Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Analysis of Parallel Microelectrode Recordings

Bc. Jiří Vošmik

Supervisor: Mgr. Tomáš Sieger, Ph.D.
Field of study: Biomedical Engineering and Informatics
Subfield: Biomedical Informatics
January 2018

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Jiří Vošmik

Study programme: Biomedical Engineering and Informatics

Specialisation: Biomedical Informatics

Title of Diploma Thesis: Analysis of Parallel Microelectrode Recordings

Guidelines:

1. Study the method of deep brain stimulation of the subthalamic nucleus as an effective treatment of advanced Parkinson´s disease. Familiarize yourself with intraoperative microelectrode recordings (MER).
2. Study methods of MER preprocessing (artifact detection and removal, frequency filtering, detection and sorting of action potentials), which yield both continuous signals representing the activity of neuronal population, and discrete sequences of action potentials of individual neurons.
3. Study analytical methods of assessing couplings between MER-derived signals.
4. Design and implement a SW tool capable of identifying, exploring, and visualizing couplings between MER-based signals.
5. Using methods above, preprocess and analyse MER data recorded from several Parkinson´s disease patients.
6. Briefly interpret and discuss your results.

Bibliography/Sources:

[1] Dayan E.P and Abbott L. F. Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. The MIT Press, 1st edition, December 2001

[2] Emery N Brown, Robert E Kass & Partha P Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. Nature Neuroscience 7 (5), May 2004

[3] Sieger T. Processing and Statistical Analysis of Single-Neuron Recordings. Doctoral Thesis. FEL CTU, Prague, 2013

[4] Quiroga R. Q, Nadasdy Z, and Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural computation, 16(8):1661-1687, 2004

[5] Bastos AM and Schoffelen J-M. A Tutorial Review of Functional Connectivity Analysis Methods and Their Interpretational Pitfalls. Front. Syst. Neurosci. 9:175, 2016. doi: 10.3389/fnsys.2015.00175

[6] West T, Farmer S, et al. The Parkinsonian Subthalamic Network: Measures of Power, Linear, and Non-linear Synchronization and their Relationship to L-DOPA Treatment and OFF State Motor Severity. Front. Hum. Neurosci. 10:517, 2016. doi: 10.3389/fnhum.2016.00517

[7] Uri Eden. Introduction to Point Processes. http://www.stat.columbia.edu/~liam/teaching/neurostat-fall15/uri-eden-point-process-notes.pdf

Diploma Thesis Supervisor: Mgr. Tomáš Sieger, Ph.D.

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 11, 2017

Acknowledgements

I would like to thank my supervisor Mgr. Tomáš Sieger Ph.D. for the guidance, suggestions and criticisms he provided me with as well as for his patience.

In addition, I would like to thank Bára Malíková and Markéta Šmejkalová for their help and support.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 9.1.2018

. . . .
Jiří Vošmik

Abstract

Human brain is one of the most complicated known structures which has been receiving a large amount of attention from researchers. However, majority of the research so far was focused on characteristics and behaviour of neuronal populations. Studying the activity of individual neurons opens new possibilities in terms of applications of neuronal research and can deepen our understanding of the brain significantly.

In this thesis, a functional neuronal connectivity is studied. A principal source of the data for this thesis are microelectrode recordings. These signals, recorded directly in the brain, can contain distinguishable spikes from individual neurons and can therefore be used to analyze their behaviour. The methods of preprocessing these signals and assessing neuronal connectivity within them a were studied and implemented as original software tools.

These tools were employed in processing and analysing a large dataset of microelectrode recordings obtained during deep brain stimulation surgery in Parkinson's disease patients. The results of this analysis show how various measures of functional neuronal connectivity perform on the whole data set and its parts.

Keywords: microelectrode recordings, Parkinson's disease, deep brain stimulation, signal processing, spike sorting, functional neuronal connectivity, phase slope index, phase lag index, generalized linear models, mutual information, correlation

Supervisor: Mgr. Tomáš Sieger, Ph.D.

Abstrakt

Lidský mozek je jedna z nejkomplikovanějších známých struktur a jako takový je dlouhodobě zkoumán. Většina výzkumu mozku se zatím zabývala výzkumem chování celých populací neuronů. Detailní výzkum chování jednotlivých neuronů tak může významně přispět k již dosaženým znalostem v této oblasti a otevřít cestu novým aplikacím.

Tato diplomová práce se zabývá funkční konektivitou jednotlivých neuronů - vliv chování jednoho neuronu na další neurony. Zdrojem dat pro tento cíl jsou mikroelektrodové záznamy. Tyto signály jsou nahrávané přímo uvnitř mozku a obsahují rozlišitelnou aktivitu jednotlivých neuronů. Metody pro přezpracování těchto signálů a analýzu konektivity v nich obsažené byly implementovány jako původní software v rámci této práce.

Implementovaný software byl použit pro předzpracování velkého množství záznamů pořízených během hluboké mozkové stimulace - léčebné procedury pro Parkinsonovu chorobu - a dále pro analýzu konektivity v těchto záznamech. Výsledky ukazují, jak se různé metody pro analýzu konektivity chovají v závislosti na různých parametrech signálů.

Klíčová slova: Mikroelektrodové záznamy, Parkinsonova choroba, hluboká mozková stimulace, analýza signálů, třídění akčních potenciálů, funkční neuronální konektivita, phase slope index, phase lag index, generalizované lineární modely, vzájemná informace, korelace

Contents

Figures

Tables

Chapter 1

Introduction

Recording and analysing neuronal activity has been a subject of scientific studies for decades and new methods and results are still emerging every year, making progressively more applications practical. These applications range from greater understanding of the brain function to curing neural disorders and developing human-machine interfaces.

In this thesis, recordings of neuronal activity made during a deep brain stimulation surgery are studied. Deep brain stimulation is an invasive treatment for Parkinson's disease where a stimulation electrode is inserted into patient's subthalamic nucleus (STN), a structure deep in the brain.

To achieve accurate positioning in the nucleus, microelectrode recordings (MER) are used during the surgery. These recordings contain distinguishable activity of individual neurons and can be made in several locations at the same time, allowing a study of neuronal interactions between these locations.

In this work, these interactions are examined. The motivation for such analysis is that a knowledge about the neuronal interactions in STN could provide us with a greater insight into this otherwise hardly reachable brain structure and might help us understand the underlying mechanisms of Parkinson's disease.

The objectives of the thesis are the following:

1. Study the method of deep brain stimulation of the subthalamic nucleus as an effective treatment of advanced Parkinson's disease. Familiarize yourself with intraoperative microelectrode recordings (MER).

2. Study methods of MER preprocessing (artifact detection and removal, frequency filtering, detection and sorting of action potentials), which yield both continuous signals representing the activity of neuronal population, and discrete sequences of action potentials of individual neurons.

3. Study analytical methods of assessing couplings between MER-derived signals.

4. Design and implement a SW tool capable of identifying, exploring, and visualizing couplings between MER-based signals.

5. Using methods above, preprocess and analyse MER data recorded from several Parkinson's disease patients.

6. Briefly interpret and discuss your results.

1.1 Organisation of the thesis

Chapter 2 introduces the medical background for the work. This chapter deals with Parkinson's disease and the methods for its management, including deep brain stimulation. Furthermore, it describes the process of creating parallel microelectrode recordings during the deep brain stimulation surgery. Finally, some anatomical and physiological properties of neurons - the sources of the signals in question - are discussed there.

Chapter 3 introduces various methods of processing mictroelectrode recordings in order to obtain signals usable for scientific analysis. These methods include artifact management, frequency filtering and spike detection and sorting.

Chapter 4 discusses the topic of functional neuronal connectivity. Several measures of connectivity are introduced and possible problems with estimating connectivity are noted.

In chapter 5, the original results are presented. This includes the description of the implemented software and the work done with the real data - preprocessing, connectivity measurement and analysing the results.

Lastly, chapter 6 summarises the achieved results and proposes directions for future research.

Chapter 2

Background

In this chapter, I will briefly describe the medical background and the motivation for this work.

The subject of this thesis is processing of parallel microelectrode recordings and an analysis of couplings between such signals. Microelectrode recordings used were made in the course of deep brain stimulation neurosurgery, where they were utilised as one of the means of navigation. Deep brain stimulation, described in section 2.2, is a surgical procedure used to treat Parkinson's disease, a neurodegenerative disease described in section 2.1.

In section 2.3, I will characterise neurons and action potentials - the anatomical and physiological sources of microelectrode recordings.

2.1 Parkinson's disease

Parkinson's disease (PD) is a progressive neurodegenerative disorder, characterized by a loss of dopaminogenic neurons in substantia nigra, a part of the basal ganglia connected to the motor cortex. This leads to decreased dopamine levels in the mentioned areas of the brain, impairing their function. The primary cause that leads to the decay of the dopaminogenic neurons is not yet fully understood [Brodal, 2010].

The disease is most commonly associated with its motor symptoms - tremor, rigidity, akinesia and bradykinesia. The tremor manifests in the form of involuntary shakiness of the hands or of the jaw, its frequency is usually in the range of 3 to 6 Hz and it disappears with voluntary movement and in sleep. The rigidity, caused by increased muscle tone, can lead to stooped posture and to problems with balance and locomotion. Bradykinesia, i.e. slowness in the execution of movement, and akinesia, a difficulty in initiating voluntary movements, further contribute to the problems with motor skills the PD patients face. In addition, the PD patients experience various non-motor symptoms, such as depression, sleep disturbances, problems with abstract thinking, working memory and attention.

The treatment options for Parkinson's disease are focused on the management of its symptoms, as there is no known cure for the disease itself. Since the disease is characterised by reduced dopamine levels in certain areas of the brain, many of the treatments try to alleviate the symptoms by increasing the dopamine levels through medication.

One such medication is levodopa, a dopamine precursor capable of crossing the blood-brain barrier and being metabolised into dopamine. This treatment does reduce the severity of some of the symptoms, but it cannot stop the progress of the disease and it can lead to various side effects, since most of the levodopa is metabolised in areas of the brain other than basal ganglia. In addition, this treatment is not effective in all patients.

Other ways to manage Parkinson's disease include surgical procedures such as deep brain stimulation or lesions. The latter method is becoming less prevalent as medicament based management is preferable in most cases [Brodal, 2010] [Sieger, 2013].

2.2 Deep Brain Stimulation

Deep brain stimulation (DBS) is a surgical procedure used to treat Parkinson's disease and other disorders of the central nervous system. In DBS, a neurostimulator is implanted into the patient's body, with electrodes inserted directly into specific brain structures, which are then stimulated by electrical pulses from the device. For PD treatment, the electrodes are most commonly inserted into globus pallidus or to subthalamic nucleus (STN). A schematic illustration of DBS is shown in Figure 2.1.

Deep brain stimulation is used to treat Parkinson's disease patients unresponsive to medication. It provides a relief from the motor symptoms of the disease and unlike some other surgical methods, such as lesions, it is reversible. However, the mechanism of its action is poorly understood and it can introduce some adverse psychological effects [Sieger, 2013] [Brodal, 2010].

2.2.1 Surgery and microrecordings

For a DBS surgery to be successful, the target brain structure must be correctly located and the stimulation electrode must be accurately implanted inside it. Since the target brain structures are typically very small, the precision requirements are fairly steep. To satisfy these requirements, the

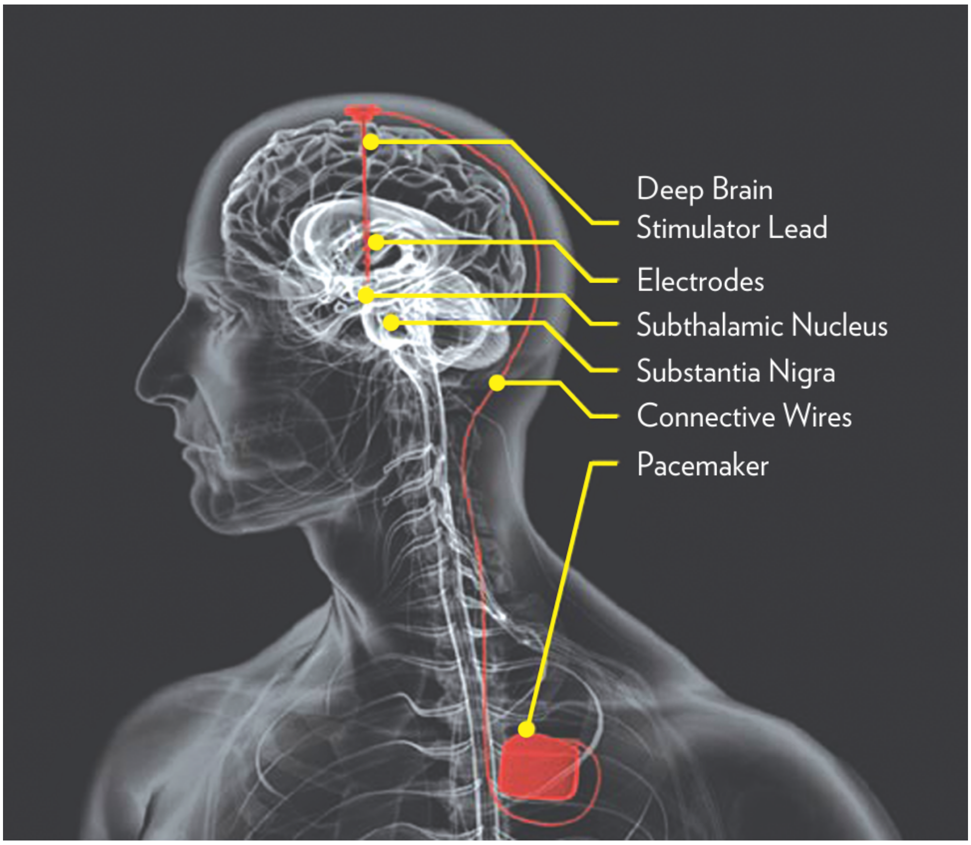


Figure 2.1: A schematic depiction of implanted DBS instruments [Peck, 2007]

patient's head is affixed to a stereotactic frame during the localisation and during the surgery itself.

Before the surgery, a magnetic resonance imaging (MRI) calibrated with the stereotactic frame is used to map the brain structures needed. With the MRI data, the stimulation electrode target is found and a trajectory of insertion which is as noninvasive as possible is planned. The target brain structure can be symmetrically located in both brain hemispheres, resulting in both having to be implanted separately. In such case, two trajectories of insertion are planned. This is true for DBS of STN.

In order to improve the accuracy of the mapping, microelectrode recordings are used and the recorded signals are displayed to the surgeon on screen and played back to him as sound. He then evaluates this information and assesses whether the correct positioning has been reached [Wild, 2015] [Lourens et al., 2013].

Signals recorded by microelectrodes in such surgeries are the principal source of data used in this work. This dataset therefore consists of signals recorded from several patients undergoing STN DBS treatment. For most of the patients, two smaller datasets were obtained - one for each STN in each brain hemisphere. These smaller datasets will henceforth be referred to as *trajectories*.

In each trajectory, five parallel microelectrodes arranged in a shape of a cross with 2 mm distance between them were used. These electrodes were gradually lowered deeper into the patient's brain. Depths along the trajectory were recorded with respect to a coordinate system where a depth of 0 mm is the expected depth of the target and positive depths representing areas of the brain deeper than the expected target. The recordings started when a depth of -10 mm was reached and the lowering of the electrodes was conducted in discrete 0.5 mm steps. A schematic representation of the process is displayed in figure 2.2.

In each depth along the trajectory, parallel signals were recorded. Due to a limitation of the recording equipment, only signals from 4 electrodes could be recorded at once. Because of this, two sets of parallel signals, each with different electrode missing, were recorded in each depth. A single set of parallel signals is referred to as a *position* from now on. The parallel signals in each position were recorded for 10 seconds.

The process of lowering electrodes and recording the signals lasts until the surgeon responsible decides that the microelectrodes have passed STN and are deeper in the brain. At this point, the mapping with microelectrodes is concluded. Afterwards, the stimulation electrode is implanted in place of the microelectrode which is evaluated as the one with the most desirable position in the STN.

In summary, the used dataset can be divided into trajectories. Each trajectory consists of positions - a set of 4 parallel 10 second signals recorded on a particular depth. Two positions were recorded in each depth.

2.3 Neurons

The nervous system consists of two main types of cells: neurons and glial cells. Glial cells or glia are the more numerous of the two types and they serve various purposes, such as structural support and protection of neurons. Neurons provide the main functions of the nervous system - receiving, processing and transmitting information. For this purpose, the neurons are connected into large networks. The human brain is one such network and it contains about 10^{11} neurons and 10^{14} interconnections between these neurons.

Neurons are separated from their environment by the neuronal membrane. This membrane is similar to the membranes of other cells in human body,

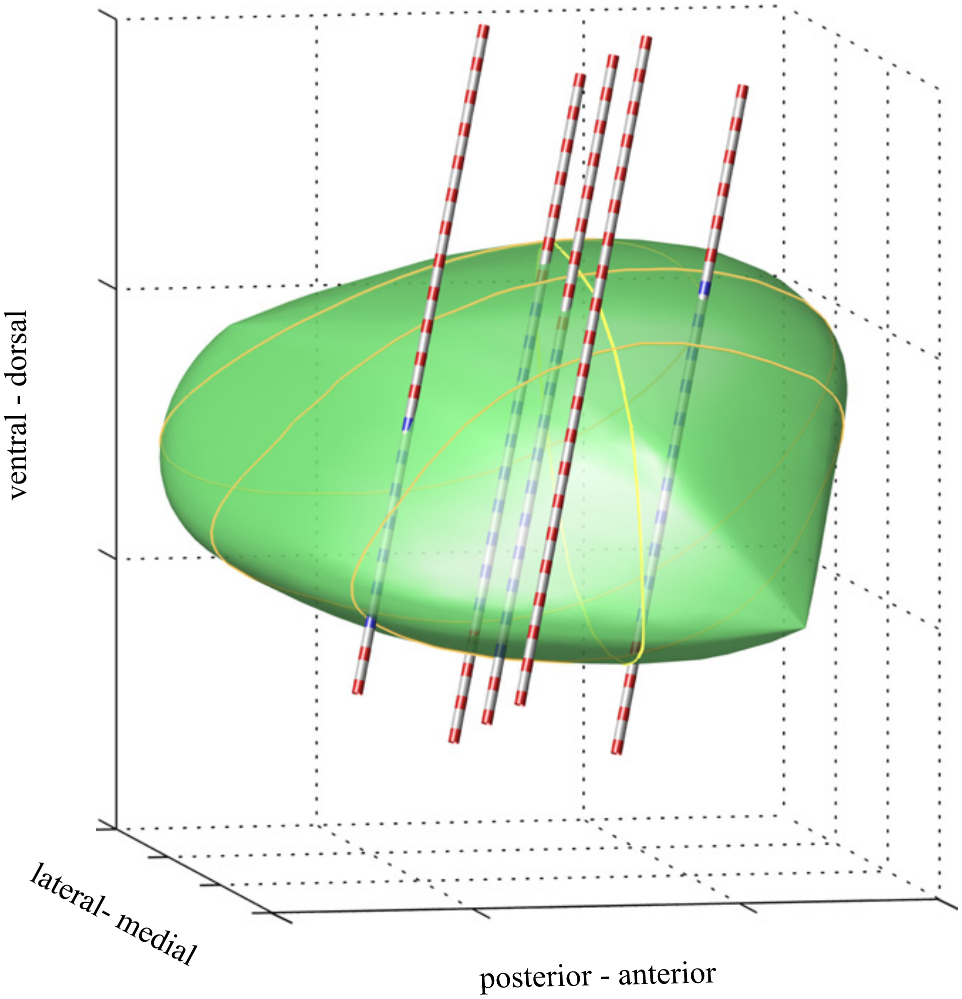


Figure 2.2: 3D depiction of the microelectrode recording process. The green volume represents STN with the electrodes running through it. The red and blue points on the electrodes represent the recording points outside and inside STN respectively [Lourens et al., 2013]

but contains specialized structures that allow neurons to accomplish their function - ion channels and ion pumps. Ion channels allow one or more types of ions to flow through the membrane into or out of the cell, while ion pumps actively transfer the ions from one side of the membrane to the other. Some of the ion channels can open or close depending on various conditions. For instance, the voltage-gated ion channels are in open or closed state based on the current membrane potential. The most important ions that play a role in neuronal membrane dynamics are sodium (Na^{+}), potassium (K^{+}), calcium(Ca^{2+}) and chloride (Cl^{-}) ions.

The concentrations of the different ions on either side of the neuronal membrane are typically not equal - some ions are more concentrated on one side of the membrane than on the other. In such situation a concentration gradient exists for this ion, causing it to move from the side with a higher concentration to the side with a lower concentration through the open ion channels. However, different concentrations of the ions also cause the neuronal membrane to become electrically charged - a nonzero potential inside the cell relative to the outside exists. The resulting voltage gradient also causes ions to pass through the membrane, according to their polarity. When the voltage and concentration gradients are balanced for all the ions, their movement stops and the membrane reaches a balanced state. The potential of the inside of the membrane relative to the outside in this state is called a resting membrane potential and it is about -70 mV.

Aside from the neuronal membrane, there are other features that distinguish the neurons from other cells. A structure of a typical neuron consists of the soma, a cell body containing the organelles standard to other animals cells and hosting most of the cell's metabolism, but also of specialized morphological features - the dendrites and the axon. A schema of a structure of a typical neuron is in Fig. 2.3.

Dendrites are short branches of the neuronal membrane capable of receiving information from other neurons. Each neuron has numerous dendrites that branch to form large dendritic trees, making neurons capable of receiving information from many other neurons.

The axon is a single branch of the neuronal membrane which is capable of propagating information in the form of a short electrical discharge, called action potential, to other neurons. Axons in humans can be over a meter long and end in axon terminals. Axon terminals are in contact with dendrites of other neurons and the point of contact is called the synapse.

There are two types of synapses: electrical and chemical. In electrical synapses, the potential of the presynaptic neuron directly influences the membrane potential of the postsynaptic neuron. In chemical synapses, action potential causes the release of a chemical - neurotransmitter - from the axonal terminal of the presynaptic neuron. This neurotransmitter binds to receptors on the postsynaptic neuron and causes depolarization or hyperpolarization of its membrane [Brodal, 2010], [Bear et al., 2007] [Dayan and Abbott, 2001].

2.3.1 Action potentials

As stated above, the neuronal membrane at rest is negatively charged at approximately -70 mV and concentration and voltage gradients are balanced for all the ions on either side of the membrane. The membrane of a neuron

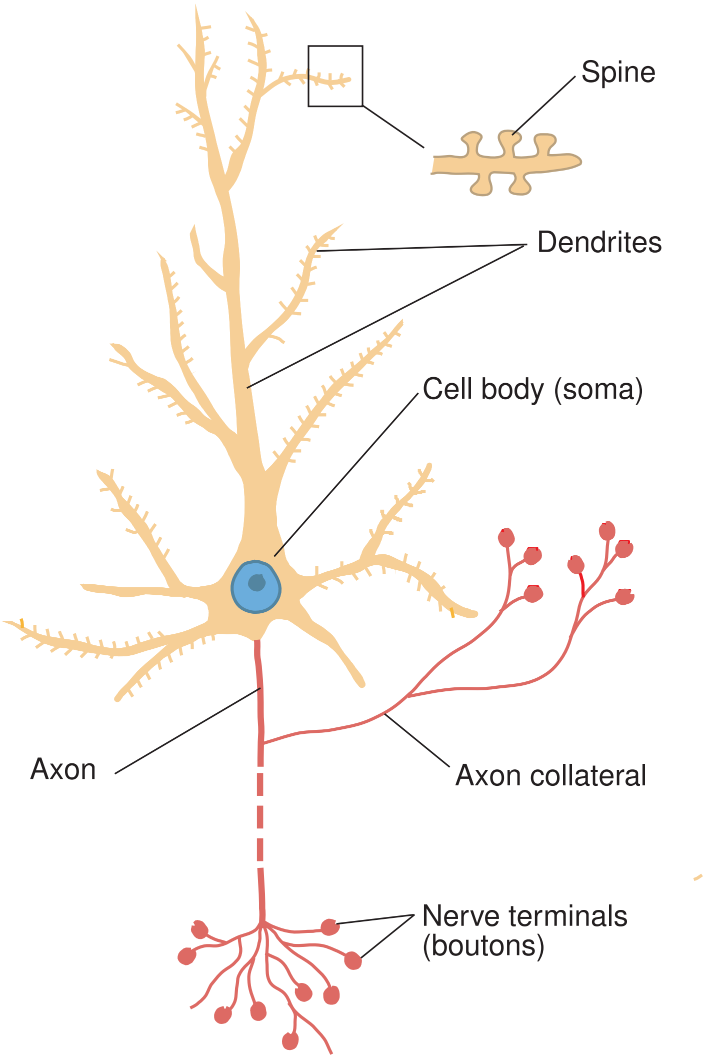


Figure 2.3: A structure of a typical neuron [Brodal, 2010]

can be polarized or depolarized as a result of other neurons acting on the neuron at synapses.

If the membrane is depolarized above certain threshold, a sequence of events causing a rapid and significant depolarization and subsequent repolarization of membrane takes place. This sequence of events results in measurable electrical signal called action potential or spike and unlike subthreshold potential changes, it propagates over large distances.

The mechanism of generating action potentials is dependent on the voltage-gated ion channels. When an action potential is triggered, the voltage-gated channels for sodium in the membrane temporarily open, followed by the potassium voltage-gated channels. This changes the permeability of the membrane for these ions and upsets the balance of forces generated by concentration and potential gradients, causing the ions to rapidly move through the membrane. First, sodium ions flow into the cell, causing a rapid depolarization of the membrane. Afterwards, potassium ions flow out of the cell, polarizing the membrane again, even beyond the resting membrane potential levels. When this process is done, energy dependent ion pumps transport some of the sodium and potassium ions back, restoring original ion concentrations and resting membrane potential. This whole process lasts several milliseconds.

Due to the necessity of restoration of the resting state of the membrane after spike, a neuron cannot generate an action potential again for a few miliseconds. This is called absolute refractory period. Even after the absolute refractory period is over, chances of the neuron firing again are significantly reduced for a while and this is called relative refractory period.

Action potential represents all or nothing response of the neuron to a stimulus. In addition, the course of the membrane potential in spikes from single neuron is always the same, independent of the input. This means that the information about the stimulus is not encoded in the spikes themselves. It is encoded in the firing times and firing frequencies of individual neurons and in the firing pattern of populations of neurons. Another important fact that can be derived from this is that individual neurons can be distinguished from each other using the shape of their spikes [Brodal, 2010], [Bear et al., 2007].

Chapter 3

Microelectrode recordings processing

In this chapter, I am going to discuss microelectrode recordings and the techniques used to process these signals.

In section 3.1 I will list some of the methods used to record neuronal activity, including microelectrode recordings. In section 3.2, I will go through some of the preprocessing methods used to detect and remove artifacts from the recorded signals. Finally, in section 3.3, I am going to describe the process of extracting spike trains from MERs.

3.1 Recording neuronal responses

Researchers have utilised many different approaches in order to gain insights about neuronal activity in the past. Traditional recording methods, such as electroencephalography (EEG), magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI) are capable of recording summarized activity of a population of neurons in a particular area of the brain, but give no information about individual neurons.

Methods capable of recording the activity of single neurons have been available for several decades and recent advances have made them a reasonably accessible source of information. One such method, microelectrode recordings (MER), or single-unit recordings, utilises very thin electrodes inserted directly into the patient's brain, enabling the analysis of the behaviour of the individual neurons.

MERs can be of two distinct types: intracellular and extracellurar. To perform an intracellular microelectrode recording, the electrode is either inserted directly into the soma of neurons or is tightly pressed against the neuronal membrane. Intracellular MERs can detect changes in the neuronal membrane potential, even if these changes remain subthreshold and the neuron does not fire. Due to the difficulty of performing an intracellular

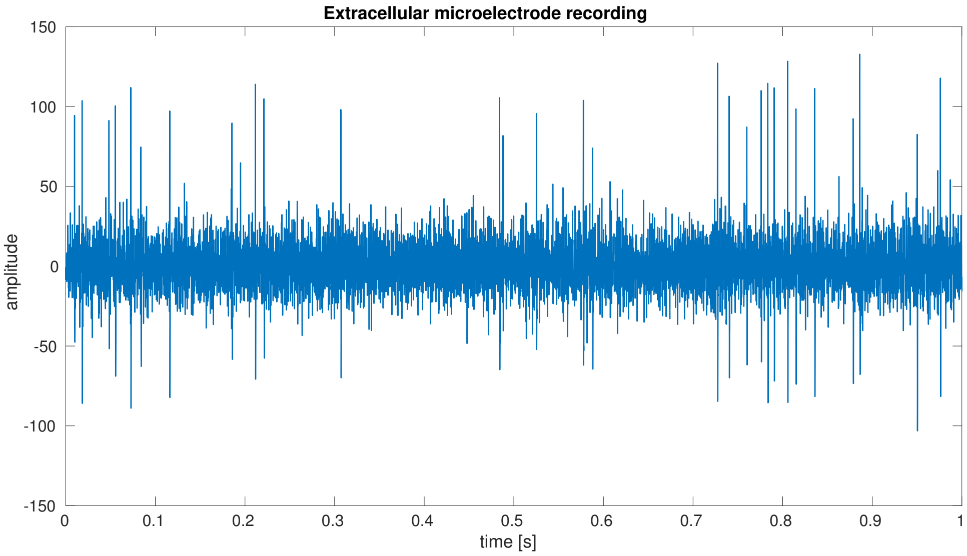


Figure 3.1: An example of time series of extracellular microelectrode recording. Individual spikes are clearly distinguishable from the background noise

MER in vivo, it is more commonly used to record in vitro preparations [Dayan and Abbott, 2001].

Extracellular microelectrode recordings utilize electrodes placed in the direct vicinity of one or more neurons. A signal recorded in such a way contains the spikes of neurons located near the tip of the electrode and a summarized activity of neurons further away from the electrode as noise, but cannot contain any subthreshold potentials. An example of such signal is displayed in figure 3.1. Extracellular microelectrode recordings are often made in vivo during behavioural experiments and can also be employed during DBS surgeries [Sieger, 2013].

Extracellular microelectrode recordings are the principal source of data utilised in this work and as such, the subsequent text will deal solely with this type of MERs.

3.2 Preprocessing

Extracellular MERs are, much like any other recording of neuronal activity, a mixture of physiological signals and of non-physiological artifacts. As described in the section 3.1, the source of the physiological part of MER are spiking neurons and this part manifests as neuronal spike trains mixed with noise. The non-physiological part of MERs, however, can have various sources and can take many different forms [Bakštein et al., 2017].

Most of the artifacts are introduced to the signals during the recording. To be able to record individual spiking neurons, the instrumentation in question has to be extremely sensitive. This means that it can also easily pick up noises from various external sources, such as the patient moving or someone talking during the recording. Furthermore, the signals can be corrupted by electromagnetic noise from nearby electrical appliances and by transient currents in the recording instruments.

These artifacts can then manifest in both the time series or in spectra of the signals. Beginnings and sometimes ends of the recordings can contain sections with amplitudes greatly exceeding the amplitudes of physiological signals, resulting from transient events on the recording instruments. Power spectral density of the signals often have spikes at 50 Hz (the power line frequency) and its harmonics or on other frequencies. Fluctuating baseline of the signals can be identified in its time series and can be seen as high power on very low frequencies in the spectrum.

Since the subject of this work is analysis of connectivity between physiological sources, the artifacts in the signals are obviously undesirable and could lead to false discoveries. Therefore, they have to be removed from the recordings before further analysis can take place. In some cases, the signals are only slightly corrupted and the artifacts can be filtered out. For instance, 50 Hz frequency artifacts in otherwise clean segments of the signals can be removed by frequency filtering. In other cases, the segments of the recordings are so heavily affected by the artifacts, that the separation of the artifacts from signal of interest cannot take place and these segments have to be discarded.

3.3 Spike detection and sorting

As described in the section 2.3.1, most of the information transmitted in the nervous system is contained within the firing times of the individual spikes and in the firing pattern of the neurons. This means that timing of the spikes is the part of MERs which carries the most of the information and it is therefore valuable to extract the spikes from raw microelectrode recordings. Furthermore, raw MERs can contain spikes from one or several neurons and sorting these spikes according to the neuron that fired them can aid further analyses [Brown et al., 2004].

Obtaining the spike trains of the individual neurons contained within a MER signal consists of two separate steps: spike detection and spike sorting.

In spike detection, specific characteristics of spikes are used to detect them in the signal. Most often, the characteristic exploited is the fact that spikes have higher amplitudes than the noise in the signal and the detection is based on simple amplitude thresholding. This can be combined with frequency filtering to eliminate possible high amplitude, low frequency local field potentials - summed electrical current flowing from neurons located close to the electrode but not close enough to generate detectable spike trains. Obviously, this procedure is based on some assumptions about the noise present and will fail if these assumptions are not met. Other methods of spike detection can be employed, such as transform-based methods. These methods might be more robust in high-noise cases. After the spikes are detected, their firing times and waveforms are extracted from the signals to be used in the sorting.

During the spike sorting, the spikes found in previous step are assigned to their putative source neurons. This is usually done by utilizing the fact that the waveforms of spikes fired from a single neuron have very little variability. Therefore, some discriminatory features can be extracted from the spikes and used to cluster the spikes using an unsupervised learning algorithm. This task is complicated by the the unknown number of actual spiking neurons contained in the signal [Wild, 2015] [Brown et al., 2004] [Oweiss, 2010].

In this work, I employ WaveClus algorithm for both spike detection and spike sorting. WaveClus was compared with other spike sorting algorithms in [Wild, 2015] and was evaluated as relatively slow, but as the most accurate method.

3.3.1 WaveClus

WaveClus [Quiroga et al., 2004] is an algorithm combining spike detection and sorting that utilises automatic amplitude thresholding, wavelet transformation and superparamagnetic clustering.

To perform spike detection, WaveClus first prepares the data by filtering it using 300-6000 Hz bandpass filter. Next, the spikes in the filtered signal x are detected by amplitude thresholding with the threshold thr_{wc} being set automatically as:

$$thr_{wc} = 4 \cdot \text{median}\left(\frac{|x|}{0.6745}\right) \tag{3.1}$$

The median in equation 3.1 is an estimation of the noise level in the signal. This way, WaveClus adapts to differing levels of background noise in signals and sets the threshold accordingly.

After the spike detection, waveforms of each detected spike are saved and wavelet transform is applied to obtain wavelet coefficients for each spike.

Coefficient selection using Kolgomorov-Smirnov test is performed in order to select only the maximally non-Gaussian coefficients that best separate the spike classes. The selected coefficients are then used as features in clustering.

WaveClus utilises superparamagnetic clustering (SPC) to sort spikes. This clustering algorithm is inspired by superparamagnetism - a natural property of some materials. It performs clustering based on nearest neighbors using several "temperature settings".

At low temperatures, very few of the spikes change clusters with temperature changes and results remain reasonably static. When the temperature changes in higher temperature ranges, many new clusters with just a few members are created. WaveClus searches for the highest temperature where some large cluster (containing at least 60 samples by default) is created and this temperature is used to perform the final clustering. In this way, the information describing how the samples change clusters together is used to find the optimal number of clusters [Quiroga et al., 2004].

Chapter 4

Coupling measures

In this chapter, I will review some of the methods that can be used to assess the presence and strength of a coupling or functional connectivity between a pair of parallelly recorded signals. Since the analysed data is available in two distinct forms - MER signals and spike trains - the methods analysing them can differ significantly. Although some of the methods used on raw signals can be applied to spike trains, specialized coupling metrics were used for both data types in this work.

In section 4.1, I will describe the first group of functional connectivity measures used:

- Pearson's correlation coefficient
- Cross-correlation function
- Mutual information

When assessing functional connectivity using these methods, several commonly encountered problems can cause spurious correlations and skew the results. Some of these problems are listed in the section 4.2 and in the section 4.3, I will describe a group of measures designed to overcome these problems:

- Imaginary part of coherence
- Phase slope index
- Phase lag index
- Weighted phase lag index

All of the measures listed so far evaluate a functional connectivity of a pair of MERs. In the context of this work, a statistically significant value of a coupling measure for a given pair of MERs implies a presence of functional connectivity. Therefore, a method for estimation of significance for each computed measure is needed. In section 4.4, I will describe a general scheme used in this work to evaluate significance of the MER coupling measures.

Lastly, in section 4.5, I will outline some mathematical notation and concepts used to model neuronal spike trains. Then, I will apply these concepts to describe generalized linear models and describe how this modelling technique can be used to assess coupling between two spike trains.

4.1 Time domain MER coupling measures

This section contains information about metrics that can be used to detect relations between MER signals and utilise the information contained in time series of these signals.

In the subsequent sections, the following common notation is used: The functional connectivity is being assessed for a pair of signals x and y. These signals are both the same length of T seconds and were sampled with the same frequency f_s. Therefore, both of the signals are $n = Tf_s$ samples long. For the sake of clarity, the signals x and y are treated simply as realizations of random variables in the following formulas.

4.1.1 Pearson's correlation coefficient

Pearson's correlation coefficient (PCC) is one of the simplest metrics and it is a measure of linear dependency of two random variables. Pearson's correlation coefficient of x and y can be computed using the following formula:

$$PC_{x,y} = \frac{\mathbf{E}\left[(x-\mu_x)(y-\mu_y)\right]}{\sigma_x \sigma_x} \tag{4.1}$$

where μ_x is the mean of x and σ_x is the standard deviation of x. The equation 4.1 shows that the order of input signals does not impact the resulting PCC, which is therefore a non-directed

The Pearson's correlation coefficient is bounded to the interval $[-1, 1]$ and the values 1 and -1 signify total positive linear dependency and total negative linear dependency respectively. On the other hand, PCC of 0 indicates no linear correlation between x and y. Note that PCC cannot detect any nonlinear relationships between the signals.

For the purposes of PCC estimation, the signals x and y are treated simply as realisations of random variables and the temporal structure of the data has no effect on the result. In other words, pairs of signal samples could be randomly shuffled before calculating the coefficient and the result would be the same. This also means that PCC cannot detect coupling in the signals which is significantly time-lagged, i.e. the coefficient might be small even in the case of two highly correlated signals which were time shifted with respect to each other [Bastos and Schoffelen, 2015].

4.1.2 Cross-correlation function

If a functional connectivity acting at a specific time lag of τ samples was investigated, PCC could be still be used by shifting one of the signals by τ samples before calculating the coefficient. Doing this for every possible lag and recording the correlation coefficients results in cross-correlation function (CCF):

$$CCF_{x,y}(\tau) = \frac{\mathbf{E}\left[(x(t)-\mu_x)(y(t+\tau)-\mu_y)\right]}{\sigma_x \sigma_x} \tag{4.2}$$

Location of the maximum of CCF is the time lag for which the two signals are the most correlated. The maximum of the cross-correlation function can be therefore used as a non-directed measure of functional connectivity that functions in a similar way to PCC, but can detect time lagged interactions. This is desirable since real interactions cannot be instantaneous.

Much like Pearson's correlation coefficient, maximum of cross-correlation function is a non-directed and its values are bounded to [-1,1]. The order of the input signals will only manifest in the maximum lag being positive or negative. Much like PPC, CCF is insensitive to nonlinear relationships. [Bastos and Schoffelen, 2015].

4.1.3 Mutual information

Mutual information (MI) is a non-directed measure based on information theory. It represents an amount of information about one random variable that can be obtained through the second random variable and unlike the correlation-based measures, it can detect even nonlinear dependencies. Mutual information of two continuous random variables X and Y is defined as:

$$MI_{X,Y} = \int_X \int_Y f_{(X,Y)} \log\left(\frac{f_{(X,Y)}}{f_{(X)} f_{(Y)}}\right) \tag{4.3}$$

where $f_{(X)}$ is a probability density function of X and $f_{(X,Y)}$ is a joint probability density function X and Y. When calculating mutual information for a pair of signals, the densities are unknown and mutual information has to be estimated in a different way.

Mutual information can be expressed in terms of entropies $H(X)$, $H(Y)$ and joint entropy $H(X,Y)$:

$$MI_{X,Y} = H(X) + H(Y) - H(X,Y) \tag{4.4}$$

To estimate the entropy $H(x)$ of a signal x, its values can be binned into k bins b_j. Probability $p(b_j)$ of a signal sample belonging to a bin b_j is then estimated as a number of values in the bin divided by the total number of values. Entropy is then calculated as:

$$H(x) = \sum_{j=1}^{k} p(b_j)\log(p(b_j)) \tag{4.5}$$

Joint entropy can be analogously estimated from a 2D histogram. This method of estimating mutual information is very sensitive to chosen number of bins k [Bastos and Schoffelen, 2015] [Grün and Rotter, 2010].

4.2 Common problems in connectivity analysis

The goal of the methods outlined in this chapter is to find connections between neuronal spike trains or between MERs - to identify the cases where the behaviour of neurons in one source influences the neurons in the other. However, several problems commonly appearing in the connectivity analysis can cause the methods to find such a functional connection even in the absence of one.

A common reference problem affects signals recorded by electrodes using the same reference channel. When multiple electrodes use the same reference, the changes of the potential on the reference electrode will manifest as simultaneous correlated changes in all measured signals. This problem can be alleviated by the use of bipolar recordings or by using a different reference for each electrode [Bastos and Schoffelen, 2015].

Another issue that can be encountered during the analysis of neuronal connectivity is the volume conduction problem [Stam et al., 2007]. Volume conduction is a term used to refer to an activity of unrelated sources being recorded by multiple electrodes. This is a problem that can affect MERs due to the low distance of the electrodes and it manifests as simultaneous or nearly simultaneous correlated changes in the recorded signals [Bastos and Schoffelen, 2015].

Two main approaches are used to reduce the effects of volume conduction. First, if the properties of volume conduction are known, the removal of its effects from measured signals can be attempted. However, this is almost never the case and the success of this method relies on the assumptions of the person in charge. Second way of dealing with volume conduction relies on the fact that it manifests in all of the affected recordings at the same time, while the real interactions cannot be instantaneous. This fact is utilised by some of the methods described in section 4.3 [Stam et al., 2007].

Neither the volume conduction problem, nor the common reference problem affect the spike train-based connectivity measures. In spike trains, all of the information is carried by the timing of the spikes. This nullifies the effect of these problems, since spikes can only be detected locally and small changes in the recorded signals caused by volume conduction and by common reference are unlikely to cause artificial spikes to be detected.

4.3 Coherency-based MER coupling measures

A cross-correlation is a measure of linear dependency of two time-domain signals $x(t)$ and $y(t)$ as a function of their relative shift τ. Analogously, one can obtain a measure of linear dependency of the signals as a function of frequency. This function is called coherency and is defined as:

$$C_{xy}(f) = \frac{S_{xy}(f)}{\sqrt{S_{xx}(f)S_{yy}(f)}} \tag{4.6}$$

where $S_{xy}(f)$ is a cross-spectrum of $x(t)$ and $y(t)$. Coherency of the signals is therefore their cross-spectrum normalised with respect to the spectral power of each signal. Values of coherency are complex numbers. Note the distinction between the term *coherency* and frequently used term *coherence*, which denotes the absolute value of coherency - a real-valued function of frequency. Cross-spectrum $S_{xy}(f)$ of the signals $x(t)$ and $y(t)$ is a function of frequency f and is defined as:

$$S_{xy}(f) = \mathbf{E}\left[X(f)Y(f)^*\right] \tag{4.7}$$

where $X(f)$ and $Y(f)$ are Fourier-transformed $x(t)$ and $y(t)$, $Y(f)^*$ signifies complex conjugate of $Y(f)$ and operator $\mathbf{E}$ denotes expectation value [Nolte et al., 2004].

To obtain a cross-spectrum estimate for the discrete signals $x(t)$ and $y(t)$, they are divided into n segments $x_i(t)$, $y_i(t)$ with a length of T_i seconds. The segments can overlap which results in more usable segments acquired from the signals. The segments then are multiplied by a window function and a discrete Fourier transform (DFT) is used to calculate their spectral images

$X_i(f)$ and $Y_i(f)$. Afterwards, a cross-spectrum of each pair is calculated and the resulting cross-spectra are averaged to obtain the result.

Since DFT is used, the resulting cross-spectrum is a vector $f_s T_i$ samples long and is symmetrical around the midpoint corresponding to a frequency $\frac{f_s}{2}$. Length of the segments T_i is an important parameter when computing coherency, since lower T_i leads to a cross-spectrum estimate less affected by noise but also to a smaller frequency resolution $\delta f = \frac{1}{T_i}$.

Absolute value or phase of coherency can be directly used as a measure of connectivity, but these values are sensitive to volume conduction problem. Measures described in the following sections use coherency in ways that negate the influence of volume conduction.

4.3.1 Imaginary part of coherency

A first coherency-derived measure discussed here is imaginary part of coherency. This measure was introduced in [Nolte et al., 2004] and it is argued there that it is insensitive to volume conduction. More specifically, a volume conduction or any other instantaneous changes in the signals cannot create a non-vanishing imaginary part of coherency, because imaginary part of coherency is only affected by the time-lagged changes.

It has been noted that discarding the real part of coherency can lead to some false rejections, especially when the time lag between the interacting sources is small [Vinck et al., 2011] [Nolte et al., 2004]. The properties of the imaginary part of coherency are also leveraged in the following measures.

4.3.2 Phase slope index

Phase slope index (PSI) is an estimate of an average phase slope in the coherency. PSI is based on the following assumptions:

- Real interactions between two sources are not instantaneous
- Speed at which interactions propagate is similar, therefore phase difference between sources increases with frequency

If these assumptions hold, then there should be a positive or negative slope in the phase of the coherency of the signals. Phase slope index quantifies the average slope as:

$$PSI_{xy} = \Im\left[\sum_{f\in F} C^*_{xy}(f)C_{xy}(f+\delta f)\right] \tag{4.8}$$

Where operator $\Im$ takes the imaginary part of a complex number. To see that PSI measures the slope of the phase spectra, it can be rewritten as

$$\begin{aligned} PSI_{xy} &= \sum_{f \in F} |C_{xy}(f)||C_{xy}(f+\delta f)| \sin(\theta_{xy}(f+\delta f) - \theta_{xy}(f)) \\ &\approx \sum_{f \in F} |C_{xy}(f)||C_{xy}(f+\delta f)|(\theta_{xy}(f+\delta f) - \theta_{xy}(f)) \end{aligned} \tag{4.9}$$

with $\theta_{xy}(f)$ representing the phase of the coherency at frequency f. It can be seen that if a consistent phase slope exists across the measured frequencies, the PSI value will be significantly nonzero. The sign of PSI may be used to deduce the direction of the interaction, but if one is only interested in assessing the presence of interaction, absolute value of PSI can be used [Nolte and Mueller, 2010] [Nolte et al., 2008].

4.3.3 Phase lag index

Phase lag index (PLI), proposed in [Stam et al., 2007], is based on similar assumptions as phase slope index. It is argued that a consistent lag between the phases of the signals is likely a result of interaction rather than a result of volume conduction or a random chance. PLI is therefore a measure of asymmetry in the phase difference distribution:

$$PLI_{xy} = |\mathbf{E}\left[\text{sign}(\Im\left[S_{xy}\right])\right]| \tag{4.10}$$

It can be seen that $0 \leq PLI_{xy} \leq 1$. Phase lag of non-interacting sources should be random and the value of PLI should approach 0 in this case. On the other hand, PLI value of 1 would signify perfect phase locking, hinting at strong interaction between the sources.

Since the PLI only considers the signs of the phase differences, it is sensitive to noise, especially in situations where the synchronisation effects are small in magnitude. Because of this, [Vinck et al., 2011] introduced weighed phase lag index (WPLI) as an improvement of the original phase lag index:

$$WPLI_{xy} = \frac{|\mathbf{E}\left[\Im\left[S_{xy}\right]\right]|}{\mathbf{E}\left[|\Im\left[S_{xy}\right]|\right]} = \frac{|\mathbf{E}\left[|\Im\left[S_{xy}\right]|\text{sign}\left(\Im\left[S_{xy}\right]\right)\right]|}{\mathbf{E}\left[|\Im\left[S_{xy}\right]|\right]} \tag{4.11}$$

In WPLI, the $\text{sign}(\Im\left[S_{xy}\right])$ is therefore weighted by $|\Im\left[S_{xy}\right]|$, reducing the effect of the imaginary parts of cross-spectra close to zero. Much like PLI, the WPLI can attain values in interval $[0, 1]$.

4.4 Estimating coupling significance

In this work, various measures were used to assess the coupling between two sources. Since the distributions of these measures are often unknown or hard to determine, parametric tests of significance are usually not usable. Therefore, to determine whether the values of the measures are significantly different from the values arising from random properties of the signals, a Monte Carlo test of significance was employed.

The goal of Monte Carlo test of significance, as used here, is to estimate the p-value of a value r of a particular connectivity measurement method applied to a pair of signals $x(t)$ and $y(t)$. To this end, m pairs of artificial signals $x_i(t)$ and $y_i(t)$ are generated, the measure r_i is calculated for each pair of the artificial signals and the estimated p-value of r under the null hypothesis of no connectivity is obtained as:

$$p_r = \frac{(\text{number of } r_i \text{ greater than or equal to } r) + 1}{m+1} \tag{4.12}$$

The artificial signals $x_i(t)$ and $y_i(t)$ for this test were generated using the original signals $x(t)$, $y(t)$ in the following way:

1. Frequency spectra $X(f)$ and $Y(f)$ are obtained from the signals using Fourier transform
2. m pairs of artificial spectra $X_i(f)$, $Y_i(f)$, $i = 1, \ldots, m$ are created using these rules:
   - $|X_i(f)| = |X(f)|$, $|Y_i(f)| = |Y(f)|$
   - $\angle X_i(f)$ and $\angle Y_i(f)$ are drawn from uniform random distribution on $\pm\pi$
3. m pairs of artificial signals $x_i(t)$ and $y_i(t)$ are generated from the spectra $X_i(f)$ and $Y_i(f)$ through the inverse Fourier transform

Artificial signals generated in this way reproduce the spectral power and autocorrelation of the original ones, but their temporal structure of the phase difference is randomized [Simpson et al., 2001] [West et al., 2016].

4.5 Spike train coupling measures

In this section I will discuss generalized linear models and their application in measuring a functional connectivity between a pair of neuronal spike trains. Describing spike trains and their modelling requires specialized mathematical methods and notation which will be introduced first.

4.5.1 Point processes modelling

Spike trains can be mathematically described as point processes. A point process is a set of discrete events occurring in continuous time. Times of the n events or spikes s_i, $i = 1 \dots n$, distributed on a time interval $(0, T]$ fully describe a point process. Additionally, a counting process $N(t)$ is defined as a number of events that have occurred up to and including the time t.

A point process can be completely described by its conditional intensity function $\lambda(t|H(t))$:

$$\lambda(t|H(t)) = \lim_{\Delta t \to 0} = \frac{P\left[(N(t+\Delta) - N(t)) = 1|H(t)\right]}{\Delta t} \tag{4.13}$$

where $P[.|.]$ denotes conditional probability and $H(t)$ denotes history of the point process up to a time t. For a small Δt, the equation 4.13 can be re-expressed as:

$$P\left[(N(t+\Delta) - N(t)) = 1|H(t)\right] = \lambda(t|H(t))\Delta t \tag{4.14}$$

In other words, the probability of a point process generating a spike in time interval $(t, t+\Delta]$ is equal to the product of its conditional intensity and of the length of the interval [Eden, 2011] [Dayan and Abbott, 2001] .

When a model is fitted to some data, it is crucial to test its goodness-of-fit before making further inferences from it. In the case of point processes, the time rescaling theorem provides a natural approach for goodness-of-fit testing. The time rescaling theorem states that if the spike times s_i were generated by a point process with conditional intensity function $\lambda(t|H(t))$, they can be transformed to new times τ_i with an exponential distribution with a unit rate. This transformation is described by the following equation:

$$\tau_i = \int_{s_{i-1}}^{s_i} \lambda(t|H(t))dt \tag{4.15}$$

It is then useful to perform a second transformation:

$$z_i = 1 - e^{-\tau_i} \tag{4.16}$$

This way, the variables z_i are uniformly distributed on $[0, 1]$. Therefore, to test a goodness-of-fit of a model with a conditional intensity function $\lambda(t|H(t))$ to spike times s_i, rescaled times z_i are calculated and their distribution is compared to theoretical uniform distribution using Kologorov-Smirnov (KS) test [Haslinger et al., 2010] [Truccolo et al., 2005].

To analyse point processes, it is often useful to discretize their time representation. To do this, the interval $(0, T]$. is divided into K bins t_k, $k = 1, \dots, K$, with equal size of $\Delta t = \frac{T}{K}$. The length of each bin Δt should be small enough so each bin can realistically contain only a single spike. The quantity $\Delta N_k = N(t_k) - N(t_{k-1})$ then describes the number of events in one bin. For discretized point processes, the equation 4.14 can be simplified further [Eden, 2011] [Truccolo et al., 2005]:

$$P\left[\Delta N_k = 1 | H(t_k)\right] = \lambda(t_k | H(t_k))\Delta t \tag{4.17}$$

In the case of discretized point processes, the equation 4.15 can be simply replaced by

$$\tau_i = \sum_{k=k_{i-1}}^{k_i} \lambda(t_k | H(t_k)) \tag{4.18}$$

However, the process of discretization can introduce significant biases to KS test. The causes of these biases and possible way to remove them are described in detail in [Haslinger et al., 2010]

4.5.2 Generalized linear models

Generalized linear models (GLM) is a class of models that generalizes the ordinary linear regression by allowing the predicted variable to have distribution other than normal. To do this, the predicted variable is related to the regressors via a nonlinear link function.

In the case of spike trains, the predicted variable is the conditional intensity function of the underlying point process, which is poisson distributed. The link function in this case is therefore a logarithm.

To model a spike train x with times s_i^x, which is functionally connected to another spike train y with times s_i^y, a following general form of the conditional intensity function can be used [Truccolo et al., 2005]:

$$\lambda(t | H(t), \theta) = \exp\left(\alpha_0 + \sum_i \alpha_i \Theta_i + \sum_j \beta_j \Phi_j + \sum_k \gamma_k \Psi_k\right) \tag{4.19}$$

where Θ_i, Φ_j and Ψ_k are the regressors and α_i, β_j and γ_k are their respective coefficients.

The meaning of coefficients and regressors in equation 4.19 is as follows:

- α_0 represents the background firing intensity
- α_i and Θ_i represent the variability of firing intensity with time
- β_j and Φ_j represent the dependency of the intensity on refractory effects
- γ_k and Ψ_k represent the dependency of the intensity on a spike train y

To fit a GLM to spike trains x and y, the regressors have to be calculated using the spike trains and then GLM can be fitted using maximum likelihood optimisation. After a model is obtained, its goodness-of-fit has to be obtained by the method outlined in section 4.5.1. Then, to find out whether a significant functional connection between x and y exists, the submodel of coefficients γ is tested against the full model.

Chapter 5

Results

In this chapter, I present the original work done in the course of this thesis and its results. This work was carried out to accomplish some of the objectives of this thesis, namely:

1. Design and implementation of the software tools for identification and visualization of couplings between MER-based signals

2. Preprocessing MER data measured on real patients and using the implemented tools to analyse this data

3. Result interpretation and discussion

I will review the software design and implementation in section 5.1 and I will describe the work performed on real data in section 5.2. The obtained results are visualized, analysed and discussed results in section 5.3.

5.1 Implemented software tools

In this section, I will describe the overall design of the implemented software tools. A structured documentation can be found in appendix A.

First, I will provide a high level overview of the architecture and then I will describe some of the implementation details for the individual components or introduce the third party tools used instead.

Most of the software in this thesis was written in the Matlab programming language [The MathWorks, 2017] with additional analyses being written in the Python programming language using Jupyter Notebook [Jupyter, 2017] environment. The software is dependent on used Matlab toolboxes (Signal Processing Toolbox and Statistics and Machine Learning Toolbox) but is otherwise meant to be standalone. Nevertheless, the developed original software is based on the software tools developed by the Computational

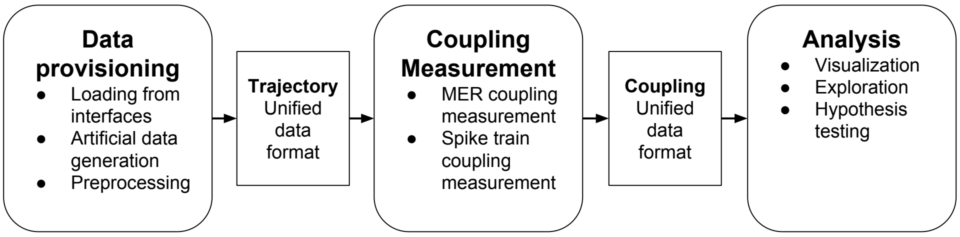


Figure 5.1: Schematic visualization of the components and dataflow

Neuroscience research group on FEE CTU and it can easily interface with these tools. To avoid namespacing conflicts with other Matlab code, names of all of the original functions are appropriately prefixed.

To organise the software necessary for accomplishing the thesis objectives, I grouped it into three distinct components.: data provisioning, coupling measurement, and analysis. Each of these components represents a set of functions and scripts that accomplish a single logical step in the overall workflow. This is schematically depicted on the fig. 5.1.

The components exchange their inputs and outputs with the other components, but are otherwise independent of each other. Therefore, it does not matter which exact functions and scripts are called in each component, as long as the outputs are in the specified format that is usable by the next component. This allows user to quickly swap or add functions to alter his workflow without worrying about compatibility. This should aid the users with rapid experimentation and might allow easier adaption for workflows beyond this thesis.

To avoid constant rewriting of commonly used data operations and ensure proper handling of the data, the components typically consist of functions with higher and lower level of abstraction. The higher level functions are meant to be called by the users and in turn call the lower level ones correctly and are responsible for recording the operations done on the data and managing the data formats. Lower level functions usually represent some atomic operations on a single signal or on a pair of signals and are expected to be called through the higher level functions, although they are still available to the users for a direct usage. Some of the lower level functions are only expected to be called by other functions and were made private using Matlab private functions mechanism.

To choose the component output data formats, several considerations were made. First, the high level output structures were chosen to be analogous

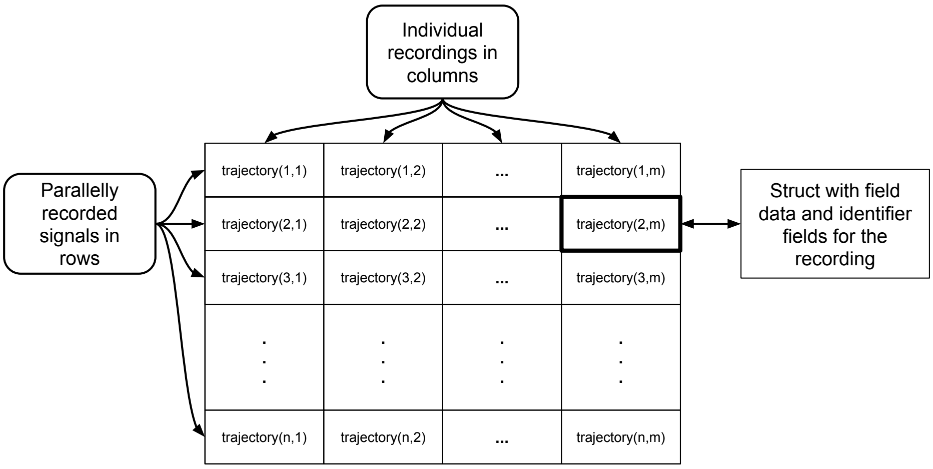


Figure 5.2: Schematic visualization of the trajectory data structure

to how the actual recordings are made and to be somewhat compatible with used data interfaces. Secondly, the analysis of the results is exploratory in nature, therefore each data point should contain as much information about itself as possible. This allows the users to pick and choose the information relevant to him or her and quickly factorize the result set according to his or her needs. On the other hand, if the user knows precisely what part of the information he wants even before the experiments, the resulting data format might be unnecessarily verbose.

The result of data provisioning - `trajectory` - should always be two dimensional Matlab structure array, where each row represents data that were recorded parallelly. In each row, every column contains data originating from a single electrode on a single position and its descriptors. Each structure must contain field `data` with the actual MER or extracted spiketrains and can optionally contain any number of other fields that identify this data. This structure is schematically depicted on fig. 5.2.

The result of coupling measurement - `coupling` - is one dimensional Matlab structure array where each row contains information about one pair of sources that could be functionally connected. Each row contains all the data descriptors of both of the sources and additionally, fields with all of the information about used coupling measurement method and the results of the coupling measurement - value of the measure, significance of the value and other information obtained during the computation. This structure is schematically depicted on fig. 5.3.

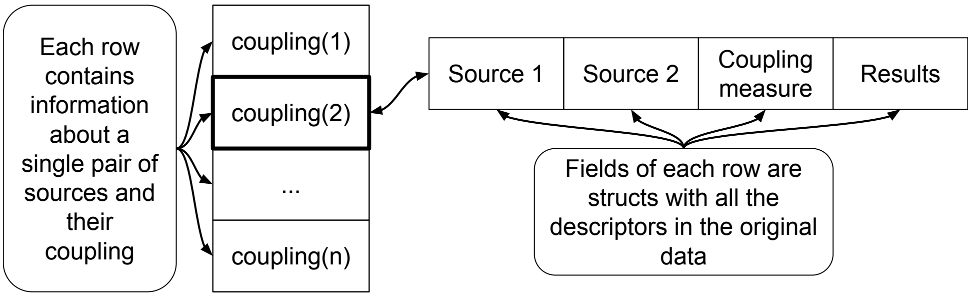

Each row contains information about a single pair of sources and their coupling
coupling(1)
coupling(2)
...
coupling(n)
Source 1
Source 2
Coupling measure
Results
Fields of each row are structs with all the descriptors in the original data

Figure 5.3: Schematic visualization of the coupling data structure

5.1.1 Data provisioning

In data provisioning, the tasks needed for this thesis were artifact detection and removal, spike detection and sorting, frequency filtering and other preprocessing.

To detect and annotate artifacts, Siginspect [Bakštein, 2015], a tool developed in Computational Neuroscience research group was used. This tool allow its users to iterate through the data and assign artifact annotation to 1 second long segments of each MER, based on the displayed signal time series and spectrograms. It also allows its users to generate automatic annotation based on machine learning methods.

Waveclus [Quiroga et al., 2004] was used for automatic spike detection and for semi-supervised spike sorting. This tool uses amplitude thresholding and superparamagnetic clustering and is described in more detail in section 3.3.1.

Another Computational Neuroscience research group tool, Data Access Objects (DAO) [Sieger, 2013] was used to load the cleaned MERs or extracted spiketrains together with the headers containing the information about the data.

To obtain a usable clean MER a simple removal of all segments containing artifacts would not be sufficient. Since the subsequent analysis requires contiguous segments of the same length that were recorded simultaneously, a capability of DAO to obtain maximal shared contigous artifact free segments in each position was used. This artifact removal scheme satisfies the following rules:

1. All returned segments must be artifact free
2. All returned signals on all channels must be contiguous
3. Returned signals from all channels in a single position must be of the same length

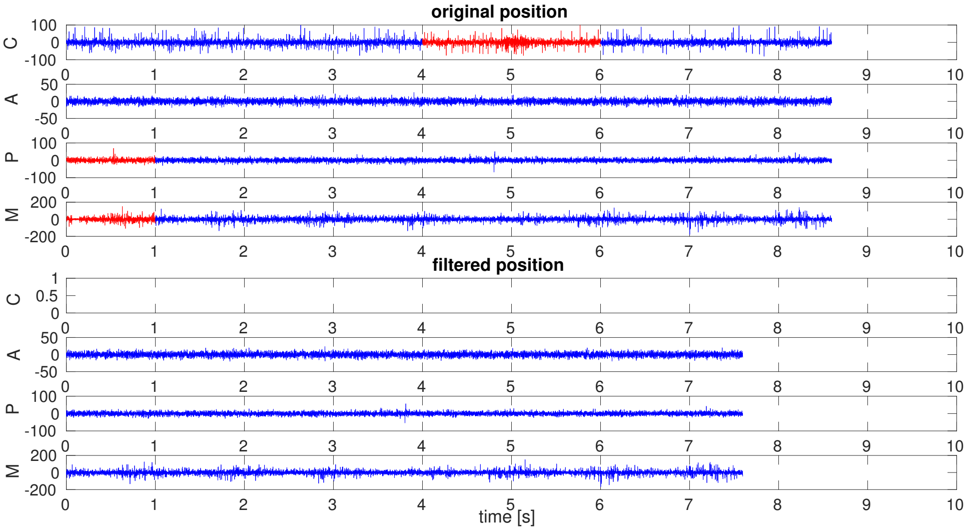


Figure 5.4: An example of artifact removal. Top half represents the original signals with artifacted segments denoted with red color. Bottom half represents clean signals without the first segment and without the signal from the central electrode. Adding the central electrode would result in reducing the total number of available segments, it is therefore discarded.

4. Returned signals from all channels in a single position must come from the same segments, i.e. they must be parallel

5. Returned segment selection must maximise the total amount of segments returned across all channels

6. If discarding one whole channel increases the total number of segments returned, it must be done

An example of such artifact removal can be found in Fig. 5.4

I have implemented additional preprocessing methods that can be applied to the loaded data in order to remove other problems with the data. These methods were removal of missing data samples, removal of short signals, various frequency filters and spike train binning and they were implemented as low level functions accessible through the high level function `couplingPreprocess`. This function will iterate through the data given, apply the user selected preprocessing methods in sequence and write the names of the used preprocessing methods into the data descriptors. The usage of `couplingPreprocess` and its associated lower level function is described in more detail in appendix A.1.2, while the remainder of this section discusses the motivation for the implemented preprocessing methods.

Missing data removal was necessary because missing data samples were not annotated during the artifact detection. This is because unlike artifact detection, every sample with missing data can be easily and correctly identified. Therefore, removing a whole second of data because of a few missing samples might waste a large amount of otherwise usable data.

Since parallelly recorded and contiguous signals are necessary for further computations, the longest segment without any missing samples in any of the signals was selected and the rest was discarded. Typically, the signals contain no missing samples or only a few missing samples near their beginnings and ends and this operation does not waste much data. In some cases, isolated missing samples in the middle of the signals could reduce the amount of available data dramatically. If this is frequent, a user can relax the requirements and allow non-contiguous segments of data.

After the removal of artifacts and missing data, some of the signals could be left very short, unusable for further analyses. The usability of the short signals is mostly constrained by the coherency-based coupling measures, which require enough segments of sufficient length to obtain adequately denoised coherency estimates with the necessary frequency resolution. In order to not introduce any false positives based on these short signals and to save computation time, such signals can simply be removed by the appropriate preprocessing method.

Following a data quality inspection which will be discussed in more detail in section 5.2.1, some filterable frequency artifacts were discovered in many of the signals. To reduce the effect of these artifacts on the analyses several predefined frequency filters were using Matlab tools. The most important filter implemented was a successive notch filter for removal of 50 Hz power line interference and its harmonics. The filters used are typically IIR with nonlinear phase characteristic and simple filtering would result in phase distortion which is unacceptable for the purposes of coupling measurement. In order to avoid this, zero phase digital filtering with Matlab function `filtfilt` was used.

A comb filter for 50Hz noise and its harmonics was also used at first, however its usage together with `filtfilt` resulted in noticeable amplitude distortion in time series of the signals. This could be a result of a combination of several factors (high ratio of filtered frequency and sampling frequency, zero phase filtering) necessitating an extremely high effective filter order. Since the components of the signals that are of biological origin are expected to be located on much lower frequencies than the sampling frequency used in the available data, techniques such as decimation could be used to obtain more reasonable filters. However, successive notch filtering on several frequencies turned to be sufficient in this thesis.

When spikes are extracted using WaveClus, the resulting spike trains are encoded using the times of the spikes. As described in the section 4.5.1, spike train analysis in this work uses binned discrete representation of spike trains. Therefore a function was implemented to transform spike time representation into binned representation. Default bin width used was 1ms. This was the only additional preprocessing applied to spike trains.

5.1.2 Coupling measurement

This section discusses implementation details of coupling measurement methods. The list and theoretical background of these methods can be found in chapter 4. The goal of each of these methods is to find whether a coupling exists between two sources. These sources can be represented by MERs or by spike trains.

Individual coupling measures are implemented as low level functions accepting a pair of signals as inputs. The low level functions can also accept optional parameters and in the following text, the term coupling measure or connectivity measure is used to refer to a single combination of a low level function and its parameters. The specification of a coupling measure passed in Matlab environment is a struct with fields specifying at least the function to call and the parameters to call it with. To save time during experiments, it became useful to define a unique name for every coupling measure and the higher-level functions therefore also support specification of the measures by their unique names.

On a higher level, two functions `couplingComputeMER` and `couplingComputeST` were implemented. These functions accept a trajectory, a specification of coupling measures to be used and other optional arguments and return the measure values and their significance for all of the possible coupled pairs in the trajectory. In practice, this means iterating through the trajectory and calling the appropriate lower level functions on every pair of parallel MERs or spike trains.

In case of MERs, this means computing the coupling between all of the combinations of non-empty electrodes within a single position for all positions given. `couplingComputeMER` accepts a trajectory and list of MER coupling measures and for every pair of electrode data, the values of the specified coupling measures are calculated. Next, a significance of these values is estimated by the Monte Carlo method described in the section 4.4.

Generating the surrogate signals for every combination of MERs in a position for every coupling measure would be very time consuming. Therefore, each MER is used to generate the desired number of artificial signals and these signals are then reused as needed when estimating significance of multiple coupling measures for different combinations of MERs. This reduces the time complexity and it also might reduce random disparities between the results of different measures.

In case of spike trains, `couplingComputeST` has to compute the coupling between all of the combinations of single neuron spike trains from all of the combinations of electrodes within a single position. A coupling between two spike trains recorded by a single electrode could also be sought, but such analyses are beyond the scope of this thesis. To avoid overfitting the models which usually had 10-15 regressors, only the spike trains with sufficient number of spikes were considered. The minimum number of spikes was chosen as 50 to avoid discarding a high number of (especially non-STN) recordings.

`couplingComputeST` accepts a trajectory and a specification of GLM models to be used. For every pair of spike trains, the prescribed GLMs are fitted and goodness-of-fit test is applied. For a model to pass, the p-value of the used KS test had to be higher than 0.1. The models which did not pass were discarded and were not used to infer connectivity between the two recordings. Typically, about 15% of the fitted models did not pass this test.

If the goodness-of-fit test passes, then the significance of the model regressors describing inter spike train interaction in the whole model is evaluated using Matlab function `coefTest`. Finally, a model of coupling of two neurons from two electrodes which is the most significant is taken as a coupling model of these two electrodes - the function is searching for the recordings where *any* contained neurons are coupled.

On a lower level, each coupling measure described in chapter 4 was implemented as a single function that accepts a pair of MERs or a pair of spike trains and returns the value of the measure. The list of these functions is in the table 5.1. These functions can also accept any number optional arguments (e.g. segment length for coherency based measures) as name-value pairs and may return some of additional information gained during the computation other than the value of the measure itself (e.g. lag of maximal cross-correlation).

Pearson's correlation coefficient and cross-correlation are already implemented in Matlab as standard functions `corr` and `xcorr`. The corresponding implemented functions therefore only call the standard functions with the right input data formats and add various optional arguments. Both of these functions can take optional boolean argument `abs`, which specifies whether

Connectivity measure	Theory	Matlab function
Correlation coefficient	4.1.1	`couplingPearsonCorrelation.m`
Cross-correlation	4.1.2	`couplingCrosscorrelation.m`
Mutual information	4.1.3	`couplingMutualInformation.m`
Imaginary part of coherency	4.3.1	`couplingImaginaryCoherency.m`
Phase slope index	4.3.2	`couplingPhaseSlopeIndex.m`
Phase lag index	4.3.3	`couplingPhaseLagIndex.m`
Weighted phase lag index	4.3.3	`couplingWeightedPhaseLagIndex.m`

Table 5.1: Implemented MER connectivity measures

an absolute value of the correlation coefficient/function should be used. This can be useful because all non-zero correlation values (even negative) might imply some connectivity. `couplingCrossCorrelation` also takes arguments specifying the minimum and maximum time lag to be considered when searching for a maximum of cross-correlation function and allows a specification of the used normalization method. Note that Matlab standard function `xcorr` computes cross-correlation using the following formula:

$$CCF_{x,y}(\tau) = \frac{\sum_{i=0}^{n-\tau-1} x(i+\tau)y(i)}{n} \tag{5.1}$$

it therefore scales the result by the number of samples n, not by standard deviations as specified in 4.2. This means that the results can attain values even outside the interval $[-1, 1]$.

`couplingMutualInformation` implements the mutual information estimatimation scheme described in section 4.1.3: the values of the input signals were binned into histograms with k bins and into mutual 2D histogram with $k \times k$ bins. These histograms were normalized and used to estimate the entropies and the mutual entropy of the signals. Afterwards, mutual information was calculated using equation 4.4. Because the choice of k greatly influences the results, it can be chosen manually or adaptively according to the attributes of the signals. I have implemented a Scott's rule, which is given by a following formula:

$$k_x = \left\lceil \frac{\max(x) - \min(x)}{3.5 s_x n^{-\frac{1}{3}}} \right\rceil \tag{5.2}$$

where s_x is a sample standard deviation of x. This rule is meant to take the variability of the signals into account when choosing the number of bins.

Note that only a moderate k can be chosen before the number of bins of the 2D histogram k^2 becomes too large in comparison to the number of samples of the signals. This can limit the usability of any rule for choosing a number of bins when estimating entropy of a signal.

For the coherency based methods, `coherency` and `crossSpectrum`, private functions for estimating these quantities were implemented. The arguments for these functions are the two signals and a length of the signal segments to be used in the estimation. The process of estimation is described in section 4.3. When the signals were divided to segments during the cross-spectrum estimation, 50% overlap was used. Any function calling these functions is responsible for deciding the right segment length for the desired frequency resolution and noise removal. For instance, if a desired frequency resolution is 1Hz, the segments were chosen to be 1s long and their number is then derived from the length of the signals and segment overlap used in coherency estimation.

Every function implementing a coherency-based coupling measure first obtains an estimate of coherency or an estimate of cross-spectrum and then performs the necessary transformations described in 4.3. Sampling frequency, length of segments for coherency estimation and maximal used frequency can be specified as optional arguments. The maximal used frequency limits the parts of cross-spectra or coherencies which are used in the calculaction of the coupling measures. This could be important since it can be argued that any signal components of biological origin should lie on relatively low frequencies. In addition to the other arguments, `couplingImaginaryCoherence` allows the specification of any aggregation function, which will create a single number from the imaginary coherency vector, with default being the maximum.

`couplingGLM` accepts a pair of spike trains and allows the user to specify the regressors, which will be used to fit the model, in predefined ways. One of the spike trains is used as the predicted variable and to calculate regressors the other spike train is only used to calculate regressors. The regressors created using the spike train are used to model refractory effects and coupling and additional b-spline regressors are used to model time varying firing rate.

Each regressor is a vector with the same length as the length of the two used spike trains and its elements correspond to either a value of a b-spline on the given position or to a number of spikes fired in one the spike trains in a number of past bins.

The b-splines are fixed as 4 equally spaced b-splines of order 4 and an example of such splines spaced over 4000 bins can be seen in Fig. 5.5. Note that the "first" spline is missing, because it can be obtained as a linear combination of the other splines and including it in regressors would cause the fitting process to fail.

The regressors calculated from spike trains used in bin i are sums of spikes in bins $i - a$ to $i - b$, $a > b$, of one of the spike trains. The numbers b and

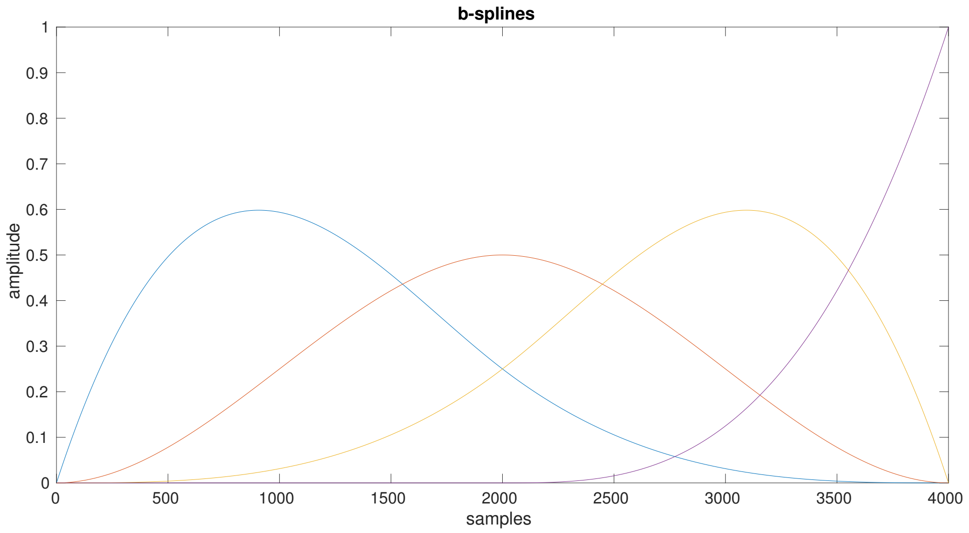


Figure 5.5: B-splines used as regressors

$a - b$ represent an offset from the current position and a length of the interval to be used respectively and fully characterize the regressor. These regressors can be specified by the user using `couplingGLM` arguments. The primary way to do this is to supply a vector of interval boundaries for both of the spike trains. For example, vector `[0,5,15,50]` specifies 3 regressors:

1. number of spikes in past 1-5 bins
2. number of spikes in past 6-15 bins
3. number of spikes in past 16-50 bins

To obtain the results, the selected regressors are calculated using the spike trains, a GLM is fitted using Matlab function `fitglm` and goodness-of-fit test is done using time rescaling theorem, described in section 4.5.1. The spike times are transformed using the model's conditional intensity function and they are compared to uniform distribution using Matlab function `kstest`.

5.1.3 Analysis and visualization tools

The results of coupling measurement is a large set of information about pairs of MERs or spike trains and their coupling which needs to be efficiently explored. Each data point in this set includes the value of the used measure and its significance, as well as identification of trajectory, position, depth, brain area and other information. The analysis of the results is exploratory in nature, but the dependency of measures on the depth along the trajectories and on areas of the brain, where the recordings were made, is a desired

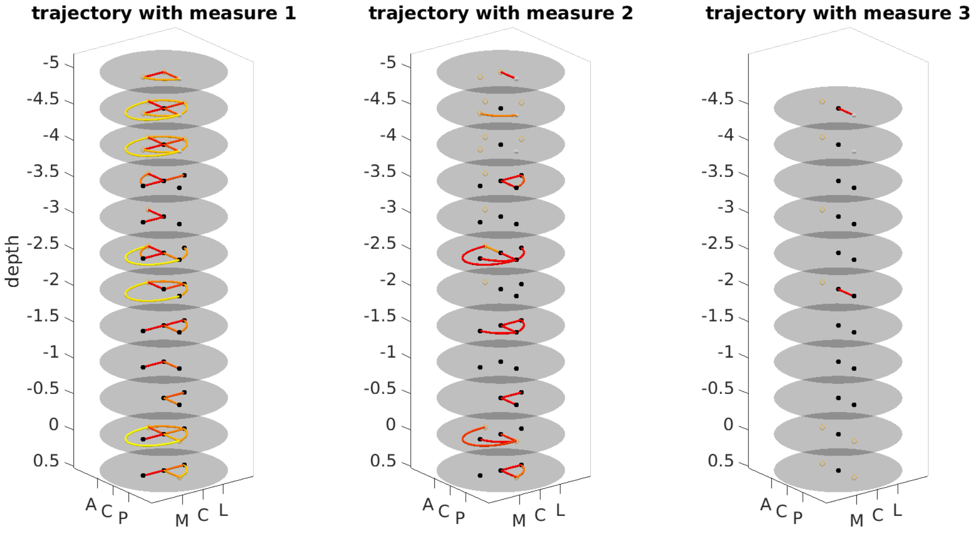


Figure 5.6: An example of the couplingVisualize output. A single trajectory with 3 different measures is shown with depths along the z-axis and individual electrodes in a single depth being represented by dots placed along x and y axes. The color of the dots represents area of the brain (black=STN, gray=other structure, no color=uknown), lines between the dots represent a significant coupling between the electrodes, color of the lines encodes the value of the measure

result. Therefore, the analysis and visualization tools were focused on these relationships.

To visualize the relationships along each trajectory, the function `couplingVisualize` was implemented. Input for this function is one or more struct arrays, each containing the results for a single measure and trajectory combination. When used, `couplingVisualize` plots an interactive 3D representation of the relationships along each trajectory. An example is shown in Fig. 5.6.

Positions along each trajectory are plotted on the z-axis and in each position, the electrodes which recorded the parallel signals are represented by points. Lines connecting a pair of electrodes denote a relationship between the two electrodes as assessed by the coupling measures. The user can choose the meaning of the lines and colors in the graph through optional arguments:

- Each electrode can be colored according to its area or its depth
- Each line can be colored according to the value or the significance of the measure
- Each line can be either displayed or hidden according to the value or the significance of the measure

Additionally, all the displayed trajectories can be rotated and zoomed and panned along the z-axis together. Lastly, the interface buttons can be hidden to allow exporting the graph when the user is satisfied with its appearance.

In order to not limit the user to analysis using Matlab, I have implemented function `couplingExportResutls`, which flattens the coupling struct array - converts structs in each row to new prefixed fields - and exports it as .csv file. This file contains every piece of information gathered for every row and allows the users to use their favorite analysis tools to inspect the data.

I have used Python programming language with Pandas and Seaborn libraries in Jupyter notebook environment to explore and analyse the exported data. The resulting notebook `couplingExploration.ipynb` contains all of the exploratory analysis done in this thesis, together with the code used to generate it and comments about the obtained results. The notebook was also exported to `couplingExploration.html` for to enable viewing outside of the Jupyter environment.

A possible workflow using all the components described so far might look like the example in listing 5.1, more detailed specification can be found in appendix A.

Listing 5.1: A complete workflow example

```
%% data provisioning
%load MERs from an interface and apply preprocessing
mer = couplingLoadTrajectoryDao(10, 2);
mer = couplingPreprocess(mer, {'removenan', 'removeshort'});

%load spike trains from an interface and apply preprocessing
st = couplingLoadTrajectoryDao(10, 2, 'dataType', 'spikes');
st = couplingPreprocess(st, {'binspikes'});

%% coupling measurement
% compute MER coupling using specified measures
mer_coupling = couplingComputeMER(mer, {'cc_max', 'pli'});

% compute spike train coupling using specified models
st_coupling = couplingComputeST(st, {'glm_1'});

%% visualization and analysis
% visualize mer coupling
couplingVisualize(mer_coupling)

% export spike train coupling as st.csv
couplingExportResults(st, 'st')
```

5.2 Real data connectivity analysis

Using the methods and tools described so far, I have processed and analysed a real data set of trajectories recorded during DBS surgeries on 12 different patients p1-p12. All of these patients were implanted in both left (sin) and right (dex) STNs, resulting in 24 total trajectories to work with. These trajectories contained 1253 positions and 5012 records in total and a quantitative summary for the individual trajectories can be found in the table 5.2. Every record in these trajectories has descriptors with information about its recording electrode, position, depth and area - a probable brain structure the record was made in.
The information about area was aggregated to three categories:

- *STN* - record made in subthalamic nucleus, should contain individual firing neurons
- *Other* - record made in nuclei and structures other than STN (e.g. thalamus), should contain individual firing neurons, but possibly with different characteristics than those in STN
- *Unknown* - record made outside any distinguishable brain area, with electrode surrounded mostly by white matter, should contain less or no distinguishable firing neurons

Unknown and STN areas were by far the most numerous in the recordings and are studied the most in this thesis.

Every record in the data set was sampled with a sampling frequency of 24000Hz for 10 seconds. Some of the signals contain missing samples and artifacts. Every recorded position contains 4 parallel MERs. Two positions were recorded in each depth in each trajectory, one with lateral electrode active, one with medial electrode active. Every position also contains recording from the central, anterior and posterior electrodes.

As stated before, a statistically significant value of a coupling measure is taken as a proof of functional connectivity between a pair of MERs or spike trains. Before the connectivity could be explored and analysed, the data had to be preprocessed and the used coupling measures were selected and calculated.

5.2.1 Preprocessing

The available raw MER recordings contained a significant amount of artifacts and artifact detection and removal was therefore employed first. I have used

Patient	Side	Positions	Records	STN	unknown	other
p1	dex	48	192	47	133	12
p1	sin	48	192	54	130	8
p2	dex	59	236	51	172	13
p2	sin	50	200	34	152	14
p3	dex	48	192	53	117	22
p3	sin	48	192	71	106	15
p4	dex	50	200	60	128	12
p4	sin	52	208	58	141	9
p5	dex	54	216	56	127	33
p5	sin	60	240	69	143	28
p6	dex	54	216	39	142	35
p6	sin	51	204	68	123	13
p7	dex	46	184	54	103	27
p7	sin	54	216	66	117	33
p8	dex	47	188	59	117	12
p8	sin	52	208	52	150	6
p9	dex	48	192	54	126	12
p9	sin	50	200	68	116	16
p10	dex	58	232	72	150	10
p10	sin	55	220	58	151	11
p11	dex	59	236	48	138	50
p11	sin	57	228	80	102	46
p12	dex	51	204	51	148	5
p12	sin	54	216	57	147	12

Table 5.2: Summary of the trajectories

the capabilities of Siginspect to generate automatic annotation for all of the trajectories and manually inspected all of the signals to verify the results and correct possible false positives and false negatives. To be able to correctly identify these cases, I was trained in artifact detection by people who were already experienced in MER processing beforehand.

The result of the annotation process was an assignment of each segment to either clean or artifacted class. However, some of the segments were manually annotated as `unsure`. This was done for a set of segments that were considered borderline in terms of their quality. The reason for this is that the total number of these segments was unknown at the beginning of the annotation process. Therefore, it could not be determined if such segments were numerous enough to skew the results in case their effect on the results was significant or unnecessarily reduce the amount of available data in case their effect was not significant.

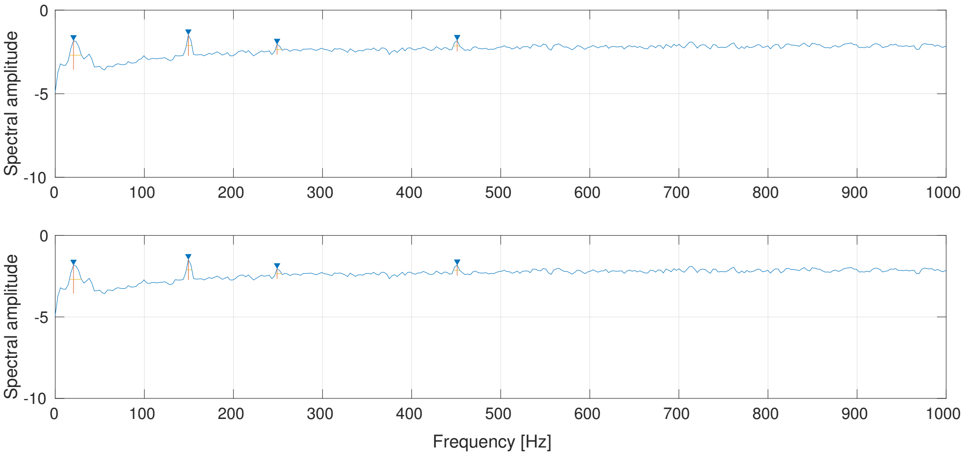


Figure 5.7: Amplitude spectra of two parallel signals showing peaks on frequency 50Hz and its harmonics

Because of this, it had to be decided whether to consider `unsure` segments as clean or not before removing the artifacts. To assess the effect of `unsure` annotation, the cleaned signals for both of the options were compared. This comparison revealed that the number of segments designated as `unsure` was relatively small and due to the used artifact removal method (see 5.1.1), the resulting cleaned signals were often the same for both of the configurations. Therefore, I subsequently considered the `unsure` segments as artifact free. Nevertheless, the `unsure` annotations remain as an additional precision in the annotations and could be used in some other analyses in the future.

After the artifact removal, the cleaned trajectories contained 3681 non-empty positions. Each position containded variable number of parallel MERs (0-4) that were all of the same length (1-10 seconds). The cleaned trajectories are quantitatively summarised in the table 5.3.

In each position, recording from any electrode can be coupled with recording from any other electrode. Therefore, if n electrodes in a position contain nonempty signals, then $\frac{n(n-1)}{2}$ possible couplings can be investigated. The total number of possible couplings is 4046 for MERs and 2737 for spike trains.

Further steps were different for the spiketrain data and for MERs. For MERs, the missing data was removed and any signals that were left with less than 2 seconds of data were discarded. The minimum length of the signal was chosen in this way to ensure that coherency estimates with 1Hz frequency resolution and 50% segment overlap were not calculated using 3 or less segments.

Afterwards, the quality of the remaining signals was assessed. Inspection of the power spectral densities of individual signals and inspection of coherency

Patient	Side	Clean records	STN	unknown	other
p1	dex	100	41	48	11
p1	sin	104	38	58	8
p2	dex	144	45	91	8
p2	sin	149	30	108	11
p3	dex	181	49	112	20
p3	sin	152	56	85	11
p4	dex	164	52	100	12
p4	sin	167	45	115	7
p5	dex	173	50	93	30
p5	sin	166	63	78	25
p6	dex	148	32	89	27
p6	sin	160	57	93	10
p7	dex	157	51	82	24
p7	sin	175	60	88	27
p8	dex	150	51	91	8
p8	sin	168	47	116	5
p9	dex	144	47	85	12
p9	sin	154	59	83	12
p10	dex	206	65	131	10
p10	sin	167	54	105	8
p11	dex	185	43	97	45
p11	sin	155	74	46	35
p12	dex	120	41	79	0
p12	sin	92	44	38	10

Table 5.3: Summary of the trajectories after the artifact removal

of pair of signals showed clear peaks at frequency 50 Hz and at its harmonic frequencies in numerous signals. Amplitude spectra of two parallel signals showing such peaks are shown in Fig. 5.7

This is likely an effect of electromagnetic noise from electronic appliances powered by 50 Hz alternating current and not a physiological effect. Therefore, these signals were suppressed using successive IIR notch filters with notches at several multiples of 50Hz and zero-phase filtering.

For spike train analysis the spikes were extracted from the artifact-free MERs and then clustered using WaveClus. Automatic extraction and clustering was employed, but the results were manually checked for obvious errors. The errors were usually a very high number of clusters (>8) found during the spike sorting with very low evidence pointing to number of cluster this high. There were 91 records where obvious spike sorting errors were found. In these cases, I have performed the spike sorting manually.

5.2.2 Coupling measurement

After the data was preprocessed, I have chosen the coupling measures to use on the whole data set and these measures are summarized in the table 5.4. Every implemented method was used with several combinations of parameters. This section describes the used parameters and the reasoning behind them.

Phase slope index, phase lag index and weighed phase lag index require a fine frequency resolution to function properly. At the same time, any noise in coherency estimates should be random and should not create consistent phase lag or phase slope across all frequencies and consequently can be largely ignored. Therefore, a segment length of 1 second and corresponding frequency resolution of 1Hz was chosen for these measures. On the other hand, maximum of imaginary part of coherency is very sensitive to noise in coherency estimates. To reduce this noise, smaller segment length for coherency calculation can be used, which leads to more segments and the noise being averaged out more. Therefore, a segment length of 0.25s was used for this measure.

Another parameter which was used with coherency based measures was maximum frequency of coherency to be used in the calculation of the measures. Since the components of the signals arising from biological processes should be low in frequency relative to the sampling frequency, constraining the used frequency band might be justified. On the other hand, using only a part of coherency up to a certain frequency during the calculations of might have a non-trivial interpretation in terms of the individual measures. Also, the measures rely on consistent effects in coherency and reducing the used number of coherency samples might increase the influence of noise on the results. In the end, an unconstrained version of each coherency based measure and a version constrained to a band 0-300Hz was used.

For the correlation based methods, versions returning the original correlation and versions returning an absolute value of the correlation were used. Mutual information was used with Scott's rule deciding the number of bins and with a fixed number of 30 bins. 30 bins were chosen because even short 2 seconds long signals should have enough samples to cover the 900 bins in 2D histogram well.

Several generalized linear models with slight variations in spike train regressors were configured. Every model had at least 4 regressors modelling refractory effects and at least 4 regressors modelling coupling. These regressors contained counts of fired spikes in past bins covered by progressively lengthening intervals.

Name	Data type	Measure	Important parameters
crosscorr max	MER	Crosscorrelation	abs=false
crosscorr abs max	MER	Crosscorrelation	abs=true
corr	MER	Pearson’s correlation	abs=false
corr abs	MER	Pearson’s correlation	abs=true
psi	MER	PSI	SegLen=1
psi f300	MER	PSI	SegLen=1; maxFreq=300
pli	MER	PLI	SegLen=1
pli f300	EMR	PLI	SegLen=1; maxFreq=300
wpli	MER	WPLI	SegLen=1
wpli f300	MER	WPLI	SegLen=1; maxFreq=300
icoh max	MER	Imaginary coherency	SegLen=0.25; aggregation=max
icoh max f300	MER	Imaginary coherency	SegLen=0.25; maxFreq=300; aggregation=max
mutual info	MER	Mutual information	binRule=scott
mutual info	MER	Mutual information	NumBins=30
glm 1	spike train	GLM	IntervalSumX=[0-2-4-10-20]; intervalSumY=[0-5-10-20-50-150]
glm 2	spike train	GLM	IntervalSumX=[0-2-4-6-10-20];intervalSumY=[0-5-10-20-50]
glm 4	spike train	GLM	IntervalSumX=[0-2-4-10-20]; intervalSumY=[0-5-10-20-50]
glm 4	spike train	GLM	IntervalSumX=[0-2-4-10-20]; intervalSumY=[0-5-10-20-40-60-100-150]
glm 5	spike train	GLM	IntervalSumX= [0-3-6-10-20]; intervalSumY’=[0-5-10-20-40-60-100-150]

Table 5.4: Summary of the used coupling measures

Measure name	Significant couplings	Possible couplings	Ratio
mutual info	3972	4046	0.9817
corr	3522	4046	0.8704
corr abs	3446	4046	0.8517
crosscorr max	2851	4046	0.7046
crosscorr abs max	2802	4046	0.6925
mutual info 30	2078	4046	0.5136
pli	1489	4046	0.3680
icoh max f300	1376	4046	0.3401
icoh max	1187	4046	0.2934
wpli	1089	4046	0.2691
psi f300	261	4046	0.0645
psi	244	4046	0.0603
pli f300	168	4046	0.0415
glm 4	106	2737	0.0387
glm 5	102	2737	0.0373
wpli f300	150	4046	0.0371
glm 1	87	2737	0.0318
glm 2	82	2737	0.0300
glm 3	79	2737	0.0289

Table 5.5: Ratio of significant couplings for all measures

5.3 Analysis and visualization results

This section contains a summary of the conducted result exploration and analysis together with selected visualizations. The analysed quantities were primarily the ratio of significant couplings and the value distribution of each measure.

For the sake of brevity, this section only contains selected parts of the obtained results, with the rest being available in the attached Jupyter notebook and as exported html document `couplingExploration`. This allows keeping the text reasonably short while still providing access to the full results along with the code used to obtain them. Typically, tables in the subsequent text contain the full results, while the graphs only contain selected measures.

The coupling measures selected for the visualizations are the following: `crosscorr abs max`, `corr abs`, `glm 2`, `mutual info 30`, `psi` and `pli`. For more information about these measures, see table 5.4. Generalized linear models as used in this thesis do not have any value, `glm 2` is therefore not used in visualizations depicting values of the measures. Similarly, `corr abs` is only used in graphs illustrating the measure values, since the other results of this measure are very similar to `crosscorr abs max`. The selected measures appropriately represent all the different measure groups implemented in this work.

The total ratio of significant couplings for every used measure is displayed in table 5.5. The table shows that majority of all couplings for correlation and mutual information based methods was evaluated as significant. This is physiologically unlikely and could be explained as an effect of volume conduction since the methods insensitive to volume conduction have a more reasonable ratio of significant couplings.

The section 5.3.1 discusses the results of different measures with respect to areas of the recording and section 5.3.2 contains the same analysis for dependency of the results on the depths of the recordings relative to the target in the trajectories.

The effect of entering the STN on the measure behaviour is of great interest. However, when investigating this effect using the depths relative to the target, it might be obscured by the fact that the depth of STN entry is different in every trajectory. In order to see the effects of entering the STN more clearly, the depths in each trajectory were aligned to make depth 0 the STN entry depth and additional analysis was conducted. The results of this analysis are displayed in section 5.3.3.

5.3.1 Dependency on brain area

The areas of the recordings were aggregated into three distinct groups before the analysis: *STN*, *unknown* (recording from an area containing no known nucleus) and *other* (recording from a nuclei other than STN). Majority of the records were either *STN* or *unknown*. Since functional connectivity can be measured for a pair of records, each coupling has two areas associated with it. All of the used measures are non-directed, so the order of the areas was not taken into consideration. Therefore, there are six possible pairs of areas, but for the sake of clarity the subsequent tables and figures leave out the least numerous *other* area and only contain three area pairs: *STN-STN*, *STN-unknown* and *unknown-unknown*. The tables and figures containing all of possible the area pairs are available in `couplingExploration`.

Table 5.6 shows how the ratio of significant couplings for each measure depends on the pair of areas the individual recordings were made in. The results in this table show a considerable difference of the ratio for most of the

measure-name	STN-STN	STN-unknown	unknown-unknown
corr	0.7171	0.8227	0.9553
corr abs	0.6746	0.7989	0.9494
crosscorr abs max	0.4303	0.5691	0.8790
crosscorr max	0.4475	0.5853	0.8823
glm 1	0.0629	0.0265	0.0127
glm 2	0.0577	0.0278	0.0116
glm 3	0.0542	0.0265	0.0116
glm 4	0.0734	0.0331	0.0169
glm 5	0.0717	0.0344	0.0159
icoh max	0.2550	0.3022	0.2834
icoh max f300	0.3081	0.3460	0.3228
mutual info	1.0000	0.9971	0.9553
mutual info 30	0.2802	0.3632	0.7055
pli	0.3453	0.3966	0.3465
pli f300	0.0305	0.0391	0.0440
psi	0.0930	0.0763	0.0408
psi f300	0.0651	0.0553	0.0664
wpli	0.1846	0.2040	0.3504
wpli f300	0.0212	0.0210	0.0533

Table 5.6: Ratio of significant couplings for all measures over specific pairs of brain areas

measures with the exception of `psi f300`, `pli f300`, `wpli f300`. This seems to indicate that the restriction of coherency based methods to 0-300Hz band negatively affects the capabilities of these measures. Additionally, `mutual info` show very little variation of the ratio in different areas and the ratio is very close to 1 in all areas. This likely means that Scott's rule for the number of bins is unsuitable for the purpose it was used for in this thesis.

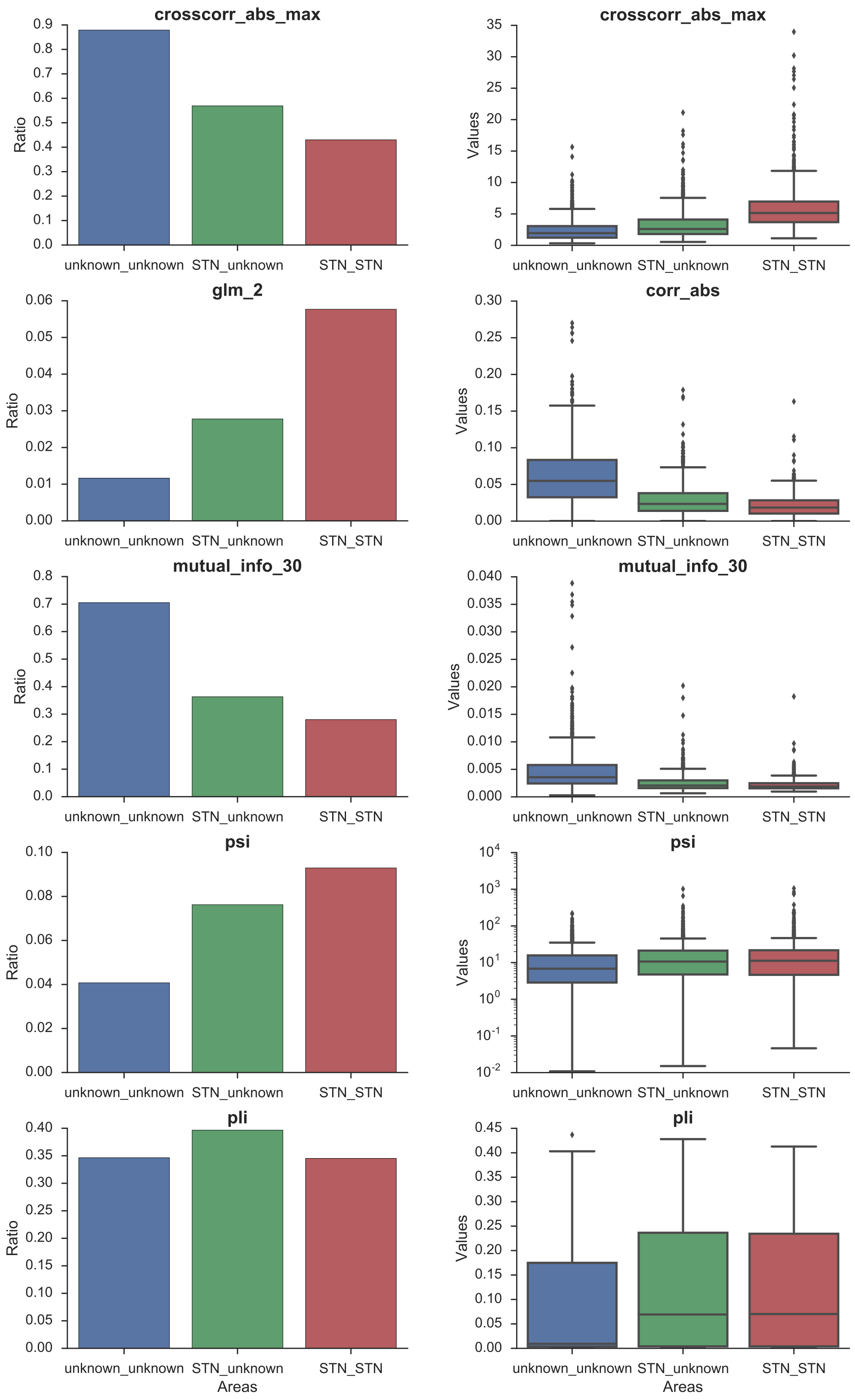
Note that some of the measures have a higher ratio of significant coupling in STN when compared to other areas, while the ratio decreases for other measures. Since the ratio of most of the coherency-based methods and spike train methods increases, this could be again attributed to volume conduction. When a signal does not contain much activity of nearby spiking neurons, the volume conduction dominates the signal and some measures might find a coupling even in the absence of one. On the other hand, when nearby spiking neurons are the sources of significant part of the signals, the effect of the volume conduction diminishes.

To provide a statistical evaluation of the dependency of the ratio of significant couplings on the brain areas, a chi-squared test of independence was conducted. The null hypothesis of this test is that the brain areas and the numbers of significant and non-significant couplings are independent of each other, a low p-value therefore indicates dependency. This test was conducted for all combinations of area pairs. The full results for the different measures are available in `couplingExploration`, the p-values for the most relevant area pairs are in the table 5.7. To keep headers short, area pair *STN-STN* is referred to as STN, area pair *STN-unknown* is referred to as mixed and area pair *unknown-unknown* is referred to as unknown in this table.

	STN vs mixed	STN vs unknown	mixed vs unknown
corr	<0.0001	<0.0001	<0.0001
corr abs	<0.0001	<0.0001	<0.0001
crosscorr abs max	<0.0001	<0.0001	<0.0001
crosscorr max	<0.0001	<0.0001	<0.0001
glm 1	0.0017	<0.0001	0.0580
glm 2	0.0095	<0.0001	0.0242
glm 3	0.0139	<0.0001	0.0369
glm 4	0.0014	<0.0001	0.0458
glm 5	0.0032	<0.0001	0.0206
icoh max	0.0321	0.1681	0.3229
icoh max f300	0.1013	0.5085	0.2355
mutual info	0.3773	<0.0001	<0.0001
mutual info 30	0.0003	<0.0001	<0.0001
pli	0.0301	0.9923	0.0108
pli f300	0.4026	0.1498	0.6055
psi	0.2384	<0.0001	0.0002
psi f300	0.4439	0.9756	0.2864
wpli	0.3352	<0.0001	<0.0001
wpli f300	0.8997	0.0006	0.0001

Table 5.7: p-values for chi-squared test of independence between a number of significant couplings and a combination of area pairs

Figure 5.8a shows the relationship between the ratio of significant couplings and the area for the selected measures using bar graph. Figure 5.8b shows the dependency of the distribution of the measure values on the areas of the recording using boxplots. The boxes in these and subsequent boxplots show the quartiles of the values while the whiskers delimit 1.5 times interquartile



(a) : Ratio of significant couplings

(b) : Distribution of measure values

Figure 5.8: Visualization of selected measures over specific brain areas

range. Note that y-axis for `psi` in figure 5.8b is logarithmic.

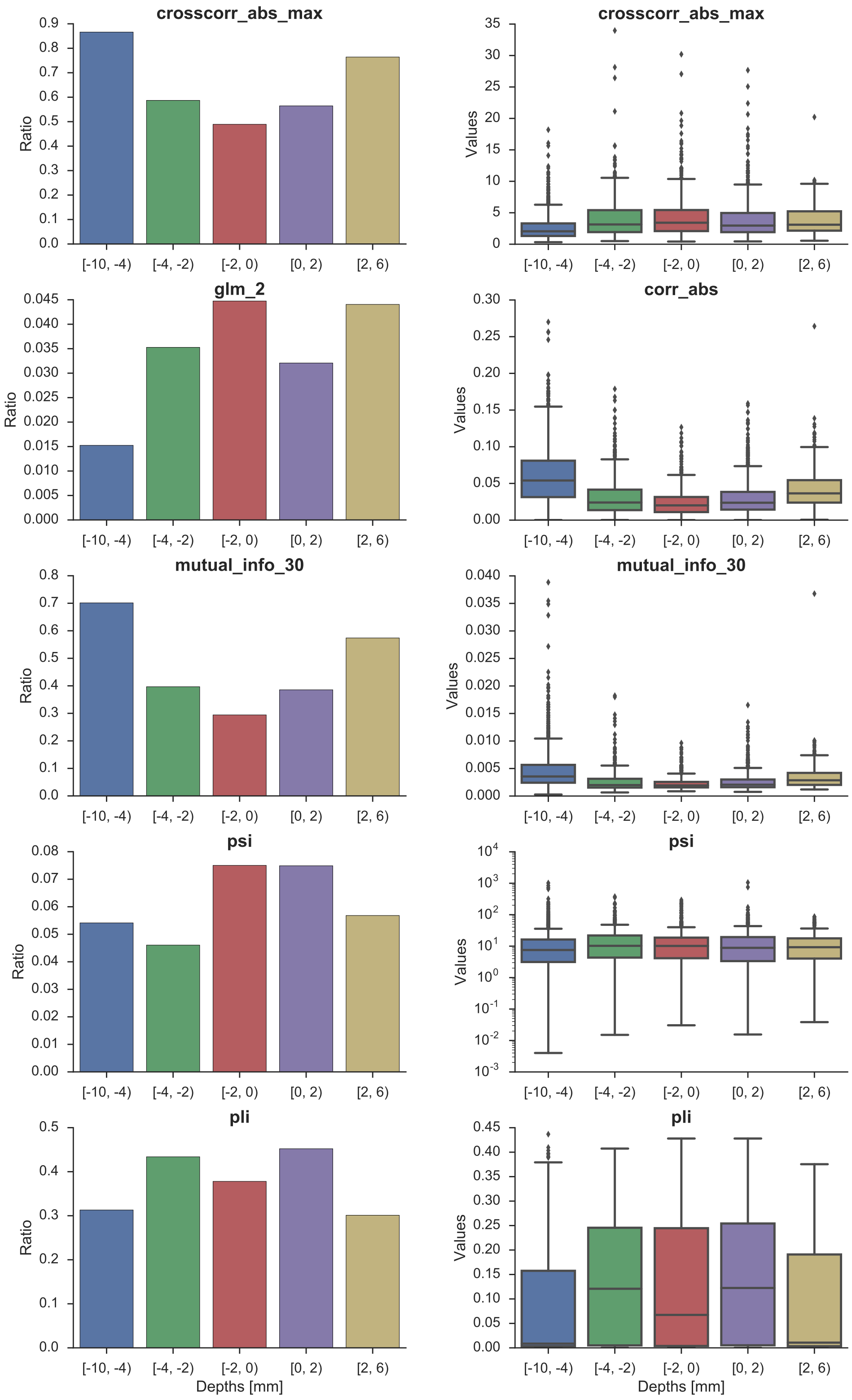
The distributions of the values in 5.8b show a clear difference of median values between *unknown-unknown* and the other area pairs. Again, the values of some of the measures increase in *STN*, while the values of other measures decrease. An interesting example of this fact is the difference between `crosscror abs max` and `corr abs`. The difference between cross-correlation and Pearson's correlation coefficient could again be taken as an evidence of instantaneous interactions dominating the signals outside the STN, while time-lagged interactions dominate inside the STN.

5.3.2 Dependency on depth relative to target

The depths relative to the target in each trajectory were grouped into several intervals for analyses and visualizations. To prevent the creation of intervals with a low number of observations inside, the outlying intervals were made wider. Recordings from all areas are used in this and the following section.

measure name	[-10; -4) mm	[-4; 0) mm	[0; 6) mm
corr	0.9558	0.7826	0.8546
corr abs	0.9446	0.7540	0.8369
crosscorr abs max	0.8662	0.5373	0.6300
crosscorr max	0.8693	0.5549	0.6486
glm 1	0.0173	0.0422	0.037
glm 2	0.0152	0.0402	0.0357
glm 3	0.0152	0.0382	0.0343
glm 4	0.0254	0.0432	0.0502
glm 5	0.0244	0.0422	0.0476
icoh max	0.3167	0.2958	0.2554
icoh max f300	0.3479	0.3499	0.3159
mutual info	0.9701	0.9912	0.9870
mutual info 30	0.7013	0.3448	0.4473
pli	0.3130	0.4056	0.4026
pli f300	0.0535	0.0337	0.0336
psi	0.0541	0.0608	0.0690
psi f300	0.0666	0.0652	0.0606
wpli	0.3367	0.2020	0.2535
wpli f300	0.0591	0.0234	0.0214

Table 5.8: Ratio of significant couplings for all measures in different depths relative to the target



(a) : Ratio of significant couplings

(b) : Distribution of measure values

Figure 5.9: Visualization of selected measures over specific target-rleative depth intervals

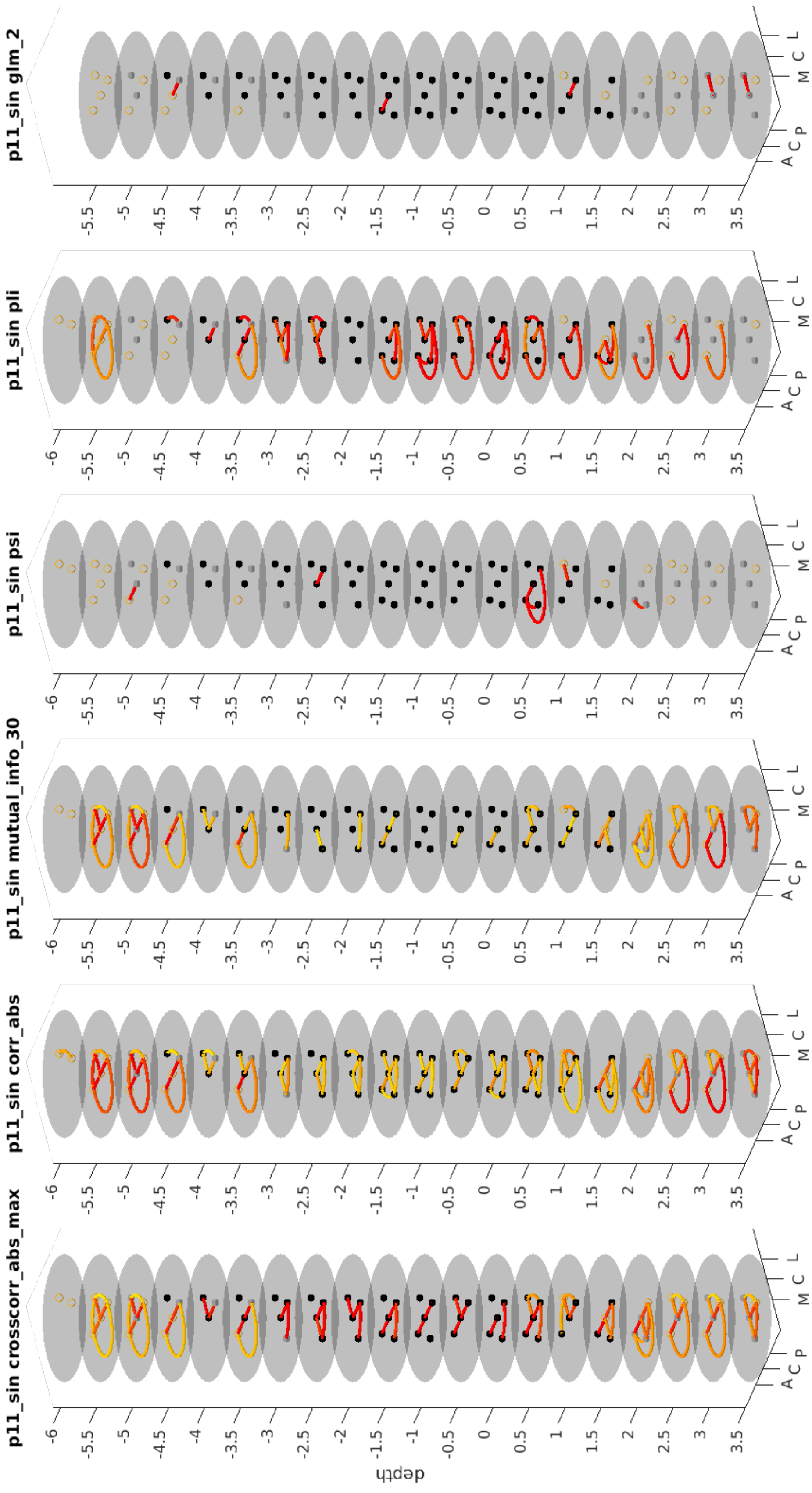


Figure 5.10: A visualization of behaviour of the selected measures for trajectory p11 sin

The table 5.8 illustrates the dependency of the ratio of significant couplings for all measures on the depths relative to target. The relationship between the depths and ratio and values of the selected measures is also shown in figures 5.9a and 5.9b. A chi-squared test of independence between number of significant couplings and depth interval for all neighboring depth intevals can be found in `couplingExploration`.

To provide a comparison between these results and the results presented in section 5.3.1, the target-relative depths the different areas are typically located on should be noted. Majority of the trajectories contain mostly *unknown* area on depths below -4, mostly *STN* on depths between -4 and 2 and mostly *other* and *unknown* on depths above 2. If this information is taken into account, then the results presented in this section closely follow the results from section 5.3.1. The ratio of the significant couplings for coherency-based methods and for spike train methods increase around depth of 0, while the ratio of correlations and mutual information decreases around 0.

Figure 5.10 show an example of 3D visualization created using `couplingVisualize` function. This visualization allows a comparison of the results of selected measures on individual depths for a single trajectory. This figure directly shows how the STN, displayed in the middle of the subplots, affects the measures. Note that inside the STN, the coupled electrodes are usually neigboring, while outside the STN, even the electrodes that are further apart are often coupled. This can be taken as another evidence of the influence of the volume conduction.

5.3.3 Dependency on depth relative to STN entry

Considering that the different trajectories enter the STN on different depths relative to target, it might be useful to shift the depths in such a way that the STN begins on the depth 0. If entering the STN significantly changes the behaviour of the coupling measurement methods, then the results should show a sharp change at the the depth 0 relative to the STN entry.

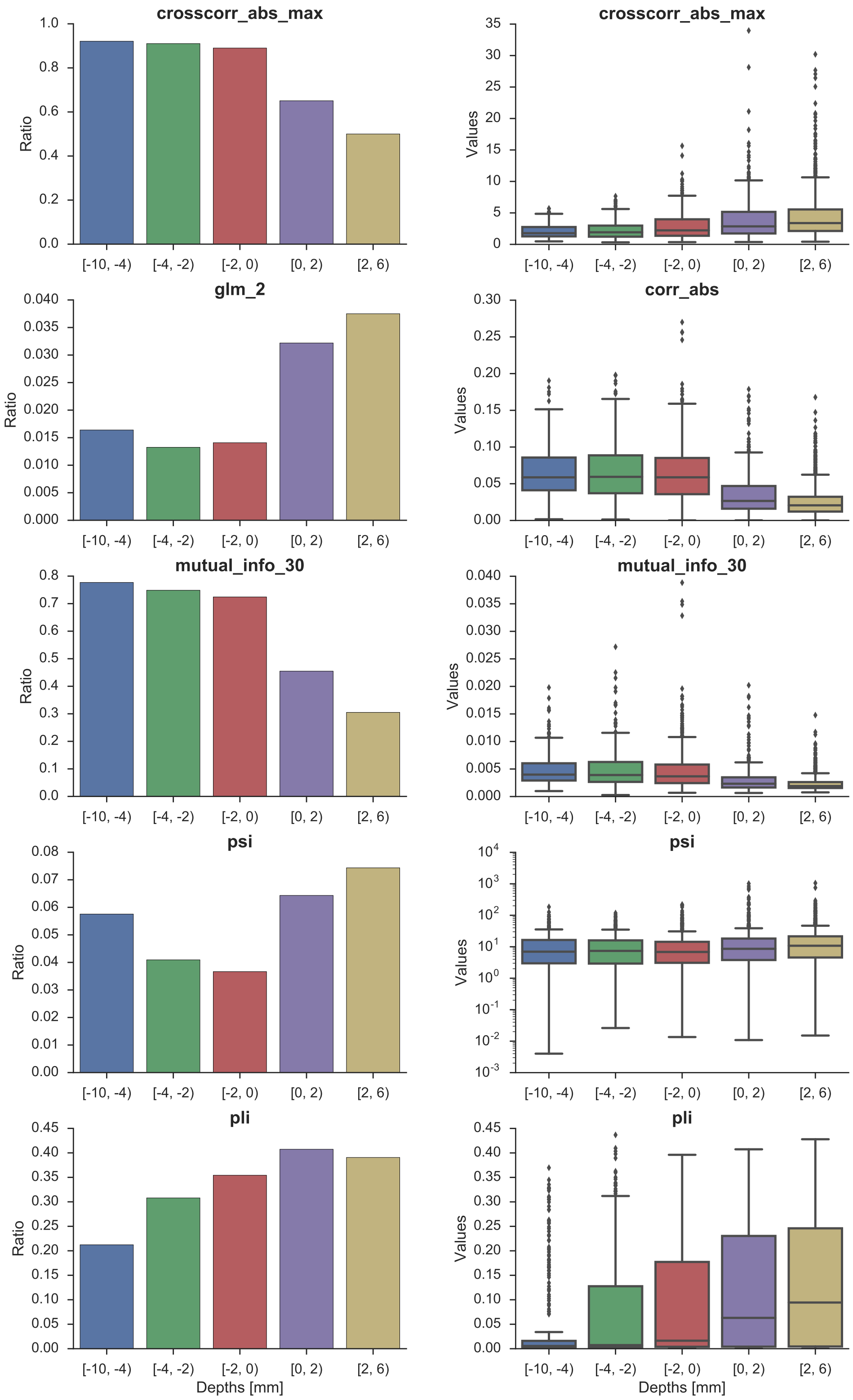
The dependency of the ratio of significant couplings can be seen in table 5.9. The visualizations of the effect of STN entry relative depth for the selected measures can be found in figures 5.11a and 5.11b. A chi-squared test of independence between number of significant couplings and depth interval for all neighboring depth intevals can be found in `couplingExploration`. The smallest p-values for most of the measures in this test were detected when testing the intervals $[-2, 0)$ and $[0, 2)$. However, the null hypothesis for these intervals could not be rejected on a significance level 0.01 for the measures

measure name	[-10; -4) mm	[-4; 0) mm	[0; 6) mm
corr	0.9748	0.9687	0.7985
corr abs	0.9676	0.9641	0.7701
crosscorr abs max	0.9209	0.8996	0.5474
crosscorr max	0.9209	0.9015	0.5652
glm 1	0.0109	0.0183	0.0372
glm 2	0.0164	0.0137	0.0359
glm 3	0.0164	0.0137	0.0345
glm 4	0.0219	0.0213	0.0471
glm 5	0.0219	0.0198	0.0452
icoh max	0.2518	0.3186	0.3001
icoh max f300	0.3129	0.3499	0.3444
mutual info	0.9604	0.9696	0.9913
mutual info 30	0.7770	0.7357	0.3521
pli	0.2122	0.3324	0.3959
pli f300	0.0432	0.0525	0.0394
psi	0.0576	0.0387	0.0712
psi f300	0.0719	0.0663	0.0645
wpli	0.3201	0.3656	0.2059
wpli f300	0.0576	0.0681	0.0202

Table 5.9: Ratio of significant couplings for all measures in different depths relative to the STN

such as `psi`, `pli` and measures based on generalized linear models.

The table and the figures indeed show a sharper change of the characteristics of most the measures around the depth 0 compared to the results in the section 5.3.2. This indicates that the properties of the signals change abruptly when the STN is reached and confirms the results presented in the section 5.3.1.



(a) : Ratio of significant couplings

(b) : Distribution of measure values

Figure 5.11: Visualization of selected measures over specific STN-entry relative depth intervals relative to STN entry

Chapter 6

Conclusion

The goals of this thesis were researching and implementing methods for processing parallelly recorded microelectrode recordings, methods for assessing a functional neuronal connectivity in these signals and applying these methods on a real data set.

A framework of software tools needed for these tasks was implemented in Matlab programming language. The framework is capable of processing both microelectrode recordings and spike trains derived from these signals. Furthermore, the implemented tools can be employed to assess a functional neuronal connectivity in both types of the signals and to analyse and visualize the results. The framework is flexible and extensible to allow work beyond this thesis to be carried out in the future. To expand on this work, more methods of coupling measurement can be added into the framework and the modelling of neuronal spike trains can be explored into a greater depth.

A dataset consisting of 5012 records was processed. The large number of records made fully manual preprocessing of each record impractical. Therefore, external tools such as Siginspect and Waveclus were used for automatic artifact detection and spike detection and sorting. The results provided by these tools were not always satisfactory and had to be manually rechecked. Before more work dealing with a large amount of data is done in the future, developing the automatic methods further would greatly reduce the time required to accomplish these tasks.

The functional connectivity in the processed data were analysed using the implemented software. A total of 4046 pairs of parallel microelectrode recordings and 2737 pairs of parallel spike trains were investigated for effects of coupling. The results illustrated how some well-known problems such as volume conduction can skew the results, but also showed a clear dependency of found functional connectivity on different brain areas the recordings were made in. Future research in this area could consider different factors influencing the results. Also, the clear differences in behaviour of different coupling measures when reaching STN might be leveraged to actually identify the STN in unannotated signals.

Bibliography

[Bakštein et al., 2017] Bakštein, E., Sieger, T., Wild, J., Novák, D., Schneider, J., Vostatek, P., Urgošík, D., and Jech, R. (2017). Methods for automatic detection of artifacts in microelectrode recordings. *Journal of Neuroscience Methods*, 290:39–51.

[Bakštein, 2015] Bakštein, E. (2015). siginspect user manual. Available at `http://neuro.felk.cvut.cz/data/uploads/software/sigInspect_UsersManual.pdf`.

[Bastos and Schoffelen, 2015] Bastos, A. M. and Schoffelen, J.-M. (2015). A tutorial review of functional connectivity analysis methods and their interpretational pitfalls. *Frontiers in systems neuroscience*, 9.

[Bear et al., 2007] Bear, M. F., Connors, B. W., and Paradiso, M. A. (2007). *Neuroscience: exploring the brain*. Lippincott Williams & Wilkins.

[Brodal, 2010] Brodal, P. (2010). *The central nervous system : structure and function*. Oxford University Press, 4th edition.

[Brown et al., 2004] Brown, E. N., Kass, R. E., and Mitra, P. P. (2004). Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–461.

[Dayan and Abbott, 2001] Dayan, P. and Abbott, L. F. (2001). *Theoretical neuroscience*, volume 806. Cambridge, MA: MIT Press.

[Eden, 2011] Eden, U. (2011). Introduction to point processes.

[Grün and Rotter, 2010] Grün, S. and Rotter, S. (2010). *Analysis of parallel spike trains*. Springer.

[Haslinger et al., 2010] Haslinger, R., Pipa, G., and Brown, E. (2010). Discrete time rescaling theorem: determining goodness of fit for discrete time statistical models of neural spiking. *Neural computation*, 22(10):2477–2506.

[Jupyter, 2017] Jupyter, P. (2017). Jupyter documentation. Available at `http://jupyter.org/documentation.html`.

[Lourens et al., 2013] Lourens, M., Meijer, H., Contarino, M., Van den Munckhof, P., Schuurman, P., Van Gils, S., and Bour, L. (2013). Functional neuronal activity and connectivity within the subthalamic nucleus in parkinson's disease. *Clinical neurophysiology*, 124(5):967–981.

[Nolte et al., 2004] Nolte, G., Bai, O., Wheaton, L., Mari, Z., Vorbach, S., and Hallett, M. (2004). Identifying true brain interaction from eeg data using the imaginary part of coherency. *Clinical neurophysiology*, 115(10):2292–2307.

[Nolte and Mueller, 2010] Nolte, G. and Mueller, K. R. (2010). Localizing and estimating causal relations of interacting brain rhythms. *Frontiers in human neuroscience*, 4:209.

[Nolte et al., 2008] Nolte, G., Ziehe, A., Nikulin, V. V., Schlögl, A., Krämer, N., Brismar, T., and Müller, K.-R. (2008). Robustly estimating the flow direction of information in complex physical systems. *Physical review letters*, 100(23):234101.

[Oweiss, 2010] Oweiss, K. G. (2010). *Statistical signal processing for neuroscience and neurotechnology*. Academic Press.

[Peck, 2007] Peck, M. (2007). Deep-brain stimulators for parkinson's disease increase impulsive decision making.

[Quiroga et al., 2004] Quiroga, R. Q., Nadasdy, Z., and Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural computation*, 16(8):1661–1687.

[Sieger, 2013] Sieger, T. (2013). *Processing and Statistical Analysis of Single-Neuron Recordings*. PhD thesis, Czech Technical University.

[Simpson et al., 2001] Simpson, D., Infantosi, A., and Rosas, D. B. (2001). Estimation and significance testing of cross-correlation between cerebral blood flow velocity and background electro-encephalograph activity in signals with missing samples. *Medical and Biological Engineering and Computing*, 39(4):428–433.

[Stam et al., 2007] Stam, C. J., Nolte, G., and Daffertshofer, A. (2007). Phase lag index: assessment of functional connectivity from multi channel eeg and meg with diminished bias from common sources. *Human brain mapping*, 28(11):1178–1193.

[The MathWorks, 2017] The MathWorks, I. (2017). Matlab documentation. Available at `https://www.mathworks.com/help/matlab/`.

[Truccolo et al., 2005] Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089.

[Vinck et al., 2011] Vinck, M., Oostenveld, R., Van Wingerden, M., Battaglia, F., and Pennartz, C. M. (2011). An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias. *Neuroimage*, 55(4):1548–1565.

[West et al., 2016] West, T., Farmer, S., Berthouze, L., Jha, A., Beudel, M., Foltynie, T., Limousin, P., Zrinzo, L., Brown, P., and Litvak, V. (2016). The parkinsonian subthalamic network: measures of power, linear, and non-linear synchronization and their relationship to l-dopa treatment and off state motor severity. *Frontiers in human neuroscience*, 10.

[Wild, 2015] Wild, J. (2015). *Spike Sorting of Microelectrode Single-schannel Recordings Evaluation and Applications*. PhD thesis, Czech Technical University.

Appendix A

Software documentation

This appendix contains a structured documentation of the original software tools. An overview of the architecture and reasoning behind it can be found in section 5.1.

The software was written in the Matlab and Python programming languages. It is organised into three components and each component is contained in a single directory. Additionally, functions and utilities that are used in more than one component are placed in a special directory. This organisation is reflected in the structure of this documentation is summarised in the table A.1.

The components typically contain public high level functions in their directories and lower level private functions in `private/` or other subdirectories. The high level functions are meant to called by the users, while the low level functions are meant to be accessed through the public functions.

Additionally, the components can contain scripts with example usage in the subdirectory `examples/` and tools for visualization of the discrete workflow steps in the subdirectory `visualization/`.

This documentation will describe the most important public functions, their associated private functions, their parameters and how to use them together.

Directory	Component	Section
`data_provisioning/`	data provisioning	A.1
`coupling_measurement/`	coupling measurement	A.2
`visualization_analysis/`	visualization and analysis	A.3
`utils/`		A.4

Table A.1: Code organisation summary

A.1 Data provisioning

The data provisioning component deals with loading or generating data in appropriate format and preprocessing it. It contains two high level functions: `couplingLoadTrajectoryDao.m` and `couplingPreprocess.m`.

A.1.1 couplingLoadTrajectoryDao

`couplingLoadTrajectoryDao.m` is a `Matlab` function that loads data for a whole trajecotry and its descriptors using the `DAO` interface.

Parameters

- `patientId` - real patient ID in `DAO`
- `trajectory` - trajectory ID in `DAO`
- `dataType` - `raw` or `spikes`, default `raw`
- `positions` - a list of positions in a trajectory to load, default all

Returns

- `trajectory` - data and descriptors for a trajectory in a format described in Fig. 5.2

A.1.2 couplingPreprocess

`couplingPreprocess.m` is a `Matlab` function that sequentially applies the user selected preprocessing methods to a trajectory. The preprocessing methods are implemented as functions in the `private` subdirectory and are meant to be acessed through the `couplingPreprocess.m` function. The functions accepts the preprocessing methods as string names, calls the appropriate `private` functions with appropriate parameters and writes the names of the applied methods into the data descriptors.

Parameters

- `trajectory`
- `preprocessingList` - a cell array containing names of preprocessing methods to be applied

Returns

- `trajectory` with the preprocessing applied and data descriptors updated

Associated private functions

- `removeNanFromRawSignals.m`
- `removeShortSignals.m`
- `binSpikes.m`
- `frequencyFiltering.m`

`removeNanFromRawSignals.m` can be called using the name 'removeNaN' in `preprocessingList` and removes the missing data from the signals by selecting only the longest contiguous segment which is free of missing samples in all of the signals in a position.

Parameters

- `position` - a single row of trajectory
- `contiguous` - whether only contiguous signal segment should be returned, default: true

Returns

- `position` - a single row of trajectory with missing samples removed

`removeShortSignals.m` can be called using the name 'removeShort' in `preprocessingList` and removes the signals which are too short.

Parameters

- `position` - a single row of trajectory
- `limFs` - minimum signal length as a multiple of signal sampling frequency, default: 2

Returns

- `position` - a single row of trajectory with short signals removed

`binSpikes.m` can be called using the name 'binSpikes' in `preprocessingList` and transforms the spike train representation into discrete binned representation.

`frequencyFiltering.m` can be called using the name 'harm50' or 'comb50' in `preprocessingList` and it applies the selected preconstructed frequency filter to all signals in a given position. The name used in `preprocessingList` selects the applied filter.

A.2 Coupling measurement

The coupling measurement component uses different coupling measures to assess functional connectivity in a trajectory.

A.2.1 couplingMeasureInformation

`couplingMeasureInformation.m` is a `Matlab` function that accepts a string representation of a coupling measure and returns all relevant information about it. This allows defining frequently used combinations of methods and parameters and accessing them using a name.
Parameters

- `name` - a string representation of a measure

Returns

- `measure` - `Matlab` structure with fields:
    - `name` - copied from argument
    - `computeFun` - a handle to a function to be called
    - `params` - cell array with additional parameters for `computeFun`
    - `cached` - whether the measure should be cached, unused
    - `useOnRaw` - whether the measure should be used on MER, unused
    - `useOnSpikes` - whether the measure should be used on spike trains, unused

A.2.2 couplingComputeMER

`couplingComputeMER.m` assesses coupling between all combinations of parallel MERs in a trajectory for all specified coupling measures.
Parameters

- `trajectory`
- `couplingMeasures` - cell array of measure names / structure array of measures to be used
- `numArtificial` - number of artificial signals to be used for significance estimation, default 999

Returns

- `coupling`

A.2.3 couplingComputeST

`couplingComputeST.m` assesses coupling between all combinations of parallel spike trains in a trajectory using all specified generalized linear models.
Parameters

- `trajectory`
- `couplingMeasures` - cell array of measure names / structure array of measures to be used

Returns

- `coupling`

A.3 Visualization and analysis

`couplingVisualize.m` accepts one or more coupling structures and visualizes them along their depths.
Parameters

- `couplingData` - `coupling` or cell array containing `coupling` structures
- `zColumn` - 'depth' or 'position', default: 'depth'
- `curvedLines` - if true, the lines between the electrodes curve, default: true
- `lineColorField` - field from the 'couplingData' to be used to color lines, default: 'value'
- `lineHiddenField` - field from the 'couplingData' to be used to decide whether a line id displayed or not, default: 'significance'
- `LineUpperBound`, `LineLowerBound` - boundary values of `lineHiddenField`

`couplingExploration.ipynb` is a Jupyter notebook (formerly known as iPython notebook) containing exploratory analysis of the data.

A.4 Utils

`couplingCacheLoad.m` loads results for specified coupling measures, preprocessing and trajectories from a specified cache folder
Parameters

- `trajectoryNames` - cell array with strings containing trajectory names
- `measureNames` - cell array with strings containing measure names
- `preprocessing` - cell array of cell arrays containing strings with names of preprocessing methods
- `concatResults` - if true, one struct array is returned, if false, cell array of struct arrays is returned, default: false
- `codeRoot` - path to a root of the code

Returns

- `results` - a coupling structure or a cell array of coupling structures

`couplingCacheSace.m` saves a result structure to a file according to its trajectory, measure, preprocessing etc.
Parameters

- `results` - coupling structure

`couplinExporResults.m` exports given results to a .csv file
Parameters

- `results` - coupling structure
- `filename` - filename to save the results as