set as the first occurring negative peak and the second event is the second negative peak, otherwise, the first event is absent and the second event is the maximum negative peak.

The detection of the rest of the events is very straightforward. It has been empirically found that the next event (slow positive peak) occurs within the next 100 ms of the second event and the last event (slow negative valley) within the 200 ms of the previous event.

Once the events are detected, latencies are calculated by subtracting the occurrence time of the events from the stimulus–onset time. The signal characteristics and the latencies are saved in a file for further processing.

After the latencies are calculated for all signal files, they are assigned to the cortical layers from where the signals were recorded from basing on a priori position information. Minimum latencies associated to each layer are then found and sorted in ascending order to determine the order of cortical layers activation.

5.3.1.2 Determining Cortical Layer Activation Order using CSD

Due to the widespread use of current source density (CSD) analysis to obtain the layer activation order, we implemented also this approach in our program. To calculate the CSDs, we considered the δ-Source Inverse CSD method (δ-Source iCSD) as explained in the next subsection. Figure 5.15 shows the MATLAB graphical user interface that generates the CSD profile from the LFPs and calculates the layer activation order.

Figure 5.15: GUI of the layer activation order calculation method using CSDs.

The δ-Source iCSD method The method, which has been adopted from (Pettersen *et al.* [2006]), divides the cortex to infinitely thin current discs each of radius R with constant planar CSD, C_p. For every recording site there is a disc with a determined C_p that lies in the xy plane. In this way, we have a δ function in the z-direction, whose value is C_p at the recording site and zero between two consecutive recording sites. The potential $\phi(z)$ at the center of a disc positioned at the position z' with recording pitch of h is given by:

$$\phi(z, z') = \frac{h}{2\sigma}(\sqrt{(z - z')^2 + R^2} - |z - z'|)C \tag{5.5}$$

where $C = C_p/h$ is the equivalent volume CSD, that corresponds to the CSD obtained if the planar current was distributed in a box of height h in which the disc is embedded and σ is the conductivity tensor in the rat brain (default value is considered as 0.42 S/m as experimentally reported by Sekino *et al.* [2009]). It is assumed that potential at position z_j is due to the sum of contributions from the various discs positioned at the recording electrode's contact points, and can be calculated using:

$$\phi(z_j) = \sum_{N}^{i=1} [\frac{h}{2\sigma}(\sqrt{(z_j - z_i)^2 + R^2} - |z_j - z_i|)]C(z_i) \tag{5.6}$$

$$= \sum_{N}^{i=1} F_{ji}C(z_i) \tag{5.7}$$

where F is an $N \times N$ matrix, denoting that from N values of LFPs we obtain N values of CSD. Furthermore, the elements of F are affected by the appropriate value of the current disc's radius (R) (experimentally reported values are: 300 μm and 200 μm by Brett-Green *et al.* [2001] and Alloway [2008], respectively; default value for the method is considered to be 250 μm) and are given by:

$$F_{ji} = \frac{h}{2\sigma}(\sqrt{(z_j - z_i)^2 + R^2} - |z_j - z_i|) \tag{5.8}$$

Now the CSD profiles can be estimated by inversing the matrix F and multiplying with the calculated potentials at various recording positions:

$$\hat{C} = F^{-1}\phi \tag{5.9}$$

Preprocessing Before applying the CSD analysis, raw LFPs are low-pass filtered using a Butterworth filter with cutoff frequency of 250 Hz. This is done to remove the high frequency components present in the LFPs that would cause unexpected oscillations in the calculated CSD profile. Moreover, resolution of CSD calculation depends on the recording pitch, with resolution improving by reducing the pitch. Therefore, to facilitate calculation of CSDs with large recording pitches, often interpolation of LFPs is performed (Rappelsberger *et al.* [1981]). During higher order interpolation high spatial frequency noise occurs especially in the neighborhood of the boundary points. To reduce this high spatial frequency noise a symmetrical, weighted average of the LFP about a given point is applied with the form of equation 6 (Hamming filter) (Szymanski *et al.* [2009], Ulbert *et al.* [2001]).

For signals recorded using neural probes with multiple recording sites separated by a small pitch (simultaneous recording producing a depth profile) the

Hamming filter is not necessary. However, it may be applied for removing spatial noise (if any) present in the recordings. In situations where recording is done at different times at different sites (using micropipettes or any other extracellular electrode), Hamming filtering is applied under the assumption that the physiological response of the animal and the stimulus do not change during the recording session, i.e., the response to the stimulus at a specific depth is reproducible over the whole experiment.

$$\phi(z) = 0.23\phi(z-h) + 0.54\phi(z) + 0.23\phi(z+h) \tag{5.10}$$

After applying the Hamming filter only $N-2$ interior recordings can be considered excluding the first and last recordings.

Detection of Layer Activation Order using CSD After the CSD profile is computed, the sources and sinks for the individual recording site can easily be viewed. The calculation of the sinks' latencies is done by subtracting the time instance of the stimulus-onset from the time instance of the peak of the first sink.

Once the latencies are calculated for the whole CSD profile, recordings for each layer are grouped together and the minimum latencies are selected. To determine the layer activation order, minimum latencies are sorted in ascending order and assigned to the different layers depending on recording depths known a priori. Latencies can be used to gain information on signal propagation within intracortical networks. Representative LFPs recorded from the rat barrel cortex under whisker stimulation and corresponding CSDs computed by the program are shown as an example in figure 5.16 (A, B). Hypothetical signal propagation pathways across the barrel intracortical network are inferred from the CSD profile and the latencies temporal pattern as seen in figure 5.16 (C) (Fox [2008], Jellema *et al.* [2004]).

In the figure 5.16 (C), the arrows indicate either signal propagation (alphabet-to-alphabet: propagation of sinks; number-to-number: propagation of sources) or directed inward current (number-to-alphabet: current flow from source to sink) according to the nomenclature adapted by Jellema *et al.* [2004]. From the analysis of the CSD profile and latencies it is inferred that, there are two dominant sink-

source complexes. The first one is initiated at the upper part of layer Va (sink 'a') and ending at layer I (sink 'f'); the second one is initiated at the lower part of layer Vb (sink 'g') and ending at layer VI (sink 'l'). The two complexes are assumed to be caused by the monosynaptic thalamic input (Fox [2008]) and are initiated through the sinks 'a' and 'g' after whisker stimulation. The first complex is supposed to be initiated by the pyramidal cells situated in the upper part of layer Va. This complex is propagated through neurons in the layer IV, III, and II. The axons of these cells are projected towards layer III (in case of Va) and layer I (in case of IV, III, and II). Propagation of this complex created huge sinks (sinks 'a' to 'f' in the CSD profile) and sources (1, 4, 3, and 6). The second complex is supposed to be initiated by pyramidal cells situated in the lower part of layer Vb and propagated through the lower portion of layer Va. During this propagation sinks 'g' to 'k' with increasing amplitude are generated. Indeed, the sink 'g' at the beginning of propagation is relatively small. Afterwards, sinks increase in amplitude and width possibly due to the fact that pyramidal cells involved in propagation receive other excitatory inputs from layer IV. Sources 2, 5, 7, 8, and 9 are associated to sinks 'g', 'h', 'i', 'j' 'c' and 'k', respectively. In layer VI and deeper polysynaptic delayed inputs caused additional sinks (sinks 'l').

Latencies indicated that the signals propagated through a pathway starting from Vb and then traversing through Va, III, IV, II, I and to VI. This order of layer activation is supported by the known neuronal architecture of the barrel cortical layers. This architecture suggests that the thalamic inputs (VPm) activate the layer Vb/IV, signals propagate through layer III and II from where the outputs are projected to layer Va, Vb. From these layers the output is sent to the layer VI and then back to the thalamus.

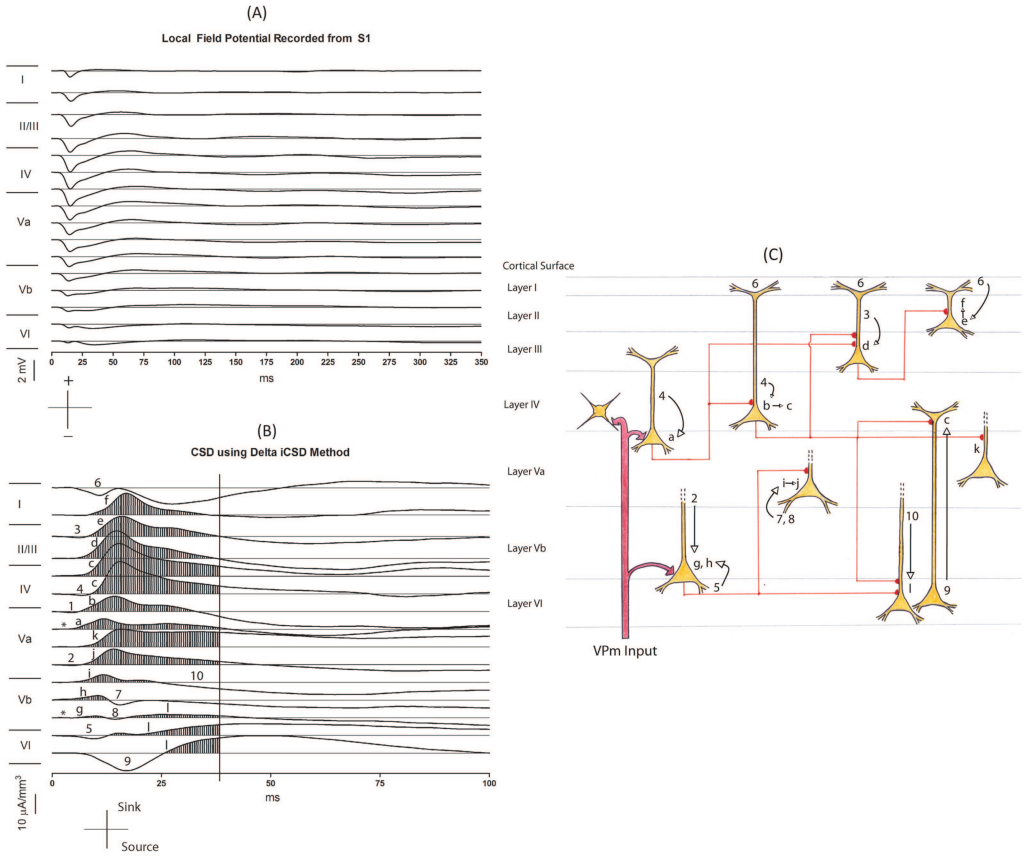


Figure 5.16: (A): Depth profile of recorded LFPs. (B): The respective CSD profile computed using δ-source iCSD from the LFPs. The hatched portions of the profile denote the sinks (a–l) and the negative portions the sources (1-10). Stars indicate the initiation sites of the current flow within the cortex. (C): Barrel column architecture derived from previous studies (Fox [2008], Jellema *et al.* [2004]) showing the possible connections among neurons in different cortical layers. Wires represent schematically excitatory connections.

A simplified architecture of a barrel column reconstructed from Fox [2008] is depicted in figure 5.17. Signal propagation estimated from CSD analysis matches, to a certain extent, the pathway through the single barrel. Of course, as the dynamics of individual events involved in signal generation and propagation through the network is largely unknown, the predictory potential of this network model regarding circuit activation remains limited.

However, a possible interpretation of the architecture in figure 5.17 is given. When a whisker is stimulated, the information first goes to the thalamus, and then from the thalamus to the corresponding barrel. As it can be seen from the picture, there are two principal thalamic inputs (VPm) that are in layer IV and at the border between layers Vb and VI. The thalamic inputs in layer IV activate both excitatory (represented by stellate cells) and inhibitory cells (represented by basket cells). These basket cells provide feedforward perisomatic inhibition from the VPm and feedback inhibition to the excitatory stellate cells. The LTS (low threshold spike) cells do not receive a thalamic input directly, so they are involved only in the feedback inhibition. The excitatory output from here is then projected to layers II/III. It should be noted that there are connections between inhibitory cells, in order to increase inhibition, and between the excitatory cells themselves. From the literature, it is known that stellate cells connect mainly with other stellate cells and pyramidal cells with other pyramidal cells. The excitatory cells of the granular layer (IV) then project to the supragranular layers (II/III). The connections between layer IV and layers II/III are numerous and strong, which may reflect the large amplitude of the sinks in these regions. Even in this case, basket cells provide both feedforward and feedback inhibition as the chandelier cells project to the axon initial segment of pyramidal cells. The excitatory outputs from these pyramidal cells are projected to layers V/VI. The output of the supragranular layers becomes the input for the infragranular layers, i.e., layers II–Va and III–Vb reciprocally connect within a column. Pyramidal cells of layers V and VI can be inhibited by inhibitory cells of same or other layers, such as the Martinotti cells. Sensory information finally reaches layer VI, whose cells are reciprocally connected with layer Vb cells, and from there comes back to the thalamus. From the picture it can be noted that layer IV connects also with layer Va and VI cells. The dash–dot–dashed and dashed lines are the feedback connections that project from the layer Vb back to layer III and from the layer VI back to the inhibitory cells of layer IV, respectively.

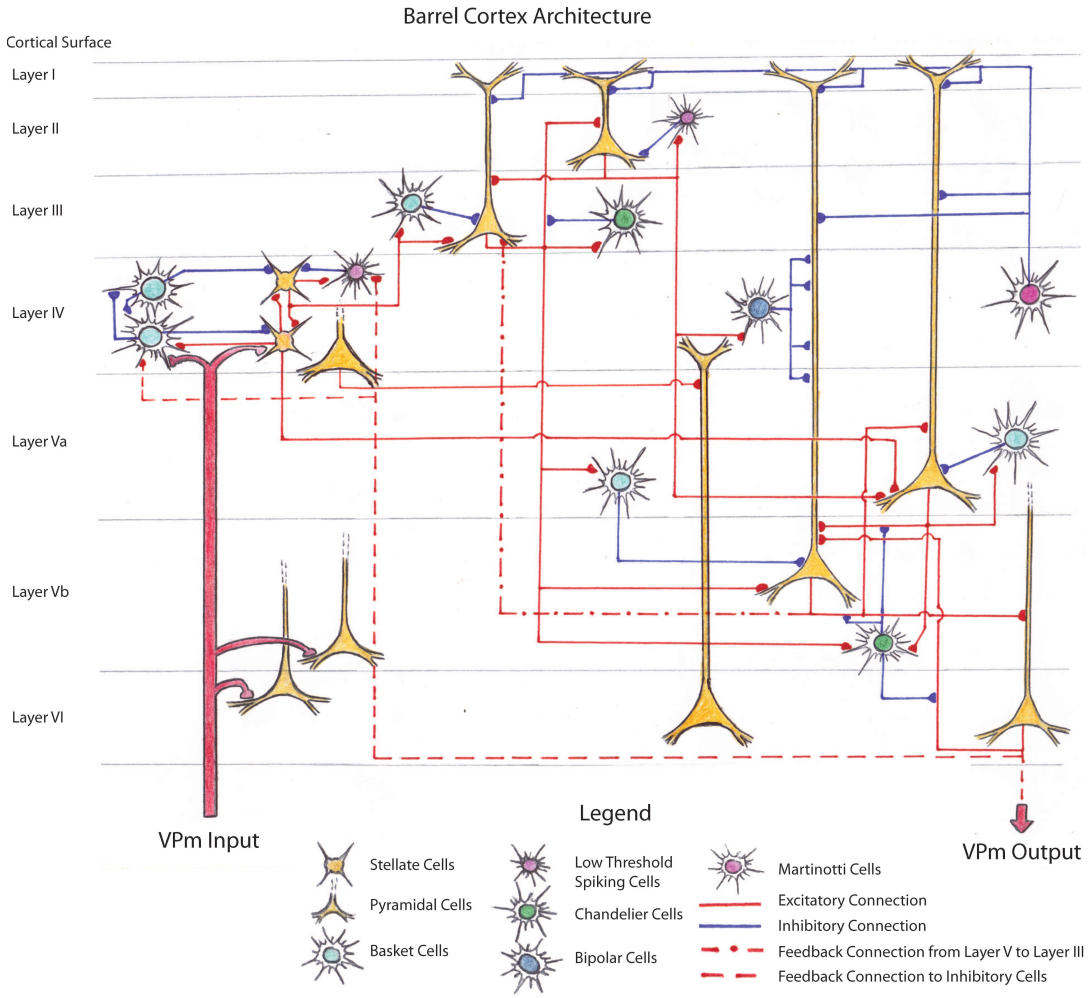


Figure 5.17: Simplified architecture of a barrel column as described in Fox [2008].

5.3.1.3 Manual Calculation of Cortical Layer Activation

As a proof of automated detection reliability, the layer activation order was also calculated manually from the LFP profile. Latencies from the stimulus–onset were estimated manually for each event (E1–E4) using a commercial software (clampfit, v.10.0, `http://www.moleculardevices.com/Products/Software/Electrophysiology/pCLAMP.html`). This software provides time instance and amplitude of the signal's data point where the data–cursor was placed. Thus, the events were pointed manually one–by–one and the time instances noted. The latencies of the events were calculated by subtracting the stimulus–onset from the time instance of each

event. Again, only E2 latencies were considered for calculation of the layer activation order. Then, E2 latencies were grouped layerwise and the minimum latency in each layer was found. These minimum latencies were then sorted in ascending order to determine the signal propagation among the layers. A comparison between automated and manual detection results is shown in the Results and discussion section.

5.3.2 Results and Discussion

The event detection algorithm described in section 5.3.1.1 has a computational complexity of $O(n^2)$. Another method proposed for event detection in LFPs (Bokil *et al.* [2006]) was based on an algorithm with higher computational complexity $O(\log n!)$. Furthermore, algorithms used to detect PQRST complexes in ECG signals (Dota *et al.* [2002], Dota *et al.* [2009], Piotrowskia & Rozanowski [2010]) can be adapted to detect the events present in LFPs; however, these algorithms have computational complexities of $O(n^3)$.

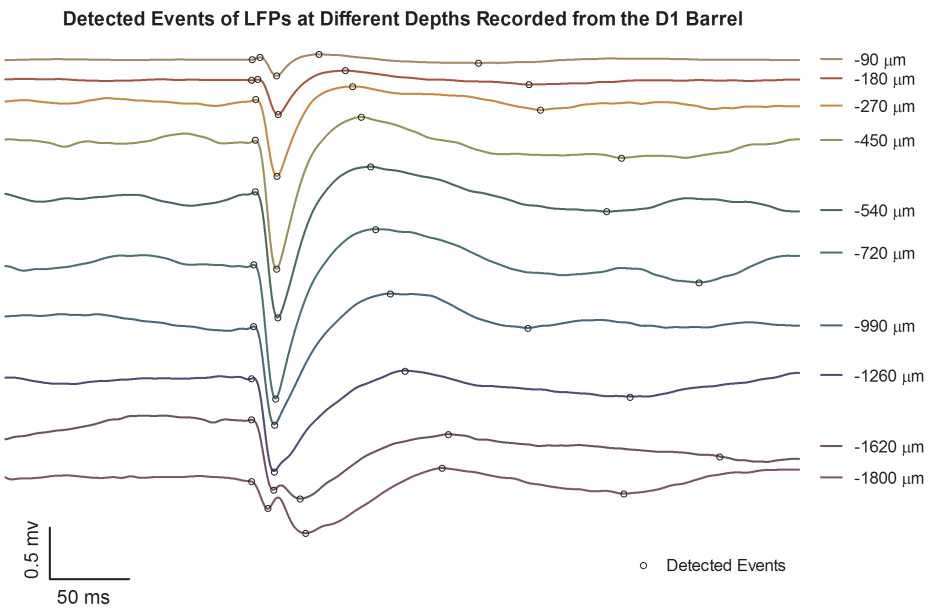


Figure 5.18: LFP depth profile with detected events using the method mentioned in section 5.3.1.1. The signals were recorded equidistantly (90 μm pitch). For better visualization only representative signals from each layer are shown.

5.3.2.1 Single Experiment

The method was applied to a number of datasets and found to be working quite well except a few situations (2% of occurrence rate) where an error of ± 300 μs was noticed in latency calculation. Particularly, latency calculation error was occurring in case of signals containing slow stimulus artifacts (with frequency components less than 250 Hz). As latencies are in terms of a few milliseconds up to hundred of milliseconds, this error can be considered negligible. Figure 5.18 shows representative signals and their respective detected events after a run of the method.

When compared, the latency results for the layer activation order obtained from the LFPs and the CSD profile (figure 5.19 and figure 5.20) are found similar in terms of activation sequence, but not with respect to their values. The layerwise latencies of CSDs are larger than those of the LFPs. This is due to the fact that, in case of the CSDs, latencies are calculated as the difference between time instance of the first sink's peak and the stimulus–onset, whereas the latencies for the LFPs are calculated as the difference between the time instance of the E2 and the stimulus–onset.

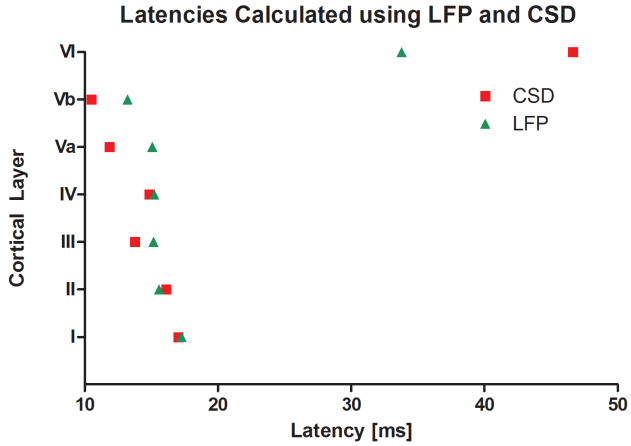


Figure 5.19: Comparison of layer-wise latencies calculated from the LFPs and CSDs.

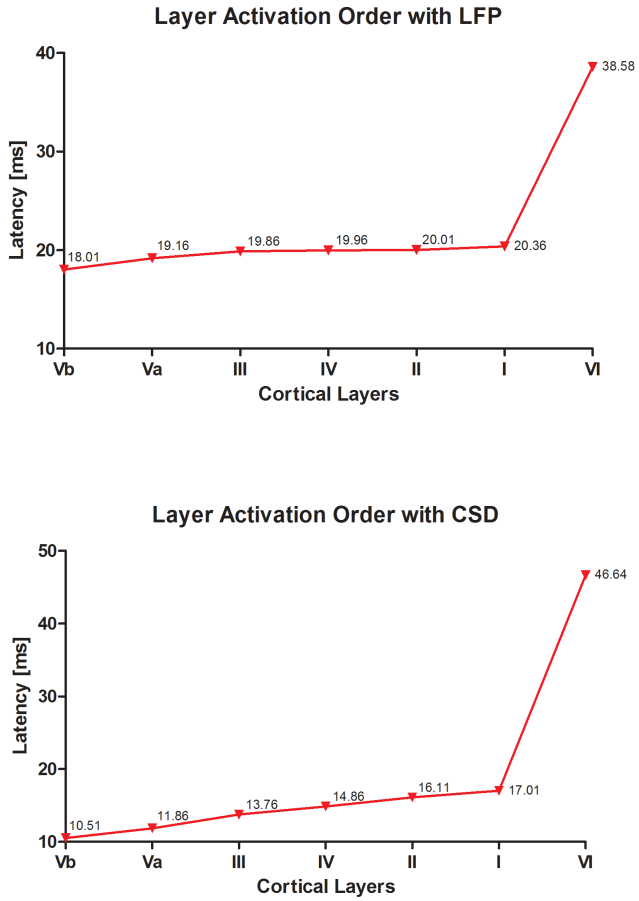


Figure 5.20: Layer activation order calculated using the LFP (top) and CSD profiles (bottom).

For determining the layer activation order using CSDs, a common reference point is required to calculate the latencies, which can be the exact initiation of the first sink (Mitzdorf & Singer [1980], Mitzdorf [1985], Kaur *et al.* [2005], Swadlow *et al.* [2002]) or the peak of the first sink (Di *et al.* [1990], Castro-Alamancos & Oldford [2002], Megevand *et al.* [2009], Staba *et al.* [2004]) or a combination of both (Szymanski *et al.* [2009], Jellema *et al.* [2004]). Due to the oscillations in the CSDs caused by calculation, it is difficult to detect the exact initiation of the first sink in each CSD using an automated method. Indeed, especially when the LFPs contain high spontaneous brain activity or oscillations the generated CSDs are too noisy and oscillatory. Filtering the LFPs or the CSDs

does not really eliminate possibility of miscalculation. Therefore, the difficulty in detecting the first sink's onset led us to consider the first sink's peak in calculating the latencies. Scientists need to perform the latency calculation manually in cases where a different reference point is required other than the first sink's peak.

Table 5.2: Comparison of manual and automatic calculation of latencies

Depth	Mode	Latencies			
		1st	2nd	3rd	4th
90 μm	M	5.384	19.784	42.934	144.954
	A	5.655	19.564	42.742	143.393
180 μm	M	Absent	19.745	60.055	174.215
	A	Absent	19.416	59.259	174.023
270 μm	M	Absent	19.905	64.795	180.965
	A	Absent	19.615	63.513	183.733
450 μm	M	Absent	20.215	69.395	232.835
	A	Absent	20.228	70.320	232.836
540 μm	M	Absent	20.075	74.205	221.595
	A	Absent	20.216	74.124	222.021
720 μm	M	Absent	20.645	79.895	283.305
	A	Absent	20.565	78.228	282.532
990 μm	M	Absent	19.375	87.805	220.125
	A	Absent	19.464	87.887	175.475
1260 μm	M	Absent	18.585	96.025	238.595
	A	Absent	18.213	96.046	239.489
1620 μm	M	16.1150	38.925	110.835	202.635
	A	16.116	38.785	112.562	198.448
1800 μm	M	10.175	38.585	118.825	234.975
	A	10.310	38.584	118.568	234.784

The latencies calculated by the automated method for the LFPs (depth profile can be seen in figure 5.18) were also compared with the manually calculated latencies and the results were found to be similar (table 5.2). 'M' denotes manual computation by hand and 'A' denotes automated calculation using the method. The '1st', '2nd', '3rd' and '4th' are the latencies of the respective events. As mentioned in section 3.4.2.2 in the upper layers we can observe the first positive peak, which gradually disappears in the middle layers and eventually becomes the first negative peak. Furthermore, table 5.3 reports average latencies for 3 different experiments evaluated manually and by the program with their root mean square errors (RMSE). The '1st', '2nd', '3rd' and '4th' are averaged latencies and RMSE of the respective events. The low RMSE indicates that the calculation of latencies using the automated method is accurate. The tables report data corresponding to representative signal(s) from depth(s) of each layer (please see figure 5.18).

Table 5.3: Average latencies of events using manual and automatic calculation with RMSE

Depth	Mode	Average Latencies 1st	2nd	3rd	4th	RMS Errors 1st	2nd	3rd	4th
90 μm	M	6.019	19.784	42.450	139.014	0.542	0.081	0.024	0.315
	A	6.592	19.564	42.201	140.047				
180 μm	M	Absent	19.745	67.547	178.850	Absent	0.092	0.021	0.221
	A	Absent	19.416	68.974	175.654				
270 μm	M	Absent	28.517	62.574	183.015	Absent	0.026	0.032	0.254
	A	Absent	28.428	65.051	187.373				
450 μm	M	Absent	25.591	74.102	221.301	Absent	0.062	0.028	0.253
	A	Absent	25.675	77.108	203.952				
540 μm	M	Absent	18.175	71.214	231.595	Absent	0.059	0.046	0.477
	A	Absent	18.318	72.980	213.741				
720 μm	M	Absent	20.145	72.985	210.745	Absent	0.048	0.094	0.351
	A	Absent	19.619	73.428	271.659				
990 μm	M	Absent	21.937	84.862	192.251	Absent	0.095	0.392	0.853
	A	Absent	22.121	90.957	183.241				
1260 μm	M	Absent	18.985	91.213	210.021	Absent	0.036	0.095	0.764
	A	Absent	19.018	91.478	228.674				
1620 μm	M	11.152	26.132	110.835	192.380	0.152	0.071	0.93	0.429
	A	10.920	25.925	112.562	181.154				
1800 μm	M	9.631	35.585	117.241	221.341	0.821	0.087	0.034	0.762
	A	9.927	35.885	113.231	214.114				

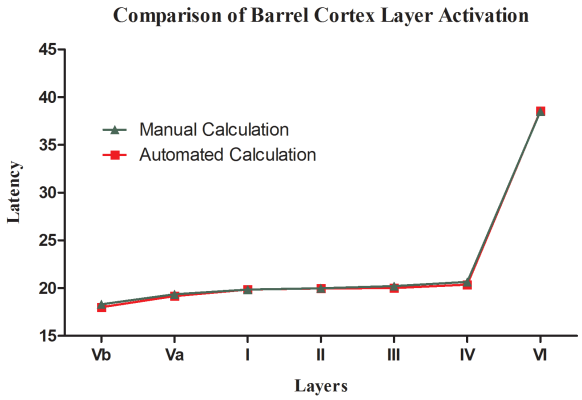


Figure 5.21: Comparison of manual and automatic method's latency calculation in finding the activation order of different cortical layers.

Also, activation orders of cortical layers based on manual calculation and by the method are exactly the same (figure 5.21). Therefore, results of table 5.2, 5.3 and figure 5.21 suggest that the automated method is accurate in detecting the various events present in the LFPs and in calculating latencies with precisions

that are sufficient for a reliable determination of the activation order of cortical layers.

In conclusion, basing on these evidences, we can assert that the automated method presented can calculate the activation order of layers in the barrel columns upon mechanical whisker stimulation in a single experiment. Both the approaches (using LFP or CSD) provide similar results; it is the choice of the user to decide which one to use based on the need.

5.3.2.2 Average Across Experiments

In addition to the single experiments we also applied our method to the grand average across three different experiments. This type of averaging technique has been adopted by previous studies to determine the temporal order of layer activation in the cortex (Di *et al.* [1990], Staba *et al.* [2004], Jellema *et al.* [2004]). During these experiments, the signals are recorded at a pitch of 90 μm by mechanically stimulating the D1 whisker. Thus, the signals were averaged across experiments (depthwise) to obtain a grand average LFP profile. This profile was then used to calculate the layer activation order using LFPs and CSDs.

To obtain the layer activation order from the grand average LFPs, the latencies of different events (E1–E4) were calculated and grouped layerwise. The minimum latency in each layer was found and sorted in ascending order, thus providing the activation order. The CSD profile was calculated using the grand average LFP profile through application of δ–iCSD method. Latencies were calculated considering the first sink's peak and were grouped layerwise. The minimum latency in each layer was found and sorted in ascending order to obtain the activation order.

Figure 5.22 shows the latencies obtained from the grand average LFP profile and the CSD profile obtained using the δ–iCSD method. Standard deviations of the means are shown as vertical bars. The latencies at different depths obtained using the grand average by the two methods show a temporal order of layer excitation comparable to previous studies (Di *et al.* [1990], Armstrong-James *et al.* [1992], Einevoll *et al.* [2007]).

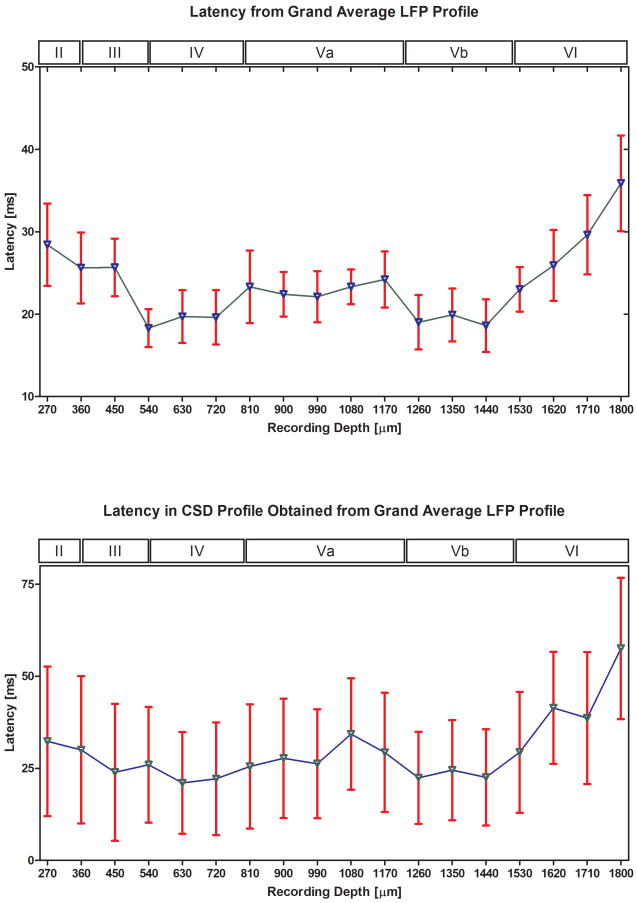


Figure 5.22: Latencies obtained from the grand average (n=3). Latencies calculated using LFP based method (top) and latencies calculated using CSD based method (bottom). The vertical bars show standard deviations of the means.

5.4 Clustering of Single Sweep LFPs

The availability of high resolution recording tools provide unprecedented information about the brain activity as a basis to understand the underlying neuronal networks (Maschietto *et al.* [2009]). These recording tools deliver huge amount of recordings containing spiking activity as well as field potentials generated in the brain area under investigation. To understand the signal propagation and processing pathways, scientists have relied on the evoked field potentials. Due

to the fact that the scientists use stimulus–locked field potentials to assess and understand the effect of stimuli on a brain area, the LFPs provide a finger–print of the stimulus's effect on the activation and inactivation of neuronal networks in the brain region under study (Legatt *et al.* [1980]). The conventional way of analyzing these LFPs is to record for a period of time and then obtain a stimulus–locked average. However, experimental studies have shown that the individual information provided by a single sweep may disappear if one considers an average over several runs under same stimulus conditions (van Hemmen & Ritz [1995]). Furthermore, to understand certain issues of the brain (for example, signal processing pathway and cortical layer activation order by Mahmud *et al.* [2010b]) and for certain operations (for example, current source density analysis) signal shape plays an important role (Okun *et al.* [2010]). It is implicated that different shapes in the single sweep signals denote different neuronal network activity. Thus a shape based classification method is necessary to extract different shapes present in a pool of single sweeps and to decipher the neuronal network activity from the LFPs. A wide range of research has been conducted in detection and sorting of neuronal spikes (Quiroga *et al.* [2004], Quiroga [2007]) but till date there is no method capable of performing similar sorting for the single sweep stimulus induced potentials.

In this work, we present a method for single sweep classification or sorting based on their shape. This sorting is done through four steps: smoothing of single sweeps, template generation, single sweep recognition, and classification of recognized single sweeps.

The LFPs recorded from a barrel column of the rat S1 cortex by stimulating the corresponding whisker can be differentiated by their layer–specific characteristics from where they are recorded (Ahrens & Kleinfeld [2004], Kublik [2004]). There are basically four events that characterize these signals. Event 1 (E1): small positive/negative peak; event 2 (E2): large and fast negative peak; event 3 (E3): relatively slow positive peak; event 4 (E4): slow negative peak returning to zero (Mahmud *et al.* [2010b]).

The smoothing is performed using nonlinear least square estimation to reduce the spatial oscillations and noise. Once the signals are estimated, the starting and end of the response is determined (response part). An average of these response

parts is usually considered as a template for signal recognition. This method makes use of the contour matching for recognition of the single sweeps. The contour of the template is compared to each of the single sweep's contour with a predefined boundary condition. If the single sweep falls within the boundary condition, the single sweep is considered to be recognized. Once the single sweep recognition is over, intelligent K–means clustering is applied on the recognized sweeps to classify them according to their shapes.

The classified or clustered single sweeps are then locally averaged and these averaged local LFPs show different shape and amplitude of the signal events that characterizes those signals. These different shapes and amplitudes provide us with an insight about the underlying neuronal network activity. Based on the shape information some of the important conclusions are made in decoding the signal processing pathways. However, the clustered averages of the single sweeps have revealed differences in event latencies and amplitudes, thus, demonstrating different network activity of the same area at different times over the same stimulus.

5.4.1 Method

5.4.1.1 Template Generation

As the single sweeps contain spontaneous neural oscillations and noise, it is often difficult to have precise information about the events of the signal. Thus, a(n) smoothing / estimation is performed to get rid of the spatial oscillations and noise. Gauss-Newton based nonlinear least square method is adapted for this purpose.

From the definition of least square by Madsen *et al.* [2004], for a given vector function $f : \Re^n \longmapsto \Re^m$ with $m \geq n$, we want to minimize $\|f(x)\|$ or equivalently find:

$$x^* = argmin_x\{F(x)\} \tag{5.11}$$

where

$$F(x) = \frac{1}{2}\sum_{i=1}^{m}(f_i(x))^2 = \frac{1}{2}\|f(x)\|^2 = \frac{1}{2}f(x)^T f(x) \tag{5.12}$$

Now adding a weight function (the covariance matrix of the prediction error, $\sum_\nu$) to equation 5.11 and equation 5.12 and a model in equation 5.13 to calculate the prediction error, an analytical solution of the problem (in equation 5.14) can be obtained.

$$x = y(x^*) + \nu \tag{5.13}$$

$$x^* = (y^T \sum\nolimits_\nu^{-1} y)^{-1} y^T \sum\nolimits_\nu^{-1} x \tag{5.14}$$

To solve the nonlinearity, parameter vector is assigned initial value at $x_k^*, k = 0$. Then using first order Taylor's expansion the model is linearized around the initial value. Thus the problem can be represented by equation 5.15.

$$\Delta x = P\Delta x^* + \nu \tag{5.15}$$

Now, linear formula can be used to estimate the parameters as in equation 5.16 and new parameter vector is obtained by equation 5.17. This iterative process is repeated until the cost function stabilizes or falls below a threshold.

$$\Delta x^* = (P^T \sum\nolimits_\nu^{-1} P)^{-1} P^T \sum\nolimits_\nu^{-1} x \tag{5.16}$$

$$x^*{}_{k+1} = x^*{}_k + \Delta x^*{}_k \tag{5.17}$$

The estimated signals are scanned for occurrence of the aforementioned events. In usual cases, the stimulus–onset defines the starting point and the end of response defines the end of the template. As all the signals don't have the same end of response, signals are zero–padded and averaged to obtain a template.

5.4.1.2 Single Sweep Recognition

Once the template is generated, the contour of the template is used to recognize the single sweeps. Boundary conditions (lower and upper bounds) are imposed to facilitate the recognition process. The variance vector (calculated using equation

5.18) of the template is necessary for calculating the boundary conditions.

$$V_{tmp} = \frac{1}{N}\sum_{i=1}^{N}[Sw_i(k) - Temp(k)]^2 \tag{5.18}$$

where Sw is the zero–padded and truncated single sweeps and $Temp$ is the template.

The upper and lower bounds are calculated using equations 5.19 and 5.20.

$$Up(k) = Temp(k) + (a * (V_{tmp}(k))^{\frac{1}{2}} + b) \tag{5.19}$$

$$Low(k) = Temp(k) - (a * (V_{tmp}(k))^{\frac{1}{2}} + b) \tag{5.20}$$

with a, b are constants; the values of a, b ($a = STD(Temp)$, and $b = 3 * STD(Temp)$) are determined empirically.

A signal is considered to be recognized, if and only if all of its data points lie within the range of the boundary conditions.

5.4.1.3 Clustering the Recognized Sweeps

For our purpose we used the intelligent K–means method of classifying recognized sweeps. It is an updated version of the classical K–means (Macqueen [1967], Bock [2007]). In the rest of the text the words classification and clustering are used synonymously.

The K–means method usually is applied to a dataset involving a set of N entities, I, a set of M features, V, and an entity–to–feature matrix $Y = (y_{iv})$, where y_{iv} is the value of feature $v \in V$ at entity $i \in I$. The method produces a partition $S = \{S_1, S_2, \cdots, S_K\}$ of I in K non–overlapping classes S_k, referred to as clusters, each with a centroid $c_k = (c_{kv})$, an M-dimensional vector in the feature space ($k = 1, 2, \cdots, K$). Centroids form set $C = \{c_1, c_2, \cdots, c_K\}$. The criterion, minimized by the method, is the within–cluster summary distance to centroids:

$$W(S, C) = \sum_{k=1}^{K}\sum_{i \in S_k} d(i, c_k) \tag{5.21}$$

with d is the Euclidean distance squared.

Given K M–dimensional vectors c_k as cluster centroids, the algorithm updates clusters S_k according to the Minimum distance rule: For each entity i in the data table, its distances to all centroids are calculated and the entity is assigned to its nearest centroid. Given clusters S_k, centroids c_k are updated according to the distance d in equation 5.21, $k = 1, 2, \cdots, K$. Specifically, c_k is calculated as the vector of within–cluster averages as d in equation 5.21 is Euclidean distance squared. This process is reiterated until clusters S_k stabilize.

However, this approach has a severe drawback that the cluster number, K is required to be supplied before start of the classification. To overcome this, we adapted intelligent K-Means (iK-Means) clustering method as proposed by Chiang & Mirkin [2010]. This iKMeans method uses an anomalous pattern (AP) to find out the appropriate number of clusters.

The AP algorithm starts from an entity, which is the farthest from the origin, as the initial centroid c. After that, a one–cluster version of the generic K–Means is utilized. The current AP cluster S is defined as the set of all those entities that are closer to c than to the origin, and the next centroid c is defined as the center of gravity of S. This process is iterated until convergence.

Finally, when the single sweeps are classified into their respective clusters, they are cluster–wise averaged for further processing.

5.4.2 Results and Discussion

The method is implemented in MATLAB (Version: 7.9, release: 2009b, `http://www.mathworks.com`). A Graphical User Interface (GUI) is also designed as seen in Figure 5.23 for the ease of non-programming background users. To check the method's workability it is applied on a number of datasets and the results are found satisfactory except some exceptional cases, when the signal morphology is completely different from that of the barrel cortex. Each dataset comprised of recordings from about 20 different depths, and each of them contained as many as 100 single sweeps. Except to demonstrate the distribution of single sweeps to different clusters, we present clustering results related to a representative set of single sweeps. However, the usefulness of this method is elaborated with some

findings revealed by the outputs of the method.

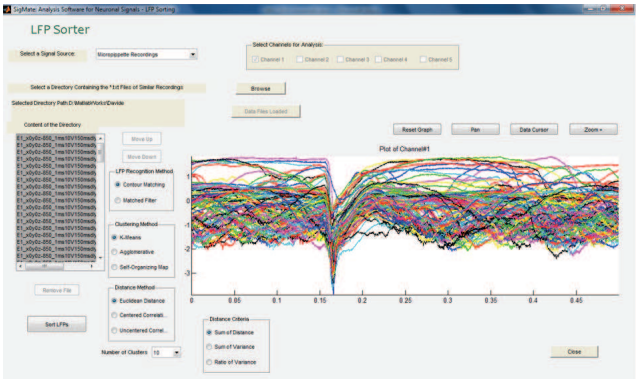


Figure 5.23: The GUI of the LFP sorting method with its components. The plotted 100 single sweeps of a recording session give an idea about the varied shapes that may be present in recordings.

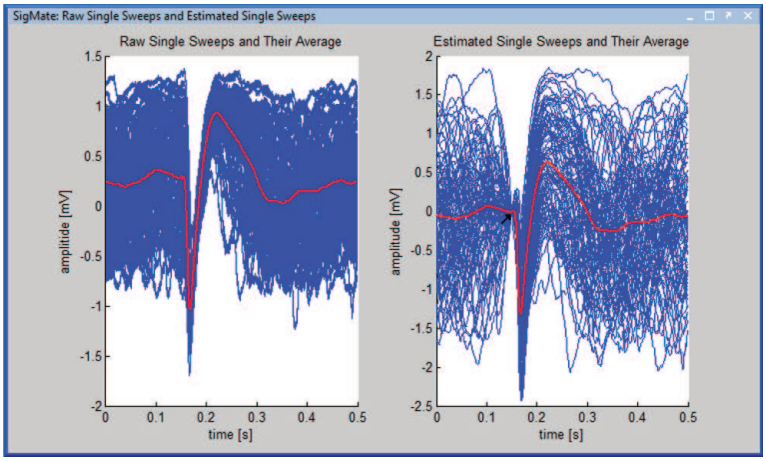


Figure 5.24: Single sweeps: on left, raw sweeps (without filtering or estimation) with average in red and on right, estimated sweeps with average in red. The arrow shows the stimulus–onset i.e., the starting point of the template. The noise in the raw single sweeps is evident in the left figure.

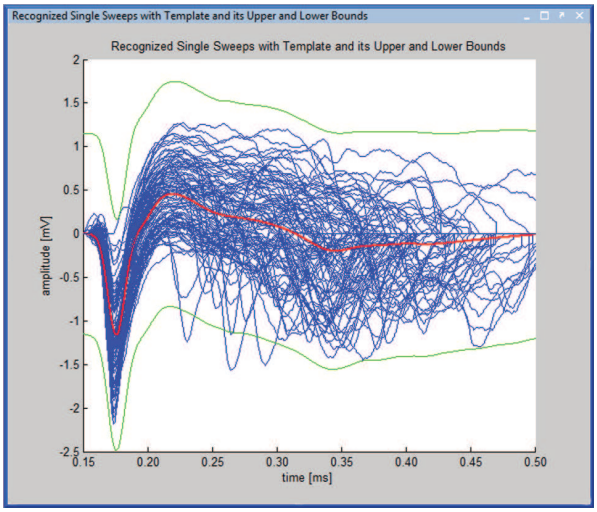


Figure 5.25: The template (in red), the upper and lower bounds (in green), and the single sweeps truncated to the size of the template.

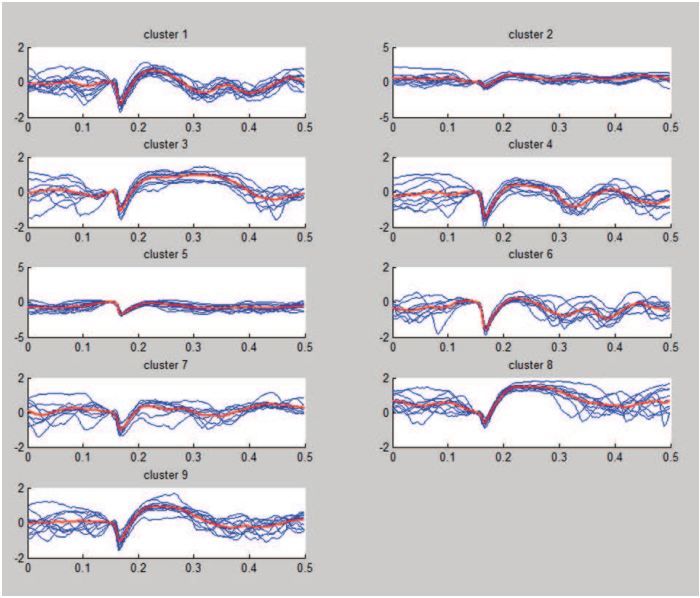


Figure 5.26: Result of the clustering. Single sweeps (in blue) and their respective averages (in red) depict clear difference in the shapes.

In Figure 5.24 we can see the raw single sweeps and their average signal (left) and the estimated single sweeps and their average signal (right). The arrow

indicates the stimulus-onset which is the starting point of the template. The main reasons behind performing the estimation are two folds. Firstly, reduction of noise without filtering out vital information, and secondly, facilitating the recognition of signal characteristics to be used as the base for selecting the feature vector for the iK–means clustering.

The Figure 5.25 shows single sweeps truncated to the size of template, the upper and lower bounds of the template. Each single sweep that fall within these bounds is considered to be recognized.

The Figure 5.26 shows the various clusters of signals with their respective averages. The N single sweeps, each represented by 200 feature points, generate a feature matrix of size $N \times 200$. The features of each single sweep are selected based on the detected events (see Section I, paragraph 3) in combination with the stimulus–onset and the end of response. Within the range of these six points 194 more points are selected arbitrarily. The choice of these 194 points is empirical, taking into consideration the trade-off between the computing overhead and an acceptable clustering of the sweeps. Clustering with 400 feature points is also done and not much difference in terms of signal classification is noticed. However, relatively more points are selected near the event peaks than that of distant locations from the peaks. This feature matrix is then classified using the iK–means clustering.

As described in the Section 5.4.1.3 the number of clusters is automatically set by the method intelligently, the Table 5.4.2 tabulates the recording depths, total number of recognized sweeps, single sweep distribution among different clusters. This table shows that the feature matrix is well classified into different clusters. In the table RS denotes total recognized sweeps; 1 through 10 are the cluster numbers; and "–" denotes absence of cluster.

Table 5.4: Total Recognized Sweeps, Single Sweep Allocation to Clusters

Depth	RS	Clusters									
		1	2	3	4	5	6	7	8	9	10
90 μm	90	5	6	12	11	11	7	8	12	6	12
180 μm	86	11	17	10	17	18	13	–	–	–	–
270 μm	87	8	8	8	11	15	7	4	10	10	6
360 μm	80	7	10	7	14	13	9	11	9	–	–
450 μm	78	10	9	11	4	7	15	9	13	–	–
540 μm	86	9	8	16	9	9	2	9	8	7	9
630 μm	85	16	6	15	14	17	17	–	–	–	–
720 μm	93	6	18	17	16	7	16	13	–	–	–
810 μm	92	10	9	13	9	13	8	14	6	10	–
900 μm	97	11	15	6	8	14	10	6	9	9	9
990 μm	96	19	15	15	10	5	17	15	–	–	–
1080 μm	92	12	12	9	9	12	9	8	10	11	–
1170 μm	99	8	13	13	9	6	11	10	9	10	10
1260 μm	100	11	20	13	16	7	7	16	10	–	–
1350 μm	100	18	13	19	19	19	12	–	–	–	–
1440 μm	98	13	10	16	8	16	14	7	5	9	–
1530 μm	100	7	9	18	7	14	7	10	16	12	–
1620 μm	99	10	5	16	10	11	8	10	12	11	6
1710 μm	99	10	12	20	13	14	12	18	–	–	–
1800 μm	100	10	12	5	15	16	14	9	19	–	–

Once the single sweep clusters are formed, the program computes local averages of each cluster for further processing. Analyses of these local averages (latency calculation based on the signal events) have revealed that the underlying neuronal network generating the signal can be different even if we are recording from the same recording site with the same stimulus.

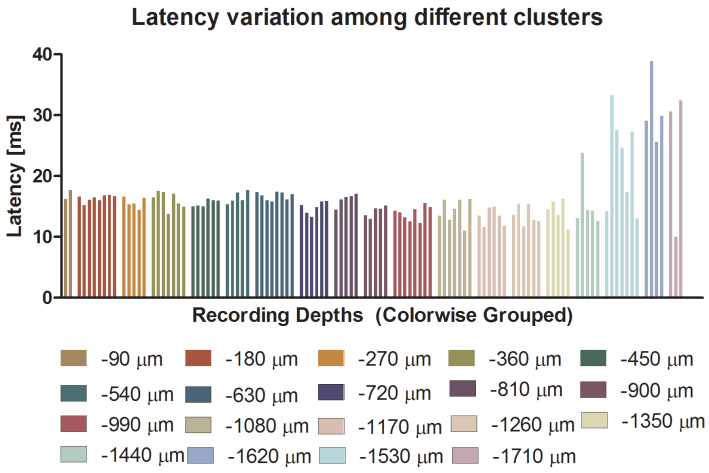


Figure 5.27: Latency variation among different clusters local averages. Each bar corresponds to a local average of a cluster and each color corresponds to a recording depth consisting of a number of clusters.

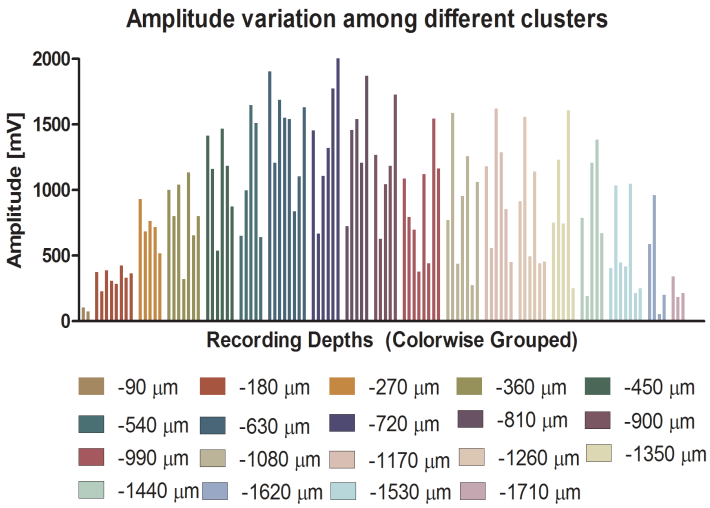


Figure 5.28: Amplitude variation among different clusters local averages.

The Figure 5.27 shows the different latencies calculated from the different local averages. The averages also show different amplitudes which can be seen in the Figure 5.28. These results imply the usefulness of the method.

5.5 EEG Based Brain–Machine Interfacing

5.5.1 The Electroencephalogram (EEG)

The Electroencephalogram (EEG) is produced by neuronal firing of the brain, which propagates through the cortex and is recorded along the scalp using Ag–AgCl electrodes. A mapping of these electrode positions is shown in figure 5.29. Depending on the position from where the signal is recorded, it provides the necessary information related to that part of the brain. For example, the visual cortex is located at the caudal portion (occipital lobe) of the brain, thus, placing an electrode at this part of the scalp will record the brain activity caused by visual stimuli.

This spontaneous activity of the brain produces the signal with amplitude usually under 100 μV and a frequency ranging from little above the DC voltage up to 100 Hz. The EEG signal is recorded from the scalp and is extracellular

in nature consisting of a few individual signals with different frequency bands, namely – delta (δ) with a frequency range of 0–3Hz, theta (θ) ranges from 4–7 Hz, alpha (α) varies from 8–12 Hz, beta (β) is from 12–30 Hz, and gamma (γ) has a band of 34–100 Hz. These different signals have their own clinical implications in disease diagnosis Niedermeyer & da Silva [2005].

In this work, the EEG signals are processed to extract the information related to the saccadic eye movement from the occipital region of the scalp (indicated by 'O1' and 'O2' in the surface view of the electrode mapping, figure 3.1). Based on the existing work done by Ohno *et al.* [2006], it is evident that the saccadic eye movement is well represented in the EEG during the memory tasks prior to the actual eye movement. These phenomena of changing EEG can be visualized by observing the signals recorded from the electrodes, 'O1' and 'O2' (see figure 5.34 for the acquired signal). Depending on the direction of the saccade, the recording from the contralateral electrode changes sharply in the EEG. These sharp changes are transformed into binary control signals that are sent to the robotic device for its navigation. In case of saccade evoked EEG, refresh rate and discrimination between left and right decisions are important, but due to our work's scope these aspects are not considered.

The choice of using the EEG evoked by saccadic eye movement is just an arbitrary decision to show the model's workability and thus does not impose a restriction for this BMI to use EEG signals generated by any other phenomena that can be transformed or conditioned using sophisticated signal processing tools to a binary decision signal.

5.5.2 Devices and Methods

5.5.2.1 Signal Processing

Both channels signal are filtered using a band–pass filter with the low–pass and high–pass cut–off frequencies to be 15 Hz and 100 Hz respectively. This filtering eliminates the possibility of recording the alpha wave generated by the occipital region of the brain. Then the signals are scanned for the occurrence of the sharp change in their amplitude based on a dynamic threshold. The system is trained before the actual experiment due to the signal variability from person to person.

This threshold gets renewed by calculating the ratio between the usual electrical activity and the existing threshold.

During the signal detection, extra care has been paid to eye blink, cardiac rhythms, noise coming from the 50–60 Hz power line and body movement artifacts. They are removed from the signal using processes proposed by Haas *et al.* [2003]; Rohalova *et al.* [2001]. These processed individual channels are then multiplexed and sent to another computer running on Linux through an UDP port using a crossover ethernet cable for interfacing with the robot to control it.

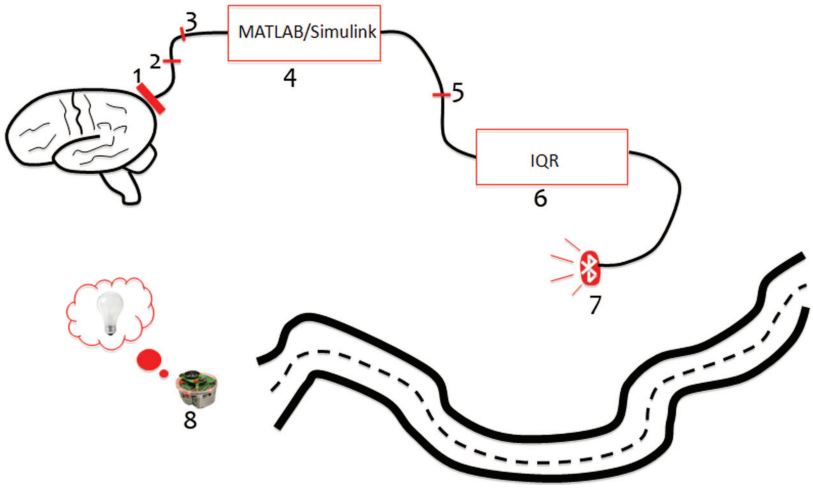


Figure 5.29: The schematic diagram of the Brain–machine interface system.

5.5.2.2 Interfacing with Robotic Device

The Figure 5.29 shows the schematic diagram of the interfaced system where each numbered box and element represent an interface through which the communication is made to the robotic system. The boxes represent computers used for this communication. The number 1 in the diagram represents the EEG recording cap with the electrodes (figure 3.3, (a)); the signals are sent to number 2, which represents the electrode connector and/or multiplexer (figure 3.3, (b)). The multiplexed signal from each channel is then sent to the preamplifier, (number 3 in the diagram, figure 3.3, (c)) where the signal is amplified with a predefined gain and transferred to the computer for digitization at 256 Hz (number 4 in the diagram) and further process them to extract the saccade information from the raw

EEG signal. The number 5 shows the UDP transfer of the conditioned signal to another computer (number 6) running on Linux for generating the command signals to the mobile robotic device. Finally these command signals are sent to the robotic device using Bluetooth (number 7). Figure 5.30 shows the flowchart outlining the various implementation steps.

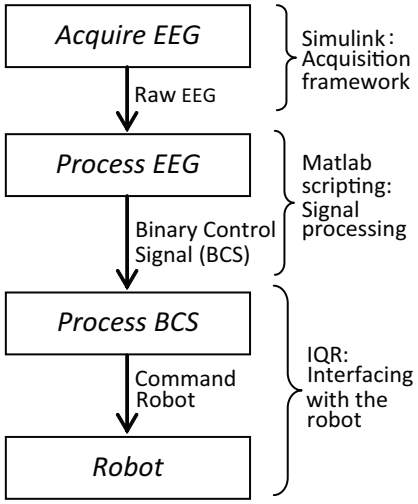


Figure 5.30: Flowchart outlining the major steps, their inputs and outputs. The curly braces categorize the steps based on the tools used in implementing those steps for the interfacing system.

iqr Bernardet *et al.* [2002], an open source software is used for interfacing and commanding the robot. The iqr provides a flexible platform for simulating neural networks and developing robust applications to interface with robotic devices.

To achieve our goal of steering wheels of the robotic device (number 8 in schematic diagram) using extracted saccade information, three modules are developed in the iqr as shown in figure 5.31 (a), (b), and (c). Two main processes are designed to receive input signals coming from the UDP port and sending a control signal based on conditioned signal to robot (figure 5.31 (a)). Each main process contains a number of neuronal populations. In figure 5.31 (c) the Simulink process group is shown where the input signals are received from the channels to control wheels of the robot. The Robot process group controls the movement of the robot's wheels. The input control signals are in form of synapse, either

excitatory or inhibitory, making a population of neurons to be active or inactive.

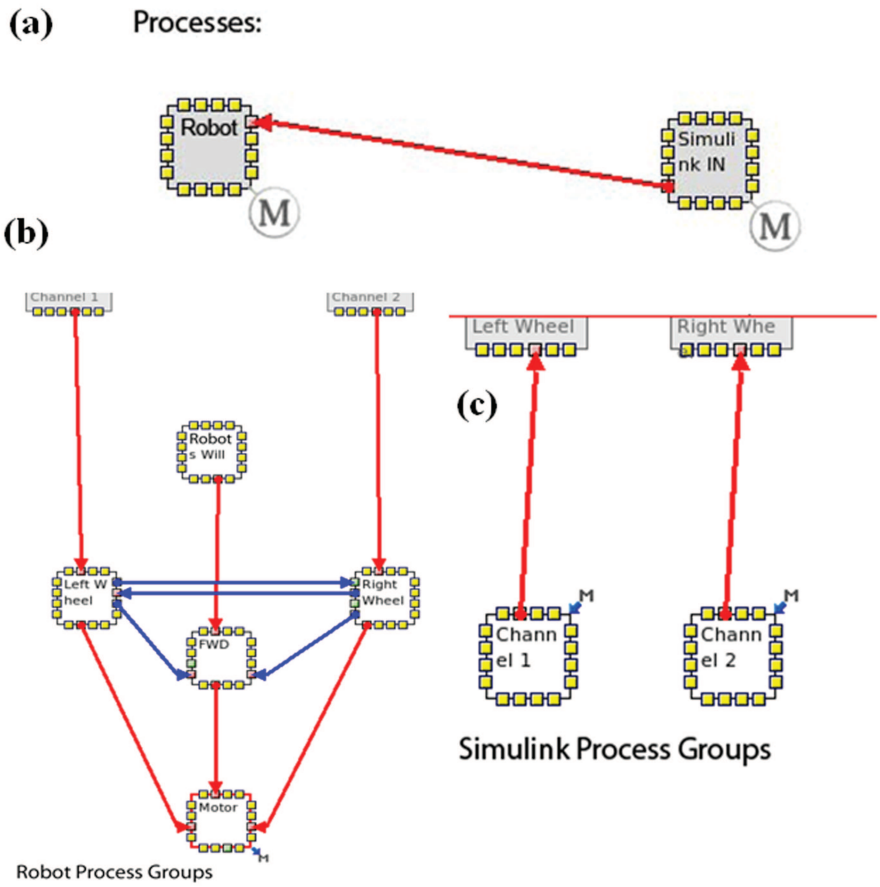


Figure 5.31: iqr modules for generating the command signal to control the robotic device.

Five neuronal populations are used to perform this work. The input control signal from each channel is fed to a neuronal population representing a wheel. These two populations are connected to the motor neuronal population (representing the motor of the robot) which initiates the movement of the individual wheels. Each time a wheel (in the rest of this section, the parts of the robot will denote a set of neuronal population) fires, it sends an excitatory synapse signal to the motor, another inhibitory synapse to the other wheel inactivating that wheel's activity; and another inhibitory synapse to the forward movement neuronal population making the motor to stop working for the previous com-

mand and get ready to adapt to the new command. Finally, there is one robot's will neuronal population that keeps generating excitatory synapse to the motor through the forward movement neuronal population that keeps the robotic device moving. The synapse from a wheel is required to stop the other wheel to steer it to follow a predefined course by taking a left and a right turn.

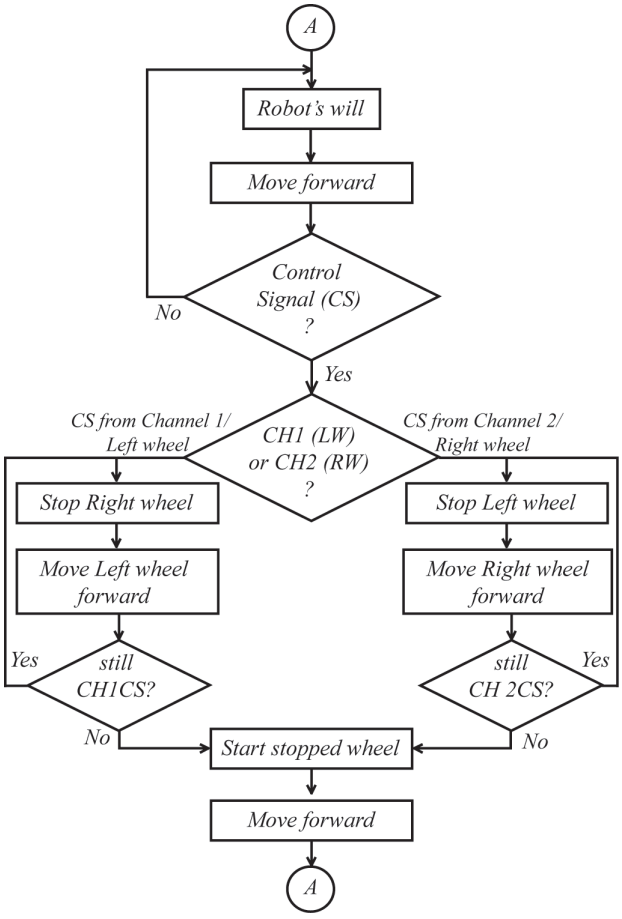


Figure 5.32: Flowchart of the IQR modules' communication for the robotic device's navigation.

The figure 5.32 depicts the flowchart of the IQR modules' communication for the robotic device's navigation. As explained in the previous paragraph, at the beginning the robot keeps on moving in the forward direction and awaits a control signal. Once the control signal is received, based on the channel from where it is generated, the wheel corresponding to the other channel is stopped

and the forward driving motor is initiated for the wheel corresponding to the control signal generating channel. This operation causes the robot to take a turn (right or left) based on the received control signal. During the turn the robot constantly scans for continuation of control signal and once the control signal is absent, the stopped wheel is restarted thus continuing the forward movement.

5.5.3 Discussion

A healthy subject is plugged in with the necessary equipments of EEG signal detection. The recorded and artifact removed EEG signal is shown in figure 5.33.

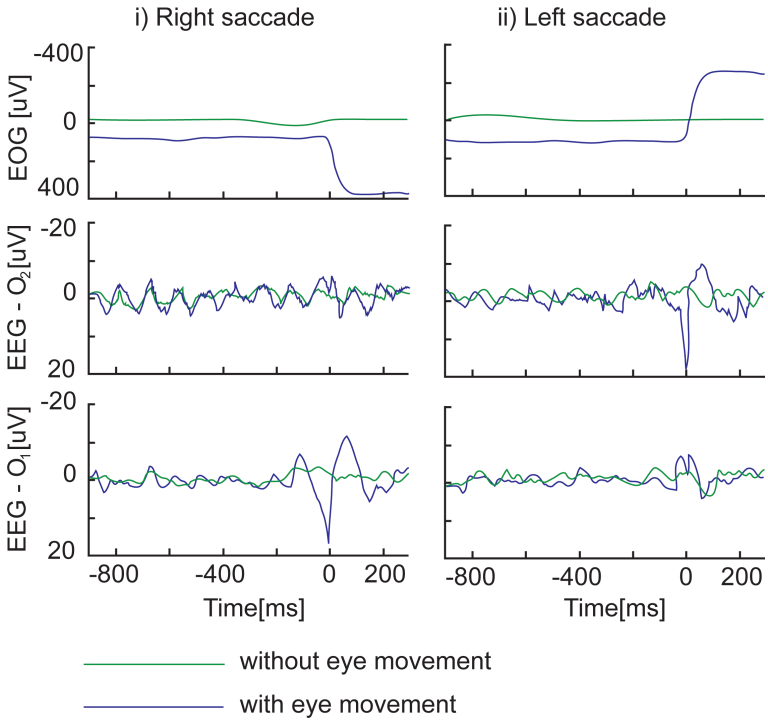


Figure 5.33: Signals recorded by EOG and 'O1', 'O2' electrodes of the EEG from one subject while performing the saccadic movement of their eyes during an experiment.

The saccadic movement direction of the subject is clearly reflected through sharp changes of amplitude in recorded signals from the contralateral electrodes. The dynamic threshold detects this sharp change and generates the controls sig-

nals which in turn trigger the activation of a particular neuronal population by providing an excitatory synapse for taking a turn. The excitatory synapse from the forward movement neuronal population keeps the robot moving forward; to make it to take a left turn the right wheel has been stopped through an excitatory synapse generated by the right wheel neuronal population, and the vice-versa for a right turn. A combination of these saccadic movements make the robot follow a predefined course, as seen in the figure 5.34, which is result of the robot's navigation during an experiment using this model.

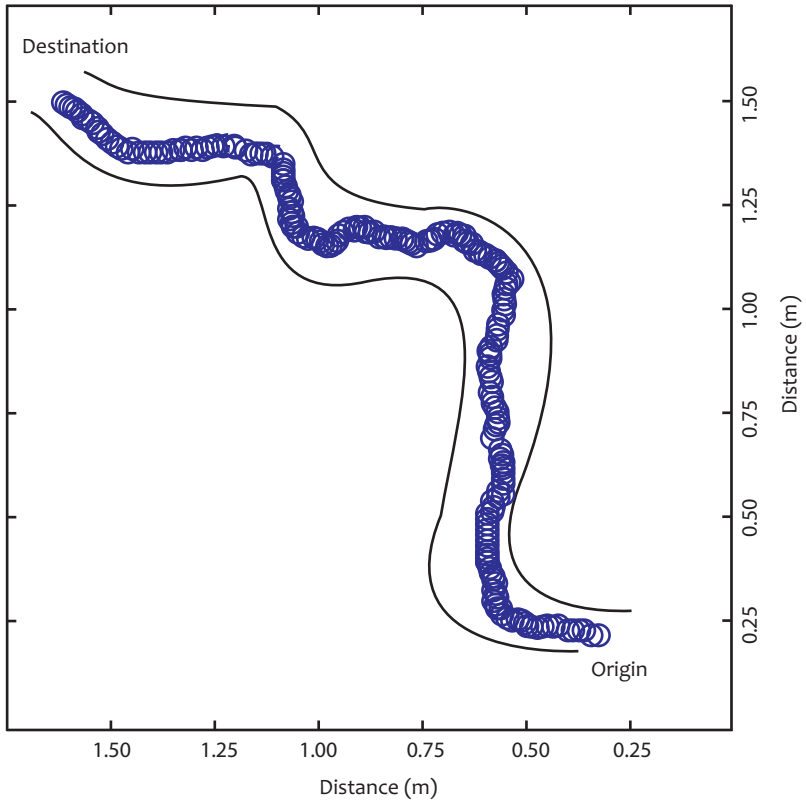


Figure 5.34: Navigation result of the robotic device during an experiment to follow a predefined course.

5.6 Neuronal Simulation Environment

In bioelectronics, the Action Potential (AP) is the basic mean of electro–chemical communication among the cells, specially, the cells of the nervous system, i.e.,

neurons. This AP can be defined as – the rapidly propagated electrical messages that speed along the axons of the nervous system and over the surface of some muscle and glandular cells (Hille [2001]). The AP is also essentially the change in membrane potential due to the flow of ions through the ion channels resembling the other message systems in the whole nervous system.

The model that describes the condition of a neuron at rest using an equivalent electrical circuit is shown at figure 5.35. The plasma membrane is formed by a thin layer (3–5 nm) of phospholipid. It separates ionic species and acts as a dielectric and is modeled as a capacitor.

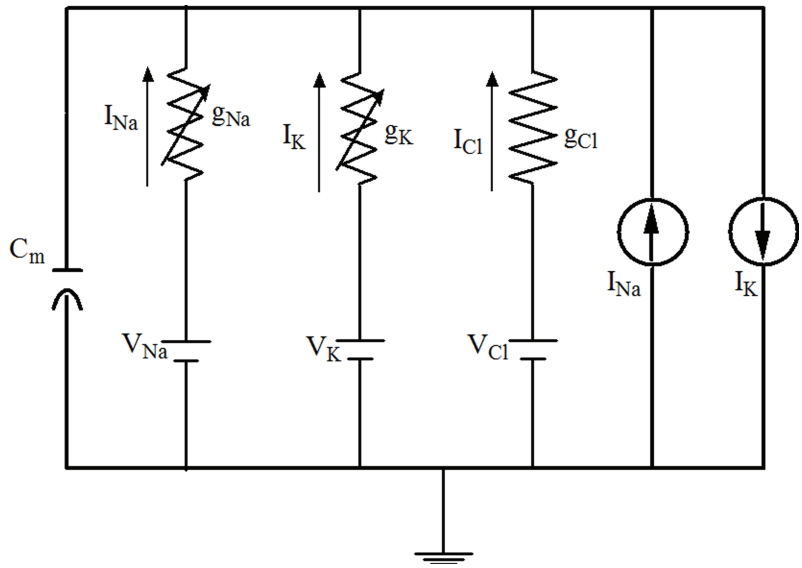


Figure 5.35: Equivalent circuit of the plasma membrane of a neuron.

Using this above electrical model of neuron, Hodgkin–Huxley (Hodgkin & Huxley [1952d]) proposed a method for the generation of the AP. Where the authors stated that the APs are generated due to: when the membrane potential is suddenly reduced (depolarization), the initial pulse of current through the capacity of the membrane is followed by large currents carried by ions (mainly sodium and potassium), moving down their own electrochemical gradients. The current carried by sodium ions rises rapidly to a peak and then decays to a low value; that carried by potassium ions rises much more slowly along an S–shaped curve, reaching a plateau which is maintained with little change until the membrane

potential is restored to its resting value (Hodgkin & Huxley [1952b]). Action potentials arise from the synergistic action of sodium channels and potassium channels, each of which opens and closes in a voltage dependent fashion (Colwell & Brenner [2009]). The figure 5.36 shows an action potential as a result of ionic channel opening and closing.

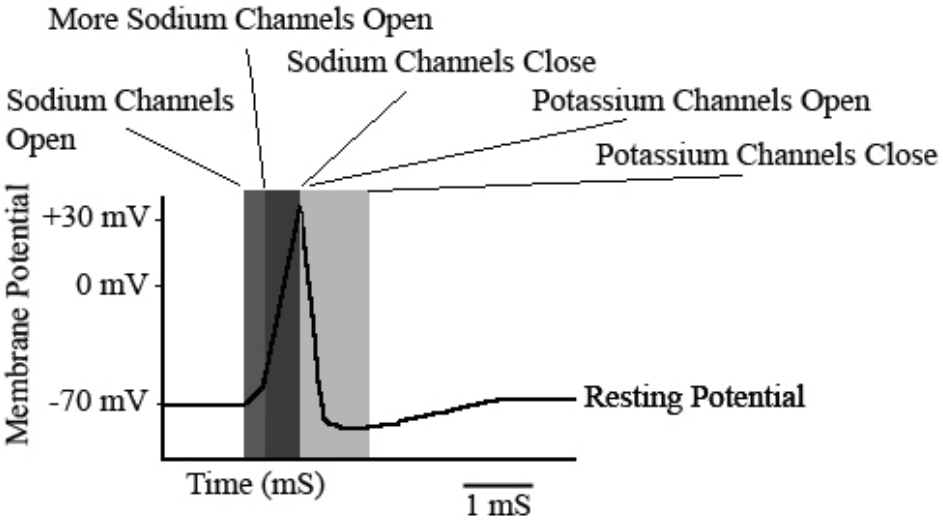


Figure 5.36: Action Potential as a result of different ionic channels' activities.

There have been many works on simulation of action potentials based on many aspects of the Hodgkin–Huxley model. They include, the changes of action potential shapes and velocity (Goldstein & Rall [1974]), simulation of the propagation of action potential in septated nerve fibers (Barach & WikswoJr. [1987]), simulation of the action potential propagation in complex terminal arborizations (Luscher & Shiner [1990]), etc. The Hodgkin–Huxley model is also modified to study unique features of action potential initiation in cortical neurons (Naundorf *et al.* [2006]), though its reliability is being questioned (McCormick *et al.* [2007]), yet this is the mostly used model in case of single neuronal behavior simulation. Therefore, in SigMate the simulation environment is provided to study the neuronal behavior.

5.6.1 The Hodgkin–Huxley Model

Hodgkin and Huxley published a series of (now classical) papers in 1952 (Hodgkin & Huxley [1952a,b,c,d]) in which they investigated membrane currents through the membrane of the squid giant axon. A mathematical analysis of this work was published in a review paper (Hodgkin & Huxley [1952d]). In their articles, they said that current flowing across the giant axon membrane may be represented by the sum of conductive components and capacitance.

In the Hodgkin–Huxley model conductive components are a function of potential difference across the cell membrane (V_m) and the equilibrium potentials (V_{ion}) of the ions. The Nernst–Einstein relationship is used to derive the equilibrium potentials. The currents through the conductive elements are expressed as:

$$I_{ion} = g_{ion} \cdot (V_m - V_{ion}) \tag{5.22}$$

They demonstrated through their results that conductance of Na, g_{Na} and conductance of K, g_K are a function of time as well as voltage, but the conductances of the other ions are constant. The Na conductance gets a transient increase due to the depolarization of the axonal membrane and a slower non–inactivating increase in K conductance. The time dependence of this conductance may be represented by an activation coefficient x which, if we consider that the conductance is represented by the opening of many individual channels, represents the probability of a gate in the channel being open. The conductance for a time dependent channel can thus be written in terms of its activation coefficient $x(0 \leq x \leq 1)$ and a maximum conductance $g_{ion,max}$:

$$g_{ion} = g_{ion,max} \cdot x \tag{5.23}$$

x may be described by the following equation:

$$\frac{dx}{dt} = \alpha_x \cdot (1 - x) - \beta_x \cdot x \tag{5.24}$$

where, α_x and β_x are rate coefficients which are non–linear functions of voltage (units are 1/time).

Now, the current may be represented by equation with the following general form due to many complex operations that usually takes place during the opening and closing of the channels, considering,

$$I_{ion} = g_{ion,max} \cdot x \cdot y \cdot (V_m - V_{ion}) \tag{5.25}$$

where, x is the activation coefficient and y is the inactivation coefficient. The Na conductance responsible for the depolarization during the action potential takes this form because it inactivates without repolarization. The Na conductance opened in response to depolarization but closed without repolarization which is because there are two control gates. One of these gates opens on depolarization and a second closes with different kinetic parameters. The rate coefficients for the inactivation variable α_y and β_y are necessarily much slower than those for the activation variable as the second closes slowly than the first opens. Consequently, in the model there are two control gates. One is activated when a threshold depolarization is achieved, the second closes more slowly to subsequently block flow of Na ions. These are represented by two variables (m and h for activation and inactivation coefficient respectively) each described by a differential equation as:

$$\frac{dm}{dt} = \alpha_m \cdot (1 - m) - \beta_m \cdot m \tag{5.26}$$

$$\frac{dh}{dt} = \alpha_h \cdot (1 - h) - \beta_h \cdot h \tag{5.27}$$

where α and β are rate constants that are functions of voltage but not of time.

The Na conductance can now be written as:

$$g_{Na} = g_{Na,max} \cdot m^3 \cdot h \tag{5.28}$$

where $g_{Na,max}$ is the maximum Na conductance.

The Na current is described by:

$$I_{Na} = g_{Na} \cdot (V - V_{Na}) \tag{5.29}$$

where V_{Na} is the Na reversal potential, calculated using the Nernst–Einstein

relation.

The K conductance may be described similarly although, instead of voltage operated activation and inactivation gates, one need only to include an activation gate:

$$\frac{dn}{dt} = \alpha_n \cdot (1 - n) - \beta_n \cdot n \tag{5.30}$$

where α and β are rate constants that are functions of voltage but not of time. The K conductance can now be written as:

$$g_K = g_{K,max} \cdot n^4 \tag{5.31}$$

where $g_{K,max}$ is the maximum K conductance.

The K current is given by:

$$I_K = g_K \cdot (V - V_K) \tag{5.32}$$

where V_K is the K reversal potential, calculated from the Nernst–Einstein relation.

Apart from the Na and K currents, there is another important current that influences the generation of the action potentials in a neuronal model: the leakage current (I_L). This leakage current can be computed using:

$$I_L = g_L \cdot (V - V_L) \tag{5.33}$$

where g_L denotes the leakage conductance and V_L is the leakage equilibrium potential.

Finally the membrane potential can be computed as:

$$C_m \cdot \frac{dV}{dt} = I - \sum_{ion} I_{ion} \tag{5.34}$$

where I_{ion} denote various ionic currents.

5.6.2 Optimization of Stimulus Protocol

Neurons in sensory regions of the brain respond to stimuli by firing one or more nerve impulses (action potentials) following stimulus presentation. How is information about the outside world encoded by the rate, timing, and pattern of action potentials? This so–called neural code is currently unknown and sensory neuroscience plays an important role in the attempt to decipher it. Looking at early sensory processing is advantageous since brain regions that are "higher up" (e.g. those involved in memory or emotion) contain neurons which encode more abstract representations. However, the hope is that there are unifying principles which govern how the brain encodes and processes information. Studying sensory systems is an important stepping stone in our understanding of brain function in general.

However, in most of the central nervous system, neurons communicate exclusively by sending each other action potentials, colloquially known as "spikes". It is therefore thought that all of the information a sensory neuron encodes about the outside world can be inferred by the pattern of its spikes. Current experimental techniques cannot measure individual spikes non–invasively, so electrodes must be used to reveal a neuron's spikes. A typical single neuron experiment will consist of isolating a neuron (that is, navigating the neuron until the experimenter finds a neuron which spikes in response to the type of stimulus to be presented, and (optionally) determining that all of the spikes observed indeed come from a single neuron), then presenting a stimulus protocol. Because neural responses are inherently variable (that is, their spiking pattern may depend on more than just the stimulus which is presented, although not all of this variability may be true noise, since factors other than the presented stimulus may affect the sensory neuron under study), often the same stimulus protocol is repeated many times to get a feel for the variability a neuron may have.

To reduce the complexities of this situation and to have preliminary information about the neuronal behavior single neuron simulation environment can be used. I have used the above mentioned Hodgkin–Huxley model to describe a single neuron. The necessary parameters for mammalian cells are used to optimize stimulus protocol. These parameters according to Awiszus [1992] can be found

in table 5.5.

Table 5.5: Hodgkin–Huxley model parameters for mammalian neurons

Parameters	Value	Unit
$g_{Na,max}$	240	mS/cm^2
$g_{K,max}$	36	mS/cm^2
g_L	68	$\mu S/cm^2$
V_{Na}	+64.7	mV
V_K	−95.2	mV
V_L	−51.3	mV
V_{rest}	−71	mV

To optimize the stimulation protocol, a stimulus generator is designed and incorporated with the simulation environment capable of providing various waveforms. As found in the literature, for studying single neuron behavior usually sinusoidal waves, sawtooth wave, reverse–sawtooth wave, square pulses, and ramped pulses are used. Thus the stimulus generator can deliver all these stimulus types at varied amplitude and frequency.

Four different types of stimulus waveforms ((i) reverse sawtooth waves, (ii) sawtooth waves, (iii) hyperbolic sine waves, and (iv) sine waves) with varied amplitude and frequency are applied to the model and the response is observed.

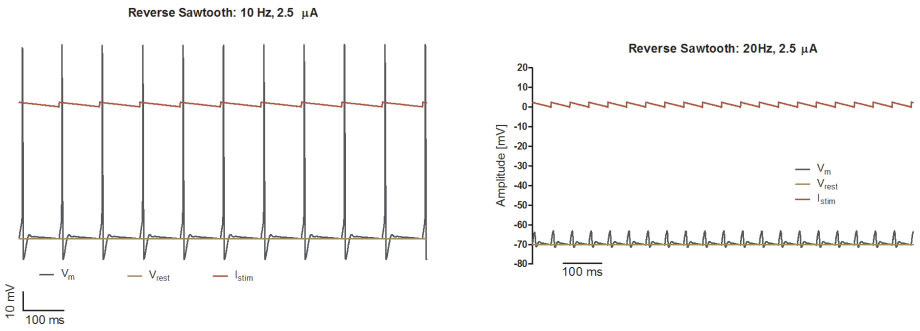


Figure 5.37: Reverse sawtooth waves applied to single neuron model.

Figure 5.37 shows two representative situations by applying reverse sawtooth waves of amplitude 2.5 μA with frequencies 10 Hz and 20 Hz. It is found that at 10 Hz the neuron starts firing action potentials at a one–to–one basis in correspondence to the waveform, however, around 15 Hz the neuron can no more generate action potentials.

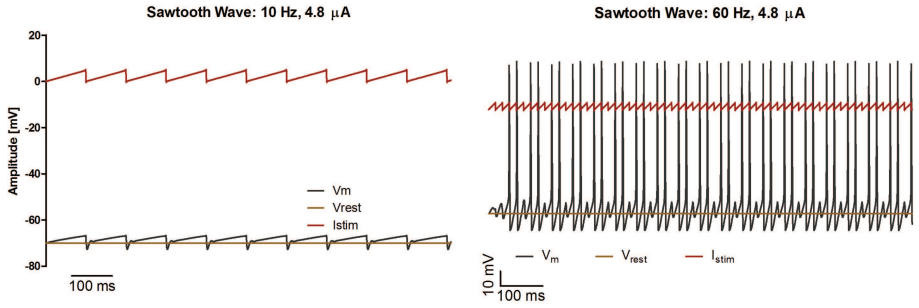


Figure 5.38: Sawtooth waves applied to single neuron model.

In figure 5.38 two situations representing neuronal response by applying sawtooth waves of amplitude 4.8 μA with frequencies 10 Hz and 60 Hz are shown. It can be seen that at 10 Hz the neuron is almost in the resting potential (just little perturbed). With increasing frequency, at the same amplitude the neuron starts firing action potentials at 60 Hz and the firing rate increases until 80 Hz with a one–to–one correspondence to the waveform. But, once the neuron reaches 100 Hz, the firing rate decreases gradually and at 150 Hz it can no more generate action potentials.

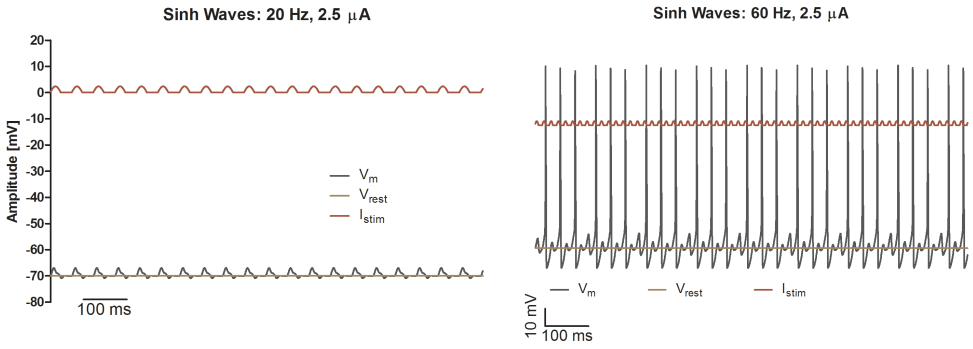


Figure 5.39: Hyperbolic sine waves applied to single neuron model.

Figure 5.39 reports the neuron's response upon application of hyperbolic sine waves of amplitude 2.5 μA and frequencies 20 Hz and 60 Hz. Similar results are noticed as in case of sawtooth waves only with the exception that the amount of stimulus current applied is almost half of the previous case.

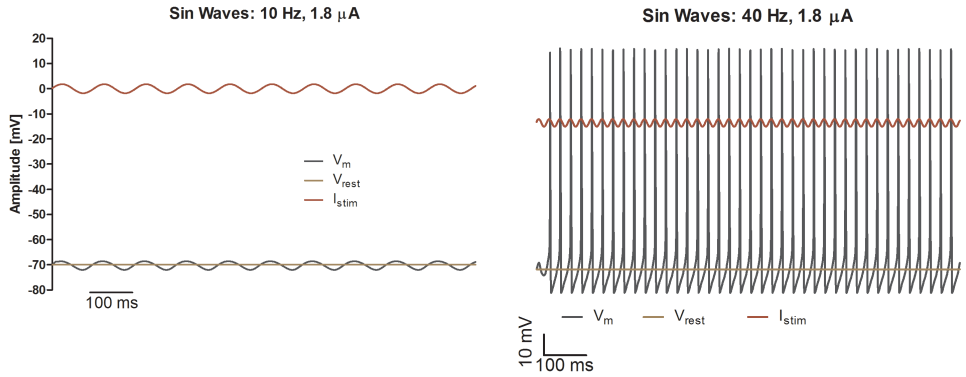


Figure 5.40: Sine waves applied to single neuron model.

As most of the neuronal responses can be represented by means of sine waves, this type of signal is used mostly for neuronal stimulation. Figure 5.40 depicts the neuronal response after application of sine waves of frequencies 10 Hz and 40 Hz with 1.8 μA amplitude. The neuron remains in the resting state until 30 Hz after which it starts generating action potentials. The firing rate of action potentials increase with frequencies until 40 Hz, where it reaches a one–to–one correspondence to the stimulus waveform. After 70 Hz, the response of the neuron gradually decays and at around 80 Hz, the neuron can not produce any more action potentials.

Table 5.6 shows frequency and amplitude dependence of four different types of stimuli on a single neuron Hodgkin–Huxley model.

Table 5.6: Effect of stimulus frequency and amplitude variation on single neuron model

Stimulus	Minimum Amplitude	Starting Frequency
Sine wave	1.8 μA	40.1 Hz
Hyperbolic Sine wave	2.4 μA	60.1 Hz
Sawtooth wave	4.8 μA	60.1 Hz
Reverse sawtooth wave	2.5 μA	10.1 Hz

As evidenced above, the neuronal simulation environment is capable of producing neuronal response upon application of various stimulation. Using the stimulus generator, it is possible to provide different types of stimuli to the neuronal model of varied amplitude and frequency and the behavior of the neuron can

be studied. This type of study is very much helpful in optimization of stimulus protocol before getting into real experiments.

5.6.3 Ca2+ Based Neuronal Simulation Environment

The thalamus has long been recognized as a key structure in the generation of brainwave rhythmicity (Steriade & Deschenes [1984]; Steriade *et al.* [1990]). Important progress in understanding the cellular basis of this pacemaker activity has come with an appreciation of the unique firing properties of thalamocortical neurons. These cells exhibit two distinct modes of firing depending on the membrane potential level (Deschenes *et al.* [1984]; Jahnsen & Llins [1984a,b]). At normal resting potential, the cells transform suprathreshold input signals into trains of action potentials with high fidelity in what is referred to as "relay mode". However, on hyperpolarization there is an abrupt transition to phasic (burst) firing in which action potentials occur grouped into bursts. During drowsiness and slow–wave sleep, the burst firing of thalamic neurons plays an essential role in the generation of synchronized 8– to 10 Hz rhythmic oscillations ("spindling") in the cortical electroencephalogram. In addition, the stereotyped spike–and–wave discharges characteristic of absence (petitmal) seizures may also be dependent on synchronized thalamic bursting (Gloor [1984]).

5.6.3.1 The Ca2+ Based Modified Hodgkin–Huxley Model

Following the work of Coulter *et al.* [1989], the researchers developed a model for the T–type Ca2+ current in analogy with the fast sodium current modeled by Hodgkin & Huxley [1952d] (Jin *et al.* [2004]; Wang *et al.* [1991],). According to this model the calcium current (I_{Ca}) is given by:

$$I_{Ca} = -g_{Ca} m_{Ca}^3 h_{Ca} (V - V_{Ca}) \tag{5.35}$$

where g_{Ca} is the maximal conductance per unit area (expressed in mS/cm$^(2)$ and V_{Ca} is the reversal potential for calcium flux, here fixed at + 120 mV. The value of g_{Ca} may be chosen to scale the amplitude of I_{Ca} to that measured in particular voltage–clamp experiments.

By hypothesis, m_{Ca} represents the fraction of channels for which one of the three activation gates is in its open state (O); and h_{Ca}, the fraction of those with the inactivation gate in its open state. A channel is open if all of its four gates are in their open states. Each activation gate satisfies a two–state transition kinetics with voltage–dependent rates α_{mCa}, and β_{mCa}, so that:

$$\frac{dm_{Ca}}{dt} = \alpha_{mCa}(1 - m_{Ca}) - \beta_{mCa} m_{Ca} \tag{5.36}$$

In several experimental studies, no effect on the kinetic behavior of T–type Ca2+ currents was observed when the intracellular Ca2+ concentration was varied, when extracellular Ca2+ was replaced by Ba^{2+} as the charge carrier, or under conditions where Ca2+ or Ba^{2+} accumulation could not occur (Carbone & Lux [1984, 1987a,b]; Hernndez-Cruz & Pape [1989]; Nowycky *et al.* [1985]). The inactivation gate of the T–Type Ca2+ channel is therefore regarded as insensitive to the intracellular Ca2+ concentration. To overcome this and include Ca2+ concentration in the model, the inactivation is defined as (Jin *et al.* [2004]):

$$h_{Ca} = \frac{K}{K + [Ca^{2+}]_i} \tag{5.37}$$

$$\frac{d[Ca^{2+}]_i}{dt} = \frac{-I_{Ca}}{2 \cdot F \cdot v} + \frac{[Ca^{2+}]_{eq} - [Ca^{2+}]_i}{\tau_{trans}} + \eta \cdot I_{glu} \tag{5.38}$$

$$\tau_{trans} = 6 \cdot e^{\frac{V}{100}} \tag{5.39}$$

$$V_{Ca} = 12.5 \cdot log(\frac{[Ca^{2+}]_0}{[Ca^{2+}]_i}) \tag{5.40}$$

with $\bar{g}_{Ca}$ =135 nS.

It has been demonstrated that the LHC of carp retina expresses AMPA–sensitive glutamate receptors, the properties of which can be regulated by Ca2+–dependent phosphorylation processes (Liman *et al.* [1989]; Mammen *et al.* [1997]; McMahon *et al.* [1994]). Thus, the conductance of the glutamate channel is a

function of intracellular Ca2+ concentration and can be calculated using:

$$g_{glu} = \bar{g_{glu}} \cdot f([Ca^{2+}]_i) \tag{5.41}$$

And therefore, the I_{glu} can be calculated using:

$$\bar{I_{glu}} = 600 \cdot Glu(t) \cdot f([Ca^{2+}]_i) \tag{5.42}$$

$$I_{glu} = \bar{I_{glu}} \frac{e^{(\frac{V+60}{95} - 1.5)}}{e^{(\frac{40-V}{125} + 0.5)}} \tag{5.43}$$

$$f([Ca^{2+}]_i) = \theta \cdot [Ca^{2+}]_i \tag{5.44}$$

Glutamate is the neurotransmitter released by photoreceptors. The transmitter release is decreased when the retina is exposed to light stimulus, which causes a hyperpolarization of the photoreceptors. Glutamate activates I_{glu}, and in the meantime suppresses I_{anom} on the postsynaptic LHC (Ishida *et al.* [1984]; Kaneko & Tachibana [1985]; Tachibana [1985]).

Eventually, the membrane potential can be calculated using the Hodgkin–Huxley formation:

$$C_m \frac{dV}{dt} + I_{Na} + I_K + I_{Ca} + I_{anom} + I_{glu} + I_L = 0 \tag{5.45}$$

This model can generate action potentials that can be controlled by controlling the glutamate release to hypothesize on the regularization of a neuron's bursting firing behavior (Milone *et al.* [2010]).

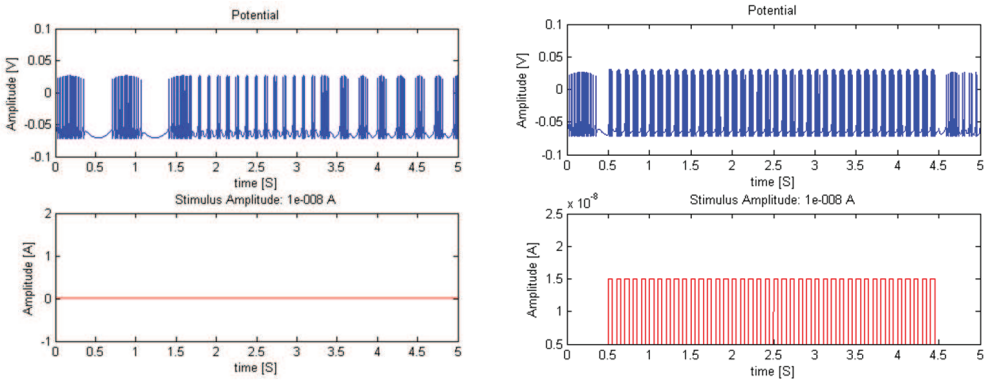


Figure 5.41: Modified Hodgkin–Huxley model including Ca2+ channel during stimulation at 10 nA and 10 Hz. Left: without glutamate and right: with glutamate.

Figure 5.41 shows the simulation of modified Hodgkin–Huxley based model incorporating Ca2+ channels. The stimulation is done using 10 nA current at 10 Hz (50 ms + 50 ms). Point to notice that the irregularity of bursting (on left, without Glutamate) can be regularized using activation of Glutamate (on right).

It now can be understood that the neuronal simulation environment can successfully simulate the single neuron behavior based on Ca2+ concentration.

Chapter 6

Conclusions

To understand brain activities with an unprecedented level, parallel high resolution recordings are required. This thesis presents a software package, "SigMate". As with the growth of neuronal probes, amount of acquired data are increasing, the need of one single software package performing all necessary processing and analysis on the data has become crucial. This is the first step towards meeting that need. As the software has been extensively tested with two possible sources of data, we believe that once it is disseminated to the community (which will happen in the near future) it will serve a good deal in analyzing extracellular neurophysiological signals. Modules for coherence and correlation based analysis to obtain more information about the brain's functionality from the recorded signals, neuronal network based on different stimuli to be able to predict the signals a priori, compare them with the recorded signals and understand the activation of underlying neuronal networks generating the signals, are under development.

The main modules of SigMate and their usefulnesses are concluded individually in the following paragraphs.

- **Artifact removal**: To be able to use the neurological evoked potentials in understanding the brain signaling pathways or to map the brain region in response to stimuli, they must be free of any kind of artifacts. Artifact shapes and frequency over-lapping are the main barriers in efficient artifact removal. Also, there are no standard methods for removing the slow stimulus artifacts, caused, for example, by air puff stimuli. The method proposed in this work is an automatic, flexible, very simple to implement,

and computationally efficient way to perform this task. The automatic selection of the threshold in detection of peaks-and-valleys extends its usage also in offset correction and batch processing of neuronal evoked potentials.

- **Noise characterization**: As multiple neuronal probes for recording neuronal responses require sophisticated electronic equipment, even after utmost care there are noises present in the recordings. It is important to have a method capable of characterizing the noise and comment about the quality of the recordings. Such a method has been proposed in this work with the advantages of being automated, and easy to implement. From the presented results, it is evident that the means of the MEs are sufficiently smaller than the means of the steady–states, suggesting the experimental recordings to be of good quality in terms of noise contamination and thus characterizing the recorded signals from a high resolution brain–chip interface. The quick analysis capability and ease in implementation of the method make it suitable to characterize the noise on real–time basis for high resolution brain–chip interface recording from cortical surface.

- **Latency estimation and layer activation order determination**: Whisking in the rodents is one of the most important ways in exploring the environments. To understand the whisking mechanism, its role in localizing objects and discriminate among them based on shape and texture are under extensive study. To perform this kind of studies, determining the signal processing pathway and, in turn, the order of activation of different cortical layers is very important. Scientists perform this task manually which is time consuming and boring. As evidenced above, the proposed method is an automated solution in performing this kind of analysis. Two methods are automated, one uses the LFPs and the other one the CSDs. Depending on the need, the user can select either of them and have a qualitative assessment of the layer activation order. Moreover, the method using LFPs is computationally efficient, quick and easy to implement. As it involves less calculations compared to the CSD based approach, this method could be preferably adapted for analysis of signals recorded using high resolution brain–chip interfaces or neural probes capable of recording large amounts

of data during an experiment.

- **Single sweep LFP clustering**: Through whisking rats perform very fine discrimination of the environment that they are in. To understand the tactile information processing pathway better scientists depend on the LFPs. And the shapes of the LFPs work as a fingerprint of the activated neural network near the recording electrode. To understand and assess these multiple networks' activity at one recording position, it is necessary to distinguish between the different shapes of signals recorded at a single recording site. Till date scientists have relied on single conventional average, thus, this method is a tool for them to extend their vision beyond the conventional network activity.

- **EEG based BMI**: This Brain-machine interface based on the EEG demonstrated here is a proof-of-principle to reduce the complexity that is gradually prevailing upon the very potential field of rehabilitation. By applying this technique it is possible to provide mobility to the people with motor dysfunction. The usage of saccade evoked EEG signals in this work is just to demonstrate the proposed model's workability. It is very much possible to adapt and extend the BMI model to use any other type of signal (voluntary or involuntary) through proper signal processing technique capable of transforming the input signal to a binary decision signal. The work in progress is to extend this technique to control assistive robotic devices (e.g., robotic wheelchair) for the disabled.

- **Neuronal simulation environment**: This simulation environment contains two components: (i) single neuron simulation based on Hodgkin–Huxley model capable of simulating neuronal behavior based on different stimuli, and (ii) modified Hodgkin–Huxley based neuronal model incorporating Ca2+ channels, which are designed using calcium concentration controlled by Glutamate release. The simulation environment can be very helpful in situations when neuronal behavior is to be known before performing experiments.

Appendix A

.1 Peak–valley Detection Algorithm

The following algorithm is used for the detection of peaks–and–valleys of the signal to have an estimation of the artifact signal.

Function: Detect_Peak-Valley()
Input: Signal file, whose peaks and valleys are to be found.
Output: Peaks-and-valleys of the input signal.
Method:

1. *Initialize, S:=signal; and Threshold:=Standard-deviation(S);*
2. *Set, Peak:=infinity; Valley=-infinity; and Flag:=True;*
3. *Current:=Current element of S;*
4. **if** *(Current > Peak), Reset, Peak:= Current;* **end if** *;*
5. **if** *(Current < Peak), Reset, Valley:= Current;* **end if** *;*
6. **if** *(Flag is True)*
   if *(Current <(Peak − Threshold))*
   Add Current to Peaks; and
   Reset, Flag:=False;
   end if*;*
   if *(Current >(Valley + Threshold))*
   Add Current to Valleys; and
   Reset, Flag:=True;
   end if*;*
   end if*;*

7. *Repeat step 3 to 6 for every element of the signal.*
8. *Return Peaks and Valleys.*

.2 Noise Characterization Algorithms

.2.1 Calculation of Measurement Errors

The following algorithm is used in calculating the MEs. As described in section 5.2.2.3, the MEs are calculated after having detected the FSS and the SSS. Thus, this method calls two other methods (algorithm is given in .2.2 and .2.3) which detect the FSS and the SSS.

Function: MeasurementErrors( )
Input: Signal files containing time and data.
Output: Statistical information (μ, σ and distribution) of the data.
Method:
// this is for the single sweep approach as described in section 5.2.2.3

1. **for** *each signal file*
   a. *Load time (t) and data (s);*
   b. *firstSteadyState[fssT, fssS]:=Call the findFirstSteadyState(t, s)*
   c. *secondSteadyState[sssT, sssS]:=Call the findSecondSteadyState (t, s, firstSteadyState)*
   d. *Fit mathematical models to the firstSteadyState and secondSteadyState;*
   e. *Calculate the data bounds around the firstSteadyState and secondSteadyState;*
   f. *Adjust the data points based on the data bound, i.e., boxing the data around $\pm 2\sigma$ from the straight line;*
   g. *Calculate MEs for first and SSSs using equation (1);*
   h. *Characterize MEs by calculating μ and σ;*
   i. *Plot the distribution of MEs using histogram and estimate the distribution using a smoothing function;*

end for

// this is for the averaged steady–state approach as described in section 5.2.2.3

2. *Calculate the average of the all FSSs;*
3. *Fit mathematical model to the average of the FSSs;*
4. *Calculate the ME of the average using equation (1);*
5. *Characterize error by calculating μ and σ;*
6. *Calculate the average of the all SSSs;*
7. *Fit mathematical model to the average of the SSSs;*
8. *Calculate the ME of the average using equation (1);*
9. *Characterize error by calculating μ and σ;*
10. *Plot the distribution of ME using histogram and estimate the distribution using a smoothing function;*

// this is for the averaged ME approach as described in section 5.2.2.3

11. *Calculate average of all MEs for the FSS;*
12. *Characterize error by calculating μ and σ;*
13. *Plot the distribution using histogram and estimate the distribution using a smoothing function;*
14. *Calculate average of all MEs for the SSS;*
15. *Characterize error by calculating μ and σ;*
16. *Plot the distribution using histogram and estimate the distribution using a smoothing function;*

.2.2 Detecting the First–Steady–State

The following algorithm is used in detecting the first–steady state.

Function: findFirstSteadyState( )
Input: The time (t) and data (s).
Output: The FSS of the signal (firstSteadyState).
Method:

1. *firstPart:=the first 10 ms of the signal, s;*

2. *stdFirst:=std(firstPart); stdS:=std(s);*
3. *intervalInt:= 3 ms; newS:=rest of the signal;*
4. *Divide newS into intervals of length intervalInt;*
5. *Initialize, newPartS:=[]; newPartT:=[]; flag:=1;*
6. **while** *currInterval isn't in evoked response*
   fitLine:=Fit a straight line to currInterval;
   sdNew:=std(currInterval);
   if *(sdNew ≥ stdCheck && flag)*
   add currInterval to newPartS;
   add currIntervalTime to newPartT;
   stdCheck:= sdNew; flag:= 1;
   else
   flag:=0;
   add currInterval to newPartS;
   add currIntervalTime to newPartT;
   if *(last point of fitLine > stdS)*
   remove currInterval from newPartS;
   remove currIntervalTime from newPartT;
   end if;
   end if;
   end while;
7. *Return firstSteadyState:=[newPartT,newPartS];*

.2.3 Detecting Second–Steady–State

The following algorithm detects and returns the second–steady–state.

Function: findSecondSteadyState(t, s, fsState)
Input: Signal time (t) and data (s), and the FSS (fsState).
Output: The SSS of the signal (secondSteadyState).
Method:

1. *fsStateLength:=length(fsState); fsStateStd:=std(fsState);*

2. *sToAnalyse:=s(fsStateLength to length(s));*
3. *sToAnalyseT:=t(fsStateLength to length(s));*
4. *numDiv:=floor(length(s)/fsStateLength);*
5. *sToAnalyseRev:=Reverse(sToAnalyse);*
6. *sToAnalyseRevT:=Reverse(sToAnalyseT);*
7. *Initialize, ssStateS:=[]; ssStateT:=[]; flag:=0;*
8. *Find which part of sToAnalyseS has to be considered as SSS*

```
for i:=1 to numDiv
    currInterval:=sToAnalyseRev[i];
    currIntervalTime:=sToAnalyseRevT[i];
    stdCurrInterval:=std(currInterval);
    if stdCurrInterval < fsStateStd
      add the currInterval to ssStateS;
      add the currIntervalTime to ssStateT;
    else
        if (i==1)
          add currInterval to ssStateS;
          add the currIntervalTime to ssStateT;
        end if;
        flag:=flag+1;
        if (flag<=2) && (length(sToAnalyseRev) < length(fsState))
          nextInterval:= sToAnalyseRev[i+1];
          nextIntervalT:=sToAnalyseRevT[i+1];
          stdNextInterval:=std(nextInterval);
          if (stdNextInterval <= fsStateStd)
            add the nextInterval to ssStateS;
            add the nextIntervalT to ssStateT;
          end if;
        else
          exit the loop;
        end if;
    end if;
end for;
```

9. *sssRevT:=Reverse(ssStateT); sssRevS:=Reverse(ssStateS);*
10. *Return secondSteadyState:=[sssRevT, sssRevS];*

References

Ahissar, E. & Knutsen, K. (2008). Object localization with whiskers. *Biol Cybern*, **98**, 449–58. 51

Ahrens, K.F. & Kleinfeld, D. (2004). Current flow in vibrissa motor cortex can phase–lock with exploratory rhythmic whisking in rat. *J Neurophysiol*, **92**, 1700–7. 21, 71

Akaike, H. (1974). A new look at the statistical model identification. *IEEE T Automat Contr*, **19**, 716–22. 44

Alloway, K. (2008). Information processing streams in rodent barrel cortex: the differential functions of barrel and septal circuits. *Cereb Cortex*, **18**, 979–98. 58

Armstrong-James, M., Fox, K. & Das-Gupta, A. (1992). Flow of excitation within rat barrel cortex on stiking a single vibrissa. *J Neurophysiol*, **68**, 1345–57. 69

Awiszus, F. (1992). Reduction of a hodgkin-huxley-type model for a mammalian neuron at body temperature. *Biological Cybernetics*, **67**, 427–432. 93

Baillet, S., Mosher, J., Leahy, R. & Shattuck, D. (1999). Brainstorm: a matlab toolbox for the processing of meg and eeg signals. In *Proceedings of the 5th Int. Conf. on Human Brain Map., NeuroImage*, vol. 9, S246. 7

Barach, J. & WikswoJr., J. (1987). Computer simulation of action potential propagation in septated nerve fibers. *Biophysical Journal*, **51**, 177–183. 89

Bernardet, U., Blanchard, M. & Verschure, P. (2002). Iqr: a distributed system for real-time real-world neuronal simulation. *Neurocomputing*, **44:46**, 1043–8. 83

Bock, H. (2007). Clustering methods: a history of k-means algorithms. In P. Brito, G. Cucumel & F. de Carvalho, eds., *Selected contributions in data analysis and classification*, vol. 888, 161–72, Springer Verlag, Heidelberg. 74

Bokil, H.S., Pesaran, B., Andersen, R. & Mitra, P. (2006). A method for detection and classification of events in neural activity. *IEEE T Bio–Med Eng*, **53**, 1678–87. 64

Bologna, L.L., Pasquale, V., Garofalo, M., Gandolfo, M., Baljon, P.L., Maccione, A., Martinoia, S. & Chiappalone, M. (2010). Investigating neuronal activity by spycode multi-channel data analyzer. *Neural Networks*, **23**, 685–697. 6, 10

Bonomini, M.P., Ferrandez, J.M., Bolea, J.A. & Fernandez, E. (2005). Data-means: An open source tool for the classification and management of neural ensemble recordings. *Journal of Neuroscience Methods*, **148**, 137–146. 6, 10

Bowman, A. & Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis.*. Oxford University Press, New York. 46

Bressler, S.L. & Freeman, W.J. (1980). Frequency analysis of olfactory system eeg in cat, rabbit, and rat. *Electroencephalography and Clinical Neurophysiology*, **50**, 19 – 24. 7

Brett-Green, B., Chen-Bee, C. & Frostig, R. (2001). Comparing the functional representations of central and border whiskers in rat primary somatosensory cortex. *J Neurosci*, **21**, 9944–54. 58

Buzsaki, G. (2004). Large-scale recording of neuronal ensembles. *Nat Neurosci*, **7**, 446–51. 5

Carbone, E. & Lux, H. (1984). A low voltage–activated, fully inactivating ca channel in vertebrate sensory neurones. *Nature*. 98

Carbone, E. & Lux, H. (1987a). Kinetics and selectivity of a low–voltage–activated calcium current in chick and rat sensory neurones. *J Physiol.*. 98

Carbone, E. & Lux, H. (1987b). Single low–voltage–activated calcium channels in chick and rat sensory neurones. *J Physiol.*. 98

Castro-Alamancos, M. & Oldford, E. (2002). Cortical sensory suppression during arousal is due to the activity-dependent depression of thalamocortical synapses. *J Physiol*, **541**, 319–31. 66

Chiang, M. & Mirkin, B. (2010). Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *J Classif* , **27**, 3–40. 75

Colwell, L.J. & Brenner, M.P. (2009). Action potential initiation in the hodgkin-huxley model. *PLoS Comput Biol*, **5**, e1000265. 89

Coulter, D., Huguenard, J. & Prince, D. (1989). Calcium currents in rat thalamocortical relay neurones: kinetic properties of the transient, low-threshold current. *The Journal of Physiology*, **414**, 587–604. 97

Cui, J., Xu, L., Bressler, S.L., Ding, M. & Liang, H. (2008). Bsmart: A matlab/c toolbox for analysis of multichannel neural time series. *Neural Networks*, **21**, 1094–1104. 6, 11

Cutmore, T. & Celka, P. (2008). Composite noise reduction of erps using wavelet, model–based, and principal component subspace methods. *J Psychophysiol*, **22**, 111–20. 41

Dagtekin, M. (2006). *A chopper modulated amplifier system design for in–vitro neural recording*. PhD thesis, North Carolina State University, USA. 41

De Cheveigne, A. & Simon, J. (2008). Denoising based on spatial filtering. *J Neurosci Methods*, **171**, 331–9. 41

Delorme, A. & Makeig, S. (2003). Eeg changes accompanying learned regulation of 12-hz eeg activity. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **11**, 133–137. 7

Delorme, A. & Makeig, S. (2004). Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of Neuroscience Methods*, **134**, 9–21. 6, 8

Delorme, A., Makeig, S., Fabre-Thorpe, M. & Sejnowski, T. (2002). From single-trial eeg to brain area dynamics. *Neurocomputing*, **44-46**, 1057 – 1064. 7

Deschenes, M., Paradis, M., Roy, J. & Steriade, M. (1984). Electrophysiology of neurons of lateral thalamic nuclei in cat: resting properties and burst discharges. *J Neurophysiol*, **51**, 1196–1219. 97

Di, S., Baumgartner, C. & Barth, D. (1990). Laminar analysis of extracellular field potentials in rat vibrissa/barrel cortex. *J Neurophysiol*, **63**, 832–40. 66, 69

Diamond, M., vo Heimendahl, M., Knutsen, P., Kleinfeld, D. & Ahissar, D. (2008). 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, **9**, 601–12. 51

Dodge, Y. (2003). *The Oxford Dictionary of Statistical Terms.*. Oxford University Press, New York. 42

Dota, C., Skallefell, B., Edvardsson, N. & Fager, G. (2002). Computer-based analysis of dynamic qt changes: toward high precision and individual rate correction. *Ann Noninvas Electro*, **7**, 289–301. 64

Dota, C., Edvardsson, N., Skallefell, B. & Fager, G. (2009). Pc-based ecg waveform recognition and validation of novel software against a reference ecg database. *Ann Noninvas Electro*, **14**, S42–7. 64

Duann, J., Jung, T. & adn T.J. Sejnowski, S.M. (2002). fmrlab: an ica toolbox for fmri data analysis. In *Human Brain Mapping, Sendai, Japan*. 7

Egert, U., Knott, T., Schwarz, C., Nawrot, M., Brandt, A., Rotter, S. & Diesmann, M. (2002). Mea-tools: an open source toolbox for the analysis of multi-electrode data with. *Journal of Neuroscience Methods*, **117**, 33–42. 6, 9

Einevoll, G., Pettersen, K., Devor, A., Ulbert, I., Halgren, E. & Dale, A. (2007). Laminar population analysis: estimating firing rates and evoked synaptic activity from multielectrode recordings in rat barrel cortex. *J Neurophysiol*, **97**, 2174–90. 69

Fox, K. (2008). *Barrel Cortex*. Cambridge University Press, Cambridge. xx, 51, 52, 59, 60, 61, 63

Friston, K. (1995). Statistical parametric mapping: Ontology and current issues. *J. Cereb. Blood Flow Metab.*, **15**, 361–370. 7

Gloor, P. (1984). Electrophysiology of generalized epilepsy. In P. Schwartzkroin & H. Wheal, eds., *Electrophysiology of Epilepsy*, 109–136, Academic press, London. 97

Goldberg, D., Victor, J., Gardner, E. & Gardner, D. (2009). Spike train analysis toolkit: enabling wider application of information–theoretic techniques to neurophysiology. *Neuroinformatics*, **7**, 165–178. 6, 11

Goldstein, S.S. & Rall, W. (1974). Changes of action potential shape and velocity for changing core conductor geometry. *Biophysical Journal*, **14**, 731–757. 89

Gunay, C., Edgerton, J., Li, S., Sangrey, T., Prinz, A. & Dieter-Jaeger (2009). Database analysis of simulated and recorded electrophysiological datasets with pandoras toolbox. *Neuroinformatics*, **7**, 93–111. 6, 12

Haas, S., Frei, M., Osorio, I., Pasik-Duncan, B. & Radel, J. (2003). Eeg ocular artifact removal through armax model system identification using extended least squares. *Communications in Information and Systems*, **3**, 19–40. 82

Harrison, R. & Charles, C. (2003). A low-power low-noise cmos amplifier for neural recording applications. *IEEE J Solid-St Circ*, **38**, 358–65. 41

Hazan, L., Zugaro, M. & Buzsaki, G. (2006). Klusters, neuroscope, ndmanager: A free software suite for neurophysiological data processing and visualization. *J Neurosci Methods*, **155**, 207–16. 6, 11

Hernndez-Cruz, A. & Pape, H. (1989). Identification of two calcium currents in acutely dissociated neurons from the rat lateral geniculate nucleus. *J Neurophysiol.*. 98

Herz, A.V., Meier, R., Nawrot, M.P., Schiegel, W. & Zito, T. (2008). G-node: An integrated tool-sharing platform to support cellular and systems neurophysiology in the age of global neuroinformatics. *Neural Networks*, **21**, 1070–1075. 6, 9

Hille, B. (2001). *Ion Channels of Excitable Membranes.*. Sinauer Associates Inc., Massachusetts, USA, 3rd edn. 88

Hodgkin, A. & Huxley, A. (1952a). The components of membrane conductance in the giant axon of loligo. *J Physiol*, **116**, 473–496. 90

Hodgkin, A. & Huxley, A. (1952b). Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo. *J Physiol*, **116**, 449–472. 89, 90

Hodgkin, A. & Huxley, A. (1952c). The dual effect of membrane potential on sodium conductance in the giant axon of loligo. *J Physiol*, **116**, 497–506. 90

Hodgkin, A. & Huxley, A. (1952d). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol*, **117**, 500–544. 88, 90, 97

Huang, Y., Li, X., Li, Y., Xu, Q., Lu, Q. & Liu, Q. (2008). An integrative analysis platform for multiple neural spike train data. *Journal of Neuroscience Methods*, **172**, 303 – 311. 6, 11

Ishida, A., Kaneko, A. & Tachibana, M. (1984). Responses of solitary retinal horizontal cells from carassius auratus to l–glutamate and related amino acids. *The Journal of Physiology*, **348**, 255–270. 99

Jahnsen, H. & Llins, R. (1984a). Electrophysiological properties of guinea–pig thalamic neurones: an in vitro study. *The Journal of Physiology*, **349**, 205–226. 97

Jahnsen, H. & Llins, R. (1984b). Ionic basis for the electro-responsiveness and oscillatory properties of guinea-pig thalamic neurones in vitro. *The Journal of Physiology*, **349**, 227–247. 97

Jellema, T., Brunia, C. & Wadman, W. (2004). Sequential activation of microcircuits underlying somatosensory-evoked potentials in rat neocortex. *Neuroscience*, **129**, 283–95. xx, 52, 59, 61, 66, 69

Jin, X., Hu, J.F. & Liang, P.J. (2004). Possible mechanism of flicking-induced short–term plasticity in retinal cone–lhc synapse: a computational study. *Biol. Cybern.*, **90**, 360–367. 97, 98

Jiruska, P., Cmejla, R., Powell, A., Changa, W., Vreugdenhil, M. & Jefferys, J. (2009). Reference noise method of removing powerline noise from recorded signals. *J Neurosci Methods*, **184**, 110–4. 41

Jung, T.P., Makeig, S., Westerfield, M., Townsend, J., Courchesne, E. & Sejnowski, T.J. (2001). Analysis and visualization of single-trial event-related potentials. *Human Brain Mapping*, **14**, 166–185. 7

Kale, S. & Dudul, S. (2009). Intelligent noise removal from emg signal using focused time–lagged recurrent neural network. *Applied Computational Intelligence and Soft Computing*, **2009**, 1–12. 41

Kaneko, A. & Tachibana, M. (1985). Effects of l-glutamate on the anomalous rectifier potassium current in horizontal cells of carassius auratus retina. *The Journal of Physiology*, **358**, 169–182. 99

Kaur, S., Rose, H., Lazar, R., Liang, K. & Metherate, R. (2005). Spectral integration in primary auditory cortex: laminar processing of afferent input, in vivo and in vitro. *Neuroscience*, **134**, 1033–45. 66

Kublik, E. (2004). Contextual impact on sensory processing at the barrel cortex of awake rat. *Acta Neurobiol Exp*, **64**, 229–38. 21, 71

Kwon, K.Y., Eldawlatly, S. & Oweiss, K. (2009). Neuroquest: a comprehensive tool for large scale neural data processing and analysis. In *Proc of the 4th International IEEE EMBS Conference on Neural Eng*, 622–5. 5, 6, 12

Landlaw, E. & DiStefano Third, J. (1984). Multiexponential, multicompartmental, and noncompartmental modeling. ii. data analysis and statistical consideration. *Am J Physiol Regul Integr Comp Physiol*, **246**, R665–77. 43

Legatt, A., Arezzo, J. & Vaughan, H. (1980). Averaged multiple unit activity as an estimate of phasic changes in local neuronal activity: effects of volume-conducted potentials. *J Neurosci Methods*, **2**, 203–17. 71

Lidierth, M. (2009). sigtool: A matlab-based environment for sharing laboratory-developed software to analyze biological signals. *J Neurosci Methods*, **178**, 188–96. 6, 8

Liman, E.R., Knapp, A.G. & Dowling, J.E. (1989). Enhancement of kainate-gated currents in retinal horizontal cells by cyclic amp-dependent protein kinase. *Brain Research*, **481**, 399–402. 98

Luscher, H. & Shiner, J. (1990). Simulation of action potential propagation in complex terminal arborizations. *Biophysical Journal*, **58**, 1389–1399. 89

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc of the Fifth Berkeley Symp on Math Statist and Prob*, vol. 1, 281–97. 74

Madsen, K., Nielsen, H. & Tingleff, O. (2004). *Methods for Non-Linear Least Squares Problems.*. Informatics and Mathematical Modelling, Technical University of Denmark (DTU), Kgs. Lyngby, 2nd edn. 72

Magri, C., Whittingstall, K., Singh, V., Logothetis, N. & Panzeri, S. (2009). A toolbox for the fast information analysis of multiple-site lfp, eeg and spike train recordings. *BMC Neuroscience*, **10**, 81. 6, 11

Mahmud, M., Girardi, S., Maschietto, M., Rahman, M., Bertoldo, A. & Vassanelli, S. (2009a). Noise characterization of electrophysiological signals recorded from high resolution brain–chip interface. In *Proceedings of the International Symposium on Bioelectronics and Bioinformatics (ISBB2009), Melbourne, Australia*, 84–7. 31

Mahmud, M., Girardi, S., Maschietto, M., Rahman, M., Bertoldo, A. & Vassanelli, S. (2009b). Slow stimulus artefact removal through peak–valley detection of neuronal signals recorded from somatosensory cortex by high resolution brain–chip interface. In *IFMBE Proceedings of the World Congress 2009 in Medical Physics and Biomedical Engineering (WC2009), Munich, Germany*, 2062–5. 30, 38, 45

Mahmud, M., Bertoldo, A., Girardi, S., Maschietto, M. & Vassanelli, S. (2010a). Sigmate: a matlab-based neuronal signal processing tool. In *Proceedings of the 32nd annual international conference of the IEEE engineering in medicine and biology society (IEEE EMBC2010)*, 1352–5. 11, 44

Mahmud, M., Bertoldo, A., Maschietto, M., Girardi, S. & Vassanelli, S. (2010b). Automatic detection of layer activation order in information processing pathways of rat barrel cortex under mechanical whisker stimulation. In *Proceedings of the 32nd annual international conference of the IEEE engineering in medicine and biology society (IEEE EMBC2010)*, 6095–8. 33, 53, 71

Mahmud, M., Pasqualotto, E., Bertoldo, A., Girardi, S., Maschietto, M. & Vassanelli, S. (2010c). An automated method for detection of layer activation order in information processing pathway of rat barrel cortex under mechanical whisker stimulation. *Journal of Neuroscience Methods*, **In Press**. 33

Mahmud, M., Travalin, D., Bertoldo, A., Girardi, S., Maschietto, M. & Vassanelli, S. (2010d). A contour based automatic method to classify local field potentials recorded from rat barrel cortex. In *Proceedings of the 5th Cairo International Biomedical Engineering Conference, Cairo, Egypt*, 163–166. 33

Makeig, S. (1993). Auditory event-related dynamics of the eeg spectrum and effects of exposure to tones. *Electroencephalography and Clinical Neurophysiology*, **86**, 283 – 293. 7

Makeig, S., Jung, T.P., Bell, A.J., Ghahremani, D. & Sejnowski, T.J. (1997). Blind separation of auditory event-related brain responses into independent?components. *Proceedings of the National Academy of Sciences of the United States of America*, **94**, 10979–10984. 7

Makeig, S., Westerfield, M., Jung, T.P., Covington, J., Townsend, J., Sejnowski, T.J. & Courchesne, E. (1999). Functionally independent components of the late positive event-related potential during visual spatial attention. *J. Neurosci.*, **19**, 2665–2680. 7

Makeig, S., Westerfield, M., Jung, T., Enghoff, S., Townsend, J., Courchesne, E. & Sejnowski, T. (2002). Dynamic brain sources of visual evoked responses. *Science*, **295**, 690–694. 7

Mammen, A.L., Kameyama, K., Roche, K.W. & Huganir, R.L. (1997). Phosphorylation of the a–amino–3–hydroxy–5–methylisoxazole4–propionic acid receptor glur1 subunit by calcium/ calmodulin–dependent kinase ii. *Journal of Biological Chemistry*, **272**, 32528–32533. 98

Maschietto, M., Mahmud, M., Girardi, S. & Vassanelli, S. (2009). A high resolution bi-directional communication through a brain-chip interface. In *Proceedings of the 2009 ECSIS Symposium on Advanced Technologies for Enhanced Quality of Life (AT-EQUAL 2009), Iasi, Romania*, 32–5. 41, 70

McCormick, D.A., Shu, Y. & Yu, Y. (2007). Hodgkin and huxley model – still standing? *Nature*, **445**, E1–E2. 89

McMahon, D.G., Rischert, J.C. & Dowling, J.E. (1994). Protein content and camp-dependent phosphorylation of fractionated white perch retina. *Brain Research*, **659**, 110 – 116. 98

Megevand, P., Troncoso, E., Quairiaux, C., Muller, D., Michel, C. & Kiss, J. (2009). Long-term plasticity in mouse sensorimotor circuits after rhythmic whisker stimulation. *J Neurosc*, **29**, 5326–35. 66

Meier, R., Egert, U., Aertsen, A. & Nawrot, M.P. (2008). Find – a unified framework for neural data analysis. *Neural Networks*, **21**, 1085–1093. 6, 8

Milone, F., Mahmud, M., Minelli, T., Rahman, M. & Vassanelli, S. (2010). Cns 10 hz led 650 nm stimulation: Measures and hypotheses on the possible mechanisms of reinforcement of the alpha brain rhythms. In *Mind Force 2010: ConVersActions on the Embodied Mind, Centre for the Study of Complex Systems, University of Siena, Italy*. 99

Milstein, J., Mormann, F., Fried, I. & Koch, C. (2009). Neuronal shot noise and brownian $1/f^2$ behavior in the local field potential. *PLoS ONE*, **4**, e4338. 42

Mitzdorf, U. (1985). Current source-density method and application in cat cerebral cortex: investigation of evoked potentials and eeg phenomena. *Physiol Rev*, **65**, 37–100. 66

Mitzdorf, U. & Singer, W. (1980). Monocular activation of visual cortex in normal and monocularly deprived cats: an analysis of evoked potentials. *J Physiol*, **304**, 203–20. 66

Mrup, M., Hansen, L.K. & Arnfred, S.M. (2007). Erpwavelab: A toolbox for multi-channel analysis of time-frequency transformed event related potentials. *Journal of Neuroscience Methods*, **161**, 361–368. 6, 7

Naundorf, B., Wolf, F. & Volgushev, M. (2006). Unique features of action potential initiation in cortical neurons. *Nature*, **440**, 1060–1063. 89

Niedermeyer, E. & da Silva, F.L. (2005). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields.*. Lippincott Williams & Wilkins, Philadelphia, USA. 81

Novellino, A., Chiappalone, M., Maccione, A. & Martinoia, S. (2009). Neural signal manager: a collection of classical and innovative tools for multi-channel spike train analysis. *Int J Adapt Control Signal Process*, **23**, 999–1013. 6, 10

Nowycky, M.C., Fox, A.P. & Tsien, R.W. (1985). Three types of neuronal calcium channel with different calcium agonist sensitivity. *Nature*, **316**, 440–443. 98

Ohno, K., Funase, A., Cichocki, A. & Takumi, I. (2006). Analysis of eeg signals in memory guided saccade tasks. In *FMBE Proceedings of World Congress on Medical Physics and Biomedical Engineering, COEX Seoul, Korea*, vol. 14/4, 2264–7. 81

Okun, M., Naim, A. & Lampl, I. (2010). The subthreshold relation between cortical local field potential and neuronal firing unveiled by intracellular recordings in awake rats. *J Neurosci*, **30**, 4440–8. 71

Oweiss, K. & Anderson, D. (2001). Noise reduction in multichannel neural recordings using a new array wavelet denoising algorithm. *Neurocomuting*, **38–40**, 1687–93. 41

Paralikar, K., Raob, C. & Clementa, R. (2009). New approaches to eliminating common–noise artifacts in recordings from intracortical microelectrode arrays: Inter-electrode correlation and virtual referencing. *J Neurosci Methods*, **181**, 27–35. 41

Perelman, Y. & Ginosar, R. (2007). An integrated system for multichannel neuronal recording with spike/lfp separation, integrated a/d conversion and threshold detection. *IEEE T Bio-med Eng*, **54**, 130–7. 41

Pettersen, K., Devor, A., Ulbert, I., Dale, A. & Einevoll, G. (2006). Current-source density estimation based on inversion of electrostatic forward solution: Effects of finite extent of neuronal activity and conductivity discontinuities. *J Neurosci Meth*, **212**, 116–33. 33, 52, 57

Pfurtscheller, G. & Aranibar, A. (????). 7

Piotrowskia, Z. & Rozanowski, K. (2010). Robust algorithm for heart rate (hr) detection and heart rate variability (hrv) estimation. *Acta Physica Polonica A*, **118**, 131–5. 64

Pouzat, C. & Laurent, O.M.G. (2002). Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *J Neurosci Methods*, **122**, 43–57. 11

Prochazka, A., Mushahwar, V. & McCreery, D. (2001). Neural prostheses. *J Physiol*, **533**, 99–109. 5, 41

Quiroga, R. (2007). Spike sorting. *Scholarpedia*, **2**, 3583. 71

Quiroga, R., Nadasdy, Z. & Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, **16**, 1661–87. 5, 6, 11, 27, 35, 71

Rappelsberger, P., Pockberger, H. & Petsche, H. (1981). Current source density analysis: methods and application to simultaneously recorded field potentials of the rabbit's visual cortex. *Pflugers Arch*, **389**, 159–70. 58

Rodriguez, E., George, N., Lachaux, J.P., Martinerie, J., Renault, B. & Varela, F.J. (1999). Perception's shadow: long-distance synchronization of human brain activity. *Nature*, **397**, 430–433. 7

Rohalova, M., Sykacek, P., Koska, M. & Dorffner, G. (2001). Detection of the eeg artifacts by the means of the (extended) kalman filter. *Measurement Science Review*, **1**, 59–62. 82

Saha, R., Jackson, N., Patel, C. & Muthuswamy, J. (2010). Highly doped polycrystalline silicon microelectrodes reduce noise in neuronal recordings *in–vivo*. *IEEE T Neur Sys Reh*, **18**, 489–97. 41

Sahin, M. (2005). A low-noise preamplifier for nerve cuff electrodes. *IEEE T Neur Sys Reh*, **13**, 561–5. 41

Schmidtner, M. & Fromherz, P. (2006). Functional na+ channels in cell adhesion probed by transistor recording. *J Biophys*, **90**, 183–9. 17

Schubert, D., Kotter, R. & Staiger, J. (2007). Mapping functional connectivity in barrel–related columns reveals layer– and cell type–specific microcircuits. *Brain Struct Funct*, **212**, 107–19. 52

Sekino, M., Ohsaki, H., Yamaguchi-Sekino, S., Iriguchi, N. & Ueno, S. (2009). Low-frequency conductivity tensor of rat brain tissues inferred from diffusion mri. *Bioelectromagnetics*, **30**, 489–99. 57

Shaw, F., Yen, C. & Chen, R. (2003). A simple and effective process for noise reduction of multichannel cortical field potential recordings in freely moving rats. *J Neurosci Methods*, **124**, 167–74. 41

Smith, L. & Mtetwa, N. (2007). A tool for synthesizing spike trains with realistic interference. *J Neurosci Methods*, **159**, 170–80. 6, 11

Snellings, A., Anderson, D. & Aldridge, J. (2006). Improved signal and reduced noise in neural recordings from close-spaced electrode arrays using independent component analysis as a preprocessor. *J Neurosci Methods*, **150**, 254–64. 41

Staba, R., Bergmann, P. & Barth, D. (2004). Dissociation of slow waves and fast oscillations above 200 hz during gaba application in rat somatosensory cortex. *J Physiol*, **561**, 205–14. 66, 69

Steriade, M. & Deschenes, M. (1984). The thalamus as a neuronal oscillator. *Brain Res.*. 97

Steriade, M., Jones, E.G. & Llinas, R.R. (1990). *Thalamic Oscillations and Signaling.*. Wiley, New York. 97

Swadlow, H., Gusev, A. & Bezdudnaya, T. (2002). Activation of a cortical column by a thalamocortical impulse. *J Neurosci*, **22**, 7766–73. 66

Swanson, L. (2003). *Brain Maps: Structure of the Rat Brain, third ed.*. Academic Press, London. 15

Szymanski, F., Garcia-Lazaro, J. & Schnupp, J. (2009). Current source density profiles of stimulus-specific adaptation in rat auditory cortex. *J Neurophysiol*, **102**, 1483–90. 58, 66

Tachibana, M. (1985). Permeability changes induced by l–glutamate in solitary retinal horizontal cells isolated from carassius auratus. *The Journal of Physiology*, **358**, 153–167. 99

Tallon-Baudry, C., Bertrand, O., Delpuech, C. & Pernier, J. (1996). Stimulus specificity of phase-locked and non-phase-locked 40hz visual responses in human. *J. Neurosci.*, **16**, 4240–4249. 7

Ulbert, I., Karmos, G., Heit, G. & Halgren, E. (2001). Early discrimination of coherent versus incoherent motion by multiunit and synaptic activity in human putative mt+. *Hum Brain Mapp*, **13**, 226–38. 58

Uranga, A., Lago, N., Navarro, X. & Barniol, N. (2004). A low noise cmos amplifier for eng signals. In *Proceedings of the IEEE International Symposium on Circuits and Systems, Vancouver, Canada*, 21–4. 41

van Hemmen, J. & Ritz, R. (1995). Neural coding: A theoretical vista of mechanisms, techniques, and applications. In S. Andersson, ed., *Lecture Notes in Computer Science, Analysis of Dynamical and Cognitive Systems*, vol. 888, 75–119, Springer Berlin / Heidelberg. 71

Vargas-Irwin, C. & Donoghue, J. (2007). Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *J Neurosci Methods*, **164**, 1–18. 6, 11

Vato, A., Bonzano, L., Chiappalone, M., Cicero, S., Morabito, F., Novellino, A. & Stillo, G. (2004). Spike manager: a new tool for spontaneous and evoked neuronal networks activity characterization. *Neurocomputing*, **58-60**, 1153–1161. 6, 11

Versace, M., Ames, H., Lveill, J., Fortenberry, B. & Gorchetchnikov, A. (2008). Kinness: a modular framework for computational neuroscience. *Neuroinformatics*, **6**, 291–309. 6, 9

Wagenaar, D., DeMarse, T.B. & Potter, S.M. (2005). Meabench: A toolset for multi-electrode data acquisition and on-line analysis. In *Proceedings*

of the 2nd International IEEE EMBS Conference on Neural Engineering, v–viii. 6, 10

Wang, X.J., Rinzel, J. & Rogawski, M.A. (1991). A model of the t–type calcium current and the low–threshold spike in thalamic neurons. *Journal of Neurophysiology*, **66**, 839–850. 97

Weiss, S. & Rappelsberger, P. (1996). Eeg coherence within the 13-18 hz band as a correlate of a distinct lexical organisation of concrete and abstract nouns in humans. *Neuroscience Letters*, **209**, 17 – 20. 7

Widrow, B., Glover Jr., J., McCool, J., Kaunitz, J., Williams, C., Hearn, R., Zeidler, J., Eugene Jr., D. & Goodlin, R. (1975). Adaptive noise cancelling: principles and applications. *Proc IEEE*, **63**, 1692–716. 41

Wise, K., Anderson, D., Hetke, J., Kipke, D. & Najafi, K. (2004). Wireless implantable microsystems: high-density electronic interfaces to the nervous system. *Proc IEEE*, **92**, 76–97. 5, 41

Woo, J., Miller, C., Abbas, P., Hongc, S. & Kima, I. (2006). Improved noise reduction in single fiber auditory neural responses using template subtraction. *J Neurosci Methods*, **155**, 319–27. 41

Yang, Z., Zhao, Q., Keefer, E. & Liu, W. (2009). Noise characterization, modeling, and reduction for *in-vivo* neural recording. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams & A. Culotta, eds., *Advances in Neural Information Processing Systems*, 22, 2160–8. 42

Zanchettin, C. & Ludermir, T. (2007). Wavelet filter for noise reduction and signal compression in an artificial nose. *Applied Soft Computing*, **7**, 246–56. 41