On the pertinence of a numerical transmission model for neural information

Zoran Tiganj

Université Lille 1, Sciences et Technologies

École doctorale Sciences pour l'Ingénieur
UFR IEEA

On the pertinence of a numerical transmission model for neural information

Zoran Tiganj

THESIS

prepared at INRIA - Lille Nord Europe
and defended at the University Lille 1
on 8 November of 2011

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Science

Specialized in Control, Informatics, Signal and Image Processing
In the field of Neural Engineering

Members of the jury

President :	Jean-Pierre Richard
Reviewers :	Eric Moreau David Guiraud
Examiners :	Ratko Magjarevic Christophe Pouzat
Invited member :	Maxime Colas
Advisors :	Mamadou Mboup Lotfi Belkoura

INRIA Lille - Nord Europe

Order number: 40603

informatics mathematics

Université Lille1 Sciences et Technologies

Université Lille 1, Sciences et Technologies

École doctorale Sciences pour l'Ingénieur
UFR IEEA

Sur la pertinence d'un modele de transmission numérique pour l'information neuronal

Zoran Tiganj

THÈSE

preparée á INRIA - Lille Nord Europe
et présentée á l'Université Lille 1
le 8 novembre 2011

Thèse présentée en vue d'obtenir le titre de

Docteur en Sciences

Discipline: Automatique, Génie Informatique, Traitement du Signal et des Images
Application: Génie Neuronal

Composition du jury

President :	Jean-Pierre Richard
Rapporteurs :	Eric Moreau David Guiraud
Examinateurs :	Ratko Magjarevic Christophe Pouzat
Invité :	Maxime Colas
Directeurs :	Mamadou Mboup Lotfi Belkoura

INRIA Lille - Nord Europe

Numéro d'ordre: 40603

Abstract

How living organisms, including us, process information has been one of the largest mysteries of the whole humankind. For almost all the animals, information processing is performed by a group of organs called nervous system, with neurons as the major information processing units. Neurons are mutually connected and communicate primarily using action potentials. How the information is actually coded within the nervous system is still unknown. The implications of a good understanding of the neural code are enormous and influence several different fields of research: from neural prosthetics, over artificial intelligence all the way to the research of consciousness. Thousands of scientific papers have addressed the neural coding problem.

In this thesis we bring together advanced mathematical tools and signal processing methods to address three very important problems in neuroscience: neural action potential (spike) detection, neural spike sorting and directly neural coding.

Starting from extracellular neural recordings, which are the most common way of acquiring the neural activity, we first address the problem of spike detection. The spike time occurrences appear (as irregularities) explicitly in the distributional derivatives of the neural signal. The problem is seen as a change point detection problem. Using operational calculus, which provides a convenient framework to handle such distributional derivatives, we characterize the time occurrence of a spike by an explicit formula. We derive from this characterization a joint change point detection and localization system implemented using a digital Volterra filter.

After spike detection we address the spike sorting problem. The detected spikes to be sorted are first processed with some feature extraction technique, such as PCA, and then represented in a space with reduced dimension by keeping only a few most important features. The resulting space is next filtered in order to emphasis the centers of the clusters which are assumed to be areas of the highest spike density in the feature vector space. Lastly, the spikes are assigned to a particular cluster using a sort of a greedy algorithm.

Next, we extend this simple sorting algorithm for case when multi-channel recordings are available. In such context, the performance of ICA is known to be limited since the number of recording sites is much lower than the number of the neurons around. The algorithm uses an iterative application of ICA and a deflation technique in two nested loops. In each iteration of the external loop, the spiking activity of one neuron is singled out and then deflated from the recordings. The internal loop implements a sequence of ICA and spike detection for removing the noise and all the spikes that are not coming from the targeted neuron.

Validation of the proposed spike detection and sorting algorithms is done using simulated and real neural recordings. As real recordings, we used simultaneous intra- extra-cellular. The detection and sorting is done on the extracellular recordings and the validation of the results is done using the intracellular counterparts.

Finally, we discuss on properties of the neural code. Inspired by the nature in general, where information is commonly present in a discrete form (from quantum of light and matter to genetic code for example), we investigate whether the nature of the neural code is discrete or continuous. Moreover, if it is discrete, whether the elements of the code are drawn from a finite alphabet. We particulary address pulse-position coding scheme, making a link between communication theory and neural code. Even though we do not give an explicit answer to these questions, we suggest a possibility that on different time scales and in different parts of a nervous system several coding schemes might coexist and that nature of the neural code might be dual.

Résumé

Comment les organismes vivants, nous inclus, traitent les informations a toujours été l'un des plus grands mystères de toute l'humanité. Pour presque tous les animaux, le traitement d'informations est opéré par un groupe d'organes appelé système nerveux, dont les neurones sont les principales unités de traitement d'information. Les neurones sont reliés entre eux et communiquent essentiellement par le biais de potentiels d'action. La façon exacte dont l'information est codée au sein du système nerveux est toujours inconnue. Les implications d'une bonne compréhension du code neural sont gigantesques et influencent plusieurs domaines de recherche différents: depuis la prothétique neurale en passant par l'intelligence artificielle et jusqu'a la recherche de conscience. Des milliers d'articles scientifiques ont abordé le problème du codage neural.

Dans cette these, nous utilisons un ensemble d'outils avancés de mathématiques et de méthodes de traitement du signal pour aborder trois problemes importants en neuroscience: la détection de potentiels d'action (spikes), leur tri, et le codage neuronal.

A partir d'enregistrements invasifs de l'activité de neurones, nous avons d'abord abordé la question de la détection de spikes. Les instants d'occurrence des spikes apparaissent (comme des irrégularités) de maniere explicite dans les dérivées distributionnelles du signal neuronal. Le probleme est alors posé en termes de détection et estimation conjointe de ces instants d'occurrence. En utilisant le calcul opérationnel, qui fournit un cadre commode pour manipuler des dirac, nous caractérisons le temps d'occurrence d'un spike par une formule explicite. Nous déduisons de cette caractérisation un système de détection et localisation de spikes implémenté par l'utilisation d'un filtre de Volterra.

Après avoir développé l'analyse de potentiels d'action, nous aborderons la partie qui traite des problèmes de triage des potentiels d'action. Ces potentiels d'action que l'on veut trier sont d'abord abordés par une technique permettant d'extraire des caractéristiques comme PCA, et de les recueillir dans un espace réduit tout en conservant seulement les caractéristiques les plus pertinentes. L'espace résultant est par la suite filtré. L'attention est alors portée sur l'augmentation des centres de parutions qui sont programmés pour être des régions de grande densité de potentiels d'action, dans cet espace où sont retenues ces caractéristiques spécifiques.

Nous élargissons ensuite ce simple algorithme de triage pour les cas où les enregistrements multi-canaux sont disponibles en utilisant l'ICA. Dans un tel contexte, la performance de l'ICA est connue pour être limitée du fait du nombre de sites d'enregistrements beaucoup plus faible que le nombre de neurones autour. L'algorithme utilise une application itérative de l'algorithme

ICA et une technique de déflation dans deux boucles imbriquées. Dans chaque itération de la boucle externe, l'activité d'un neurone est ciblée, puis isolée et ensuite élaguée des enregistrements. La boucle interne met en ouvre une séquence d'applications de ICA et de détection afin de supprimer tout ce qui ne releve pas du neurone cible.

La validation de détection de potentiels d'action et du tri a été réalisée en utilisant des enregistrements de neurones simulés et réels. En tenant compte des enregistrements réels, nous avons utilisé simultanément ceux intra- extracellulaires. La détection et le tri ont été réalisés sur l'enregistrement extracellulaire et la validation des résultats a été faite en utilisant l'équivalent intracellulaire.

Enfin, nous menons une discussion sur les propriétés du code neuronal. Inspiré par la nature en général, où l'information est souvent présente sous une forme discrète, nous examinons l'hypothese d'un code reposant sur un alphabet fini. Cette hypothese correspond a un schéma de modulation PPM qui est déja suggéré par le codage temporel et la présence de jitter. Même si nous ne donnons pas une réponse explicite à ces questions, nous proposons une possibilité qui est qu'à différentes échelles de temps et dans différentes parties d'un système nerveux, plusieurs schémas de codage peuvent coexister et que la nature du code neuronal pourrait être double.

Acknowledgments

I owe my deepest gratitude to my thesis advisor Mamadou Mboup. His positive spirit, great patience and amazingly clear way of explaining things made conduction of this thesis possible, but also taught me valuable life lessons. He encouraged me to develop my own ideas, but always directed me towards the right path.

A special thanks goes to my second thesis advisor Lotfi Belkoura. Lotfi was always there for me and kindly helped me in many ways to successfully conduct my thesis.

I would like to thank Jean-Pierre Richard, head of INRIA-Lille project NON-A (former ALIEN) for warmly welcoming me to his team and for supporting me during my thesis. Jean-Pierre, as well as the entire team, inspired me through our many creative and productive discussions. I thank also Marie-Benedicte Dernoncourt and Anne-Claire Binetruy for their kind help in dealing with a large number of administrative issues.

I am deeply grateful to Ratko Magjarevic, my master thesis advisor, with whom I have made my fist steps into science. Thank you Ratko for your constant moral and scientific support and your precious advices during my master's and PhD thesis.

I am heartily thankful to Christophe Pouzat for the inspiring discussions and for sharing his wise perspective which helped me to understand the bigger picture and to pose the right questions in my research. Christophe also kindly allowed me access to the data he obtained using the extracellular recordings and introduced me to this technique.

I would also like to thank Velimir Isgum for guiding me through my first contacts with electroencephalography and for his profound career advices. Together with his PhD students Magdalena Krbot and Ana-Branka Sefer he provided me with access to their EEG recordings, for which I am very grateful.

It is my great pleasure and honour to thank the reviewers, David Guiraud and Eric Moreau, for many insightful and helpful comments they provided me with, also to all the members of the jury, whose work I greatly admire.

Last but not the least; I would like to thank my family and friends for their strong moral support during my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

How living organisms, including us, process information has been one of the largest mysteries of the whole humankind. Different animals process all kinds of stimuli in a proficient way, obtaining amazing outputs. Today, we believe that we generally understand how the information is transmitted within a nervous system. However, the exact way how it is coded is still unknown, even though there has been a large number of studies that addressed this question. Few decades ago, the coding problem seemed to be very well understood, but recently it has been suggested that the coding process is likely much more complex than previously assumed. Apart from the scientific curiosity, fully cracking the neural code would open the doors widely for a great development of neuroprosthetics applications and artificial intelligence. One important problem in decoding of the neural code is obtaining reliable recordings from the nervous system. That problem will be closely analyzed in this dissertation, together with discussion on what coding schemes might exist in a nervous system.

This chapter starts with a brief overview on the biological organization of the system that performs this sophisticated neural information processing, starting from the top with the nervous system and ending by describing neurons and their activity. The aim is to give to a reader an impression of complexity and variety of the biological elements and mechanisms that are the subject of interest of the following chapters. For details on the anatomy of the complete nervous system the interested reader should see [Netter 2002] and for description of the principles of neural science we suggest the great

books [Purves 2008], [Kandel 2000], [Rose 2006] and [Izhikevich 2006] which have been the main source of information in this overview.

Next, we discuss on problems and challenges towards neural coding. We analyze currently available recording techniques and discuss on how better processing of these recordings can lead to a better understanding of the neural code. Particulary we present problems of neural action potentials detection and their sorting, which stands as important challenge in neuroscience and which will be central topic of chapters two, three and four. Finally, in chapter five, we briefly present different neural coding schemes, particulary discussing the finite alphabet coding scheme.

1.1 Biological background

This section gives a brief overview of the most interesting biological elements, from perspective of neural information processing.

1.1.1 Nervous system

For almost all the animals, information processing is performed by a group of organs called *nervous system*[1]. The main processing unit of a nervous system is a special type of cell, called *neuron*. Typically, the nervous system consists of two parts: central and peripheral[2]. The *central nervous system* of vertebrates (such as humans) contains a brain, a spinal cord, and a retina. The *peripheral nervous system* consists of sensory neurons, clusters of neurons called ganglia, and nerves connecting them to each other and to the central nervous system.

1.1.2 Brain

The most complex and the most advanced part of the nervous system is the brain. This short overview will give only some brief insight in its complexity.

1.1.2.1 Cell types

The brain is composed of two broad classes of cells: neurons and glial (also called neuroglial or supporter cells). In humans, the brain contains roughly 100 billion neurons[3] and around the same number of glial cells.

[1] Only sponges and a few other simpler animals have no nervous system.

[2] Some primitive animals such as jellyfish and starfish have a decentralized nervous system.

[3] The number depends on many factors such as age and gender.

Glial cells come in several types, which perform a number of critical functions, including structural support, providing of nutrition and energy, maintaining homeostasis, destroying pathogenes and removing dead neurons.

Until recently neurons have been considered as the only cells directly responsible for an information processing in the brain. However, some recent studies have shown that glial cells could also have a role, even though much smaller than neurons, in processing of information[4]. The role of the glial cells is not going be investigated in this thesis; we will concentrate on the role neurons.

1.1.2.2 Major parts

The brain of a vertebrate can be divided into six main regions: the telencephalon (cerebrum), diencephalon (thalamus and hypothalamus), mesencephalon, cerebellum, pons, and medulla oblongata (figure 1.1). In mam-

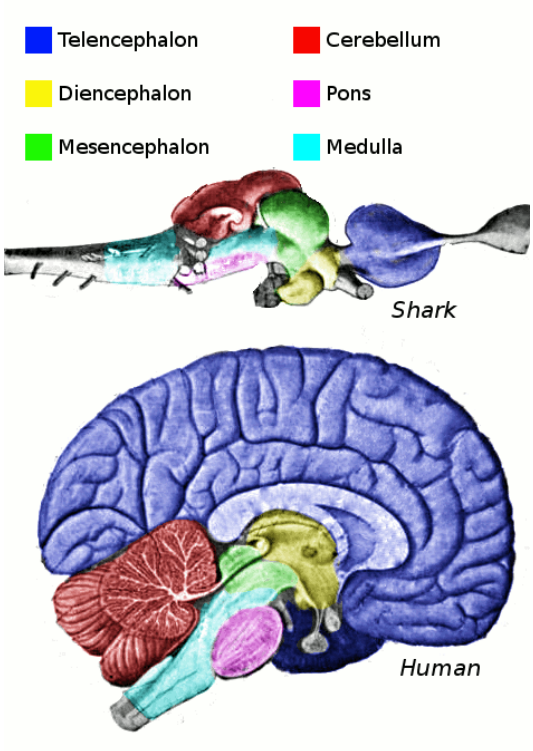


Figure 1.1: Regions of human's and shark's brain. From Wikipedia.

malian brain, the outer layer of the telencephalon, called *cerebral cortex*, plays a key role in memory, attention, perceptual awareness, thought, language, and

[4]See e.g. [Huxtable 2010] and [Eric 2003] where it is shown how glia can modulate neuronal activity by releasing transmitters in a process called gliotransmission.

consciousness. As it is so important the cortex is often in the focus of the studies about neural information processing. Even though we will investigate some general principles in this thesis, most of our simulations and discussions will be based on the data recorded from cortex, either human or other animals. To illustrate the cortex complexity we give several statistical information we find fascinating. The human cerebral cortex is $2 - 4mm$ thick. In a cubic millimeter it contains more than 10^4 cell bodies and several kilometers of 'wires'. Overall, it contains more than 10 billion neurons, where each is usually connected with more than 10000 others. Organization of the cortex is quite complex. Its laminar structure is usually described through 6 layers (figure 1.3), however the number of layers, their thickness and organization are not the same over the whole surface of the cortex. In addition, cortex may

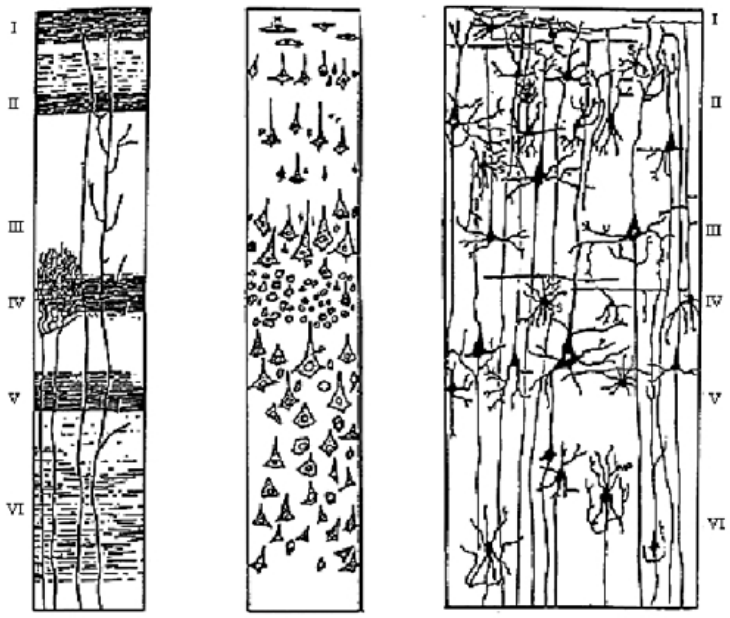


Figure 1.2: Layer organization in different parts of cerebral cortex, given through three different visualization and coloration techniques: (a) Weigert's coloration (shows cell bodies and axons inside and between the layers) (b) Nissl's coloration (only reveals cell bodies) (c) Golgi's coloration shows the whole cells. From [Nolte 2002].

be classified on the basis of gross topographical conventions into four lobes: temporal lobe, occipital lobe, parietal lobe and frontal lobe.

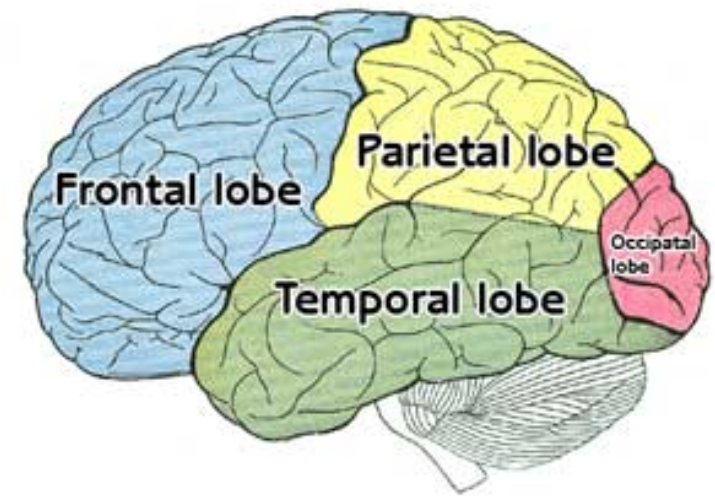


Figure 1.3: The four lobes of cerebral cortex. From Wikipedia.

1.1.3 Neurons

The role of a neuron is to receive, process and transmit information by electrical and chemical signaling. Neurons communicate mutually making the large networks and systems that have been described before in this chapter.

Neurons are morphologically very complex. The main parts of a typical neuron are shown on figure 1.4. Most of neurons receive stimuli through *synapses* (from outside world or from other neurons). The stimuli provoke an electrical responses (postsynaptic potential) which then through dendrites come to the cell membrane.

The membrane is populated by voltage activated ion channels. These channels help to transmit the signal. The membrane bounds the body of a cell (called *soma*) which is the metabolic center with *nucleus* that stores genetic information of the cell. When the stimulus is large enough, a signal, called action potential, is generated in *axon hillock* (see following subsection for a detail description of this process). The action potential is carried away from the body of a neuron by *axon* which can have length from 0.1mm to more than a meter. At the end of an axon are *axon terminals*. They contain synapses, each synapse conducts signal between two neurons. A signal is emitted from a *presynaptic cell* and received by a *postsynaptic cell*. The cells are usually not anatomically connected. The small space between the communicating cells is called *synaptic cleft*.

Even though the majority of neurons have the parts described above, morphologically and functionally neurons can be very different among each other. Different classifications divide neurons into even more than 1000 different

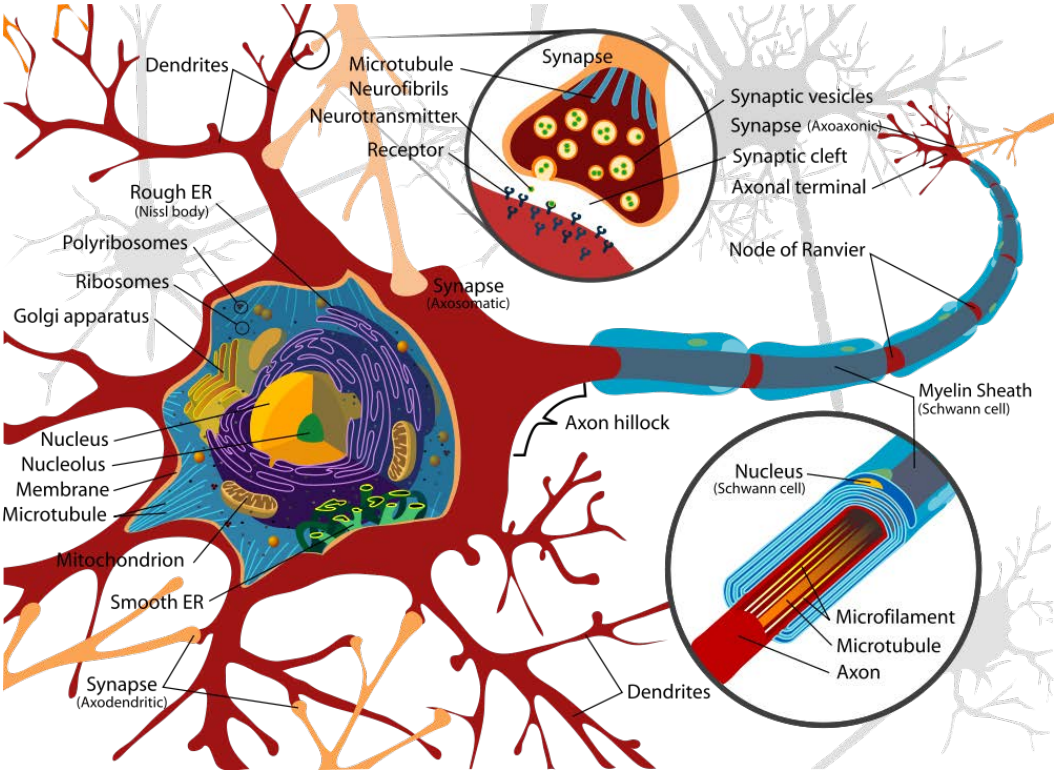


Figure 1.4: Main parts of a neuron. From Wikipedia.

groups. Only to illustrate this diversity, several main criteria used for such classifications will be described here. Mainly, neuron are classified by the function they play in a nervous system:

- *Sensory* (*afferent*) neurons convey information from tissues and organs into the central nervous system.
- *Motor* (*efferent*) neurons transmit signals from the central nervous system to the effector cells, *e.g.* muscle cells.
- *Interneurons* connect neurons within specific regions of the central nervous system.

Sensory neurons and interneurons that are close to the sensory neurons will be particulary interesting for us in this thesis, especially in the, neural coding related chapter five. Even though we will not conduct any biological experiments we will often use data recorded from these neurons since we can try to relate their response with some controlled stimuli. Such experimental paradigm is a very useful insight in neural information processing, widely performed in today neuroscience.

To illustrate better the anatomical diversity of neurons we mention another important classification, by the number of processes that originate from the cell body (figure 1.5):

- *Unipolar* or *pseudounipolar*: dendrite and axon emerging from same process.
- *Bipolar*: axon and single dendrite on opposite ends of the soma.
- *Multipolar*: more than two dendrites.

1.1.4 Action potentials

An *action potential* is a short-lasting (around $1ms$) event in which the electrical membrane potential of a cell rapidly rises and falls, following a consistent trajectory. Its purpose is transmission of information from one cell to others. Action potentials occur in several types of animal cells, called excitable cells, which include neurons, muscle cells, and endocrine cells, as well as some plant cells. In this dissertation neural action potentials will be in the focus.

1.1.4.1 Resting potential

To understand the mechanism of generation of action potentials we first have to consider resting state, when no stimulus comes to a neuron's membrane.

Four ionic currents are mostly responsible for the electrical activity of a neuron: sodium (Na^+), potassium (K^+), chloride (Cl^-), and calcium (Ca^{2+}). The concentrations of these ions are different on the inside and the outside of a cell. These ionic gradients are the major forces driving neural activity.

The extracellular medium has a high concentration of Na^+ and Cl^- ions (it is salty, like seawater), and a relatively high concentration of Ca^{2+}. The intracellular medium has high concentrations of K^+ and different negatively charged molecules (denote generically A^- for anions). Concentrations of these molecules inside and outside of the cell are illustrated on figure 1.6.

The cell membrane has large protein molecules forming channels through which ions (but not A^-) can flow according to their electrochemical gradients. The flow of Na^+ and Ca^{2+} ions is not significant, at least at rest, but the flow of K^+ and Cl^- ions is. This, however, does not eliminate the concentration asymmetry for two reasons:

- *Passive redistribution*: The impermeable anions A^- attract more K^+ into the cell and repel more Cl^- out of the cell, thereby creating concentration gradients.

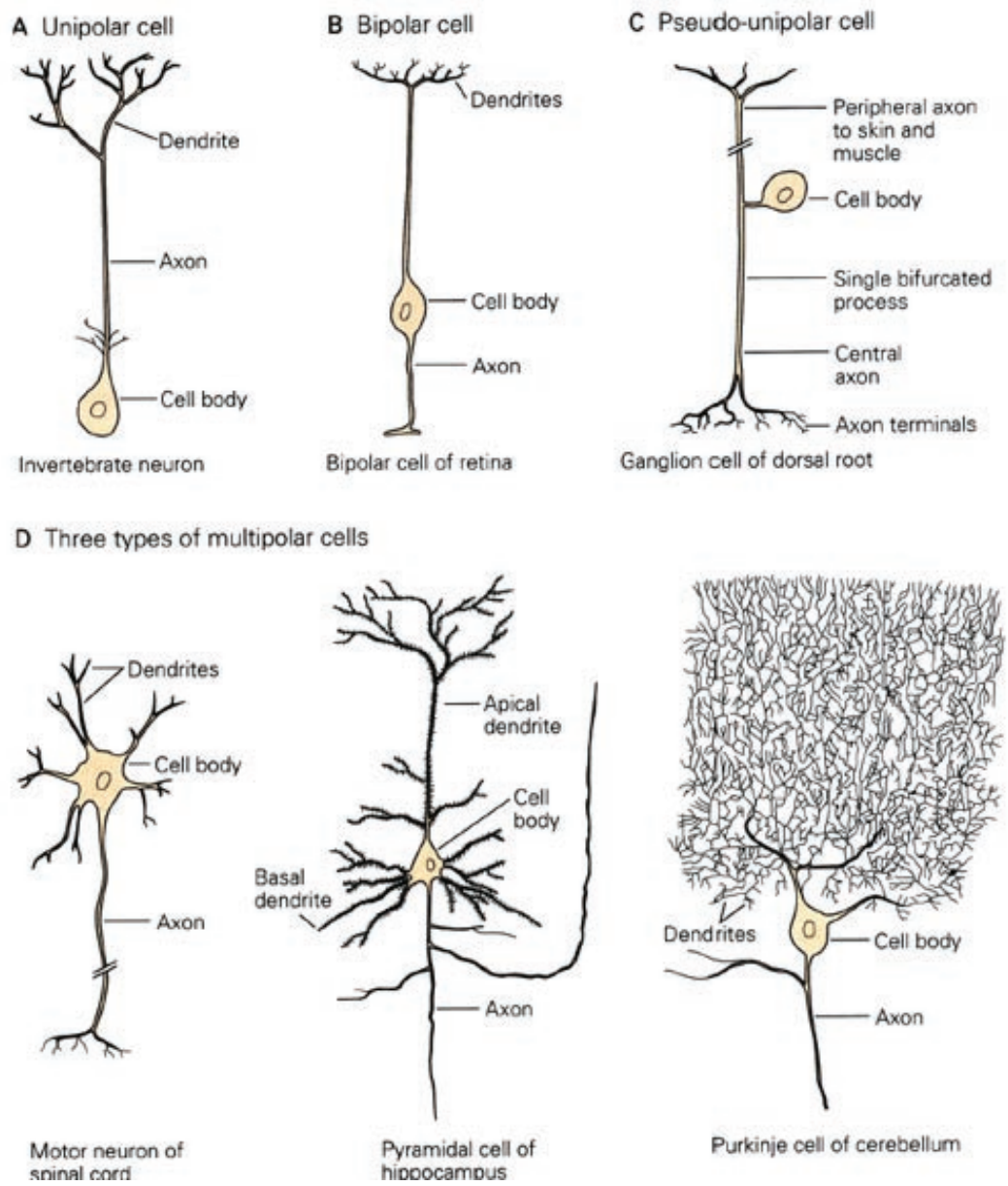


Figure 1.5: Neurons can be classified as unipolar, bipolar, or multipolar according to the number of processes that originate from the cell body. From [Kandel 2000].

- *Active transport*: Ions are pumped in and out of the cell via ionic pumps. For example, the Na^+-K^+ pump, shown on figure 1.6, pumps out three Na^+ ions for every two K^+ ions pumped in, thereby maintaining concentration gradients.

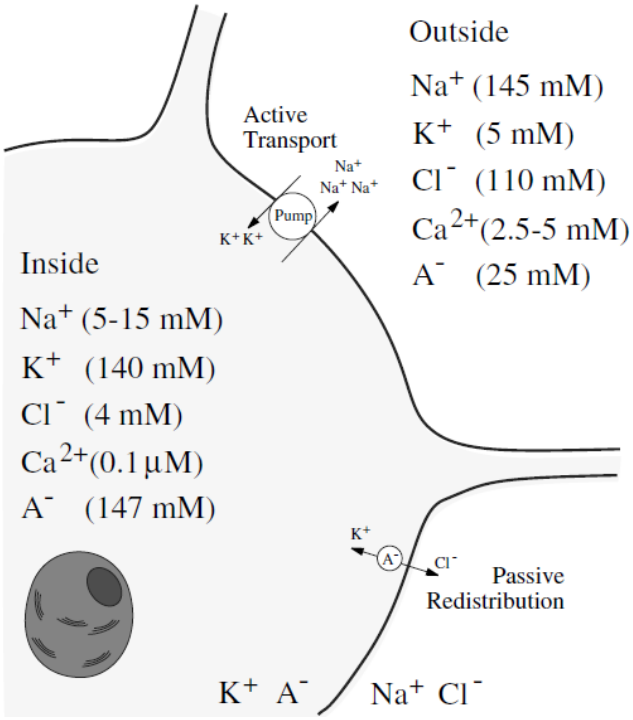


Figure 1.6: Ion concentrations inside and outside of the neuron. From [Izhikevich 2006] where it is taken and modified from [Johnston 1995].

Thanks to these mechanisms, in resting state, the difference of potentials between inside and outside of a neuron is quite stable. It can be different from neuron to neuron, but it is generally around $-70mV$. Maintaining this equilibrium, as we will see later in this section, is a very important property of neurons. Half of the metabolic energy consumed by the human brain is used for moving of ion pumps.

1.1.4.2 Synaptic activation

For the majority of neurons, excitation happens through synapses. As mentioned earlier, each neuron is usually connected with a large number of other neurons through synapses (one synapse makes a connection between two neurons). The strength of the synaptic transmission can be enhanced or reduced over time, depending on the history of the cellular activity. This is called *neural plasticity* and it is crucial for memory and learning. Depending on ions that will be released as a consequence of the synaptic transmission, it can either increase (excitatory) or decrease (inhibitory) the chance of firing the action potential by the postsynaptic neuron. There are generally two basic types of synapses:

- Electrical synapses (figure 1.7 A) where the presynaptic and postsynaptic cell membranes are connected by channels that are capable of passing electrical current, causing voltage changes in the presynaptic cell to induce voltage changes in the postsynaptic cell. The separation between two neurons is very small only around $3.5nm$. The most interesting property of theses synapses is that the transmission is very fast - it lasts only around $0.01ms$. Thus, such type of synapses is found in systems where very fast information processing is essential, such as in escape mechanisms. Another insetting property of electrical synapses is that they are bidirectional, *i.e.* they allow signal transmission in either direction.

- Chemical synapses (figure 1.7 B) are far more numerus than electrical synapses. The separation between two neurons is here much larger and ranges between $20nm$ and $40nm$ (called *synaptic cleft*). The transmission is based on the release of neurotransmitters by a presynaptic neuron. The neurotransmitters bin to the receptors located in the postsynaptic cell, inducing generation of ions. The number and type of the ions depend on the strength and type of the synaptic transmission (excitatory or inhibitory). Chemical synapses take much more time than electrical to transmit a stimulus, from 0.5 to $40ms$

1.1.4.3 Breaking the equilibrium

An action potential is initiated by breaking the equilibrium that keeps the resting potential. The development of an action potential can be observed through 6 phases (figure 1.8):

1. Initiation of the action potential typically happens by injection of extra sodium cations, which is commonly caused be the synaptic activation and brought to the cell membrane through the dendrites. The initial membrane permeability to potassium is low, but much higher than that of other ions, keeping the value of resting potential the same. The depolarization opens both the sodium and potassium channels in the membrane, allowing the ions to flow into and out of the axon, respectively.

2. If the depolarization is small (if the stimuli from the dendrites are not big enough), the outward potassium current overwhelms the inward sodium current and the membrane repolarizes back to its normal resting potential. However, if the depolarization is large enough, the inward sodium

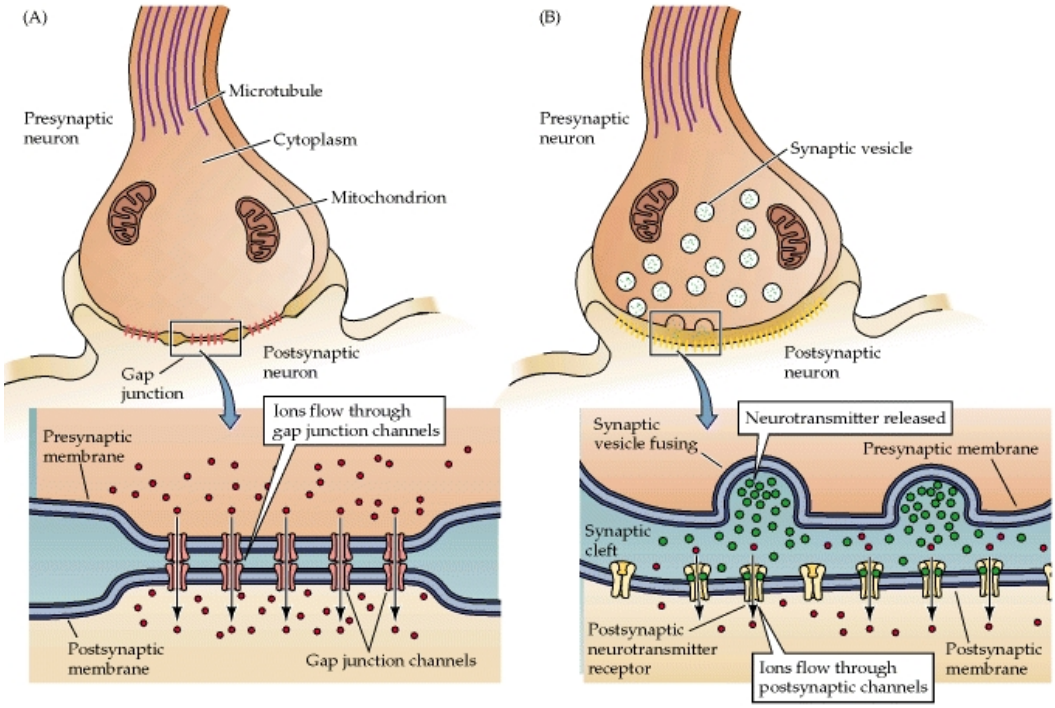


Figure 1.7: Electrical (A) and chemical (B) synapses. From [Purves 2008].

current increases more than the outward potassium current and a positive feedback results: the more inward current there is, the more sodium channels open. This positive feedback continues until the sodium channels are fully open. The critical threshold voltage for this positive feedback is usually around $-45mV$, but it depends on the recent activity of the axon.

3. The positive feedback of the rising phase slows and comes to a halt as the sodium ion channels become maximally open. The same raised voltage that opened the sodium channels initially also slowly shuts them off. The sodium channels become inactivated. This lowers the membrane's permeability to sodium relative to potassium, driving the membrane voltage back towards the resting value. At the same time, the raised voltage starts to open voltage-sensitive potassium channels. Since the potassium channels are much slower to open, the depolarization has time to be completed. Combined, these changes in sodium and potassium permeability cause the potential to drop quickly.

4. With the potassium channels open, the membrane begins to repolarize back toward its rest potential. A membrane that has just fired an action

potential cannot fire another one immediately, since the ion channels have not returned to their usual state. The period during which no new action potential can be fired is called the *absolute refractory period*. It includes phases from 1 to 4 and lasts usually around $1ms$. The refractory period ensures that the action potential travels in only one direction along the axon.

5. The repolarization typically overshoots the resting potential to about $-90mV$. In this phase the axon can be stimulated to produce another action potential, but only with a much stronger depolarization. The period during which action potentials are unusually difficult to provoke is called the *relative refractory period* and lasts about $1 - 2ms$. This phase obviously has a strong importance in information processing and it will be discussed later in this dissertation.

6. After hyperpolarization, the Na^+/K^+ pump eventually brings the membrane back to its resting state of $-70mV$.

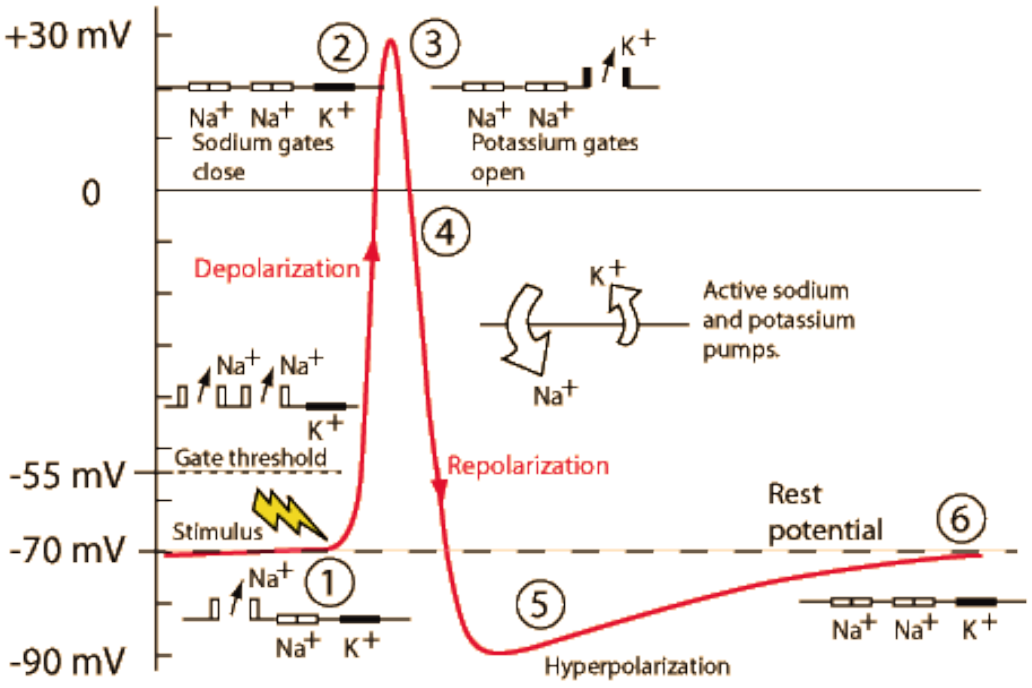


Figure 1.8: Generation of action potential through six phases. From Wikipedia.

1.2 Challenges in understanding of neural communication

The above overview of the nervous system shows that we have quite an extensive knowledge about the anatomy of a nervous system. The anatomical structure is generally well understood on both macroscopic and microscopic scale. On the macroscopic scale we are able to identify the major parts of the nervous system and their functions. On the microscopic scale the general physiology of individual neurons is well understood. Why then we still do not understand exactly how neurons code the information they transmit?

First, we have to notice that understanding of the neural code is currently much better for parts of a nervous system that consist of a small number of neurons. For example, the escape system of a cockroach specie *periplaneta americana* consists of only 8 neurons. The escape system detects an approach of a toad (which is the cockroach's natural enemy) and codes the direction in which the cockroach should escape. Current evidences, based on recordings of the activity of these neurons, suggest that the direction of escape is coded using combined activity from all 8 neurons. Each neuron is associated with a particular angle and its activity can be represented as a vector of that angle and with length proportional to the number of fired action potentials. The direction is determined by a population vector which is a sum of all the 8 individual vectors.

Contrary to this example, understanding of the neural code is not on such level in more complex parts of a nervous system. As we said in the previous section, the cortex, the most interesting part is also the most complicated one. The reason why the coding problem is still unsolved for the cortex seems to raise from the incomplete knowledge on the mesoscopic scale. Even though we are able to divide the cortex in different functional regions and we generally know the anatomy of individual neurons, we are still not able to understand the interactions that are happening among the cortical neurons. With an extremely high density of neurons in a cortex, neural communication is very hard to precisely analyze and model.

To understand why the high density of the neurons makes the problem of decoding much more difficult, we have to analyze some side effects of having such high neural density. Processes of generation of action potentials, the flow of the action potentials through an axon and the synaptic transmission, among else, result in the change of electrical potential in the medium which surrounds neurons - the extracellular fluid. The reason for this interaction with the extracellular medium is the ion leakage which happens during these processes. Such change of the potential in the extracellular medium has a

feedback influence on neurons, affecting their firing patterns. This feedback can only happen when the density of neurons is very high since the extracellular fluctuations decay with distance. We can say that in the cortex the interactions between neurons are not happening only by action potentials, but also by extracellular potentials. This complicates very much the understanding of the neural code, because it is very difficult to understand and model the exact effect of neurons on the extracellular potential, and vice versa.

When talking about the interactions between neurons through the extracellular potentials, we have to raise a question whether these interactions are really only a side effect or a way of communication. In other words, are these extracellular potentials only a stochastic noise, or they exhibit some correlation and actually convey an information? In fact, several recent studies suggested that the second mentioned is correct. This obviously complicates the problem of cracking the neural code.

There are several different aspects on how the code cracking problem could be approached. One approach is by computational modeling of the neural networks (developing an *internal model*). Hypothetically speaking, if the model is a perfect representation of some biological neural network, observing the model behavior could help to have a better understanding of the coding process. The benefit of using the model is the ability to identify a change of the electric potential in any point of the model. For comparison, as we will see later, such extensive ability is currently not possible to obtain from the biological measurements.

Making the computational models that are the exact replica of the particular parts of the cortex is an extremely difficult task, due to the enormous complexity, which does not rise only from the large number of neurons but from their individual differences.

Let us now discuss a little these individual differences. As we said in the previous section, some classifications say that there are more than 1000 anatomically different types of neurons. However, even though the neurons that are considered to be of the same type are in fact not the same. Their anatomy has similar structure, but there are still some significant mutual differences. Some of the differences are direct implication of the natural selection, but some of them occur as a consequence of so-called *transcriptional noise*. The term transcriptional noise refers to the mistakes in interpretation of the genetic code which lead to variations in the protein structures. Manifestation of such noise in a nervous system is visible, for instance, in the different number and distribution of the ion channels among the neurons. So two neurons, even if they are classified in the same class, will usually have different number and distribution of the ion channels. This will lead to, for example, different resting potentials, but more general to the different dynamic properties of the

neurons.

Since neurons are generally mutually different it is currently impossible for the artificial neural networks to make an exact replica of some real neural network. This has a very important implication when we talk about using the artificial neural networks to crack the neural code. The implication is that it is not possible to verify if the artificial model is correct or not. For example, if the artificial neural network is an exact copy of some biological neural network, by sending the same stimuli on the input of the both networks and observing the output, it would be possible to say weather the artificial network is a valid representation of the biological.

Instead of using the computational models to crack the neural code, we can use a more direct approach and try to deduce the code by observing inputs and outputs, either of neural networks or individual neurons. If we assume that we know the input and the output we can test different coding hypothesis (a *black box* model).

To have some realistic idea of what kind of signal we are currently able to record form the nervous system, particulary the cortex, we give in the next section a brief overview of the common recording techniques.

1.3 Neural recordings

The focus in this section will be on recording techniques that measure directly the neural electrical activity. Some other techniques based on indirect recording of the neural activity (electroneurogram (ENG)), e.g. certain metabolic processes also exists. For example, functional magnetic resonance imaging (fMRI) which measures change in the blood flow. As such recordings have generally low temporal resolution, they are not very useful in understanding of the neural code.

The recordings based on direct measurement of the neural electrical activity in the brain can target on recording the activity of a large neuron assemblies from brain/skull surface (surface recordings), smaller populations of neurons (extracellular recordings) or single neurons (usually intracellular recordings).

1.3.1 Surface recordings

Three frequently used devices for such recordings will be described here.

Skull EEG Large neural assemblies are most commonly recorded with electrodes placed on a skull (skull electroencephalography - skull EEG). Since the

technique is non-invasive it is often performed on humans, in order to diagnose or treat (in neurofeedback systems) neurological diseases. Skull EEG is nowadays recorded with systems that have up to 256 electrodes. Since the number of neurons in the human brain is around 100 billion, it is clear that this technique will record some cumulative activity of a very large, synchronously active neural networks. Because the skull has a frequency response of a low-frequency filter, skull EEG can not register frequencies larger than about 80Hz.

Cortical EEG In the cortical EEG the electrodes are placed under a skull, directly on the neural tissue of the cortex. Such procedure is clearly much more difficult to perform. Usually the cortical EEG is measured on humans only when the skull has to be opened due to some surgical procedure. The benefit of the cortical EEG is that the skull filtering is completely eliminated.

Magnetoencephalography - MEG MEG is based on recording of small magnetic fields produced by electrical currents occurring naturally in the brain, using arrays of SQUIDs (superconducting quantum interference devices). Magnetic fields are less distorted than electric fields by the skull and scalp, which results in a better spatial resolution of the MEG than skull EEG. Thus MEG is typically used to localize large synchronously active neural networks. Since magnetic fields are much more difficult to register than electrical, the MEG devices are much more sophisticated and more expensive than EEG devices.

With any of such recordings we are obviously not able to isolate the individual action potentials, but only to see their combined activity. Even though they might not directly help in cracking the neural code, these recordings can give an interesting insight on how large populations of neurons generally react on different stimuli. To illustrate this with an example, let us mention a common experimental paradigm performed during surface recordings, especially the skull EEG. The paradigm consist of an iterative stimulation with light during the recording. In each iteration, similarly shaped patterns can be observed from certain regions of the cortex, particulary from the occipital region where the center for processing of visual stimuli appears to be. So even though in this particular case we do not know how the stimuli are actually coded, we do know what kind of response on a large population level we can expect. This has already been used in applications such as brain-computer interface (BCI), where brain recordings are used to enable an examinee to control a computer, *e.g.* through a virtual keyboard. From the biological point of view, the general

concept of BCI is not difficult to explain roughly. A light stimuli, converted into an electrical current by the retina, is carried through an optical nerve to the occipital region of the cortex. There, the neurons in charge for processing the visual stimuli are excited and they start to fire action potentials. This results by synchronous increase of the activity in the occipital region which is then visible on the recording. So even though we do not understand the coding process itself, we know that delivering of stimuli will result in increase of the firing rate (number of action potentials per a time unit) for some neurons. So to understand the coding process we have to understand what is the exact correlation between the particular stimuli and the spike rate and timings. To do so, we need the recordings and we need to isolate individual neural activity.

1.3.2 Recordings on a single neuron

To obtain recordings of individual neural activity we need to record directly from the inside of the brain. An example of such recordings are recordings on a single neuron.

Recordings on a single neuron have generally very good signal to noise ratio (SNR), the activity of the recorded neuron is usually obvious from such recordings. In the sequel we first briefly describe the recordings and then analyze if they can help in understanding of the neural code.

Single neuron recordings are typically performed intracellularly. An intracellular recording involves measuring voltage and/or current across the membrane of a cell. To do an intracellular recording, the tip of a fine (sharp) microelectrode must be inserted inside the cell, so that the membrane potential can be measured. Most intracellular electrodes are glass micropipettes, with a tip diameter lower than $1\mu m$, and a resistance of several $M\Omega$. The micropipettes are filled with a solution that has a similar ionic composition as the intracellular fluid of the cell. A chlorided silver wire inserted in the pipet connects the electrolyte electrically to the amplifier. The voltage measured by the electrode is compared to the voltage of a reference electrode, usually a silver chloride wire in contact with the extracellular fluid around the cell. In general, the smaller the electrode tip, the higher its electrical resistance, so an intracellular electrode is a compromise between size (small enough to penetrate a single cell with a minimum damage of the cell) and resistance (low enough so that small neural signals can be discerned from thermal noise in the electrode tip).

Another technique called *patch clamp* is also popular. It is similar to the one described above, but instead of inserting the microelectrode inside the cell here it is sealed onto the surface of the cell membrane.

An experimental paradigm that could help in understanding of the neural

code would be to simultaneously record a large number of neurons, while they are stimulated with repeated and controlled stimuli. Here we face the main drawback of intracellular recordings: difficultly of simultaneous performing such recordings on more than one cell. This technical limitation raises from the complex recording setup, which makes the intracellular recordings very difficult to perform, even on a single neuron.

Notice also that intracellular recordings are generally very difficult to perform on moving animals, especially for long time. This significantly limits their applicability in different neural prothesis, which will be discussed later in this chapter.

1.3.3 Extracellular recordings

Intracellular recordings give a reliable insight in a single neuron activity, thus they can be useful when neural coding problem is addressed. However, we will now consider extracellular recordings that could lead to a successful record of activity of more than one neuron.

Extracellular recordings are performed by placing the electrode in the medium around the neurons - as close as possible to the bodies of the neurons, but without touching them. They measure difference of the electric potentials between the electrode and some reference point which is considered to have a stable potential. An extracellular electrode can have just one or more recording sites. The distance between the neighboring recording sites is usually similar as the mutual distance between the recorded neurons. For example, in [Mathieson 2004] an extracellular recording system with 512 electrodes is described. The diameter of each recording site is $5\mu m$, and the distance between two neighboring sites is $60\mu m$.

With extracellular recordings we obtain, on each recording site, the simultaneous activity from a large number of neurons within a small part of the neural tissue. We can say that each recording site records a mixture of the projections of the activity from a vast number of the neurons around. The amplitude and the shape of a projection depend primarily on the distance between the corresponding neuron and the recording site, but also on the morphology of the neuron and on the electrical properties of the medium between the neuron and the recording site. Even though subthreshold fluctuations are part of activity of any neuron (when stimuli are not large enough to initiate an action potential), the most important neural activity is characterized by action potentials, due to their large amplitude. An example that demonstrates the change in amplitude and shape of an action potential recorded at four different places, all very close to the neuron, is shown of figure 1.9. An action potential recorded with an extracellular electrode is usually called

spike. Due to the imperfection of the amplifiers and filters, but also due to the quantization effects, the measurement process introduces a noise.

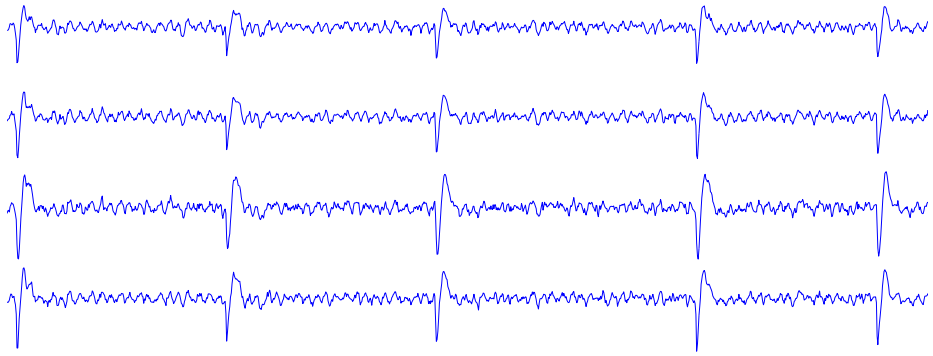


Figure 1.9: Simultaneous recording of extracellular spike waveforms shows how amplitude and shape of action potentials varies with distance from the neuron.

Generally speaking, extracellular and surface recordings both record some cumulative activity of a large number of neurons. An important advantage of extracellular recordings is that the electrodes can be placed at an arbitrary location. This is very important because it makes possible to conduct an experiment in which we can stimulate a part of a nervous system and during the stimulation record extracellularly from that region. In such way we can obtain a much more direct relationship between the stimuli and the response, which is in this case a sum of the activity of a large number of neurons.

Obtaining the response to stimuli is not enough to understand the neural code. As we discussed in the previous section, neuroscientists generally agree that the principal way of transmitting an information within a nervous system is by action potentials (spikes). Thus to address the neural coding problem on the lowest level we need some information on the individual neural activity. Now we come to a very important question which has been the subject of many neuroscience studies: how from an extracellularly recorded noisy mixture of neural activity obtain the activity of individual neurons that can be used to investigate the neural code? This question will be discussed in the next two sections, but also in the next three chapters.

1.4 Processing of extracellular recordings

Extracellular recordings consist of projection of activity from a large number of the neurons around the electrode and some additive noise introduced by the recording process itself. To make such recording useful for understanding

of the neural code we want to find the activity of the individual neurons. Activity of a single neuron consists of subthreshold fluctuations and action potentials. Even though the subthreshold fluctuations could have a role in the information transmission by influencing the extracellular medium, action potentials are the main element of the neural communication. Thus, activity of single neurons is usually simplified, by saying that a neuron is either firing an action potential or it is in the resting state.

While traveling down the axon all the action potentials have almost the same shape and amplitude, regardless of neuron which fired them. So we can say that no information is conveyed either in the shape or in the amplitude, but only in the time instant when an action potential is fired. This makes the problem of isolation of individual neural activity much simpler, since all we need to find, for each neuron which additivity we want to isolate, are the time instants when the action potentials occur. The process of finding these time instants is generally called *spike detection* and it will be widely discussed in the following chapter of this thesis. After the spikes are detected, in order to access the individual activity of the neurons that fired the detected spikes we have to assign each of the spikes to the corresponding neuron (the one that fired the spike). This process in called *spike sorting* and it will be the topic of the third and the fourth chapter of the thesis.

One of the main questions that naturally raises when talking about spike detection is: from which neurons do we want to detect the spikes? To answer this question we have to bare in mind that the main purpose of the spike detection is to obtain spike locations in order to do the spike sorting. As we will see later, the spike sorting is based on the fact that the amplitude and the shape of the recorded spikes are different when the spikes come from different neurons. In extracellular recordings, due to the simultaneous activity of a large number of neurons and presence of the noise it is hard to differentiate the spikes fired by the same neurons from the rest of the spikes. Generally, in spike detection, we are interested only in the detection of the spikes which shape is relatively well preserved, so it will be possible to sort them. Since the amplitude of the projection of neuron activity depends dominantly on the distance between the neuron and the electrode, neurons which activity will be easiest to find are obviously the one located close to the electrode. The precise number of these neurons or their maximal distance from the electrode can not be explicitly given, since they depend on the number and the activity of the more distant neurons as well as on the noise level. The spike detection task is thus obviously not defined very precisely.

When only a handful of neurons are very close to the electrode, their spikes are distinguished very clearly from the firing activities of the distant neurons, which we consider as a background noise. In such situations, spike

detection can be performed by simply setting a threshold. Unfortunately, these situations do not always prevail and a simple threshold is not always a good solution. In the following chapter, we will address the spike detection problem in a more realistic setting, when the separation of spike and noise is not always obvious.

In spike detection we are only interested in finding time instances when spikes occur: even though a spike lasts about $1ms$, we can represent it by only one time instance. Moreover, we can say that a spike, an abrupt change in its nature, is an irregularity in otherwise smooth signal. Even though this is not empirically correct, it in spirit describes well the nature of spikes and makes the problem of their detection easier. In fact, it makes the spike detection problem equivalent to a change point detection problem. One popular way to solve a change point detection problem is by using derivative estimators, either in explicit or implicit form.

The spike time occurrences appear (as irregularities) explicitly in the distributional derivatives of the neural signal. The problem is seen as a change point detection problem. Using operational calculus, which provides a convenient framework to handle such distributional derivatives, we characterize the time occurrence of a spike by an explicit formula. We derive from this characterization, a joint change point detection and localization system, implemented using a digital Volterra filter. Notice that the proposed approach is not limited only on detection of neural spikes, but results in general purpose change point detection algorithm. One demonstration of this generality is shown by successful application of the method on EEG signal in order to detect artifacts caused by eye-blinks.

Assuming that we have detected the spikes, the next mandatory step in the analysis of the neural signal is to separate the firing activities of the different neurons. This is the so-called spike sorting problem. We address this problem in chapters three and four.

As we mentioned earlier, spike sorting exploits the fact that, when recorded extracellularly, action potentials from different neurons have different shapes and amplitudes. After spike detection each spike is usually represented by few features, *e.g.* the positive and the negative peak amplitude. This is done to facilitate further computing by having a reduced-dimension signal, but also to emphasis some features considered to be discriminative among the spikes fired by the different neurons. When projected in the feature vector space, spikes that belong to the same neuron are usually non-uniformly distributed around the same center. An important problem in spike sorting is that we do not know a priori how many neurons did fire the spikes that we have detected; so we do not know how many centers the feature vector space has. However, assuming that neurons fire with some given minimal firing rate, using simple

spacial filtering, we emphasis and detect the cluster centers. We determine the borders between the clusters by finding some minimal spike density in the feature vector space. This approach generally leads to a good detection results when spike features are significantly different. If this is not the case, the clusters can be overlapped and possibly merged. The complete approach and the implementation are described in chapter three.

Improved results of spike sorting can be obtained when multi-site extracellular recordings are available and when the two following criteria are satisfied:

- The recordings sites are close enough, so the projections of action potentials from a neuron close to one electrode are visible (amplitude higher than the value represented by the lowest bit of the amplitude resolution) on the other recording site(s).

- The recordings sites are far enough, so the projections of the action potentials from a neuron close to one electrode is different (either by shape, amplitude or the both) on the other recording site(s).

If this is true, we demonstrate in chapter four, that by applying independent component analysis (ICA; a well known blind source separation technique) on such a recording we can obtain a signal which is more suitable for spike sorting techniques (particulary the one proposed in chapter three) than it would be the original recording.

Notice that since ICA is a technique for separation from a mixture of mutually independent sources, it is not perfectly suited for neural mixtures as the neural activity is not mutually independent. However, the neural activity is generally less dependent than the activity on the recordings sites. Even though ICA will not lead to the perfect isolation, it will make the signal easier to separate with the conventional spike sorting techniques.

Another problem with applying ICA in neural recordings is that ICA is designed to separate independent activity from such mixtures where the number of the recording sites is greater or equal to the number of sources (neurons in the mixture). This is obviously not true in neural recordings. The algorithm proposed in chapter four addresses this problem. The algorithm uses an iterative application of ICA and a deflation technique in two nested loops. In each iteration of the external loop, the spiking activity of one neuron is singled out and then deflated from the recordings. The internal loop implements a sequence of ICA and spike detection for removing the noise and all the spikes that are not coming from the targeted neuron. We show in chapter four that with such algorithm we can obtain much better sorting results than by applying only ICA.

1.5 Using extracellular recordings for understanding of spike coding

We are nowadays able to record neural activity with multi-site extracellular electrodes. In this section we will discuss how such recordings could be used to understand the neural code.

What neuroscientists has already noticed is some relation between the stimuli strength and the number of emitted spikes by the recorded neurons. Generally, stronger stimulus results in more generated spike. For a long time this was considered to be the only coding scheme of a nervous system (called *rate code*), but as we will see later, more recent research indicated that some other coding schemes are likely to be involved in the coding process, together with the rate code.

Benefits of better understanding of the neural code are enormous in many different research fields. They are even hard to imagine and often reach in what is today considered as science fiction. Neuroprosthetics, pharmacology, artificial intelligence and research of consciousness are just some of the examples. We will in chapter five particulary discuss possible influence of some recent findings related with neural code on development of artificial intelligence. However, even this rough knowledge about the proportionality of the stimuli strength and the number of the spikes has been widely exploited in several different applications among all the research fields mentioned above. We will try to illustrate now, on two different application (brain computer interfaces and deep brain stimulation), how the current knowledge is already practically exploited, but also how better understanding of the neural code can be very beneficial.

1.5.1 Brain computer interface - BCI

BCI is a direct communication pathway between the brain and an external device. BCIs are usually aimed at assisting, augmenting or repairing human cognitive or sensory-motor functions.

BCI is typically realized non-invasively, using skull EEG or MEG, but much more accurate results can be obtained with invasive BCI, where an electrode is implanted in the brain of a patient. The electrode is used for extracellular neural recording, and contains a large number of recording sites, *e.g.* 500. It is usually implanted in the part of the cortex where most of the motor tasks are processed. Analysis of the number of the spikes per time unit for each recording site gives some rough estimation of what movement the patient wants to do. Depending on the application, the electrode is connected with some artificial mechanic/computer devices. Such BCI can be used for

example to type and move computer mouse or to control wheelchairs. A schematic diagram of the standard invasive BCI system is given on figure 1.10.

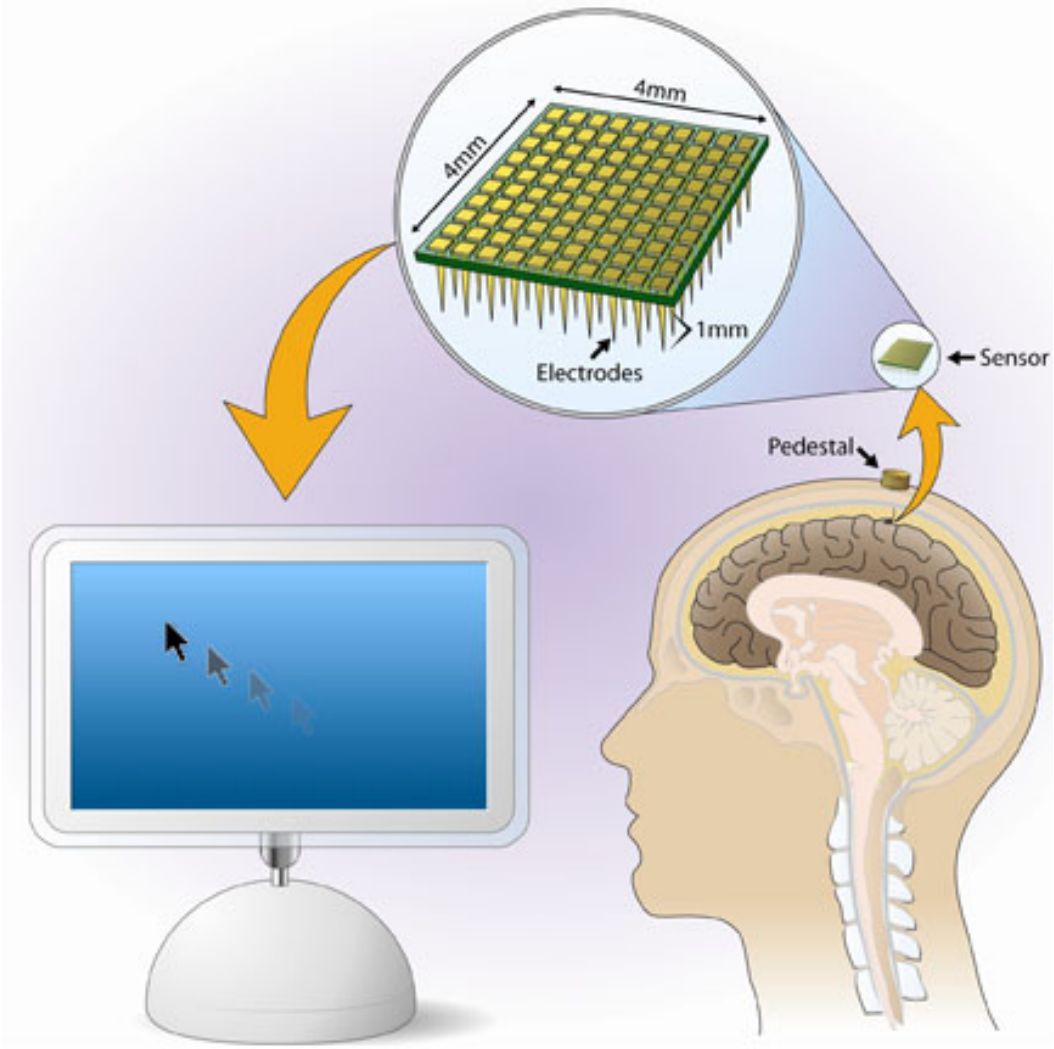


Figure 1.10: Schematic diagram of BCI system. From [Simeral 2011]

A better understanding of the neural code and generally a more efficient processing of the recorded signals (more neurons isolated from the mixture) are the most important challenges for the construction of more accurate BCI devices.

1.5.2 Deep brain stimulation - DBS

DBS is a surgical procedure used to treat a variety of disabling neurological symptoms such as the debilitating symptoms of Parkinson's disease, *e.g.* tremor, rigidity, stiffness, slowed movement, and walking problems. DBS uses a surgically implanted, battery-operated medical device called a *neurostimulator* or *brain pacemaker*. This device delivers electrical stimulation to the targeted areas in a brain. A brain areas where DBS is commonly applied are those that control movement or block the abnormal nerve signals that cause tremor and Parkinson's disease symptoms. This electrical stimulation consists of impulses which duration and frequency are usually tuned experimentally, observing the individual feedback from the patient. A schematic diagram of the DBS circuit is given on figure 1.11.

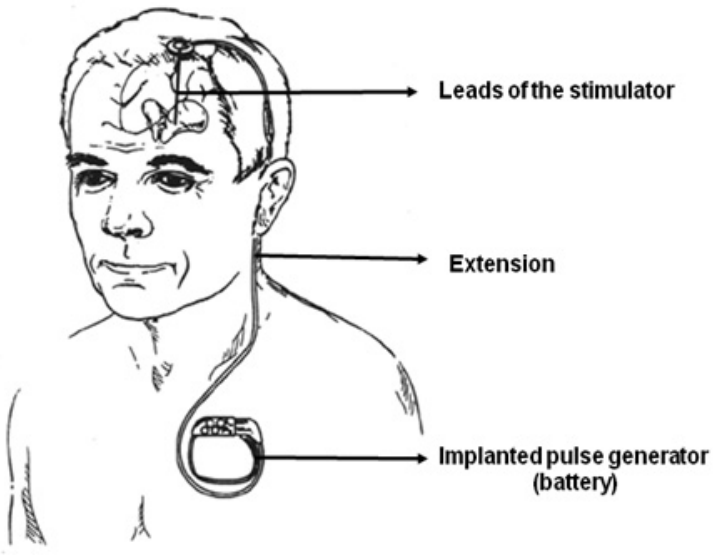


Figure 1.11: Schematic diagram of DBS system. Includes battery, connecting cable and stimulating electrodes. From Wikipedia.

Before starting with DBS, magnetic resonance imaging, computed tomography scanning and extracellular recordings are usually used to identify and locate the exact target within the brain where electrical nerve signals generate the undesirable symptoms.

Even though the results of DBS are very promising, the stimulation process in not very precise. Time structure of the delivered impulses is not estimated on some knowledge on the neural code, but rather using experience and some feedback information. Thus, how exactly DBS actually works is completely unknown. Any progress towards the challenging question of understanding the neural code would have a significant implications on DBS. For instance,

it could be performed with a large number of stimulating sites, each of which could take a role of one or a few neurons.

1.5.3 Coding schemes

The rate coding scheme was considered as the only one until it was observed that some parts of the nervous system are able to process the stimuli so fast that only one or two spikes per neuron could be fired. An example is, as already mentioned, the cockroach escape system. It has been noticed that the stimuli in that system are coded by simultaneous activity of population of neurons, what is referred as *population code*. Later on, a high correlation is observed between some stimulus property (*e.g.* intensity) and the time elapsed from the stimulus onset until the first spike after the stimulus was fired by the recorded sensory neuron; such way of representing information is called *latency code*. Moreover, a correlation between the type of the stimulus and elapsed time between two spikes, fired either by the same neuron or by different neurons, is observed as well. This coding scheme is called *interspike-interval code*. Each of these coding schemes are described in the fifth chapter; concrete examples are used to support each coding scheme.

As we mentioned earlier, it is still not clear whether the interactions between neurons through extracellular medium convey any information. By analyzing extracellular recordings, when we neglect spikes from nearby neurons, we can notice relatively high energy in low frequency oscillations (mostly below 100Hz), with some significant spatial coherence, these are called *local field potentials*. Phase of these oscillations is sometimes noticed to correlate with neural spiking activity. Some recent studies suggest that sometimes firing with respect to phase of the oscillations conveys more information about some given stimuli than the firing rate.

Inspired by the nature in general, where information is commonly present in a discrete form (from quantum of light and matter to genetic code for example), we investigate whether the nature of the neural code is discrete or continuous. Moreover, if it is discrete, whether the elements of the code are drawn from a finite alphabet. We particulary address pulse-position coding scheme, making a link between communication theory and neural code. Such coding scheme would permit that, despite the time jitter which is present in neural responses, an information about the stimulus can be extracted from a single neuron. Conformation that the neural code is based on finite alphabet would change drastically the current perspective of neural information processing and would set a completely new pathway in several research fields related with neural information processing. Even though we do not give an explicit answer to these questions, we suggest a possibility that on different

time scales and in different parts of a nervous system several coding schemes might coexist and that nature of the neural code might be dual. As one very interesting implication of the case that neural code is discrete and, more generally, that neural information is not transmitted only by action potential but as well with local field potentials we discuss influence on projects such as *Blue brain*, where the brain activity is modeled regardless of the extracellular potentials.

The thesis are finally concluded by the chapter six, which brings some final discussion and conclusion.

Chapter 2

Spike detection

Contents

When a neuron receives a sufficiently large stimulus, which exceeds some threshold level, it emits an electrical discharge called *action potential*. The action potential travels through the axon and delivers the stimulus to other neurons. It is a known fact that neurons use action potentials to communicate between themselves [Rieke 1997].

One common way of recording action potentials is with extracellular electrodes. Such type of recording can be performed for long time periods on moving animals, without destroying the cells being recorded. An action potential recorded extracellularly is typically called *spike* and a sequence of spikes from a single neuron is usually named *spike train*.

Extracellular electrodes record projections of activities from a large number of neurons. From extracellularly recorded signal it is usually possible to localize spikes fired only be a few neurons closest to the electrode. Finding locations of spikes fired by the neurons around the electrode is commonly referred as spike detection and it is in the focus of this chapter. Spike detection is obviously not a very precise task, since the number of neurons that can be sorted is not priori given. The number depends on spatial distribution of neurons with respect to the electrode, what is different in every recording. Also, the task is rather difficult due to a commonly large cumulative activity of distant neurons, which is often hard to distinguish from spiking activity of neurons that are close to the electrode.

Spike detection is frequently followed by spike sorting, in which detected spikes are grouped into clusters in such way that spikes fired by the same neuron belong to the same cluster (see [Delescluse 2005], [Roa 2007] and also [?] for a review).

Additionally, the results of spike detection can be used directly for applications such as brain-computer interface. Here as well the number of neurons which spikes should be detected is not precisely defined.

The picture[1] in figure 2.1 illustrates a spike train extracellular recording experiment, from the nervous olfactory system of the locust. Each of the two

Figure 2.1: Recording from the locust antennal lobe, from [Pouzat 2005]

shanks of the intracranial probe carry two groups of four recording sites. The recording sites are the bright spots and the side length of each is $13\mu m$. Each group of four recording sites is called a *tetrode*. The probe is next pushed gently into the antennal lobe in order to record the action potentials of the surrounding neurons. Figure 2.2 shows 1s data from a single tetrode, sampled with 15kHz sampling frequency and bandpass filtered between 300Hz and 5kHz. More details on these records are shown in figure 2.3, representing a zoom on a segment of $y_1(t)$ and $y_4(t)$, of $40ms$ duration from the point τ.
A comparison of the data from different sites shows that the amplitudes of the recorded action potentials vary markedly with the distance of the neuron from the recording site (compare b_1 and b_4 in figure 2.3). In general, such attenuations also vary in time. The background additive noise is thus composed essentially of action potentials from remote neurons [Gerstner 2002]. The zoom in figure 2.3 also exhibits different action potential waveforms, corresponding to different neurons [Delescluse 2006], [Nenadic 2005] (compare a_1–a_4, b_4 and d_1–d_4). Notice that the given is an example of signal with a very good signal

[1]This picture is reproduced from [Pouzat 2005] which we refer for more details on the recording setting.

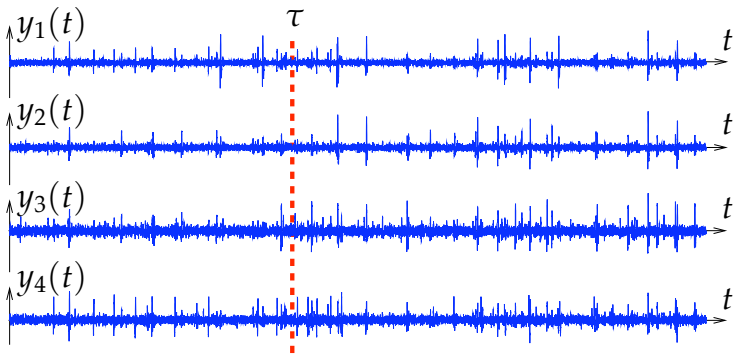


Figure 2.2: One second signal recorded from the locust antennal lobe, filtered in [300Hz, 5kHz]

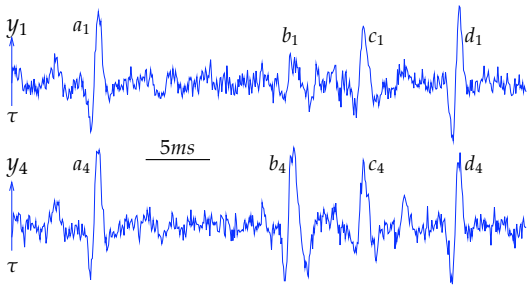


Figure 2.3: Signal recorded from the locust antennal lobe: zoom on a segment of $40ms$ duration

to noise ratio (SNR), since the amplitude of the spikes that obviously come from neurons close to the electrodes is very large in comparison to the rest of the signal, which we consider as noise.

For the recordings where the SNR is relatively high, satisfactory results of spike detection are possible to obtain simply by setting a threshold [Quiroga 2004]. Such example is signal shown on the bottom plot of figure 2.3. For the top plot, the spike $b1$ does not have some large amplitude, thus finding an appropriate threshold level would be much more difficult in this example. Another example, even more challenging for spike detection, is given on figure 2.4. The second plot shows the extracellular recording and the first plot shows the simultaneous intracellular recording of one of the neurons close to the extracellular electrode [Henze 2000]. Such type of recording is a great insight on structure of an extracellular signal. The action potentials fired by the neuron recorded intracellularly are visible in the extracellular recording, but it is obvious that their detection by placing a threshold would not be very convenient, because the noise level is too high. For such recordings, with relatively low SNR, using only spike amplitude for the detection does not lead to good

results. Some other properties of spikes should be also exploited in order to make a clear difference between the spikes fired by the closest neurons and the noise.

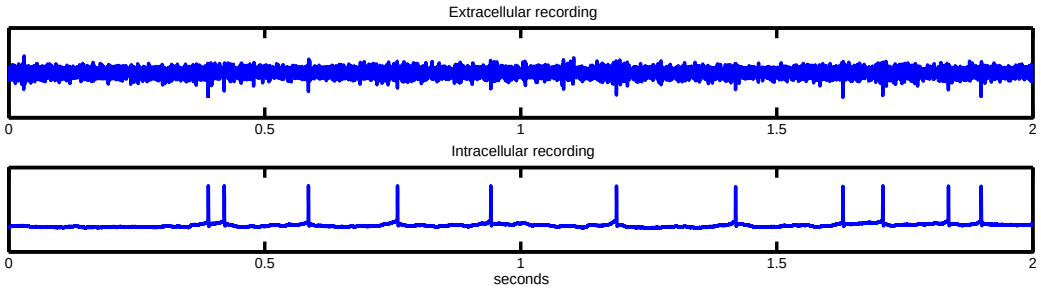


Figure 2.4: Simultaneous intra- extra-cellular recording.

A spike is the result of a sudden polarity inversion of the electrical charges across the neural membrane. Even though the spike waveforms are different, they always contain an abrupt change of electric potential. Even better example of such abrupt change is given on figure 2.5 where a zoom of the signal shown on figure 2.4 is given. On figure 2.5 we can see that both, intracellularly and extracellularly recorded action potentials do contain an abrupt change. Later in this chapter a physical model of such changes will be analyzed more precisely. Apart from the amplitude, this abrupt change of potential seems to be a property which discriminates spikes from the rest of the neural signal. In this chapter we will use this property to find the spikes within a noisy environment.

We represent such an abrupt change as an irregularity in the neural current. And since we do not require to distinguish between the spiking activities of different neurons (we do not address the spike sorting problem for the moment), the precise waveform of an action potential is not so relevant: the irregularities are our only concern. We therefore consider a piecewise regular model

$$y(t) = \sum_{i \geqslant 0} H(t - t_i)\, f_i(t - t_i) + \varpi(t), \text{ with } t_0 = 0, \tag{2.1}$$

where a change from f_{i-1} to f_i represents the occurrence of a spike at time t_i. $\varpi(t)$ is the background noise, composed mostly of activity from neurons which are far enough from the electrode so we are not interested in detection. Each $f_i(t)$ is a smooth segment and the unit step (Heaviside) $H(\cdot)$ is used for causality purpose. Based on the observation $y(t)$, we want to detect the spikes and estimate their locations t_i. This is a *change-point* detection and estimation problem. A large amount of literature is devoted to the problem,

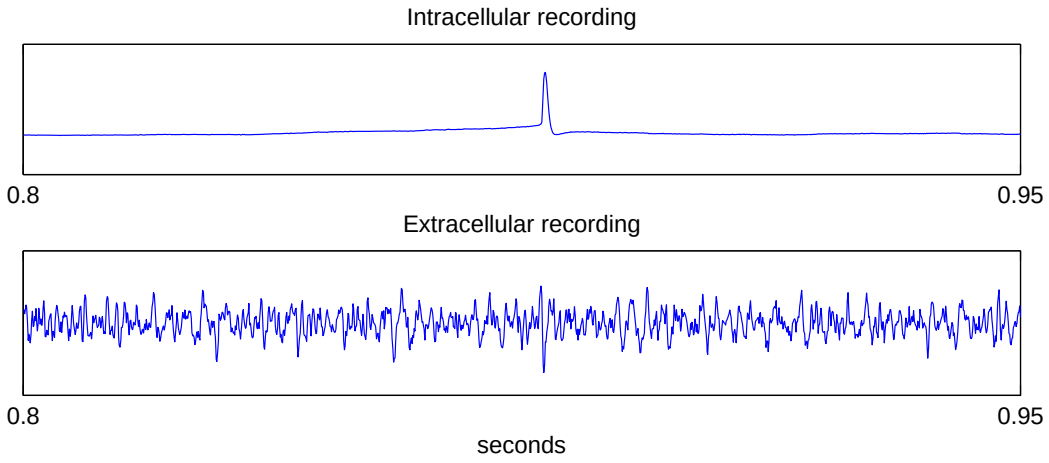


Figure 2.5: Simultaneous intra- extra-cellular recording; zoom of part of figure 2.4

in its general form. The reader is referred to the monograph of M. Basseville and V. Nikiforov [Basseville 1993] for a nice tutorial presentation and many application examples. When a statistical model description of the observation data is available, then it is very common to approach the problem via hypothesis testing. The classical cumulative sum statistic [Basseville 1993] (see [Vaswani 2007] for other statistics) is essential in most of the algorithms devised within this framework. Model selection is another common approach and therein, the Bayesian theory plays an important role. An example using penalized contrast may be found in [Lavielle 2005] (see also [Fan 2006]). As opposed to the test statistic based methods which are sequential, the ones developed in the model selection framework are rather global: the change-points are estimated simultaneously. Clearly, this is not suited for online implementation.

We may also mention the kernel based [Gijbels 1999], [Desobry 2005] and other more specific spike detection approaches such as those using morphological filters [Xu 2007] and the method using a nonlinear energy operator (NEO) [Kim 2000].

Beside the large variety of approaches, we note that wavelet decomposition is a recurrent tool for detecting irregularities in a signal (see *e.g.* [Vaswani 2007]). This is due to its ability to isolate them. The reason of this ability is that a wavelet transform Wf with n vanishing moments is a differential operator [Mallat 1999]. More precisely, if f is continuously differentiable up to order n in the vicinity of t then, the wavelet transform satisfies [Mallat 1999, eq.

(6.15)]

$$\lim_{\alpha \to 0} \frac{Wf(\alpha, t)}{\alpha^{n+1/2}} = f^{(n)}(t),$$

where the superscript (n) denotes order n differentiation.

The spike detection technique that will be presented in this chapter is close in spirit to this idea. The main advantage of the proposed approach is that we deal with a change-point (spike) location explicitly, finally obtaining the explicit solution for the time instants of a spike occurrence (explicit solution for each t_i). The approach is not limited to neural spike detection, but it can be considered as a general change-point detection and localization framework.

The main arguments of the approach are elaborated in the following section. They rely on the original continuous-time nature of the spike train as with wavelet based approaches, although the available observation data are in discrete time form. In particular, the problem is casted into a delay estimation. Section 2.2 describes the different steps leading to the detection and estimation algorithm. Given a time interval, a possible spiking instant t_r will be characterized explicitly from the distributional derivative of the signal. We then take advantage of the algebraic framework of [Fliess 2003] (see also [Mboup 2009a] and [Fliess 2010]) to handle such expression. This algebraic framework was adopted in [Rezk 2008] for change point detection in biological signals (electroencephalogram (EEG) and electrocardiogram (ECG)), but in the frequency domain and also in [Mboup 2009b] for numerical differentiation (see also [Liu 2011]). This leads to an estimator of t_r from a linear system, based only on iterated integrals of the spike train in the corresponding interval. However, if no spike occurs in the considered time interval, then the linear system becomes degenerated. Now a test of whether the linear system is degenerated or not through sliding intervals provide us with a joint spike detection and localization method. The obtained solution is then implemented in discrete time, in terms of very classical finite impulse response (FIR) filters. The decision functions are expressed as outputs of classical Volterra filters. Results of simulation on real and synthesized recordings are given in section 2.3. The approach is compared with the wavelet based method of [Nenadic 2005]. That the proposed method is not limited only on detection of neural spikes is shown in section 2.4, where it is successfully applied for detection of the spike shaped eye blink artifacts in EEG signal. Finally, the results are discussed in section 2.5.

2.1 Main ideas

We denote $x(t)$ as the unobserved noise-free spike train signal: $x(t) = y(t) - \varpi(t)$. Let $T > 0$ be given. To any $\tau \geqslant 0$, we associate the interval $I_\tau^T = [\tau, \tau + T)$. The detection of an irregularity in such interval will indicate the presence of a spike. Although it is possible that two or more neurons which are located close to the electrode fire an action potential at very close time instants, we assume that the interval I_τ^T does not contain more than one spike. Respecting the duration of a spike (not only the part we model as a discontinuity), by choosing T small enough we can make sure that if I_τ^T contains more than one spike they are overlapped for at least some given time. As spikes are after detection typically extracted in order to perform some additional processing *e.g.* spike sorting, for significantly overlapped spikes it is sufficient to detect only one spike; since the other will be automatically extracted as well. Therefore and in order to obtain a simple model we say that any I_τ^T can contain either only one irregularity or be smooth.

In the sequel, we define

$$x_\tau(t) \stackrel{\triangle}{=} H(t)x(t+\tau), \quad t \in [0, T), \tag{2.2}$$

for the restriction of the signal x in I_τ^T and we redefine the possible spiking instant, say t_τ, relatively to I_τ^T with:

$$\begin{cases} t_\tau = 0 & \text{if } x_\tau(t) \text{ is smooth} \\ 0 < t_\tau < T & \text{otherwise.} \end{cases}$$

The presence of a spike is thus interpreted as a discontinuity of $x_\tau(t)$, at an interior point of I_τ^T: an irregularity at the origin of the current interval, caused by the multiplication with the Heaviside function, is not considered as spike.

To illustrate the main ideas behind the proposed approach, let us consider an interval I_τ^T which contains a signal with a jump at time t_1 (figure 2.6). Such input signal on I_τ^T can be described as follow:

$$x_\tau(t) = a(t)H(t - t_0) + (b(t) - a(t))\, H(t - t_1), \tag{2.3}$$

where t_0 is the origin of I_τ^T. $H(t-t_0)$ is introduced to describe the irregularity at the origin of I_τ^T, caused by limiting the signal on that interval. t_1 is the location of the discontinuity that we want to find. Denote $\beta(t) = b(t) - a(t)$. The first order derivative of $x_\tau(t)$ reads as:

$$\dot{x}_\tau(t) = \dot{a}(t)H(t - t_0) + a(t)\delta(t - t_0) + \dot{\beta}(t)H(t - t_1) + \beta(t)\delta(t - t_1). \tag{2.4}$$

Now we have t_1 within delta function, not only within Heaviside function; this will help us to obtain an explicit solution of t_1.

Using the relation $\alpha(t)\delta(t-\lambda) = \alpha(\lambda)\delta(t-\lambda)$ in equation (2.4) we obtain:

$$\dot{x}_\tau(t) = \dot{a}(t)H(t-t_0) + a(t_0)\delta(t-t_0) + \dot{\beta}(t)H(t-t_1) + \beta(t_1)\delta(t-t_1). \quad (2.5)$$

Denote $[\dot{x}_\tau](t) = \dot{a}(t)H(t-t_0) + \dot{\beta}(t)H(t-t_1)$, what is the regular part of $\dot{x}_\tau(t)$ and insert it in equation (2.5):

$$\dot{x}_\tau(t) = [\dot{x}_\tau](t) + a(t_0)\delta(t-t_0) + \beta(t_1)\delta(t-t_1). \quad (2.6)$$

So all we have in equation (2.6) is a superposition of derivative of the regular part and the jumps. It is important to notice that, when we have a jump at t_1, $\beta(t_1)$, which is actually the amplitude of the jump in the signal at location t_1, will be different from zero. If there was no jump at t_1, $\beta(t_1)$ would be equal to zero.

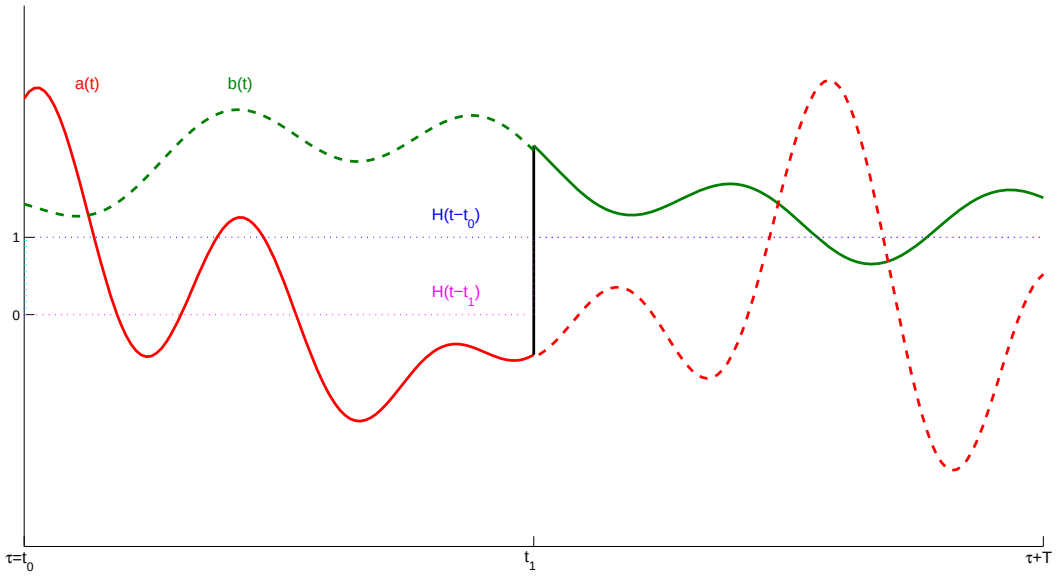


Figure 2.6: Input signal given as a piecewise polynomial on interval I_τ^T with jump in t_1 (the fat line). The signal expressed in equation (2.3) through functions $a(t)$, $b(t)$, $H(t-t_0)$ and $H(t-t_1)$. The fat black line at t_1 represents the jump.

Let us now analyze a bit different situation, where the signal on interval I_τ^T contains a jump in its first derivative, but not in the signal itself (figure 2.7). The first derivative of such signal can be again described with equation

(2.5), but now $\beta(t_1) = 0$, since there is not jump in the signal. To obtain t_1 within the delta function we need to take another derivative:

$$\ddot{x}_\tau(t) = \ddot{a}(t)H(t-t_0)+2\dot{a}(t)\delta(t-t_0)+a(t_0)\dot{\delta}(t-t_0)+\ddot{\beta}(t)H(t-t_1)+\dot{\beta}(t_1)\delta(t-t_1). \tag{2.7}$$

By denoting $[\ddot{x}]_\tau(t) = \ddot{a}(t)H(t-t_0) + \ddot{\beta}(t)H(t-t_1)$ we can express equation (2.7) as a sum of the regular part and the discontinuities.

$$\ddot{x}_\tau(t) = [\ddot{x}]_\tau(t) + 2\dot{a}(t)\delta(t-t_0) + a(t_0)\dot{\delta}(t-t_0) + \dot{\beta}(t_1)\delta(t-t_1). \tag{2.8}$$

Now $\dot{\beta}(t_1)$, the jump in the first order derivative of the signal at location t_1 within the interval I_τ^T is different from zero. Notice also that $\ddot{\beta}(t_1)$, which is a jump in the second order derivative, will be different from zero if $\dot{\beta}(t_1) \neq 0$; this analogy is true for any higher order derivatives as well.

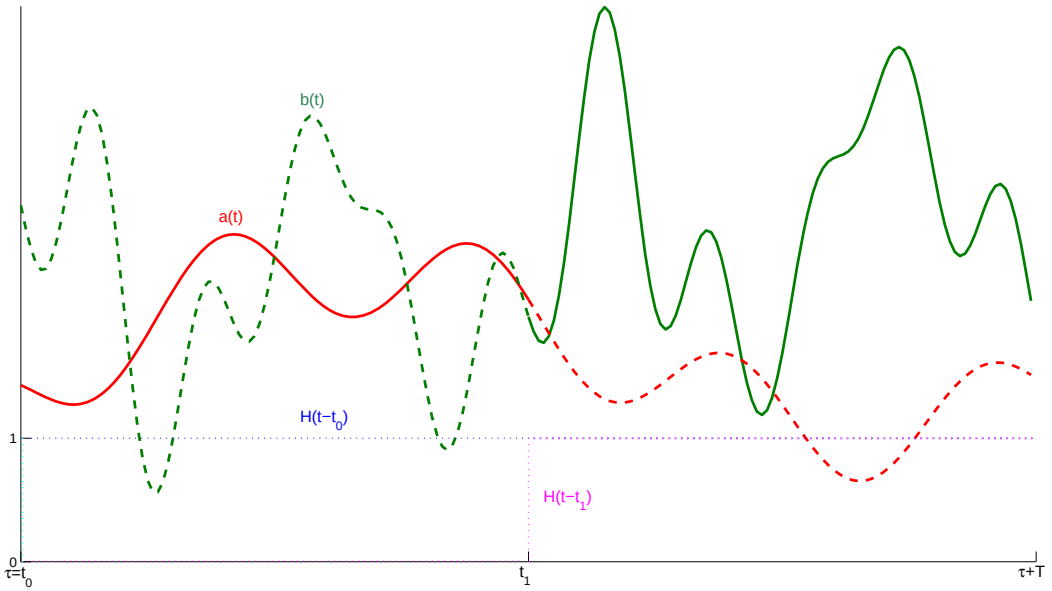


Figure 2.7: The fat line represents the input signal given as a piecewise polynomial on interval I_τ^T with jump in the first derivative of the signal at t_1.

In the sequel, we will show the general form of the above examples. For any integer $N \geqslant 0$, the N^{th} order (generalized) derivative [Schwartz 1998] of $x_\tau(t)$ satisfies

$$\frac{d^N x_\tau}{dt^N} = [x_\tau^{(N)}](t) + \sum_{k=0}^{N-1} \left(\mu_k^0\, \delta(t)^{(N-k-1)} + \mu_k\, \delta(t-t_\tau)^{(N-k-1)}\right), \tag{2.9}$$

where the superscript (k) denotes order k differentiation. We have set $\mu_k \stackrel{\triangle}{=} x_\tau^{(k)}(t_\tau+) - x_\tau^{(k)}(t_\tau-)$ and $\mu_k^0 \stackrel{\triangle}{=} x_\tau^{(k)}(0+) - x_\tau^{(k)}(0-)$ - the jumps in the k^{th}

order derivative of the signal. The sum $\sum_{k=0}^{N-1} \mu_k \, \delta(t - t_\tau)^{(N-k-1)}$ represents the irregular part of the derivative inside I_τ^T, if such irregular part exists (if there is a spike in that particular interval). The equation (2.9) is the cornerstone of the approach we are presenting that consists in a joint detection/estimation. Using the equation (2.9), let us generalize conclusions based on analysis of the derivatives of the signals from figures 2.6 and 2.7 for a given N.

Let say first that we model the spike as a jump in the signal:

1. No spike occurs during the time interval I_τ^T if, and only if,

$$\mu_0 = \cdots = \mu_{N-1} = 0.$$

2. If one (and only one, as assumed above) spike is present in I_τ^T, then $\mu_i \neq 0,\ i = 0, \ldots, N-1$ and one can identify the spiking instant $0 < t_\tau < T$ directly from equation (2.9),

instead, if we model a spike as a jump in the first derivative of the signal then:

1. No spike occurs during the time interval I_τ^T if, and only if,

$$\mu_1 = \cdots = \mu_{N-1} = 0.$$

2. If one (and only one, as assumed above) spike is present in I_τ^T, then $\mu_i \neq 0,\ i = 1, \ldots, N-1$ and one can identify the spiking instant $0 < t_\tau < T$ directly from equation (2.9).

This analogy is the same if we model a spike as a jump in any higher order derivative of the signal. Obtaining a solution for t_τ from any of such models requires very similar calculations, but of course the final expression for t_τ will be different. In the following section we will concentrate on finding an optimal spike model and then finding an explicit solution for t_τ from equation (2.9) for such model.

2.2 Spike detection and estimation

To express t_τ from equation (2.9) we need to select the value of parameter N. Since $N > 0$, an infinite number of change-point estimators may be devised. We will now discuss how the choice of N will affect the obtained results. Related with this, we will also discuss on the choice of a spike model.

The goal of the proposed approach is to detect spikes in an extracellular neural recording. By using the proposed piecewise model we idealized the

situation so the spike instants are now considered as discontinuities. Also, for the moment, we are taking into account only the noise-free signal (*i.e.* $x(t)$ the part of the input observation which contains only the spikes). In such $x(t)$ we want to find discontinuities. Since our model is idealized, we know that, especially after adding the noise, the estimated locations of the discontinuities will be subject to error.

Let us now observe the form of the regular part in order to understand better how we can model it and consequently, what value of N should we chose. The form of a typical action potential (see *e.g.* [Aksay 2001], [Thomas 2000]) is as depicted on figure 2.8 (left plot). Such form results from the basic

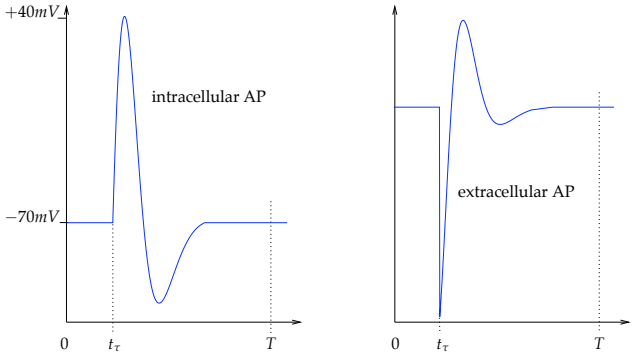


Figure 2.8: Intra- and extra-cellular Action potential waveforms

electric and chemical mechanism which gives rise to a neural spike. In accordance with this mechanism, one would suggest *e.g.* a model like $w(t) = Ke^{-\alpha t}\sin(at^2 + bt)$ (this is actually part of the function plotted on figure 2.8-left). Now a recorded spike, as the extracellular counterpart of an action potential is well represented (at least in the rising phase) by the first derivative [Aksay 2001] as on figure 2.8-right. Therefore, one would model the signal as

$$x_\tau(t) = H(t - t_\tau)f(t - t_\tau) \quad f(t) = \frac{d}{dt}w(t),$$

when the corresponding interval I_τ^T contains a spike. But then, the estimation of the additional parameters K, α, a, b would be necessary in order to obtain t_τ, regardless to the choice of N. Although such an estimation is feasible within the algebraic framework [Fliess 2003] adopted here, it does not correspond to an adequate solution: it burdens the computations while the precise values of the parameters are irrelevant. Instead we will prefer idealized but much simpler models as shown on figure 2.9. Obviously, the location of the discontinuity does not correspond to that of the signal model on figure 2.8-right. However, we mention that a spike is meant by the sudden-onset pulse in the point t_r and we idealize this with the jumps in our models. Refereing to

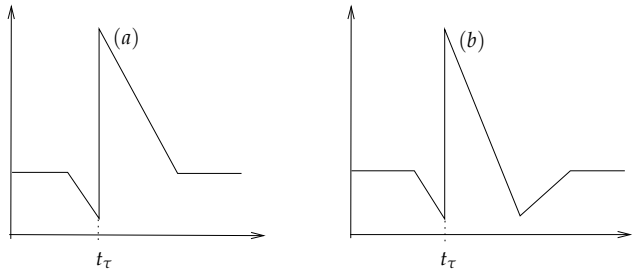


Figure 2.9: Spike piecewise affine models.

the previous section, such model which contains a jump in the signal (not *e.g.* only in its derivative) corresponds with the statement that if spike is present in I_τ^T then $\mu_i \neq 0,\ i = 0, \ldots, N-1$.

Considering the choice of N, the order of the derivative, it is evident that for the particular models on figure 2.9 the choice of $N > 1$ will annihilate the regular part, as it is composed only from first order polynomials. We could as well present models similar to the ones on figure 2.8 using higher order polynomials. By choosing N greater than the highest order of polynomials in the model, the regular part will always vanish. As we mentioned before, this will alow us to concentrate on estimation of what we are interested in - the jumps in the signal, without the need to estimate additional model parameters, which are irrelevant for us. In section 2.2.3 we will discuss through several examples on importance of choice of the correct model and we will see that it is actually not critical; *e.g.* even if the regular part indeed contains polynomials of order four, assuming that it contains only polynomials which do not exceed order one and choosing $N = 2$ (what would annihilate the assumed regular part) still leads to good detection results.

To proceed, we henceforth assume for $x_\tau(t)$ a piecewise model of degree not exceeding one. Let $N = 2$ so that the regular component $[\ddot{x}_\tau(t)]$ vanishes and equation (2.9) becomes:

$$\frac{d^2 x_\tau}{dt^2} = \mu_0^0\, \delta(t)^{(1)} + \mu_0\, \delta(t - t_\tau)^{(1)} + \mu_1^0\, \delta(t) + \mu_1\, \delta(t - t_\tau). \tag{2.10}$$

Even though t_τ can be expressed from equation (2.10) using only the time domain calculus (see *e.g.* [Belkoura 2009]), we find that terms like $\delta(t - t_\tau)^{(k)}$ are much easier to handle using operational calculus [Mikusiński 1987]. To transfer equation (2.10) into the operation domain we use the following Laplace transform pairs:

$$\delta(t) \circ\!\!-\!\!\bullet\ 1$$

$$\mathtt{If}\, f(t) \circ\!\!-\!\!\bullet\ F(s) \mathtt{then} f(t - t_0) \circ\!\!-\!\!\bullet\ F(s) e^{t_0 s}$$

$$f^{(n)}(t) \circ\!\!-\!\!\bullet\ s^n F(s) - \sum_{m=0}^{n-1} s^{n-m-1} \frac{d^m}{dt^m} f(0)$$

and express equation (2.10) in the operational domain as:

$$s^2 \hat{x}_\tau(s) - s x_\tau(0) - \dot{x}_\tau(0) = (\mu_0 s + \mu_1) e^{-t_\tau s}. \tag{2.11}$$

Here $s x_\tau(0)$ and $\dot{x}_\tau(0)$ replace other initial conditions given by $\mu_0^0\,\delta(t)^{(1)}$ and $\mu_1^0\,\delta(t)$.

By differential elimination, we will annihilate the unknown and irrelevant coefficients μ_i, and initial conditions $x_\tau^{(k)}(0)$ in (2.11). This will lead to a linear estimator for t_τ. Once translated back in the time domain, we will get an explicit expression depending on integral terms as $\int_0^T p(\lambda) x_\tau(\lambda) d\lambda$. The next algorithm describes these manipulations which stem from the estimation theory due to [Fliess 2003] (see also [Mboup 2009a]).

Step 1 - First, we get rid of irrelevant unknowns: μ_0, μ_1, $x_\tau(0)$ and $\dot{x}_\tau(0)$. To do this we first multiply the equation (2.11) with $e^{t_\tau s}$:

$$e^{t_\tau s}[s^2 \hat{x}_\tau(s) - s x_\tau(0) - \dot{x}_\tau(0)] = \mu_0 s + \mu_1. \tag{2.12}$$

To shorten the equation (2.12) we denote:

$$\hat{u}(s) \stackrel{\triangle}{=} s^2 \hat{x}_\tau - s x_\tau(0) - \dot{x}_\tau(0).$$

Equation 2.12 is now simply:

$$e^{t_\tau s} \hat{u}(s) = \mu_0 s + \mu_1. \tag{2.13}$$

We annihilate μ_0 and μ_1 by applying differential operator $d^i/ds^i, i \geq 2$:

$$t_\tau^2 e^{t_\tau s} \frac{d^{i-2}}{ds^{i-2}} \hat{u}(s) + 2 t_\tau e^{t_\tau s} \frac{d^{i-1}}{ds^{i-1}} \hat{u}(s) + e^{t_\tau s} \frac{d^i}{ds^i} \hat{u}(s) = 0. \tag{2.14}$$

Then we divide by $e^{t_\tau s}$ and return the substitution $\hat{u}(s)$:

$$t_\tau^2 \frac{d^{i-2}}{ds^{i-2}} \left(s^2 \hat{x}_\tau - s x_\tau(0) - \dot{x}_\tau(0)\right) + 2 t_\tau \frac{d^{i-1}}{ds^{i-1}} \left(s^2 \hat{x}_\tau - s x_\tau(0)\right) + \frac{d^i}{ds^i} \left(s^2 \hat{x}_\tau\right) = 0. \tag{2.15}$$

To annihilate the remaining initial conditions we apply differential operator d^2/ds^2 and obtain:

$$t_r^2 \frac{d^i}{ds^i}(s^2 \hat{x}_\tau) + 2 t_r \frac{d^{i+1}}{ds^{i+1}}(s^2 \hat{x}_\tau) + \frac{d^{i+2}}{ds^{i+2}}(s^2 \hat{x}_\tau) = 0. \tag{2.16}$$

Step 2 - By the very classical rules of operational calculus, multiplication by s corresponds to time derivation: $s\hat{x}_\tau \bullet\!\!-\!\!\circ \frac{d}{dt} x_\tau(t)$. Now, numerical differentiation is difficult and ill-conditioned (it will result by amplification of the

noise, which is the consequence we want to avoid). Thus, we divide the both members of (2.16) by a s^ν, for an integer $\nu > 2$ (we will later discuss how the choice of ν will affect the spike detection results). Only negative powers of s, and hence integral operators, will intervene in the resulting equation. We replace the unobserved signal x_τ by its noisy observation counterpart y_τ. Gathering the resulting equations for $i = \kappa + 2, \kappa \geqslant 0$ and $i + 1$, we obtain the linear system

$$\underbrace{\begin{bmatrix} \hat{v}_\kappa(s,\tau) & \hat{v}_{\kappa+1}(s,\tau) \\ \hat{v}_{\kappa+1}(s,\tau) & \hat{v}_{\kappa+2}(s,\tau) \end{bmatrix}}_{\widehat{P}_\kappa(s,\tau)} \underbrace{\begin{bmatrix} \widetilde{t_\tau^2} \\ \widetilde{2t_\tau} \end{bmatrix}}_{\widetilde{\boldsymbol{\Theta}}_\tau} = \underbrace{-\begin{bmatrix} \hat{v}_{\kappa+2}(s,\tau) \\ \hat{v}_{\kappa+3}(s,\tau) \end{bmatrix}}_{\widehat{Q}_\kappa(s,\tau)} \tag{2.17}$$

where $\hat{v}_\kappa(s,\tau) = \dfrac{1}{s^\nu}\dfrac{d^{\kappa+2}}{ds^{\kappa+2}}\left(s^2\hat{y}_\tau\right)$. The solution $\widetilde{\boldsymbol{\Theta}}_\tau$ of (2.17) is an estimator of $\boldsymbol{\Theta}_\tau = [t_\tau^2 \;\; 2t_\tau]^T$ because x_τ has been replaced by the noisy observation $\hat{y}_\tau$.

Step 3 - The numerical estimates are now obtain by expressing the system (2.17) back in the time domain. For this, let us first recall that differentiation with respect to s correspond to multiplication by $-t$: $\frac{d}{ds}\hat{x}_\tau \bullet\!\!-\!\!\circ\; -tx_\tau(t)$, and division by s^m, $m > 0$ maps to the m^{th}-iterated time integration:

$$(m-1)!s^{-m}\hat{x} \bullet\!\!-\!\!\circ\; \int_0^t (t-\alpha)^{m-1}x(\alpha)d\alpha.$$

With these rules in mind and using integration by parts, we express the time domain analog of $\hat{v}_\kappa(s,\cdot)$ as:

$$v_\kappa(t,\tau) = \frac{(-1)^\kappa}{(\nu-1)!}\int_0^t \frac{d^2}{d\lambda^2}\left\{\lambda^{\kappa+2}(t-\lambda)^{\nu-1}\right\} y_\tau(\lambda)d\lambda. \tag{2.18}$$

The analog of (2.17) is thus in the form $P_\kappa(t,\tau)\widetilde{\boldsymbol{\Theta}}_\tau(t) = Q_\kappa(t,\tau)$ so that we have an estimate $\widetilde{\boldsymbol{\Theta}}_\tau(t)$ for each estimation time $t \leqslant T$. To be consistent with the signal model in (2.2), we now on fix this estimation time to the width of I_τ^T: $t = T$. We can therefore simplify the notations above by dropping the argument t, and write $v_\kappa(\tau)$ instead of $v_\kappa(T,\tau)$.

2.2.1 Joint detection-estimation

Using the equations (2.17) and (2.18) we can easily calculate an explicit solution for $\widetilde{t_\tau}$ for any interval I_τ^T - we do not have any unknown parameter; we need only the noisy input observation and the free parameters $\kappa \geq 0$ and $\nu \geq 2$. However, due to the mismodelling error and the noise such explicit solution will not be robust: the explicit solution of the system of equations that we have obtained will be very sensitive on any variation of the coefficients and these depend on the noisy observation. In section 2.2.3 we will demonstrate this using a concrete examples with a piecewise input signal.

Instead of calculating the explicit solution for $\widetilde{t_\tau}$, we will concentrate on testing whether an interval I_τ^T contains a spike or not. If an action potential happens at time $t^\star$, then it will be detected in all intervals I_τ^T for which $t^\star - T \leqslant \tau \leqslant t^\star$. This observation shows how the detection process also leads to the estimation of the spiking instants.

Recall that in equation (2.11) we have $\mu_0 = \mu_1 = 0$ when the interval I_τ^T is devoid of an action potential. When this is the case, in idealized situation (no mismodelling error, no noise), $v_\kappa(\tau) = 0$ on I_τ^T for all κ. In other words, the estimator $P_\kappa(\tau)\widetilde{\mathbf{\Theta}}_\tau = Q_\kappa(\tau)$ degenerates into $P_\kappa(\tau) = 0$ and $Q_\kappa(\tau) = 0$ (in real situations this is actually up to the output noise level).

Apart from the mathematical, we can observe this as well from the physical point of view. $v_\kappa(\tau)$ involves $s^2\hat{y}_\tau$ which, in the time domain, translates into 2^{nd} order differentiation of the signal. Then for a 1^{st} order polynomial signal model, this differentiation is zero and hence v_k is zero. We consider now an interval I_τ^T. If the interval does not contain any discontinuity the signal will be annihilated: $v_\kappa(\tau) = 0$. If an interval I_τ^T contains a discontinuity, the discontinuity will not be annihilated by applying the derivative, what results in $v_\kappa(\tau) \neq 0$.

We can conclude that we are not limited on explicit calculation of the t_τ by solving the system of equations, but we can deduce the spike location by using the property that $v_\kappa(\tau)$ is a derivative estimation of the input observation. We can use $v_\kappa(\tau)$ for any $\kappa \geq 0$ as a decision function, but since $v_\kappa(\tau)$ is zero only in the idealized case the condition, $|v_\kappa(\tau)| > 0$ should be replaced by $|v_\kappa(\tau)| > \gamma_\mathcal{J}$, where $\gamma_\mathcal{J}$ is a threshold value. Instead of using only one $v_\kappa(\tau)$ we can design a more complex decision function that will be more robust with respect to the noise or which will have some particular desired properties. For example, $\prod_{i=0}^{L-1} v_{\kappa+i}(\tau)$, where $L > 0$ is one such decision function which is more selective as greater L we chose. As we will see later, we can affect the trade-off between falsely detected spikes and missed spikes by selecting different values of L.

Since we have obtained the system of equations, we will use some properties of that system to determine a decision function $\mathcal{J}$ along with an associated threshold $\gamma_\mathcal{J}$, and devise a detection system,

$$\mathcal{J}(\tau) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma_\mathcal{J} \tag{2.19}$$

to test the degeneration of the system $P_\tau\widetilde{\mathbf{\Theta}}_\tau = Q_\tau$ (null hypothesis $\mathcal{H}_0$), against the alternative $\mathcal{H}_1$: *one action potential occurs in* I_τ^T. we consider the family of functions $\mathcal{J}_\kappa, \kappa \geqslant 0$,

$$-\det P_\kappa(\tau) = \mathcal{J}_\kappa(\tau) = [v_{\kappa+1}(\tau)]^2 - v_\kappa(\tau)v_{\kappa+2}(\tau). \tag{2.20}$$

This also coincides with the discriminant of the quadratic equation formed by the first line of the system $P_\kappa(\tau)\mathbf{\Theta}_\tau = Q_\kappa(\tau)$. The condition $\mathcal{J}_\kappa(\tau) \geqslant 0$ is thus necessary to have real solutions while $\mathcal{J}_\kappa(\tau)$ must be different from zero. The noise and the local mismodelling apart, we then have $\mathcal{J}_\kappa(\tau) > 0$ if, and only if, an action potential happens in I_τ^T. Due to the noise presence, the condition $\mathcal{J}_\kappa(\tau) > 0$ must be replaced by $\mathcal{J}_\kappa(\tau) > \gamma_\mathcal{J}$

The results of the simulations, which will be presented later in this chapter, suggest that using different detection systems leads to similar results.

2.2.2 Discrete nonlinear filter

The proposed solution to the joint detection and estimation problem was devised based on the continuous-time nature of the spike train. Now, we give a discrete-time implementation of that solution. Using the very classical numerical integration machinery, we show how each $v_\kappa(\tau)$, $\kappa \geqslant 0$ in (2.20) can be implemented as the output of a discrete finite impulse response (FIR) filter, with the sample observation y as input. As a consequence, the corresponding decision function $\mathcal{J}_{\kappa,n}, n \in \mathbb{N}$, will read as the output of a discrete quadratic Volterra filter.

To begin, let us define

$$h_\kappa(\lambda) = \begin{cases} \frac{(-1)^{\kappa+1}}{(\nu-1)!} \frac{d^2}{d\lambda^2} \left\{(T-\lambda)^{\kappa+2}\lambda^{\nu-1}\right\}, & 0 \leqslant \lambda \leqslant T \\ 0 & \text{else,} \end{cases}$$

and redefine the intervals I_τ^T by their causal counterparts: $I_\tau^T \triangleq [\tau - T, \tau]$. Accordingly, consider (2.18) for $t = T$ and with the change of variable $\lambda \to T - \lambda$. Then, we may directly rewrite v_κ in (2.18) as

$$\begin{aligned} v_\kappa(\tau) &= \int_0^T h_\kappa(\lambda) y_{\tau-T}(T-\lambda) d\lambda \\ &= \int_0^\infty h_\kappa(\lambda) y(\tau-\lambda) d\lambda. \end{aligned} \tag{2.21}$$

We turn now to the discrete-time setting where only time samples of the observation, $y_m = y(t_m)$, are available. We assume a uniform sampling, $t_m = mT_s, m = 0, 1, \ldots$, with sampling period T_s. Let $T = MT_s$ and $\nu \geqslant 2$ be given. Set $\tau = nT_s$ and compute the finite sequence $\{h_{\kappa,m} \triangleq h_\kappa(t_m)\}_{m=0}^M$. Then, by a numerical integration method with abscissas t_m and weights W_m, $m = 0, \cdots, M$, we obtain the approximation of (2.21):

$$v_\kappa(\tau) \approx v_{\kappa,n} = \sum_{m=0}^M g_{\kappa,m} y_{n-m},$$

where

$$g_{\kappa,m} \stackrel{\triangle}{=} W_m h_{\kappa,m}.$$

In all the sequel, we use the trapezoidal method for which $W_0 = W_M = 0.5$, and $W_m = 1, m = 2, \cdots, M-1$. As $\tau = \tau_n = nT_s$ varies (sliding windows I_n^M), $v_{\kappa,n}$ reads as the output of the FIR filter, with impulse response sequence $\{g_{\kappa,m}\}_{m=0}^{M}$. Accordingly, we obtain for each $\kappa \geqslant 0$, a discrete approximation $\mathcal{J}_{\kappa,n} \approx \mathcal{J}_\kappa(nT_s)$ of (2.20)

$$\mathcal{J}_{\kappa,n} = [v_{\kappa+1,n}]^2 - v_{\kappa,n} v_{\kappa+2,n}, \tag{2.22}$$

as the output of the Volterra filter with input y_n,

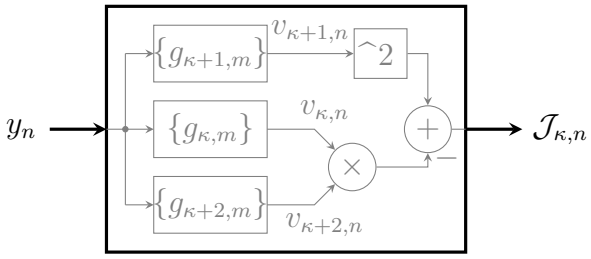


Figure 2.10: Digital Volterra filter representation

In the simulations to follow, a detection will be validated if it meets the agreement of a given number, K, of elementary decision functions $\mathcal{J}_{\kappa,n}$. The corresponding final decision function is thus given by the combination

$$\mathcal{J}_n = \prod_{\kappa=0}^{K-1} \widetilde{\mathcal{J}}_{\kappa,n}, \quad n = 0, 1, \ldots, \tag{2.23}$$

where $\widetilde{\mathcal{J}}_{\kappa,n} = \max(0, \mathcal{J}_{\kappa,n})$ for each κ and n. This makes the detection more robust to fake spikes as the decision function is more rigorous.

2.2.3 Examples of change-point detection in piecewise signals

We will now analyze the performance of different detection systems developed within the proposed framework. To do so, we create a piecewise, noise-free signal composed from polynomials of order zero and one, which contains 5 irregularities: 3 jumps in the signal and 2 irregularities corresponding to jumps in the first derivative of the signal. The input signal is shown on the top plot on figure 2.11. From the set of free parameters we chose $N = 2$, $\nu = 3$, $\kappa = 0$ and $M = 20$ (the impulse response of each filter is 20 samples long).

First, we analyze accuracy of the explicit solution t_τ, in order to validate that our theoretical assumptions and the calculations were correct. Since $N = 2$ we obtain 2 solutions for t_τ, they are shown on the second and the third plot on figure 2.11. In the ideal case, both solutions for t_τ would be

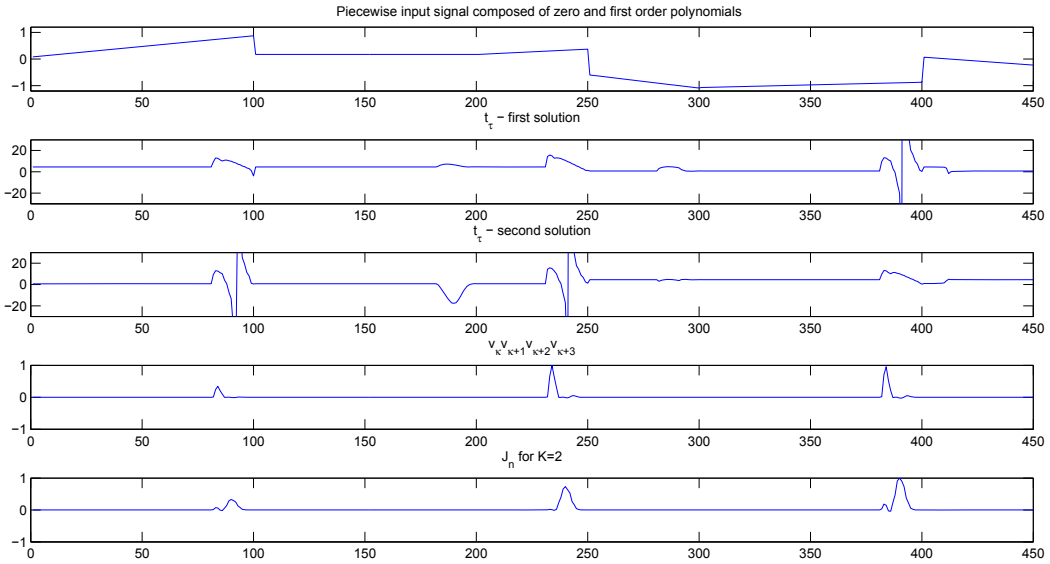


Figure 2.11: Top plot: piecewise, noise-free input signal composed from polynomials of order zero and one. The signal contains 3 jumps in the signal (at 100^{th} to 101^{st}, 250^{th} to 251^{st} and 400^{th} to 401^{st} sample) and two additional jumps in the first derivative of the signal (at 200^{th} to 201^{st} and 300^{th} to 301^{st} sample). Second and third plot: the two solutions for t_τ with $N = 2$, $\nu = 3$, $\kappa = 0$ and $M = 20$. Fourth and fifth plot: two normalized decision functions.

identical, and they would be zero except during the 20 samples before each jump in the signal, where they would linearly decrease from 20 to zero (as the jump location becomes closer to the origin of the interval I_τ^T). Since we are using discrete computation of the integrals the obtained explicit solutions for t_τ do not represent the ideal theoretical case. As we can see from the second and the third plot on figure 2.11, the two solutions are during the most of the time different, but one of them approximately linearly decays from 20 to zero as we approach to any of the jump locations. This is better illustrated on figure 2.12 which is a zoom of a part of the first three plots from figure 2.11.

On figure 2.11 we also show the results obtained when $\prod_{i=0}^{3} v_i(\tau)$ and $\mathcal{J}_n$ for $K = 2$ were used as the decisions functions (fourth and fifth plot respectively). As we expected, these two decision functions are equal to zero (or very small) except when I_τ^T contains a jump.

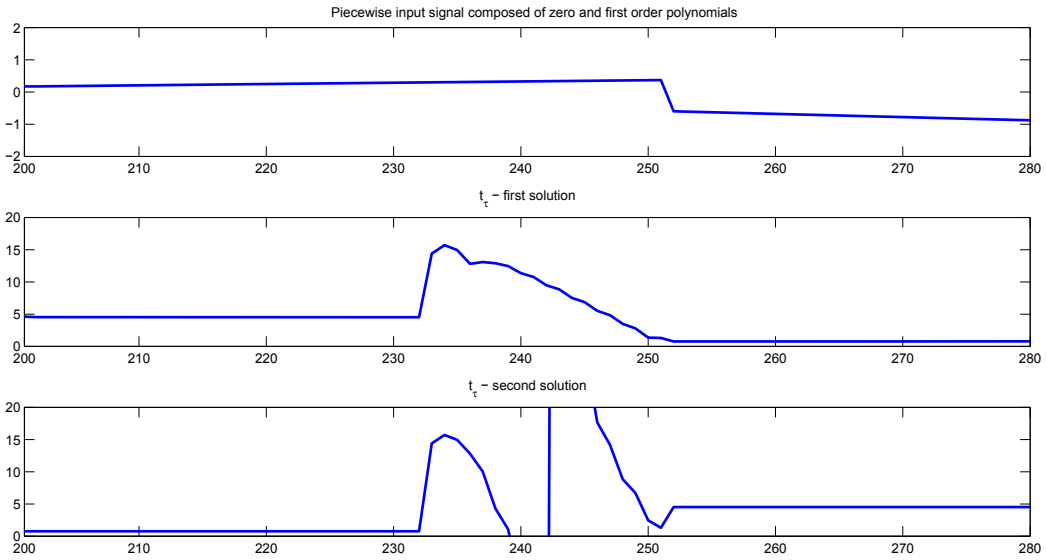


Figure 2.12: Zoom of the first three plots from figure 2.11

Notice that jumps in the first derivative of the signal are generally not detected. This is particulary the case for the two decision functions shown on the two bottom plots. To calculate these two we used products of different $v_{\kappa,n}$ for different κ. Such products are more selective than single $v_{\kappa,n}$ since it is enough that one of $v_{\kappa,n}$ from the product is close to zero to make the whole product close to zero. This result was expected, since we model spikes as jumps in the signal, not as jumps in the signal derivative.

We mentioned before that the explicit solution for t_τ will be very sensitive to the noise. On figure 2.13 we show the analog comparison as on figure 2.11, only this time with added some noise to the input signal. One of the solutions for t_τ still decays from around 20 to around zero when I_τ^T contains a jump (see zoom of figure 2.13 on figure 2.14). However, it is now very difficult to find an actual jump location since both solutions for t_τ often have a value between 0 and 20 when I_τ^T does not contain a jump. On the other side, the two decision functions shown on the fourth and the fifth plot and significantly different from zero only when jump occurs in I_τ^T. As we mentioned earlier, the reason for low robustness of the explicit calculation of t_τ lies in a generally very high sensitivity of the solutions of a square equation on noisy perturbations in the coefficients. Obtaining results using the knowledge that the derivative estimators, or their different combinations, are greater than some threshold only when a jump occurs in I_τ^T leads to much more accurate detection, but also localization. Localization is easily obtained by knowing that the derivative estimators detect the jump firstly when $t_\tau = \tau + T$.

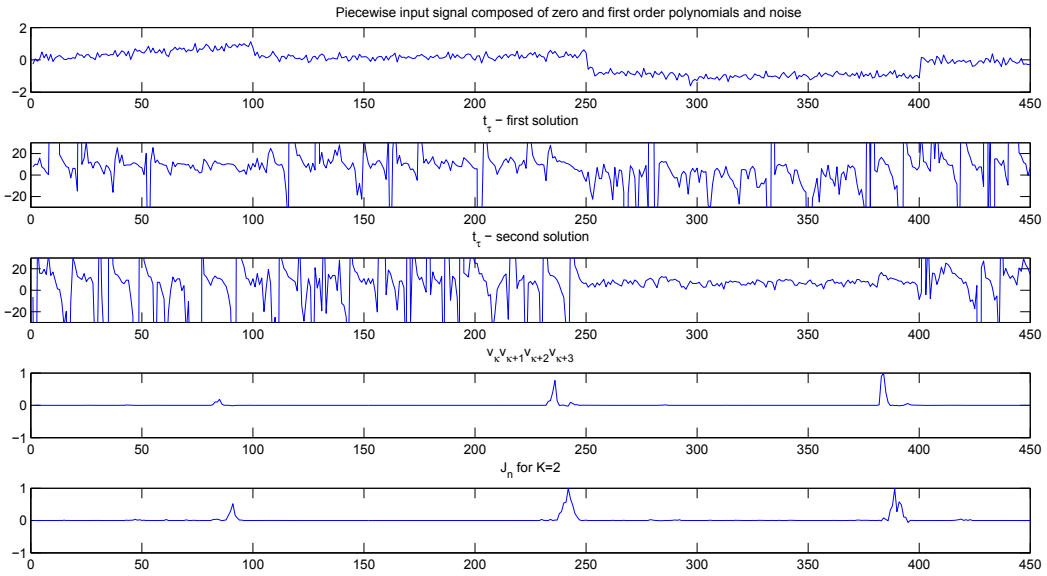


Figure 2.13: Top plot: piecewise, noisy input signal composed from polynomials of order zero and one. The signal contains discontinuities that occur at the same time as in the signal on the top plot of figure 2.11. The four bottom plots correspond to those from figure 2.11.

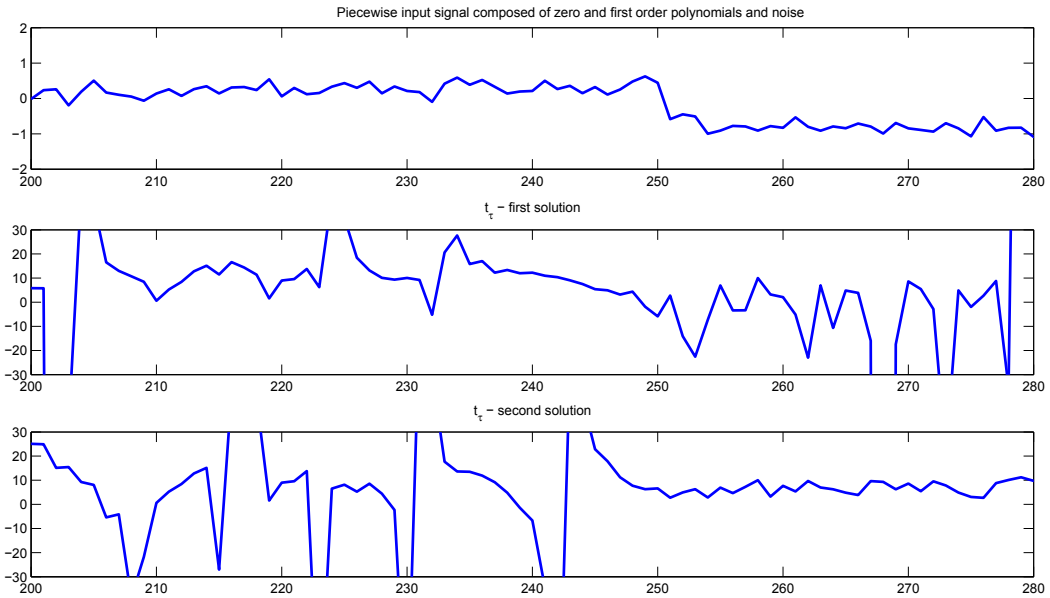


Figure 2.14: Zoom of the first three plots from figure 2.13

Let us now illustrate how the mismodelling error affects the results. We perform a similar simulation as on figure 2.11, but only this time with up to the fourth order polynomials. Since our detection system is devised with $N = 2$, the regular part of the signal will not be annihilated. Results of the simulation on the noise-free input signal are shown on figure 2.15 and on the noisy input signal on figure 2.16. The figure suggests, that the mismodeling error does not significantly affect the results. The two decision functions shown on the fourth and the fifth plot are again quite accurate and robust to the noise perturbations.

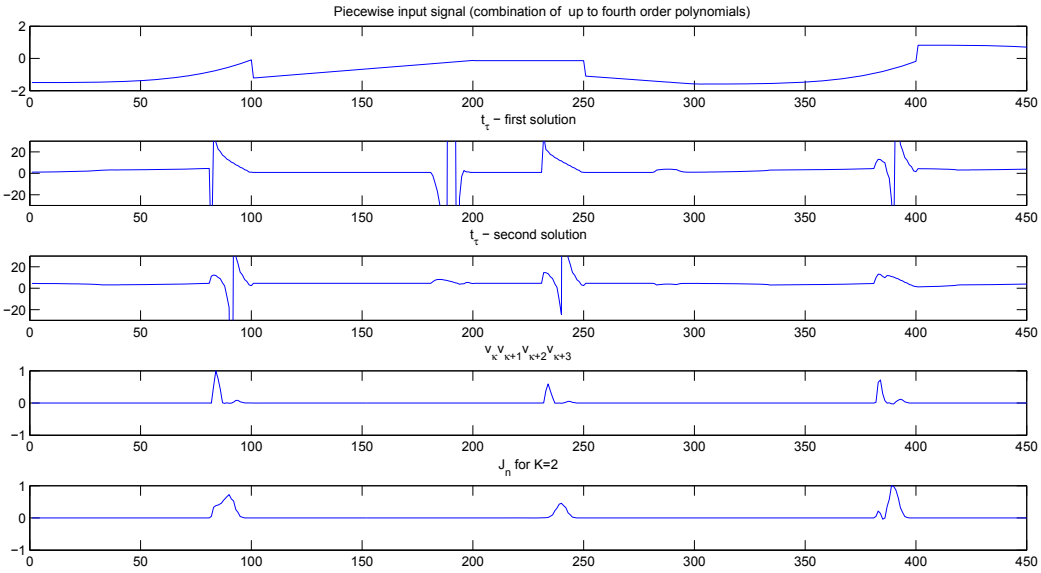


Figure 2.15: Top plot: piecewise, noise-free input signal composed from up to fourth order polynomials. The signal contains discontinuities that occur at the same time as in the signal on the top plot of figure 2.11. The four bottom plots correspond to those from figure 2.11.

To illustrate how $v_{\kappa,n}$ changes with change of N we show on figure 2.17 a piecewise signal composed of zero, first and second order polynomials (top plot) and $v_{\kappa,n}$ with $\kappa = 0$ and $\nu = 3$ for N from 1 to 4 (next four plots). Each time when the jump in the signal occurs (at 100^{th}, 250^{th} and 400^{th} sample) $v_{\kappa,n}$ changes significantly. Notice also that $v_{\kappa,n}$ for each consecutive N looks like the derivative for the preceding.

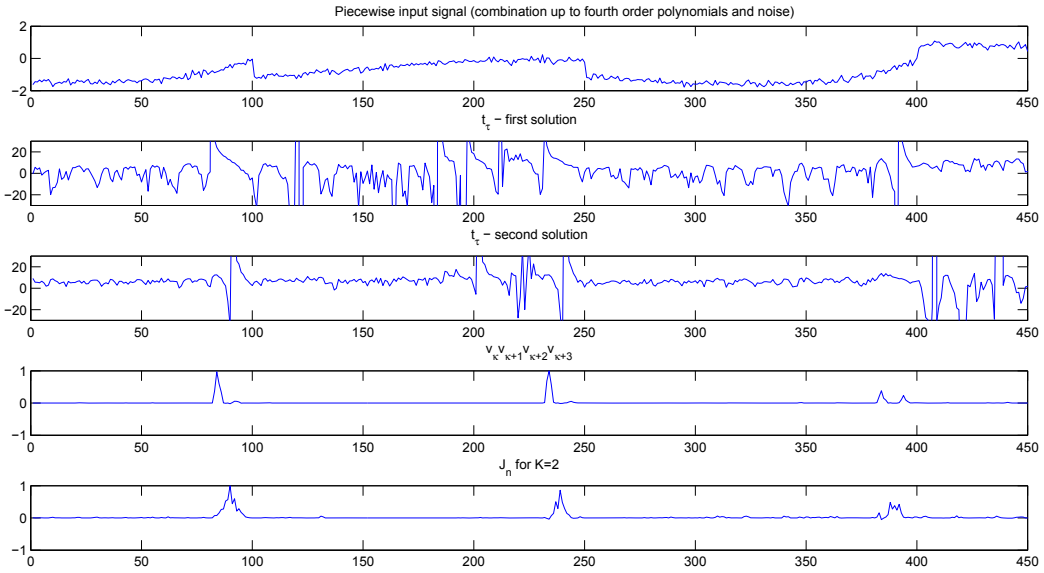


Figure 2.16: Top plot: piecewise, noisy input signal composed from up to fourth order polynomials. The signal contains discontinuities that occur at the same time as in the signal on the top plot of figure 2.11. The four bottom plots correspond to those from figure 2.11.

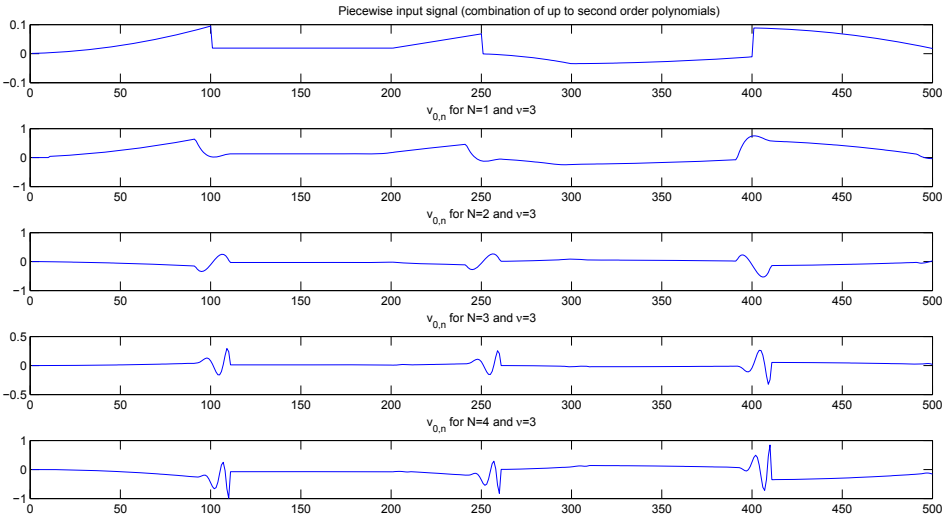


Figure 2.17: Top plot: piecewise, noise-free input signal composed from up to second order polynomials. The signal contains discontinuities that occur at the same time as in the signal on the top plot of figure 2.11. The four bottom plots show $v_{\kappa,n}$ with $\kappa = 0$ and $\nu = 3$ for N from 1 to 4.

2.2.4 Selection of the free parameters

In this subsection we will give some physical meaning to the selection of the free parameters: T (which corresponds to the filter impulse length M), κ and ν. The selection of the parameters should not be independent, *e.g.* κ and ν should be chosen depending on the choice of T.

To emphasis the spike shaped waveforms it is preferable that the effective duration of the filters (defined by the samples of the impulse response that are significantly different from 0) is similar to the spike duration. In this way we are sure that during the filtering, for some n, each spike will be fully overlapped with the filters. Figure 2.18 shows the impulse response and figure 2.19 the frequency response of the four filters ($K = 4$) with the impulse length duration $M = 60$ and ν that varies from 4 to 10 in steps of 2 (we have mentioned earlier that $\nu > 2$). If the spike length is *e.g.* 30 samples, a good choice would be $\nu = 6$ and $\kappa = 1$ or $\nu = 10$ and $\kappa = 1$, 2 and 3, since these choices would lead to approximate filter length of 30 samples.

Generally, choosing smaller ν makes the effective impulse length larger and of course the filter passes a narrower frequency band. On the other side, increasing κ will introduce filters that will have wider pass-band.

Larger ν or larger κ shifts the frequency characteristics to the right - the central frequency is increasing. Good choice for the central frequency can be calculated knowing the spike duration and assuming that *e.g.* the positive half-period of the spike on the right plot of figure 2.8 is actually a half-period of a `sin` function. Let say that the positive half-period of a spike is $0.5ms$ long, than it corresponds to 1kHz `sin` wave. Using figure 2.19, any choice seems to be good except $\nu = 6$ and $\kappa = 4$, since for that one the frequencies around 1kHz appear to be attenuated. Such computations are, of course, not exact, however they might help to chose the free parameters which will lead to efficient spike detection.

In the following simulations we will show that the proposed method is quite robust to the parameters choice. However, we do not provide an explicit way to compute the free parameters. Exact values of the parameters are best to be chosen experimentally, as their optimal values depend on the given data and on the desired detection selectivity.

2.3 Simulations and results

We first demonstrate the performance of our joint estimation and detection method on the simultaneous intra- extra-cellular recordings. Later, we compare it with one of the most successful spike detection methods: the wavelet detection method (WDM) presented in [Nenadic 2005].

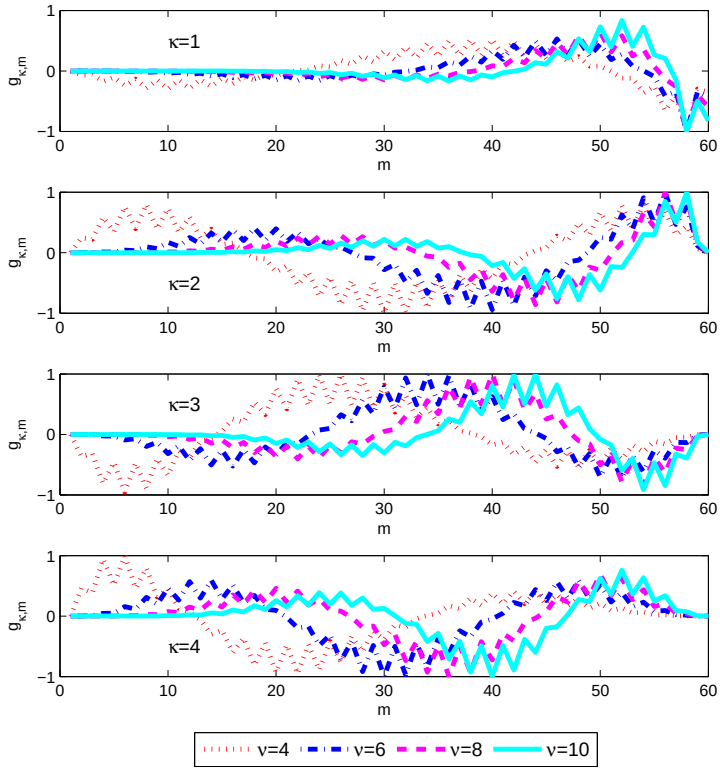


Figure 2.18: Impulse responses of the filters for different values of κ and ν.

2.3.1 Simulation on the real signal

Comparison of spike detection methods using a real signal is very difficult, since we are not able to validate the obtained results. To do the validation we should know all the spike locations, which is not possible for real recordings. The problem of the validation can be only partially solved by using simultaneous intra- extra-cellular recordings. If the extracellular and the intracellular electrode are close enough, the spikes fired by the neuron recorded intracellularly are visible in the extracellular recording as well. A drawback of such validation is that, due to a very difficult experimental setup, currently no more than one neuron can be recorded intracellularly at the same time. This is not enough for comparison of spike detection methods, since we do not know whether parts of the signal when the intracellularly recorded neuron was inactive contain spikes or not. We use such recording just to visually demonstrate results of the proposed method.

Figure 2.20 shows an example of simultaneous intra- extra-cellular recording (first and second plot) from rat hippocampal area CA1, described in [Henze 2000]. It is evident that in the particular example action potentials recorded intracellularly are visible in the extracellular recording. The third

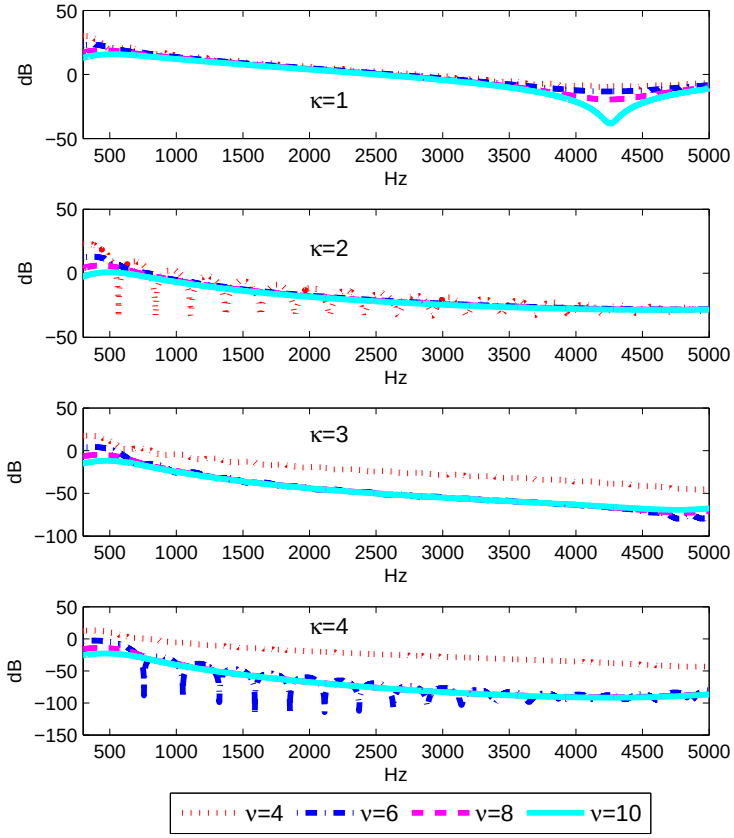


Figure 2.19: Frequency responses of the filters for different values of κ and ν.

and the fourth plot on the figure 2.20 show $\mathcal{J}_n$ which is the result of applying the proposed approach on the extracellular signal, for $K = 2$ and $K = 3$. Parameter ν is set to 7 and $T = 2.4ms$. The value of $\mathcal{J}_n$ obviously gets high when the intracellularly recorded neuron fires, which is what we wanted to obtain. $\mathcal{J}_n$ gets high at some other time instants as well, what indicates firing of other neurons close to the electrode. Since this is a real recording we are not able to verify this assumption. However, it is evident that the proposed method makes the detection of the spikes fired by the intracellularly recorded neuron much easier than it would be if it was done directly from the extracellular recording. This is particulary apparent on figure 2.21 which gives a zoom of a part of figure 2.20. We can notice also that the choice of bigger K leads to more conservative results. This is expected, since K determines the number of products of $\widetilde{\mathcal{J}}_{\kappa,n}$ that we are taking. Unfortunately, similar plot can not be shown for the WDM, since for that method the results are obtained as a binary combination of all the scales.

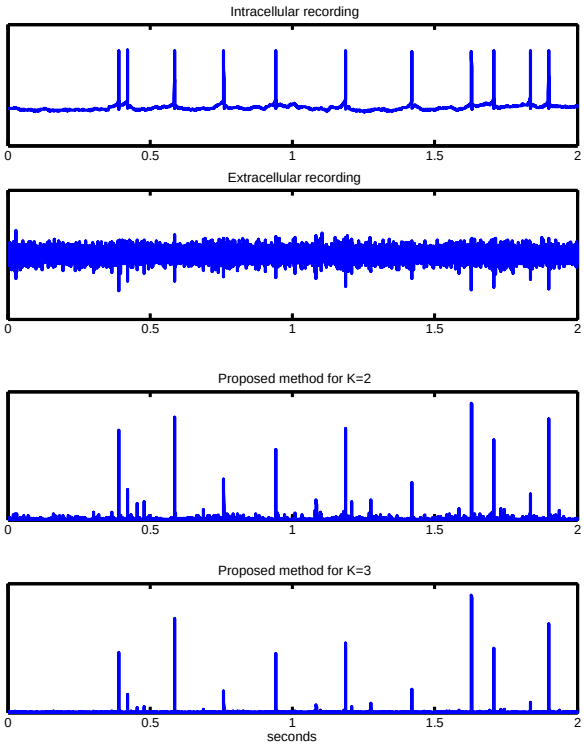


Figure 2.20: Simultaneous intra- extra-cellular recording and the results of the proposed spike detection method for $K = 2$ and $K = 3$.

2.3.2 Simulation on the artificial signal

Real recordings obviously can not be used to quantify the detection performance, thus we have to use artificially synthesized neural recordings. To make the simulation as realistic as possible we synthesize the data by using spike templates and noise extracted from the real neural recording with high SNR. By scaling the amplitudes of the spikes we are able to control the SNR of the simulated signal. The data we are using for the simulation are recorded using extracellular tetrode from locust antennal lobe, they are shown on figure 2.2.

2.3.2.1 Spike templates

An average of 256 spikes have been extracted from the channels and aligned into 5 clusters using Lloyd's algorithm [Lloyd 1982]. The centroids S_i of the clusters are then used as spike templates. They all have the same duration of $3.33ms$ and are normalized: $\|S_i\|_\infty = 1$.

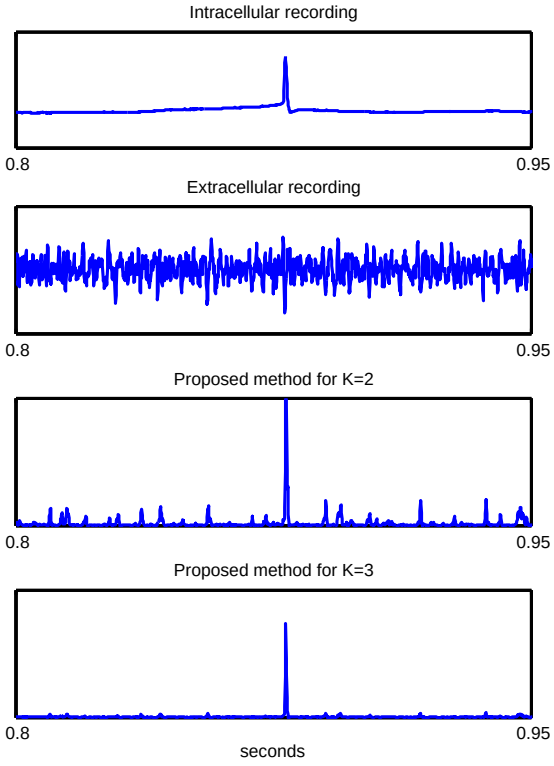


Figure 2.21: Zoom a of part of figure 2.20

2.3.2.2 Noise template

Once all the spikes have been extracted from the experimental data, the remaining background noise is used as a neural noise template. We form a single long (285000 samples) observation by concatenating the data from the four recording sites. As we already discussed, a significant part of the noise is composed of remote spikes.

2.3.2.3 Noisy spike train simulation

Each run is based on a signal segment of $^2/_3$ s (10000 samples). For each run, the spike locations are drawn from the successes of 10000 samples Bernoulli trials with a parameter p corresponding to a given mean firing rate FR. The number of simulated spikes thus varies from one run to another. Next, the noise-free spike train is generated by assigning randomly the spike templates to the locations. The polarity of each spike template also is chosen randomly. Finally, a randomly selected segment of the neural noise template is added. The noise variance σ^2 is set according to the desired SNR defined as

in [Nenadic 2005]:

$$\text{SNR} = \frac{\|S_i\|_\infty}{\sigma} = \frac{1}{\sigma}. \tag{2.24}$$

Figure 2.22 displays $0.5s$ of such a simulated signal. The spike locations are indicated by the vertical bars on the bottom.

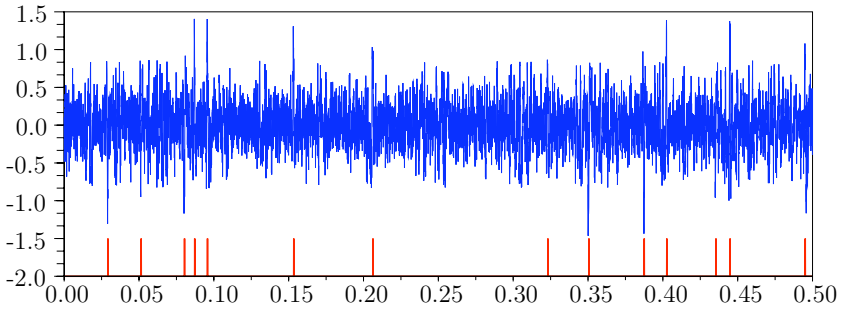


Figure 2.22: Noisy spike train simulation: SNR = 3.5, FR = 15Hz (blue); Spike locations (red)

2.3.2.4 Comparison for different choice of the free parameters

In subsection 2.2.4 a physical meaning to the selection of the free parameters was given. In this subsection the results of the spike detection for different choices of these parameters are compared.

The results are presented by the receiver operating characteristics (ROC) curves: probability of correct detection (P_{cd}) against probability of false alarm (P_{fa}). Such type of validation of detection algorithms is very classical. To obtain the ROC curves we first generate 500 noisy spike trains as described above. Then for each spike train we apply the spike detection algorithm and calculate the decision function. Two different decision functions were used in the comparison:

- The first decision function was calculated as the absolute value of the product of the three filters: $|v_\kappa(\tau)v_{\kappa+1}(\tau)v_{\kappa+2}(\tau)|$ (figure 2.23).
- The second decision function was $\mathcal{J}_n$ given in equation (2.23) (figure 2.24).

Given a spike, correct detection is meant when the distance between the exact location and the estimated one is less than half the spike duration, *i.e.* $1.66ms$. On each decision function through 12 iterations we set 12 different threshold levels. These thresholds were chosen in such way that for the minimal of the 12 values all or almost all the spikes were detected, while for the maximum none or almost none false spike was detected.

For i^{th} simulation the probability $P_{cd}^{(i)}$ of correct detection is estimated by $\hat{P}_{cd}^{(i)}$ which is given by:

$$\hat{P}_{cd}^{(i)} = \frac{N_{cd}^{(i)}}{N_t^{(i)}},$$

where $N_{cd}^{(i)}$ is the number of correct detections in i^{th} simulation and $N_t^{(i)}$ is the number of generated spikes in i^{th} simulation. Notice that $N_t^{(i)}$ is not constant over trials. Thus the estimation of P_{cd}, $\hat{P}_{cd}$, reads as:

$$\hat{P}_{cd} = \sum_{k=0}^{n} \hat{P}_{cd|k} P(N_t = k).$$

$\hat{P}_{cd|k}$ is the estimate of P_{cd} given that the number of total spikes is equal to k; $\hat{P}_{cd|k}$ can be expressed as follows:

$$\hat{P}_{cd|k} = \begin{cases} \frac{\bar{N}_{cd|k}}{k}, & k \in N \\ 0 & k = 0 \end{cases}$$

where $\bar{N}_{cd|k}$ is the average number of correct detections given that the number of detected spikes is k with n different realizations.

We compute the estimation of $P_{fa}^{(i)}$ - probability of false alarm in i^{th} simulation, in analog way:

$$\hat{P}_{fa}^{(i)} = \frac{N_{fa}^{(i)}}{N_d^{(i)}}$$

Here $N_{fa}^{(i)}$ and $N_d^{(i)}$ are the number of false alarms and the number of detected spikes (no matter correct or false), respectively. Notice that N_d is not constant across the simulations. The calculation of P_{fa} is estimated by $\hat{P}_{fa}$ which given by:

$$\hat{P}_{fa} = \sum_{k=0}^{n} \hat{P}_{fa|k} P(N_d = k)$$

where $\hat{P}_{fa|k}$ is the estimate of P_{fa} given that the number of detected spikes is equal to k; $\hat{P}_{fa|k}$ can be expressed as follows:

$$\hat{P}_{fa|k} = \begin{cases} \frac{\bar{N}_{fa|k}}{k}, & k \in N \\ 0 & k = 0 \end{cases}$$

where $\bar{N}_{fa|k}$ is the average number of false alarms given that the number of detected spikes is k with n different realizations.

The values of the parameter ν were varied from 5 to 20, with steps of 5, while the values of the parameter κ were 2, 3 and 4. For $\nu = 5$ and $\nu = 20$

the results are, for most of situations, significantly worse than for $\nu = 10$ and $\nu = 15$. The parameter κ was varied in relatively small interval so its change does not affect the results significantly. This was generally the case for all the SNR and the FR and it suggests that the method will have a stable performance for relatively large range of values of the free parameters, independently on SNR and FR. Choosing the free parameters out of this range will very significantly decrease the performance of the proposed algorithm (*e.g.* selecting too small or too large ν as in this example).

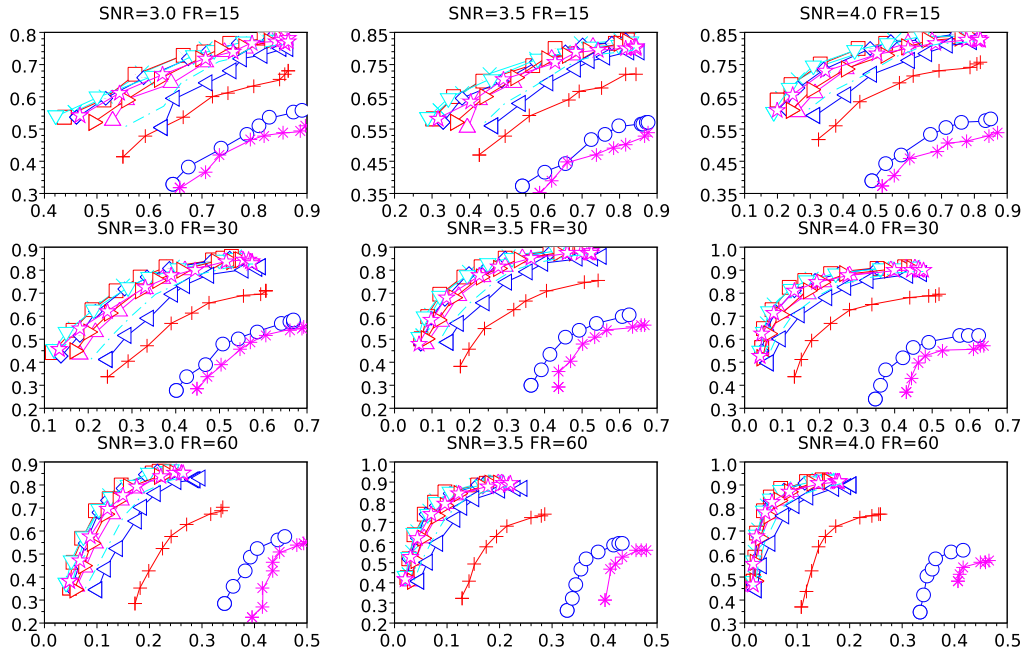
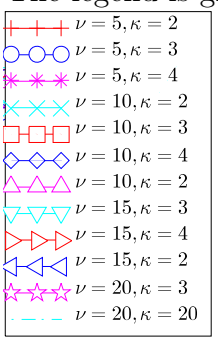


Figure 2.23: ROC curves: Comparison of the result obtained with the decision function $|v_\kappa(\tau)v_{\kappa+1}(\tau)v_{\kappa+2}(\tau)|$ for different choice of K and ν. Probability of the false alarm is given on the horizontal axis and the probability of correct detection on the vertical axis. The legend is given below.



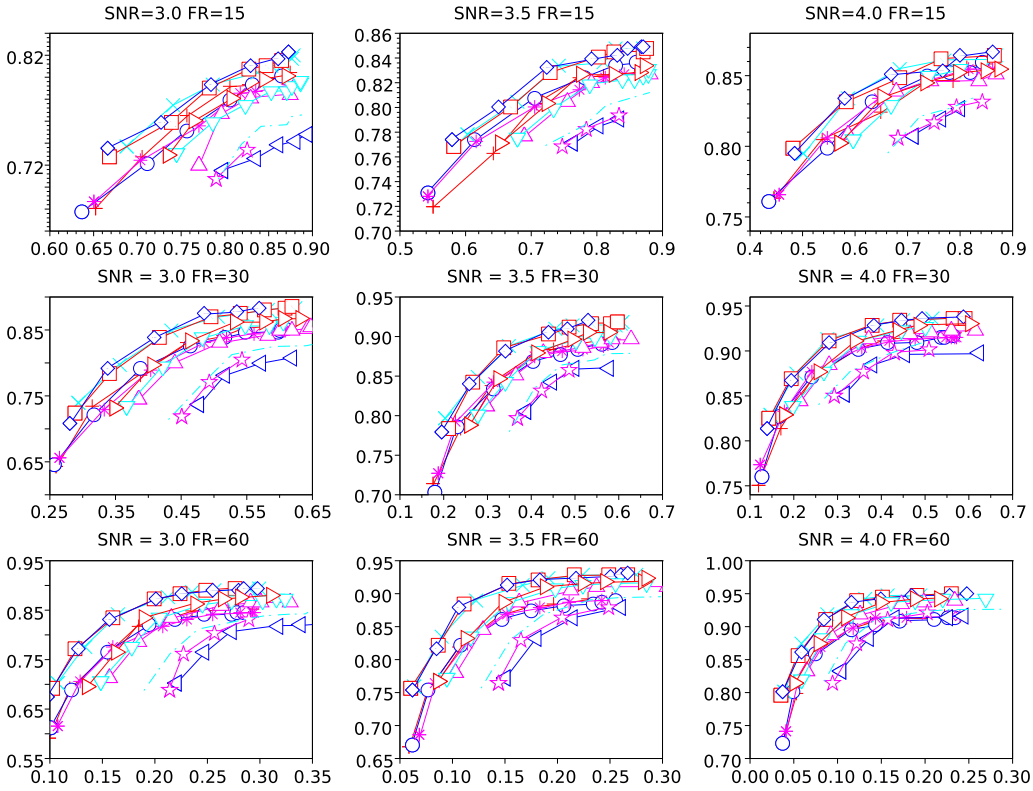


Figure 2.24: ROC curves: Comparison of the result obtained with the decision function $\mathcal{J}_n$ for different choice of K and ν. Probability of the false alarm is given on the horizontal axis and the probability of correct detection on the vertical axis. The legend is the same as for figure 2.23

.

2.3.2.5 Other approaches

The WDM, which main ideas are briefly described in the introduction to this chapter, compares favorably with currently commonly used approaches for the spike detection in neuroscience [Nenadic 2005]. In [Nenadic 2005] the wavelet method has been compared with a method based on signal power [Harris 2000], [Bankman 1993], Nonlinear energy operator (NEO) based method [Kim 2000] and a classical amplitude thresholding (single and double threshold), see e.g. [Quiroga 2004] .

The power based method computes the root-mean-square power of the filtered signal using a sliding window. The mean and standard deviation of the power are also computed, and a spike is detected when the power exceeded the threshold derived from the standard deviation from the baseline mean.

The output of the NEO is proportional to the product of the instantaneous amplitude and frequency of the input signal, thus, highlighting the action potential peak. NEO is known to perform generally well in lower SNR (see [Kim 2000]).

An amplitude thresholding is the simplest approach, which can be implemented by setting a threshold only for positive or negative polarity (single), or for both (double). This approach is suitable only for recordings with high SNR.

For the simulation setup in [Nenadic 2005], which is very similar to the one used here, the results presented through the receiver operating characteristic (ROC) curves (obtained in analog way as the one here) show that the WDM significantly outperforms all the other mentioned approaches. The method shows especially great performance for low SNR, where good results of the spike detection methods are especially important. Thus, a detail comparison with the WDM is given, using the code provided from authors of [Nenadic 2005].

2.3.2.6 Comparison with WDM

The free parameters for the both methods are chosen so as to have the best performance. For the proposed method we set $\nu = 7$, $K = 3$ and $T = 4ms$ for the estimation time. For WDM we set the number of scales to 6, used biorthogonal mother wavelet function and set the minimum spike width to $1ms$ and the maximum to $3.33ms$ (see [Nenadic 2005] for details on practical meaning of these parameters).

Figure 2.25 below displays the ROC curves. These are again computed by averaging results over 500 runs, for each pair (FR, SNR). The ROC curves of the proposed method are almost always higher than that of the WDM for all pairs (FR, SNR). This is more apparent at high noise level (SNR = 3, first

column on figure 2.25) for $P_{cd} > 0.5$. The proposed method thus shows good robustness to noise perturbations. Moreover, it is simple: for a firing rate of FR = 45Hz, the average time per run was $0.48s$ against $1.52s$ for the WDM, *i.e.* more than three times.

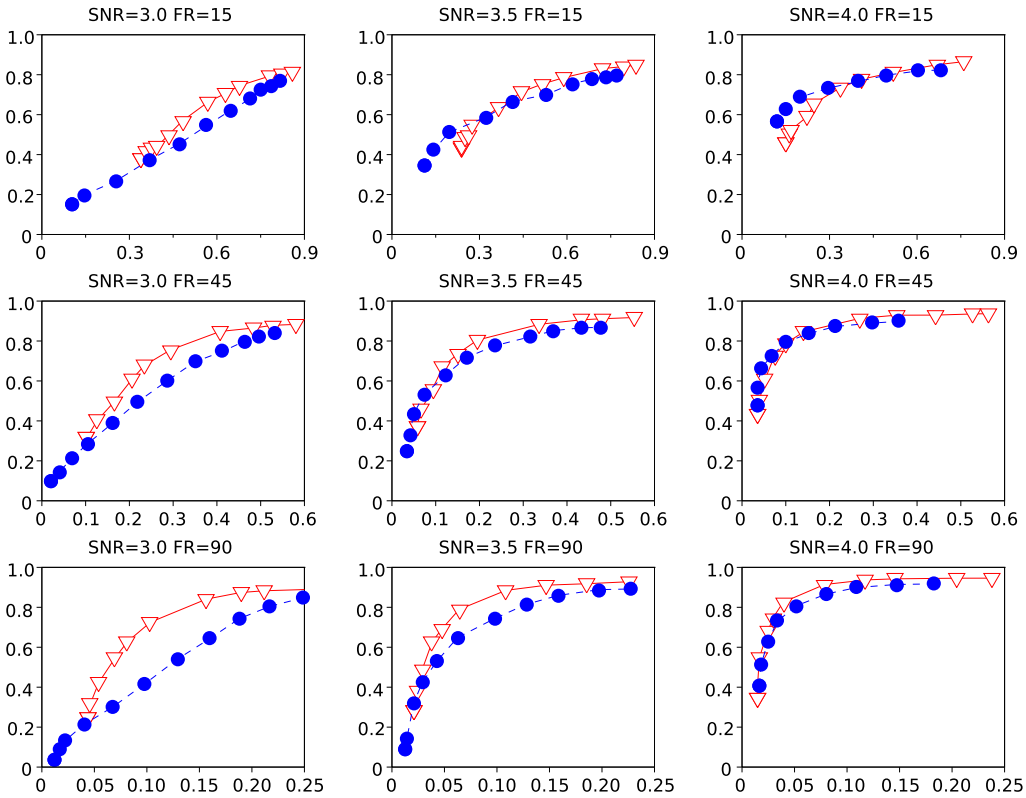


Figure 2.25: ROC curves: Comparison of the proposed method (▼ mark) with WDM (• mark). Probability of the false alarm is given on the horizontal axis and the probability of correct detection on the vertical axis

Next, on the top graph of figure 2.26, we fix the thresholds and plot P_{cd} against P_{fa} for SNR $= 3.5, 3.6, \cdots, 4$ and FR $= 15, 30$ and 45Hz. Although the two methods show quite different behaviors, it is possible to compare their performance via the ratio P_{fa}/P_{cd} as a function of the SNR. This is displayed in the bottom graph, for FR $= 15, 30$ and 45Hz. Not only this ratio is always smaller for our method but also, it decreases faster when the SNR increases. The implications of these results will be discussed in section 2.5.

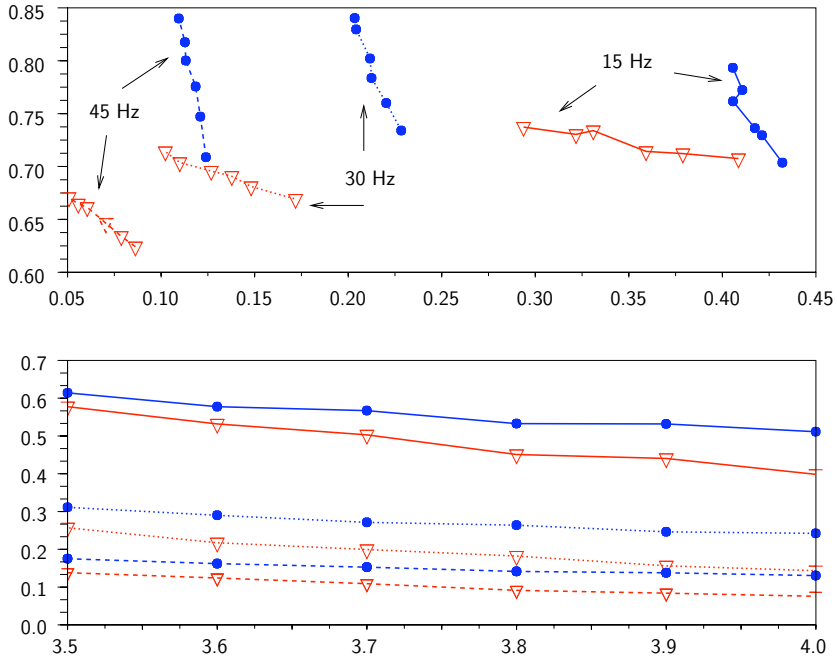


Figure 2.26: Comparison of the proposed method (▼ mark) *vs* WDM (● mark). The top graph shows P_{cd} against P_{fa} for SNR $= 3.5, 3.6, \cdots, 4$ and FR $= 15, 30$ and 45Hz. The bottom graph show P_{fa}/P_{cd} as a function of the SNR for the same FR as the top graph.

2.4 Case study: eye blink artifacts detection in EEG

We proposed the approach for spike detection, but this is not the only possible application of the devised computational framework. Actually, it can be applied in general change point detection problems. As an example, in this section the algorithm will be applied on EEG recordings (see chapter 1 for more information of EEG recordings in general). The practical purpose of the particular application is to detect what segments of an EEG recording are affected by artifacts caused by eye blinking. This case study has also been described in [Tiganj 2010].

2.4.1 Problem description

In EEG recordings the electrical activity of the levator muscle (which controls the upper eyelid and is responsible for eye blinks), projects on vectors pitched by the EEG electrodes causing artifacts. The eye blink artifacts affect more then 10% of the average human EEG (human in average blinks once in 5s,

and the duration of an artifact is more then $0.5s$), thus these artifacts are a very important problem in the EEG signal processing.

The amplitude of the artifacts depends on the amplitude of the levator muscle activity and on the brain region where the EEG is recorded. Typically, the amplitude has the same order of magnitude as the recorded brain activity. Thus, during the eye blink EEG recording is useless for most of applications. On the other hand, the artifacts can not be detected only by setting a threshold.

Common approach for eye blinks detection is placing extra electrodes close to an eye. Signal from these electrodes is amplified with differential amplifiers, then setting an appropriate threshold is enough for the artifacts detection. Moreover, if several EEG channels are recorded, blind source separation is often used, not only to detect the artifacts, but also to reject them from the EEG data [Iriarte 2003].

However, placing extra electrodes close to the eye or using EEG with several channels is not suitable in some applications. Upcoming consumer application based on EEG such as brain-computer interface [Dornhege 2007] [Thulasidas 2004] or biofeedback devices [Monderer 2002] and [Othmer 2000] are mostly intended for non-clinical, everyday home use. Simplicity of use and a low price are essential for those devices. Therefore, use of singe-channel EEG systems is very common in such applications [James 2003]. Some clinical applications such as recording on babies [Walls-Esquivel 2007] or Alzheimer's disease recognition [Cho 2003] are often based on single channel EEG systems as well. In this case study the problem of the eye blink artifacts detection in single channel systems is addressed.

2.4.2 Results

The proposed algorithm has been validated in the eye blink artifacts detection using 7.5 minutes of 32-channel EEG recording, at the sampling frequency of 1kHz. The electrodes were placed according to the international 10-20 system (figure 2.27, reproduced from [Malmivuo 1995]). The recording was unipolar with the reference calculated as the averaged sum of all 32 electrodes. The ground electrode was placed on the forehead. All electrode impedances were less then $5k\Omega$. In order to have the referent information on when the examinee actually blinks, apart from the EEG, two additional electrodes were used. One was placed above and the other below the examinee's right eye. In 7.5 minutes of the recording we detected 65 eye blinks. On figure 2.28 100 seconds of the recording (all 32 channels, together with the eye activity recording *i.e.* electrooculogram (EOG)) are shown. The eye activity is clearly visible on the most of the recording channels. The projection is the largest on the channels

FP1 and *FP2*, which are closest to the eyes. On the other side, the projection is hardly visible on *FC1*, *FC2*, *FC5* and *FC6*.

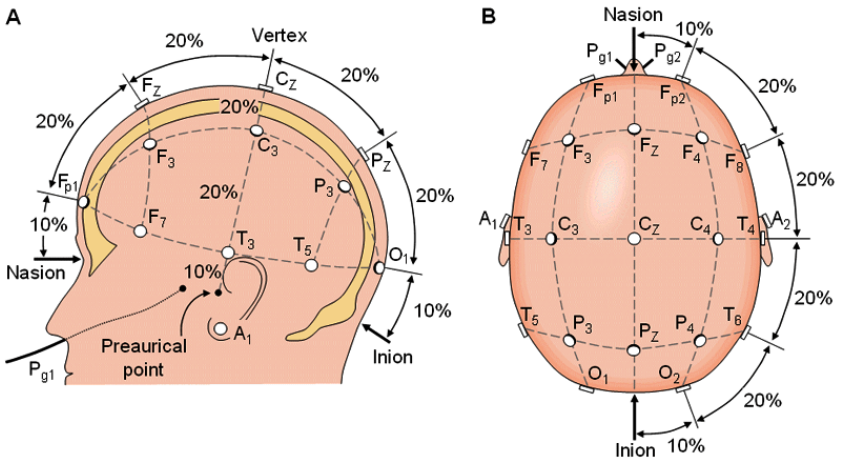


Figure 2.27: The international 10-20 system seen from (A) left and (B) above the head. From [Malmivuo 1995].

The described spike detection algorithm, with the decision function $|v_\kappa(\tau)v_{\kappa+1}(\tau)v_{\kappa+2}(\tau)|$, is applied on this recording. The free parameters were chosen as follows: Parameter T is set to 0.5s, what is a typical duration of the eye blink artifacts; Parameters ν and κ were varied in order to find which combination leads to the best results; Finally they are set to $\nu = 4$ and $\kappa = 2$.

Since we address the problem of the eye blink detection and localization in single channel EEG, the algorithm is applied on each of the 32 channels separately. Because the recorded signals are from different brain regions, we are able to evaluate how different electrode positions affect the results.

The results of the spike detection with the proposed algorithm on the recording from figure 2.28 are given on figure 2.29. The original EOG plot is again given on the top as the reference information. For most of the EEG channels, we can say that the parts of the signal that correspond to the eye activity (spikes) are emphasized with respect to the rest of the signal. Only for the channels *FC1* and *FC2* the proposed algorithm did not significantly help in the spike detection and localization.

In order to asses the efficiency of the proposed algebraic approach we compare it with three other approaches:

- The basic threshold approach, where it is assumed that the eye blink artifacts have the largest amplitudes in the recorded signal and only a threshold is applied to detect them.

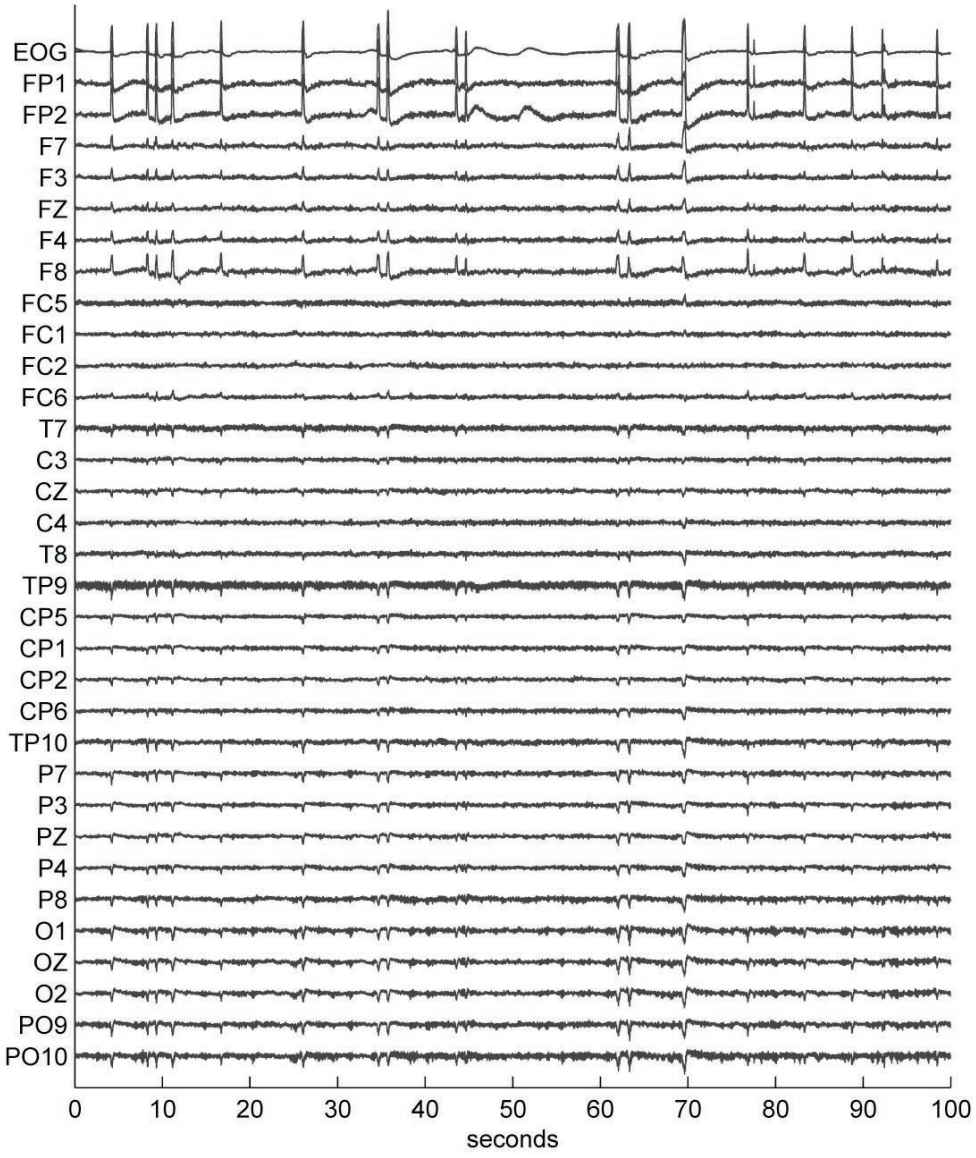


Figure 2.28: 100 seconds of EOG and EEG (32-channel) recordings.

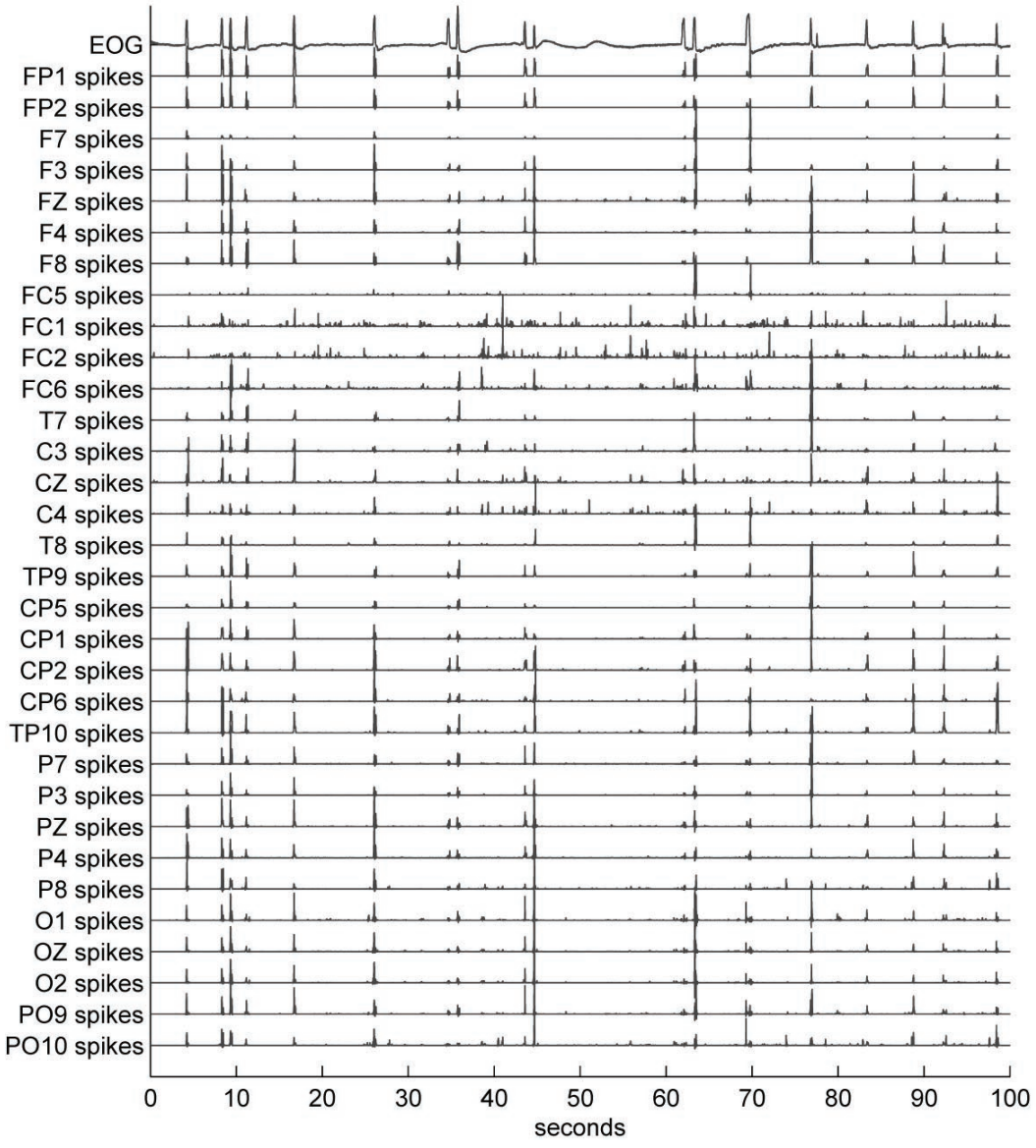


Figure 2.29: Results of the proposed spike detection algorithm on 100 seconds of 32-channel recording given on fig 2.28. EOG signal on the top is plotted as a reference.

- The approach that combines median filtering, matched filtering and NEO (see [Land 2002]).
- WDM, already mentioned and described in [Nenadic 2005].

For the second and third approach all the free parameters were varied in order to find which combination leads to the most accurate results; this combination was used in the simulation. The final results for each of the four approaches are presented with 32 ROC curves (figure 2.30), one for each channel. Every curve is made from 50 points that are calculated for different threshold levels. The detection is taken as correct if the eye blink occurred in 1s window around the point that is found to be above the threshold level. For each of the 50 points, the probability of correct detection is defined as the ratio between the number of correct detections for a given threshold and the total number of spikes. The probability of false alarm is defined as the ratio between the number of false alarms and the maximal possible number of false alarms for a given threshold. From the results shown on figure 2.30 we conclude the following:

- Since electrodes FP1 and FP2 are located very close to the eyes all four approaches have almost perfect detection rates (all eye blinks are detected with almost no false alarms).
- Poor detection results for channels FC5, FC1, FC2 and FC6 suggest that the projection of the eye muscle activity is very low in this region, what leads to conclusion that eye blink detection is not necessary in some cases.
- Advantages of using either algebraic approach, wavelet approach or combination of median filter, matched filter and NEO are the greatest when the algorithms are applied on signals from the electrodes that are placed further from the eyes (from TP9 to PO10 on figure 2.30). Here we notice the property of the algorithms to emphasis the spike shaped waveforms in the noisy environment (since threshold approach gives poor results the recorded signal is obviously very noisy in these brain regions). For the electrodes F7, F3, FZ, F4 and F8, where the projection of the eye activity is larger (since they are located closer to the eyes), threshold approach performs similar to the other three approaches.

2.5 Discussion

In this chapter we launched a discussion on spike detection and developed a computational framework which leads to a simple detection and estimation

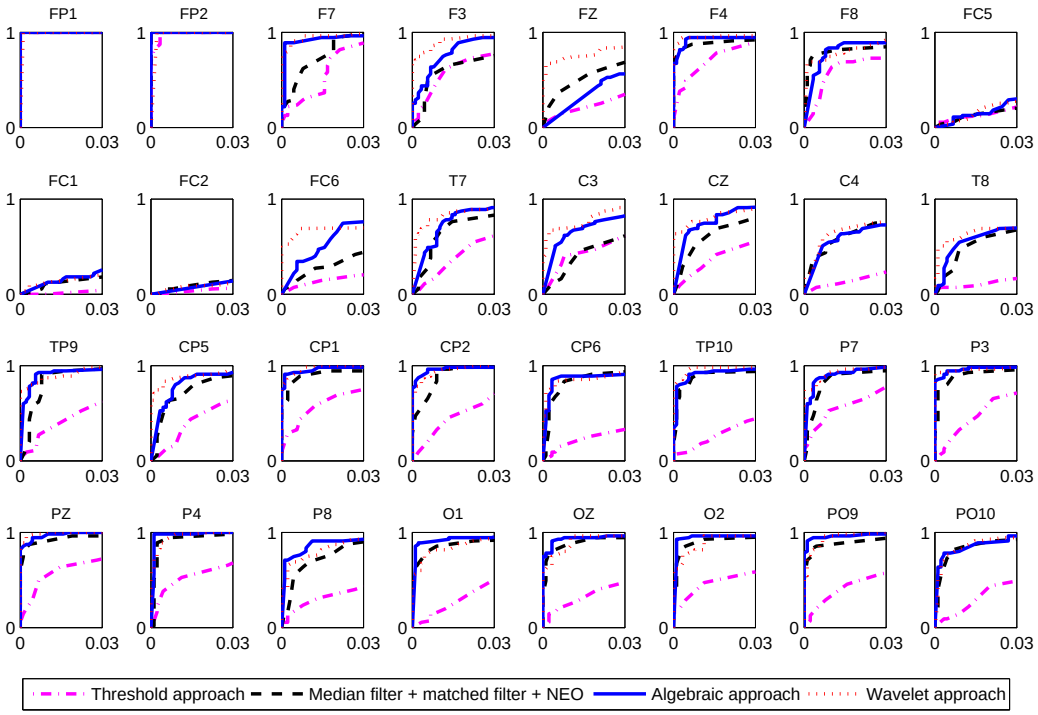


Figure 2.30: ROC curves: Comparison of the proposed method with WDM, a method based on combination of matched filtering, median filtering and NEO and simple threshold approach. Probability of the false alarm is given on the horizontal axis and the probability of correct detection on the vertical axis.

method. The approach is based on modeling polarity change within a spike as a jump in the signal. By taking derivatives of the noise-free part of the signal, using the algebraic framework, we obtain an explicit solution for each spiking instant. Large number of the free parameters enables us to adjust the detection system to our particular needs (*e.g.* trade-off between false alarms and correct detections). Our explicit solution is given through a linear system of equations which coefficients are in some sense derivative estimators of the signal. Even though many different detections systems are feasible we base ours on observing when the system will be degenerated, what indicates that there is no jump in the observed interval of the signal. Such detection system is implemented as a discrete Volterra filter. We found that it shows good robustness to the noise perturbations. We compared it with one of the most successful neural spike detection method, based on wavelet transformation, described in [Nenadic 2005]. The method compares favorably, especially for low SNR, with WDM which stands as one of the most successful neural spike detection methods.

Visualization of the detection results is given using simultaneous intra-extra-cellular recording. On such recordings it is not possible to quantify the detection accuracy, since we do not know all the spike locations. However, the obtained results give quite a good insight on applicability of the proposed method on real signals.

We continued the simulations with the synthesized recordings. For the synthesized recordings the spikes are given as spike templates, created by averaging spikes from the real recording. The noise is extracted from the real recording as well. It is very likely that the noise does not contain any spikes from the neurons close to the electrode, since we removed those from the recording by setting a very low threshold. The difference between the spikes and the noise is thus very clear. Of course, using synthesized recordings we know precisely which spike is fired by which neuron, so the results of spike detection can be quantified.

By changing the SNR the problem of spike detection can be made more or less difficult, but the actual spike locations are still always known. Drawback of this way of comparison lies in a possibility that by decreasing the SNR some spikes from distant neurons, that are considered as the noise, will become as large as the spikes from the neurons considered to be close to the electrode.

One interesting result of the performed simulations is that the proposed method has approximately the same P_{CD} for different noise levels, while the P_{FA} is changing (opposite to the WDM where P_{FA} is stable). Such property is preferable for some of spike sorting techniques (which use the results of spike detection as their input). This is because the cluster that contains the spikes with the lowest average amplitude is often neglected in spike sorting, since it is likely that it has a large number of falsely clustered spikes. Such philosophy will be used in the next two chapters where we will address the spike sorting problem. On the other side, if spike sorting technique does not neglect the cluster with the smallest spikes, the properties of the WDM approach would be preferable.

From ROC curves given on figures 2.23 and 2.24 we can see that the performance of the proposed algorithm remaind stable, even for very different SNR. It is important to emphasis that for each of the two ROC simulations the same set of the free parameters for the proposed method has been used. This suggests that the method is rather robust. However, we do not provide an explicit way to compute the free parameters, but only give their physical interpretation on the intuitive level. Exact values of the parameters should be chosen experimentally, since their optimal values depend on the given data and on the desired detection selectivity.

Regarding the simulation setup, using higher or lower SNR would not contribute much to usefulness of the results with respect to any practical

application. For lower SNR even if more spikes would be detected it would be very difficult to sort them, since their shape would be almost entirely destroyed. For higher SNR the comparison is also not necessary because all the algorithms, even simply setting a threshold, would perform with almost perfect accuracy.

Finally, we can say that even though the calculus behind the proposed method seams complex, the implementation of the method is rather simple as a discrete Volterra filter. The method is general purpose, thus no particular model is required, only an approximate spike duration should be known. The detection selectivity can be tuned with free parameters T, κ and ν. Overall, the performance of the algorithm is fast and the obtained results suggest relatively high accuracy, especially for low SNR. Since we have proposed the whole framework in this chapter, there are lot of possibilities for different variations, maybe development of more accurate detection systems.

Chapter 3

Spike sorting in single-site recordings

Contents

In the previous chapter we addressed the problem of spike detection. Now having the spikes detected we usually want to sort them: The sorting amounts find which neuron fired which spikes. This problem will be addressed in this and the following chapter. In this chapter we will concentrate on spike sorting from single-site recordings. In the next chapter we will investigate the problem of spike sorting when multi-site recordings are available.

Even though the basic form of an action potential is the same for all neurons, the spikes recorded with an extracellular electrode have different shapes and amplitudes if they are coming from different neurons [Henze 2000]. This property, which is due to the propagation and the velocity effects, is on the basis of spike sorting methods (for review, see: [Lewicki 1998] and [Gibson 2008]). Therefore, it is argued that the information transmitted by a given neuron is carried by the time occurrences of the action potentials and not by their amplitude or shape. This makes the problem of spike sorting easier, because it is not necessary to recover the original waveforms of the action potentials, but only to find the firing instants of the sorted neurons.

As we said, in this chapter we consider to have only a single-channel recordings, but the spike sorting algorithms that are commonly applied on single channel recordings are applicable on multi-channel recordings as well, usually as a combination with some Source Separation methods (for examples see [Takahashi 2003a] and [Mamlouk 2005]) - we will see this closely in the next chapter. Say we have a single-channel extracellular neural recording that consists of N samples and let each spike has a duration of M samples. A generic spike sorting algorithm consists of several steps:

1. `Spike detection`: In context of spike sorting, the goal of spike detection is to extract from the recorded signal all the spikes fired by the neurons close to the electrode. Assume that K spikes are detected in this step.

2. `Feature extraction`: Principal Component Analysis (PCA) [Adamos 2008], wavelet decomposition [Quiroga 2004], [Hulata 2002] or some other technique (e.g. [Ghanbari 2009], [Chah 2011] and [Rogers 2007]) are commonly used to reduce dimensionality of the $M \times K$-matrix by extracting the most important features of the detected spikes. The result is a new matrix of reduced dimension, $L \times K$, where $L < M$ is the number of extracted features per spike.

3. `Finding the number of clusters (the number of neurons close to the electrode)` *i.e.* `model selection`: K detected spikes could be fired by one or more neurons. This step is very often done manually, where user after observing the $L \times K$ matrix (often, L is equal to 1, 2 or 3 so the matrix can be visualized) chooses the number of clusters. But some general purpose parametric model selection algorithms [Lavielle 2005] based on penalization such as Bayesian Information Criterion (BIC) are also proposed as more efficient (see examples in [Novak 2009], [Fraley 1998] and [Pouzat 2002]). The result of this step is the estimated number of different neurons, call it P, that fired the K detected spikes.

4. `Sorting`: Finally, the K detected spikes are sorted into P clusters. Again general purpose techniques, such as K-means [Takahashi 2003a] or Expectation maximization (EM) [Wood 2004], are usually applied [Lewicki 1998].

Some other, very different approaches, also exist, (e.g. [Delescluse 2006]), but they are mostly applicable only in special cases when some specific prior knowledge (e.g. interspike interval histogram) is available. Because such knowledge is available only for some special types of neurons and particular tissues, we will not analyze those methods here, since we tend to address the spike sorting problem in its general form.

Due to a typically unbalanced structure of the clusters, finding their number and sorting the spikes into the clusters (steps 3 and 4) are usually very ill-conditioned. The unbalanced structure is mainly caused by two reasons: 1) The firing rates of different neurons are usually very different (up to hundred times) 2) The distances between the centers of the clusters can also be very different.

To illustrate better the difficulty of model selection in neural recordings let us discuss on implementation of BIC. Consider that sorting is done with

EM, k-means or any other clustering algorithm. To do the model selection with BIC the clustering algorithm is run several times, once for each realistic number of clusters. For each run the corresponding maximized expected likelihood is calculated. Since increasing the number of clusters naturally increases the maximized expected likelihood, it is necessary to penalize the increase of the number of clusters. Finding an optimal penalization is particularly ill-conditioned since it is usually chosen only once for one type of recordings while the morphology of the feature space can vary significantly from recording to recording.

On the other side accuracy of manual determination of the number of clusters is also not satisfying [Harris 2000]. Moreover manual approach is very expensive and hard to perform on large sets of data.

An example of bad results when general purpose clustering methods are applied is shown on figure 1. We used artificial neural data to be able to verify the clustering results. The dimensionality of the spikes is reduced to 2, using PCA. The first plot on the figure 3.1 displays the original clusters, corresponding to three different neurons (spikes coming from the same neuron are marked with the same shape and color). For model selection we used BIC, as described above, which is known to often overestimate the number of clusters, and obtained result that the data contains 6 clusters. The results of clustering in 6 clusters with k-means is shown on the second plot on figure 3.1 and with EM on the last, third plot.

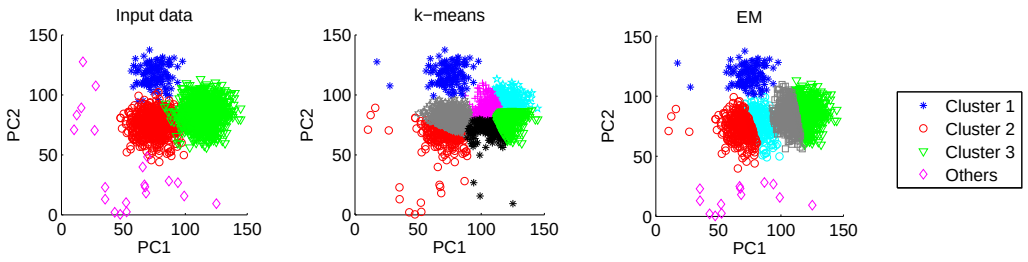


Figure 3.1: Sorting example on simulated neural data with combination of BIC with k-means and EM.

Even if the number of clusters is determined correctly, sorting the spikes into the corresponding clusters is another very difficult problem for neural data. The principal reason for this is that neural data can not be accurately clustered based only on the distance to the center of a cluster, now this is the most important parameter among the parametric clustering methods. This is illustrated on figure 3.2 where the same data sample consisting of only three clusters was used. The correct number of clusters was given to both

k-means and EM methods, but still the sorting results are not as desired. We must mention that here and in the later simulations in this chapter k-means and EM are run only once, with random positions of the initial centers. Running the algorithms several times with different positions of the initial centers and choosing the result with the largest likelihood would lead to better performance of these algorithms.

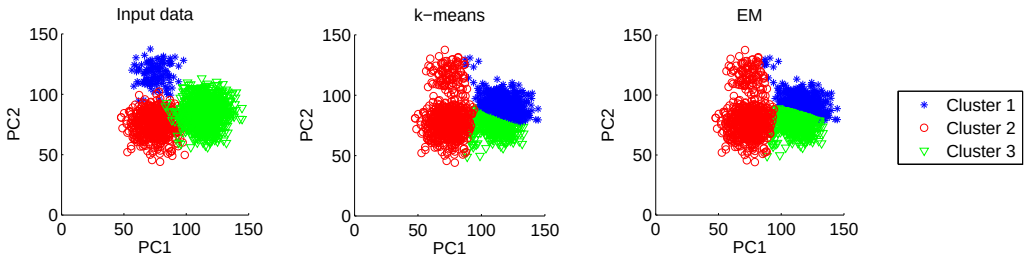


Figure 3.2: Sorting example on simulated neural data with using k-means and EM with the number of clusters priori fixed to 3.

In this chapter we propose a new method for finding the number of neurons that fired the detected spikes and for associating each of the spikes with a corresponding neuron (steps 3 and 4 above). The method is also described in [Tiganj 2011b]. We do not address questions of *e.g.* spike detection or feature extraction, but to demonstrate our results we will use techniques commonly applied for these problems. The proposed method is non-parametric. It is designed particularly for the neural recordings, with the aim of high robustness with respect to mis-classification.

The method is based on two assumptions on the recorded signal:
a) A lower bound on the number of spikes fired by a single neuron during the recording time is known. This is a very realistic assumption since neurons (especially those whose activity we actually want to detect) fire usually at least once per second.
b) The recorded signal has a unimodal distribution. This is easy to verify experimentally, but it can also be shown theoretically upon invoking the central limit theorem.

The basic idea behind the proposed method is that most of the spikes fired by the same neuron, after projection on feature extraction vectors, will be concentrated around the center of the corresponding cluster. As we go to the edges of each cluster the density of spikes decreases. Smoothing the vector space that contains the K detected spikes (represented in L-dimensions) with moving average filter allows one to emphasis the differences between clusters and leads to easy automatic detection of the centers of the clusters. To assign

spikes to a corresponding cluster we use a simple greedy algorithm: each cluster is initialized by its singleton center and the clustering is performed by classifying one spike at a time, in such a way that if an element belongs to a cluster then so is its nearest neighbor. This procedure leads to a very robust solution, which is not sensitive to unbalanced sizes of clusters.

The property of unimodal distribution of neural data is exploited in several neural spike sorting algorithms. In [Vargas-Irwin 2007] it has been used to find spike templates which are then correlated with all the detected spikes in order to emphasis the differences between clusters. Such template matching can be used as a preprocessing step of the method proposed in this chapter, since template matching can generally improve the results obtain by feature extraction methods. Different peak-valley search algorithms, such as histogram peak count, are also based on distributional properties of neural data (see *e.g.* [Sebald 2007] or the implementation in a commercial system described in [Bla 2009]). Typically, these algorithms sort all the spikes at once by estimating the morphology of the area with lower spike density, giving the borders between clusters explicitly. The implementation can thus be very complex, especially in higher dimensional space. Also, when more than two clusters exist, the vector space becomes very difficult to analyze with such methods. By clustering one spike at a time, where the borders are given implicitly, with only the filter length and height to set, the proposed method gives a simple and yet robust alternative.

A detailed description of the method is given in the following section. We apply the method on extracellular recordings and use the simultaneous intracellular recordings to verify the results of the sorting for the neuron recorded intracellularly. Such way of validation represents another contribution of this chapter, since spike sorting methods are typically validated only on the simulated signal. In section 3.2 we describe the signal that we use for the validation, which is available online [Henze 2000]), give all the simulation parameters and results of the comparison with the state-of-the-art approaches. Conclusions and some future perspectives for development of this method are given in section 4.5.

3.1 Methodology

In this section we will describe our method step-by-step. For the demonstration we use a simulated signal for which the actual activity of each neuron is known. We can therefore display the results of the proposed method. As we said in the previous section, we assume to have an N-samples long single-channel extracellular neural recording. Our goal is to find the firing patterns

of as many individual neurons as possible (for single-channel recordings this usually means activities of two to three neurons closest to the electrode). Example of simulated extracellular recording is displayed on figure 3.3. We labeled the spikes that come from different neurons with different symbols.

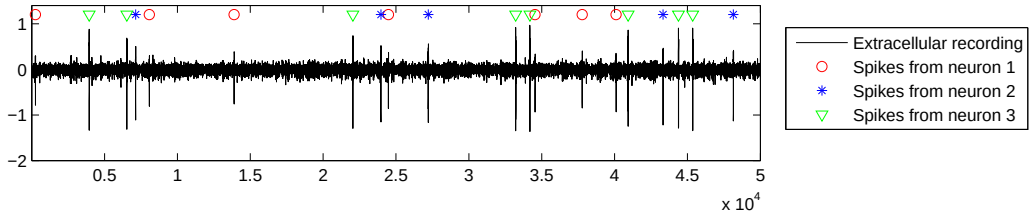


Figure 3.3: Simulated extracellular neural recording with marked spike locations from three neurons closest to the virtual electrode.

3.1.1 Preprocessing: Spike detection and feature extraction

The first preprocessing step is the spike detection. Depending on the SNR, setting a threshold directly on the extracellular recording or applying some more advanced spike detection technique can be considered. We will use the method described in the previous chapter.

Assume that we detected K spikes that come from a total of P neurons (for us P is unknown). To each detected segment $x_k, k = 1, \cdots, K$, we associate one neuron, say neuron $\#j_k, 1 \leqslant j_k \leqslant P$. This is, among all neurons which are active during that segment, the closest one to the electrode. We can then write each segment as: $x_k = w_{j_k} + n_k$, where w_{j_k} is the spike waveform coming from the associated neuron and n_k is the contribution of all remaining neurons that were active during the detected segment x_k. Figure 3.4 shows the first 5 segments x_k extracted from the signal on figure 3.3 and displays them as a sum of w_{j_k} and n_k.

Since each extracted segment has the same length of M samples, all the segments can be given in $M \times K$ matrix. So now we have each detected spike described by M samples (M is usually between 50 and 100, depending on the spike duration and the sampling period). But each spike can be described by *e.g.* its amplitude, or its minimal and maximal value. Using only these information it can be easier to find which spikes are fired by the same neuron than by observing all the M samples. Generally speaking, we can try to extract some features that give a compact description of the spikes. There are many techniques for feature extraction that try to extract one or more

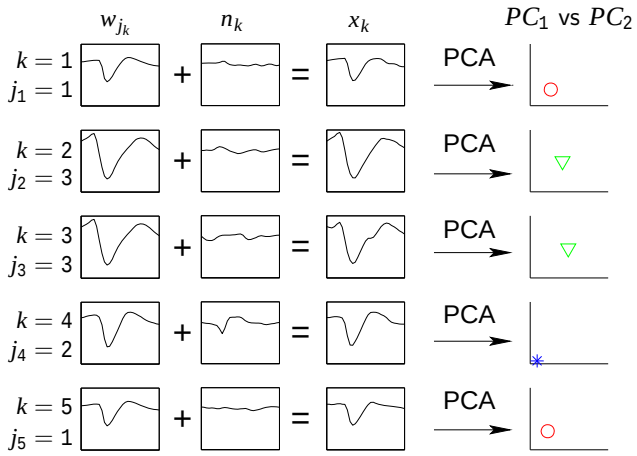


Figure 3.4: First 5 spikes from figure 3.3 shown as a sum of original spike waveforms and noise. Each spike is then projected on the first two principle components. The last column shows the values of these projections in PC1 vs PC2 basis.

of the most important features. In this chapter we will use PCA, which seems to be the most popular one, but also some other techniques could be used (good results are reported *e.g.* when wavelet decomposition is used [Quiroga 2004] and new techniques for feature extraction in neural recordings are under constant development see *e.g.* [Ghanbari 2010]). The algorithm we propose in this chapter is compatible with all of these feature extraction methods. PCA takes the variability in the data as the most interesting feature and projects the data in such a way that the first basis vector (principle component) has the variance as high as possible. Each succeeding principle component in turn has the highest variance under the constraint that it is orthogonal to the preceding components. We project our $M \times K$ matrix on L principal components. This projection results in new $L \times K$ matrix. For the simplicity of this description, we will assume the choice of $L = 2$ (only two features are extracted from each detected segment) and call the corresponding matrix $D_{2\times K}$. An example is shown on figure 3.4 where the last column gives the projection of x_k on the first two principle components. Notice that the projections are similar for spikes coming from a same neuron. All the extracted spikes projected on the first two principal components (all the elements of matrix $D_{2\times K}$) are shown on the left plot on figure 3.5. Same as for figure 3.1, the projections of the spikes that belong to different neurons are marked with different symbols and colors.

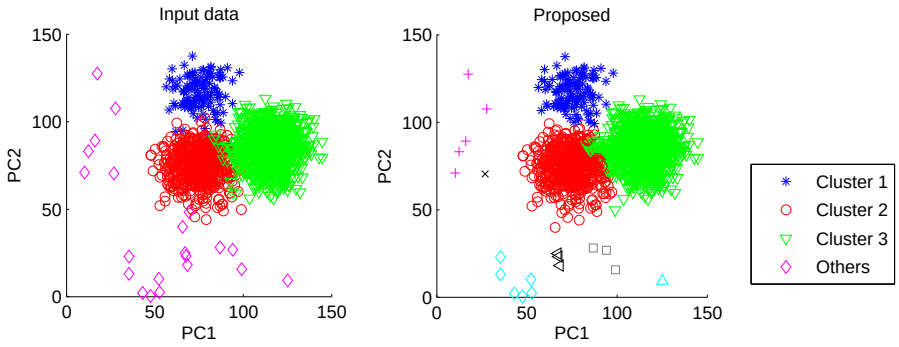


Figure 3.5: The left-side plot shows the initial data set (same as for figure 3.1). The right-side plot shows sorting results with the proposed method. The elements of each found cluster are labeled with symbols of the same shape/color.

3.1.2 Unimodal distribution

Now having $D_{2\times K}$, we need to find out how many neurons fired the spikes we detected and to associate each spike to one of the neurons.

To find the centers of the clusters we will first filter the 2D vector space. One key point of the method is to assume that the maxima of the filter's output correspond to the centers of the clusters. Such assumption obviously implies that each cluster has a unimodal shape (in L dimensional space) and also that the centers are far enough so that the unimodal shape is not destroyed by an overlapping of clusters. If this is not the case we would find multiple centers in a single cluster. We will now discuss whether these properties are realistic for neural data.

The application of PCA on each detected segment x_k can be written as a scalar product between x_k and, in our case, only the first two principal components (PC_1 and PC_2):

$$\langle x_k, PC_i\rangle = \langle w_{j_k}, PC_i\rangle + \langle n_k, PC_i\rangle = \alpha_{j_k,i} + \beta_{k,i}, \quad i = 1, 2.$$

Each spike waveform w_{j_k}, for the same neuron j, will be transformed with PCA into exactly the same point in the 2D vector space ($\alpha_{j_k,i}$). As we have mentioned earlier, we consider that these points will be mutually distant enough so that the dispersive effect of the noise cannot induce any overlapping between clusters. So the success in finding the centers of the clusters depends on the difference between the spikes from the neurons we want to sort and the noise level. When performing a recording it is generally possible to influence on both of these parameters by adjusting the electrode position.

Considering the noise part, n_k is actually a weighted sum of activities from

a very large number of neurons. Even though the distribution of the activity of each individual neuron is obviously not unimodal, the distribution of the sum of a large number of these activities tends to be unimodal. This stems from the central limit theorem, assuming that the spiking activity of each neuron is independent to the others. Recall that the many spike sorting algorithms that we have mentioned in the introduction also assume the unimodal property. Moreover, this property is simple to verify experimentally by observing the cumulative distribution function (CDF) of a neural recording. We plot the empirical CDF of a neural recording and a Gaussian CDF as a reference on figure 3.6. The neural recording has unimodal distribution which is in fact close to Gaussian (notice that a proximity to the Gaussian distribution is not required by the proposed algorithm).

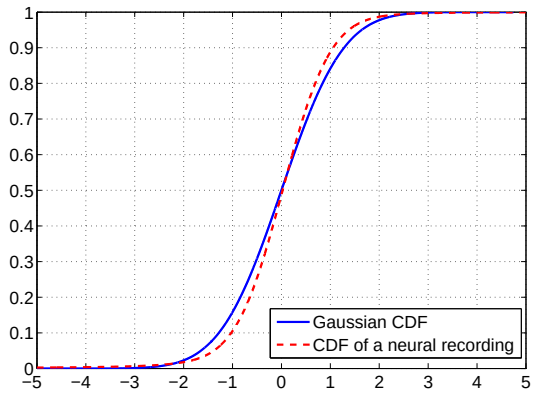


Figure 3.6: Gaussian CDF and empirical CDF of a neural recording.

Since PCA is a linear transformation $\beta_{k,1}$ and $\beta_{k,2}$ are just linear combinations of the elements of n_k, so the distribution of $\beta_{k,1}$ and $\beta_{k,2}$ is of the same type as that of n_k.

Finally, depending on the conditions mentioned above, we can expect that the density of $(\langle x_k, PC_1\rangle, \langle x_k, PC_2\rangle)$ will have a single maximum for a single cluster. Such morphology suggests the idea of sorting all the spikes into their corresponding clusters using a simple greedy algorithm which grabs the nearest neighbors: since the density of spikes decreases as we go further from the maximum, gathering spikes to the same cluster using the criteria of their proximity to nearest neighbor seems to be a simple solution. This conclusion is on the basis of the proposed spike sorting algorithm.

3.1.3 Finding the centers of the clusters

In this subsection we will describe the algorithm for finding the centers of the clusters. According to the previous subsection this amounts to find the areas of the highest density of the points $(\langle x_k, PC_1\rangle, \langle x_k, PC_2\rangle)$.

For this, we first assume that each active neuron fires at least, say, G spikes during the observation window. Most of the neurons fire at least once per second, so depending on the duration of the recording we can easily calculate G. Now knowing G, we construct a moving average filter that will smooth the obtained $2D$ K-elements space without merging different clusters. We assume that the $2D$ space is normalized so that we can consider a square domain for the filter's support. The filter should emphasis the differences between the centers and the border of each cluster. Its length (height), call it R, should be adjusted in a way that the filter window does not contain more than *e.g.* $G/2$ elements. In subsection 3.2.3 we will show that in practical situations R can be taken from a large range of values without affecting the clustering results. The result of filtering the data from figure 3.5 is shown on figure 3.7. Call the resulting matrix $D^{filt}_{2\times K}$. The centers of the clusters in the data matrix $D^{filt}_{2\times K}$ are easily found: any local maximum 1) which is centered in a domain at least larger than the filter window and 2) which is global in that domain, is a center of a cluster. In such way we are again sure that two clusters will not be merged into one. The centers found in this way are labeled with black dots on figure 3.7. The figure demonstrates that one center is found for each of the three big clusters. Several centers are found in the area distant from the biggest clusters, these actually represent the activities of neurons whose spikes are only partially detected. As we will see in the next subsection, since only a small number of spikes will be assigned to these clusters (less than G) they will be neglected and the spikes assigned to them left unsorted.

3.1.4 Associating each spike with the corresponding center

After finding the centers of the clusters, we have to associate each of the spikes with one of the centers.

To cluster the data we propose a greedy algorithm close in spirit to the agglomerative hierarchical clustering [Hastie 2003]. In spike sorting the agglomerative hierarchical clustering has already been considered in [Fee 1996] where single spikes are first sorted into an overly large number of clusters and after the clusters are progressively aggregated into a minimal set of clusters. Our approach is close in spirit, but yet significantly different from hierarchical clustering since we sort spikes one by one, instead of merging clusters. Thus

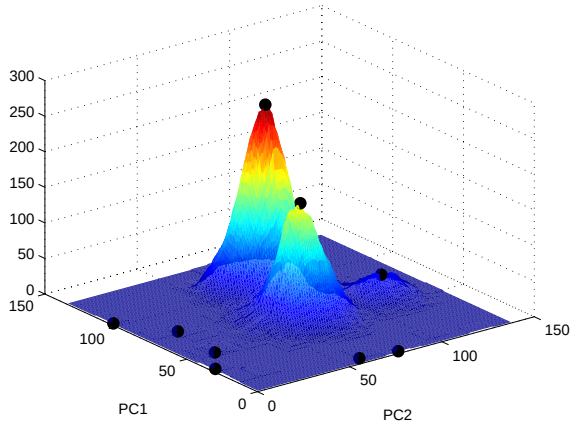


Figure 3.7: Filtered vector space from figure 3.5. Black dots denote clusters maxima.

we avoid the difficulty of devising a merging criterion for the clusters. Hence, the proposed approach is simpler and it can be implemented in an unsupervised way. We will exploit the fact that we know the centers of the clusters and that they are actually points of the highest density. Our basic statement is that if an element belongs to a given cluster, then its nearest neighbor also belongs to the same cluster. Throughout, we use the Euclidian distance.

Let $\tilde{x}_k$ be the projection of x_k on the L feature extraction vectors: $\tilde{x}_k = (\langle x_k, PC_1\rangle, \langle x_k, PC_2\rangle)$ and call $\mathfrak{D}$ the set of all $\tilde{x}_k$. Let $c_i, i = 1, \ldots, P$ denote the centers of the clusters, in an L dimensional space. The pseudocode 1 describes the algorithm for the construction of the cluster $\mathfrak{C}_i$ that corresponds with the center c_i. Therein, we use the notation $\texttt{dist}$ for $\texttt{dist}(x, \mathfrak{C}) = \min_{y\in\mathfrak{C}} \|x - y\|^2$. For the initialization, we set $\mathfrak{C}_i = \{c_i\}$ for all clusters.

Algorithm 1 Proposed method for associating spikes with corresponding centers

while $\mathfrak{D}$ not empty **do**
  for $i = 1$ to P **do**
    $\tilde{x}_i^{min} = \arg\min_{\tilde{x}\in\mathfrak{D}} \texttt{dist}(\tilde{x}, \mathfrak{C}_i)$ *For each cluster* $\mathfrak{C}_i$*, find the closest element* $\tilde{x}_i^{min}$ *of* $\mathfrak{D}$
  $j = \arg\min_{1\leqslant i\leqslant P} \texttt{dist}(\tilde{x}_i^{min}, \mathfrak{C}_i)$ *Find which cluster* $\mathfrak{C}_i$ *to update*

  $\mathfrak{C}_j \leftarrow \mathfrak{C}_j \bigcup \tilde{x}_j^{min}$ *Add the spike to the corresponding cluster*
  $\mathfrak{D} \leftarrow \mathfrak{D} \backslash \tilde{x}_j^{min}$ *Remove the sorted spike from the set* $\mathfrak{D}$

The right-hand side plot on figure 3.5 was obtained by applying this algorithm, using the centers obtained in the previous step. Comparing with the

left-hand side plot on figure 3.5, we can conclude that the proposed algorithm resulted in a quite good reconstruction of the three clusters. Several elements are misplaced but the result is quite satisfactory, especially in comparison with the one presented on figures 3.1 and 3.2. The spikes labeled as "Others" are broken into few clusters any of which is not large enough to exceed G, so they are all left unsorted. We consider this as a correct solution, since we do not want results that would contain only partially detected activity of some neurons.

In cases where we found only one cluster large enough to be kept, we can go back to the spike detection step and lower down the threshold value aiming to detect more spikes. If detecting much more spikes (extremely low threshold) does not result in detection of more than one large cluster, then it is a good indication that the recorded data are too noisy for such processing. This iterative approach can be used to find an optimal threshold in a given context for the proposed algorithm. In general, we can say that the cluster which contains most of the spikes with lowest amplitudes should be neglected, especially when high reliability of the results is needed.

3.1.5 Possible situations

Depending on the measurement conditions, we can have in practice three possible situations:

1) Spikes fired by two given neurons are significantly different. Consequently the extracted features are significantly different too (example is the first plot on figure 3.8). It is evident that the proposed approach will accurately find the centers and then sort the spikes into their corresponding cluster.

2) Spikes fired by the two neurons do not differ significantly (the second plot on figure 3.8). In this case the clusters are (slightly) overlapped: some spikes will be wrongly sorted, but we can expect that the majority will be sorted correctly. Since the density of the projected spikes is higher in the cluster's center than in the areas between the clusters, the proposed method will accurately find the centers of the clusters.

3) Spikes fired by the two neurons are very similar (the third plot on figure 3.8). Now finding the centers of the clusters is not possible with the proposed method. Even if the centers were known, an accurate sorting would still not be possible with the proposed greedy algorithm. This situation is typical for very noisy recordings. In such data, the proposed algorithm will usually detect only one cluster. As we have mentioned before, this can be used as an indicator that the recording is too noisy. A potential solution could be to project the data on a different number of principle components, reduce the noise sources,

do reposition of the electrode or use multi-channel measurement system.

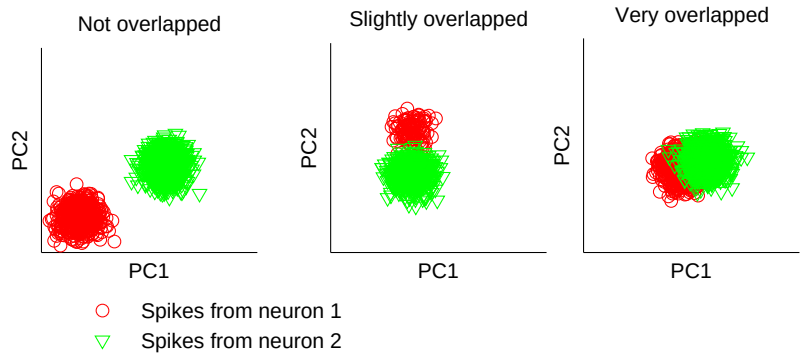


Figure 3.8: Three generally possible situations considering mutual distance between clusters of the neural data projected on the first two principle components.

It is important to mention that the results of the proposed method will be relatively conservative. This conclusion comes from the way of finding the centers of the clusters. Since, due to the small value of R, two clusters can hardly be merged into one, the number of spikes wrongly associated to a particular cluster (false spikes) will be relatively small. Since false spikes can be very confusing and misleading for different algorithms that aim to decode the communication between neurons, in the trade-off between the number of missed spikes and the number of false spikes it is highly preferable to have more missed spikes. See [Ilan 2008] for detailed argumentation and proof of this statement.

3.2 Simulation and results

To analyze the performance of the proposed method we apply it on real extracellular neural recordings. We validate the results using intracellular recordings made simultaneously with the extracellular ones. These results are compared with the results of the state-of-the-art approaches.

3.2.1 Use of simultaneous intra- extra-cellular recordings for the validation and comparison

As in the previous chapter, we use simultaneous intra- extra-cellular recordings from rat hippocampal area CA1 done by Henze et al., described in [Henze 2000]. These data are particularly interesting because they consist of simultaneous intra- extra-cellular recordings.

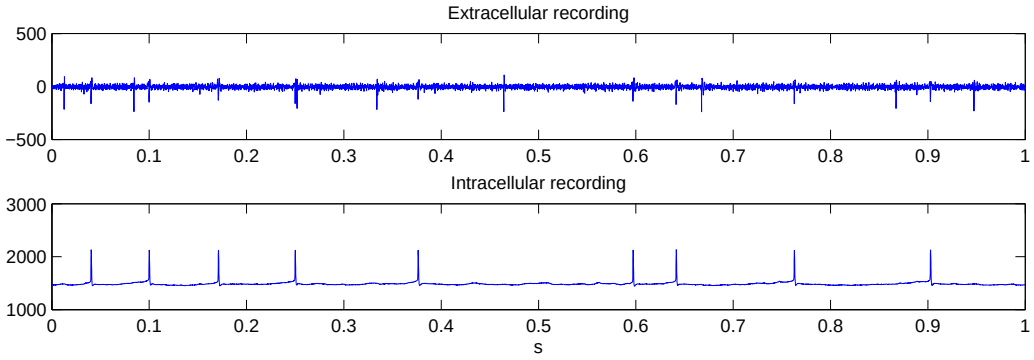


Figure 3.9: Simultaneous intra- extra-cellular recordings. The two electrodes are very close so the action potentials recorded with the intracellular electrode are clearly visible in the extracellular recording.

Single-channel extracellular spike sorting techniques usually result in sorting activity of two to three neurons. When the extracellular and the intracellular electrode are close enough, one of the neurons which activity is sorted is probably the one recorded intracellularly. Figure 4.1 shows such simultaneous recording. The top plot shows the extracellular recording and the bottom plot shows the corresponding intracellular recording. The action potentials fired by the neuron recorded intracellularly are clearly visible on the extracellular recording. This indicates that the electrodes are located very close to each other.

On the plot of the extracellular recording we can also see the activity of other neurons around the extracellular electrode.Notice that many of these spikes have similar amplitudes as the extracellularly recorded spikes that come from the neuron recorded intracellularly. Thus, the task of spike sorting is obviously not trivial in this case.

Finally, we see that once we apply the spike sorting techniques on the signal recorded extracellularly we can validate the sorting results for one of the sorted neurons, the one which is recorded intracellularly.

From the available database we selected 8 recordings that satisfy the condition that the action potentials from the neuron recorded intracellularly are clearly visible on the extracellular recording. Another criteria for the selection was that visual inspection of the feature vector space, after projection of the first two principle components, suggests that the sorting problem is not trivial (clusters far from each other) or too challenging (clusters completely overlapped). Even though the recordings were done with multi-site electrodes (4, 6 and 8 channels) since the algorithm proposed in this chapter is designed for single-site spike sorting we picked only one channel from each recording.

Table 3.1: Variances of the data projected on the first three principle components.

Recording	Variance of PC1	Variance of PC2	Variance of PC3
d533101 1	1.37	0.73	0.42
d533101 2	1.70	0.62	0.35
d533101 3	1.63	0.59	0.34
d533101 4	1.94	0.60	0.38
d12821001 3	5.51	1.88	1.34
d12821001 4	6.15	2.15	1.62
d1122109 4	0.99	0.30	0.24
d1122109 4	1.54	0.43	0.28

Of course the same algorithm can be applied on the signal from each recording site separately. However, in the next chapter we will combine independent component analysis with the algorithm proposed here to exploit the data redundancy in multi-site recordings. The recordings we used here have from around 2.5 millions to around 10 millions of samples with 287 to 1197 spikes from the intracellularly recorded neuron and from to 1409 to 4788 detected spikes in total. They are recorded with different types of electrodes and sampling frequency either 10kHz or 20kHz, what gives an important diversity of the data.

3.2.2 Preprocessing of the extracellular recordings

First we detect the spikes from the extracellular recordings. To emphasis the spikes coming from neurons located close to the electrode we apply a spike detection technique described in the previous chapter (see also [Tiganj 2009] and [Mboup 2008]). For each of the detected spikes we extract a corresponding 50 samples long data vector.

As the second step, we apply PCA for feature extraction. Figure 3.10 shows 8 plots, each corresponding with one of the signals (we kept the original signal names as in [Henze 2000]). The plots show the projections on the first three principal components. The spikes that are recognized as coming from the neuron recorded intracellularly are marked with red stars. The rotation angle is adjusted for each plot to make the differences between the clusters more obvious. We can see that the projections on the third principle component do not contribute significantly to the separation of the spikes fired by the neuron recorded intracellularly from the rest of the spikes. On the other side, the projections on the first two principle components are much more informative and enable us to see differences between clusters. This is also elaborated with table 3.1 which gives variances of the first three principle components for each of the 8 signals. Thus, in the following, we keep only the projections on the first two principle components, that is we set $L = 2$.

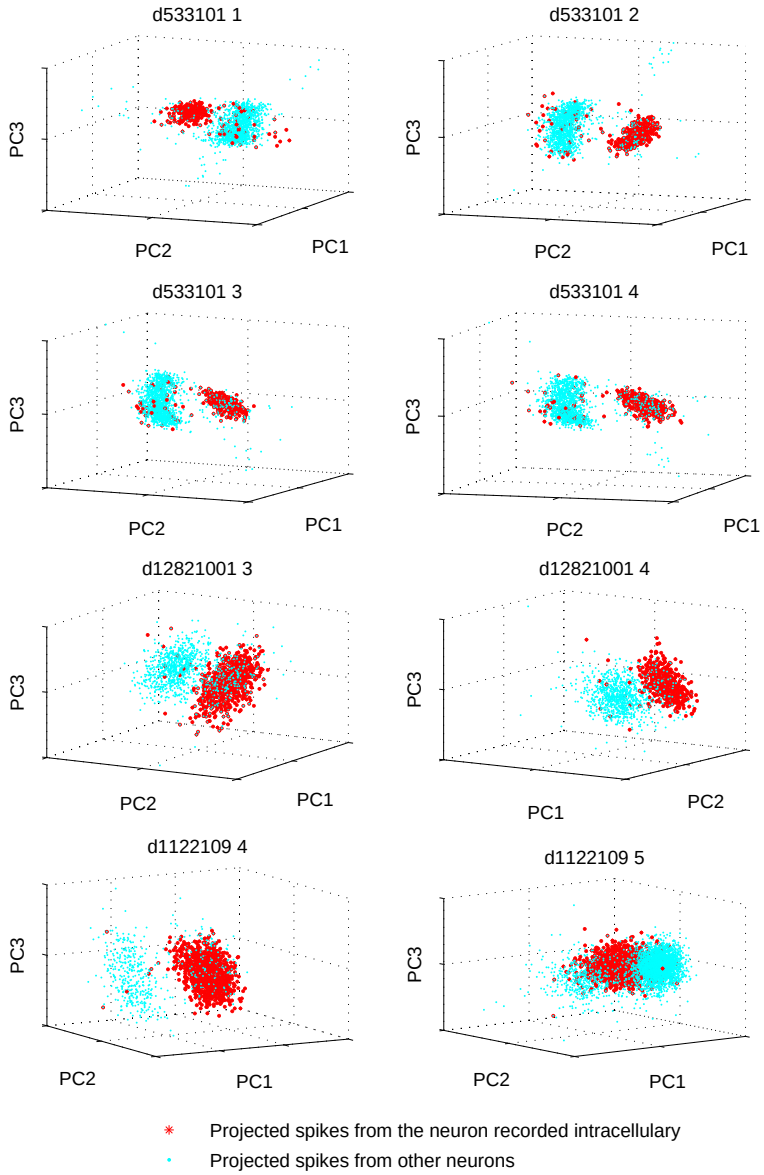


Figure 3.10: All the spikes from all 8 recordings projected on the first three principle components. Each plot is rotated in order to show differences between spikes from the intracellularly recorded neuron and all other spikes.

3.2.3 Clustering: applying the proposed algorithm

To ease the comparisons, we scale $PC1$ and $PC2$ in the range $[0, 100]$.

We assume a minimal firing rate of 1 spike per second and use a 2D filter with size (window width and length) corresponding to $R = 8$. For all the recordings this value of R and this minimal firing rate meet the requirement

of not merging two different clusters.

The centers of the clusters are found as the maxima of the filtered data, with a condition that two maxima are far enough from each other, as it is defined in subsection 3.1.3. All the spikes are finally clustered using the algorithm 1.

3.2.4 Results

We compared the proposed non-parametric approach with a combination of BIC with k-mean and EM algorithms and also with super-paramagnetic clustering [Quiroga 2004] which is *e.g.* used in spike sorting software called Waveclus. These stand as the state-of-the-art approaches and are the most widely used among all the spike sorting algorithms.

Figure 3.11 gives comparison of performance of the four algorithms on each of the 8 recordings. The spikes from the neuron which is recorded intracellularly are again represented with red stars in the first column. To quantify the clustering performance we observe how well the different algorithms managed to isolate that cluster. Since we do not have any knowledge on the activities of the other neurons we can compare the algorithms only in their ability to sort the spikes fired by the neurons recorded intracellularly (one neuron for each recording).

By observing closely the fist column on figure 3.11 we see that, after PCA, several spikes from the neuron recorded intracellularly are located far from the area that obviously contains most of the spikes from that neuron. This is especially the case for recordings `d533101 1`, `d533101 2`, `d533101 3` and `d533101 4`. We can also notice that several spikes that do not come from the neuron recorded intracellularly are in the middle of the area that contains most of its spikes. The best example is recording `d1122109 4`. Thus, we can not expect from any sorting algorithm to sort these spikes correctly, since they are obviously misplaced by the feature extraction technique. The results from figure 3.11 are also presented in table 3.2.

K-mean and EM based approach separate well spikes from the intracellularly recorded neuron for recordings `d533101 1`, `d533101 3` and `d12821001 3`. EM based approach performs solid also on recordings `d533101 2` and `d12821001 4`. For all the other recordings these approaches tend to split a single cluster into two or more clusters (*e.g.* results for recording `d1122109 4`). Using k-means and EM, isolated elements located far from the cluster are often wrongly associated with that cluster, *e.g.* for k-means recording `d12821001 3` and for EM `d533101 2`. Notice that we could actually obtain better results if we just fixed the number of clusters to two. However, those results would still be worse than the one of the proposed algorithm since there

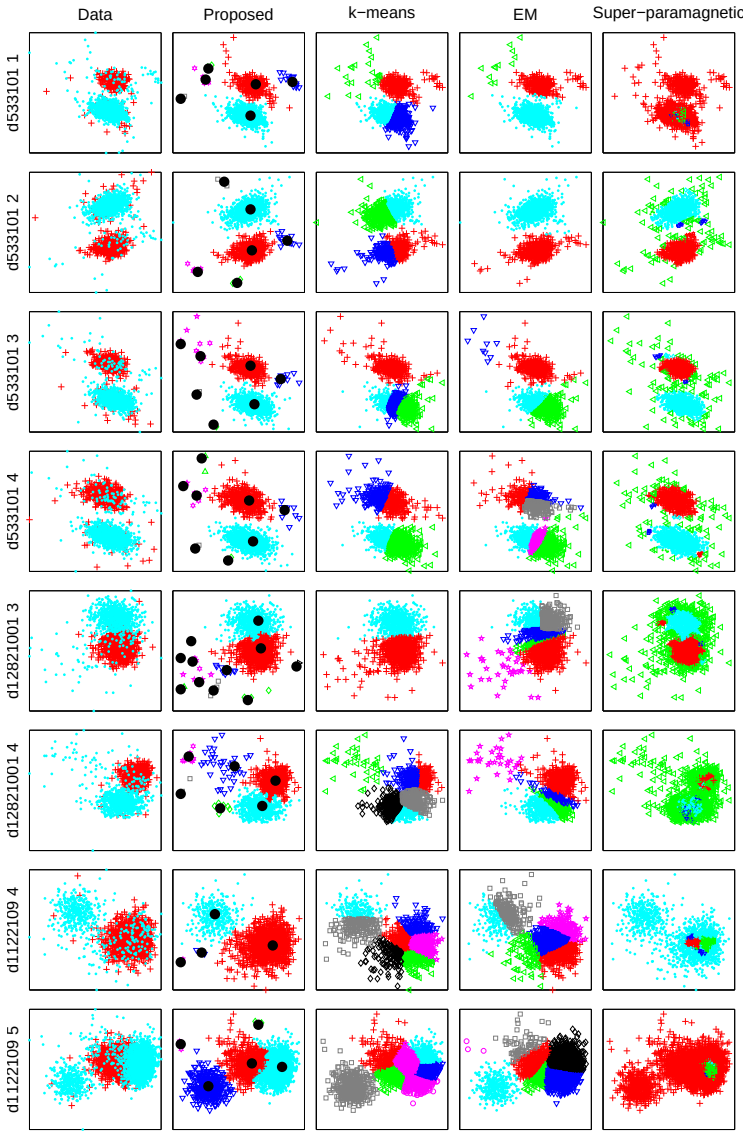


Figure 3.11: Comparison of performance of the four sorting algorithms on 8 recordings. Rows on the figure show results on different recordings. The first column displays projection of each of the recordings on the first two principle components. Red stars represent spikes that come from the neuron recorded intracellularly while cyan dots represent all other spikes (from unknown number of neurons). The other columns give the sorting results of four different algorithms. Different clusters are marked with different symbols/color. For the proposed algorithm (second column) the black dots mark centers of the detected clusters.

would be many falsely clustered spikes: all the spikes which seem to naturally

Table 3.2: Comparison of spike sorting methods on 8 different recordings. C is the largest number of spikes from the neuron recorded intracellularly that are sorted into the same cluster (correct detections). F is the number of spikes from other neurons that are wrongly placed into the same cluster for which C is calculated (false spikes). T is the total number of spikes fired by an intracellularly recorded neuron. Sorting accuracy ($SA = 100 * C/(F + C)$) is given in percentage (see *e.g.* [Chah 2011] for details). We also give the percentage of missed spikes as $SM = 100 * (T - C)/T$

recording	proposed				k-means				EM				superparamag.				
	C	F	SA %	SM %	C	F	SA %	SM %	C	F	SA %	SM %	C	F	SA %	SM %	T
d533101 1	240	64	80	16	237	61	80	17	234	62	79	18	287	1358	17	0	287
d533101 2	356	39	90	12	235	43	85	42	356	57	86	12	339	27	93	16	403
d533101 3	333	61	85	13	335	74	82	12	333	69	83	13	304	36	89	20	381
d533101 4	392	76	84	10	239	66	78	45	151	27	85	65	368	55	87	16	437
d12821001 3	714	69	91	5	724	97	88	4	600	43	93	20	599	28	96	20	753
d12821001 4	438	65	87	4	232	18	93	49	420	57	88	8	134	3	98	71	456
d1122109 4	1065	182	85	0	305	35	90	71	476	60	89	56	157	7	96	85	1070
d1122109 5	795	141	85	34	480	36	93	60	497	33	94	58	1197	3401	35	0	1197

fall out of both clusters would be necessarily clustered in one of the two clusters. Results on figure 3.11 suggest that increasing the number of clusters seems to rather lead to split of the clusters through the area of the high spike density, than grouping the spikes which are far from the areas of the high density into separate cluster.

Results of super-paramagnetic clustering are significantly different from the one of k-means or EM based approaches. They are sometimes very good (`d533101 2`, `d533101 3`, `d533101 4`) so for example isolated spikes far from the cluster are left outside the cluster. For recording `d12821001 3` the results are still satisfactory, but for recordings `d533101 1`, `d12821001 4`, `d1122109 4` and `d1122109 5` they are very bad, since almost all the detected spikes are sorted into same clusters.

The proposed approach performed generally very well. Even though in few situations several spikes that are located a bit far from the cluster are wrongly placed into the cluster (*e.g.* `d533101 3` and `d1122109 5`), the bulk of the spikes that corresponds with the intracellularly recorded neurons are always recognized. Most of the isolated spikes are assigned to their own clusters (notice the black dots that show the locations of the centers of the clusters), which are after neglected since they contain less than G spikes.

Finally, we can say that the proposed approach compares favorably to all the three approaches. On these data the proposed approach always gave reasonably good results, without big oscillations.

Of course, to obtain a single-channel data where such separation is generally possible, the SNR has to be satisfactory and good performance of feature extraction technique is necessary. These are requirements of all the four ap-

Table 3.3: Analysis of robustness of the proposed approach. Columns `min` R and `max` R show the results obtained with the minimal and the maximal filter width and height respectively, such that the results are still considered as satisfactory (relatively well clustered activity of the intracellularly recorded neuron). G is the number of the elements that contained the largest filter window for the given R and, as before, C is the number of the correct detections, F the number of false alarms and T the total number of the spikes fired by the intracellularly recorded neuron. Sorting accuracy (SA) and sorting mistake (SM) are given in percentage.

recording	min R						max R						
	R	G	C	F	SA %	SM %	R	G	C	F	SA %	SM %	T
`d533101 1`	4	158	237	43	85	17	14	959	240	61	80	16	287
`d533101 2`	4	132	351	37	90	13	22	1196	357	60	86	11	403
`d533101 3`	3	96	311	46	87	18	19	1143	334	80	81	12	381
`d533101 4`	6	211	390	74	84	11	23	1141	396	92	81	9	437
`d12821001 3`	7	132	714	49	91	5	22	668	714	100	88	5	753
`d12821001 4`	5	93	438	64	87	4	19	622	438	65	87	4	456
`d1122109 4`	5	77	1055	161	87	1	22	781	1065	182	85	0	1070
`d1122109 5`	5	373	790	111	88	34	8	808	795	141	85	34	1197

proaches.

To analyze the robustness of the proposed approach we change R, the filter width and height, to see what are the limits for which the algorithm will still give satisfying results. The comparisons are given on figure 3.12 and in table 3.3. Small R leads to much more conservative results, since most of the individual spikes, located a bit further from the area of highest density, are detected as clusters. These clusters contain less then G spikes so they are neglected. For large R, usually only the most important clusters are detected. Therefore, the results are likely to contain more false alarms (lower SA). Table 3.3 indicates that satisfying results can be obtained for a large range of G. On vector space scaled on values from 0 to 100, choices of $R = 7$ and $R = 8$ lead to good results in all examples even though the clusters have different morphologies. Choosing too large value of R will usually result in detection of only one cluster what, as we said in subsection 3.1.4 should indicate that the data are too noisy for the clustering. This significantly increases reliability of the algorithm, since it is usually preferable to characterize the data as too noisy than to obtain results that merge several clusters into one.

3.3 Discussion

In neural recordings the differences between some clusters may be very small. Meanwhile, as the firing rate can vary significantly from one neuron to another,

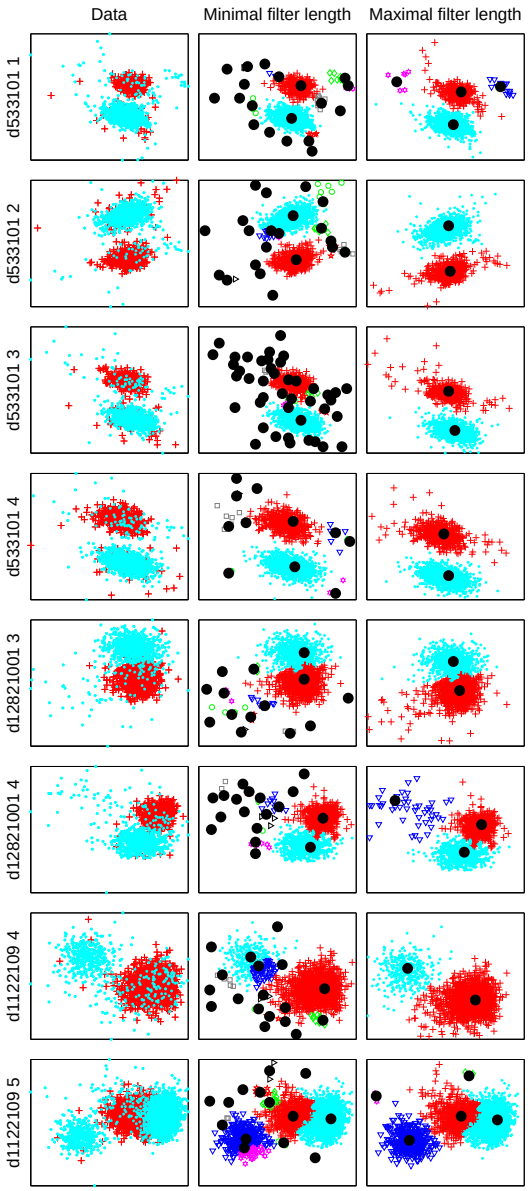


Figure 3.12: Analysis of robustness of the proposed approach. The first column is the same as on figure 3.11. The second column gives the sorting results for minimal R and third column for maximal R which still led to satisfying results.

the sizes of the clusters are usually strongly unbalanced. Also, in the spike detection phase, the firing activities of most of the remote neurons are very partially captured. Each of those neurons is thus represented by so few spikes that they can not be considered as separate clusters; they rather should be left unsorted. However, the classical solutions for the spike sorting problem

are mostly based on the application of a general purpose model selection and sorting algorithms. Now, it is very unlikely that such a general purpose algorithm could comply with all the above mentioned specificities of neural recordings.

Deviating from these solutions, we proposed a non-parametric method that exploits some common properties of the neural signal and thus is particularly adapted for neural spike sorting. We assume that some minimal number of spikes per neuron is given as a prior information (minimal firing rate). This is not a strong requirement since generally we are not interested in isolating the activity of a neuron that fired only few spikes. We also assume that the recorded signal projected on some principle components has a unimodal distribution.

The method we devised has the common preprocessing part as most of spike sorting methods (spike detection and feature extraction). Afterwards, we start with the non-parametric approach by applying a moving average filter in the space projected by the feature extraction vectors (in our simulations principle components). This enhances the centers of the clusters which are then easy to find. We associate all the spikes with their corresponding centers using a greedy sorting algorithm close in spirit to the agglomerative hierarchical clustering and developed particularly for this application. The method can perform with no supervision, using the values of the parameters as given in this chapter. However, to increase the accuracy, the user can optimize the filter width and height to the structure of the actual vector space. This optimization can be done for each recording or for a set of recordings when similar properties of the feature vector spaces are expected. Moreover, the filter width and height can be adjusted to obtain more or less conservative results.

We tested the proposed method on a specific type of neural recordings: simultaneous intra- extra-cellular recordings. We applied the method on the extracellular data and validated the results using the intracellular data. Comparing with the state-of-the-art approaches our method leads to significantly better results and shows very reliable performance.

The complexity of the proposed algorithm highly depends on the number of extracted features per spike, since this number defines the dimensionality of the filter. For feature extraction with PCA a good choice was to keep only the projections on the first two principle components which means that we needed to apply a $2D$ spacial filter. When other feature extraction techniques are used this number of features could be insufficient, so to keep the processing fast an alternative to filtering that would still enhance the centers of the clusters should be implemented. The complexity also highly depends on the scale of the feature extraction space *e.g.* in the presented simulations we scaled the

space to values from 0 to 100. If we denote lower and upper bound of n^{th} scaled feature extraction dimension as y_n^{min} and y_n^{max} then the number of operations NO_{filt} needed for the filtering can be expressed as $\mathsf{NO}_{filt} = \prod_{n=1}^{L}(y_n^{max} - y_n^{min})$. The same number of operations is needed for finding the maxima as we need to analyze each element in the vector space. So the total number of operations is given by $\mathsf{NO}_{total} = 2\prod_{n=1}^{L}(y_n^{max} - y_n^{min})$.

In its current form, the proposed method is obviously not implementable online. However, we may point out some possible modifications that would allow an online implementation. These concern essentially two points: 1) an initialization period, that provides all the centers of the clusters; 2) an online feature extraction method.

In this chapter we addressed a question of spike sorting from single-channel recordings. In the following chapter we will present an algorithm based on iterative application of independent component analysis for sorting from multi-channel recordings, which uses the proposed method to separate the activity of one neuron in each iteration. As we will see in the next chapter, the algorithm presented here will help us to obtain more reliable sorting results, especially in low SNR conditions. There the property of automatic clustering will be particularly important, since in an iterative algorithm any manual verification of the results within each iteration is not practical. Even though the algorithms compared with the proposed one could obtain similar results with help of manual parameter adjustment within some feedback process, they would not be suitable for many applications, among else for such iterative algorithm.

Chapter 4

Spike sorting in multi-site recordings

Contents

The ultimate goal of spike detection and sorting is to provide data from which it is possible to better understand the neural information. Since the neural networks, particulary in the cortex, are extremely complex (see chapter one) we need an individual activity of as many neurons as possible in order to try to understand the neural code. More precisely we need the simultaneous activity of neurons that are close to each other. To obtain recordings of such activity extracellular neural recordings are commonly used. The main advantages of extracellular recordings is that they give access to individual neural activity (after successful spike sorting) and they can be performed using multiple recording units.

Currently one common way to obtain recordings of such activity is to use a tetrode, an extracellular electrode which consists of 4 recording sites. However the electrodes with much greater number of channels are penetrating more and more to the market (see e.g. [Simeral 2011] where 100-channels recording system, described in [Bla 2009], has been used). Each recording site of the tetrode, or any multi-electrode, will record mixture of activity from a vast number of neurons around the electrode. The recordings sites are usually located close enough so that the activity of the neurons around the electrode is projected on all the recording sites. However, spikes that are fired by different neurons are, due to propagation and velocity effects, projected on each site with different shape and amplitude.

Since all the recording sites record the activity from the same neurons, the information redundancy in very large. Developing a sorting algorithm that will use this redundancy to increase the SNR is on the basis of spike sorting methods adopted for multi-site recordings. This is actually a Blind Source Separation (BSS) problem. A well known BSS technique, called Independent Component Analysis (ICA), is often used to find a new basis for the data that will minimize the mutual information between the recording sites [Shiraishi 2009], [Brown 2001]. ICA is generally known to be a very efficient technique, but it is necessary to satisfy several conditions for its successful application. Some of these conditions are only partly or not at all fulfilled in neural recordings. The most problematic condition is that the number of the recording sites (channels) should not be lover than the number of sources (neurons). This is clearly not true for a vast majority of neural recording where the number of neurons largely exceeds the number of the recording sites (so-called overcomplete problem) [Brown 2001]. The condition that the recording should be a linear mixture of the activity of the sources, is also not true as well, due to a nonhomogeneous medium between the neurons and the electrode and also generally complicated spatial form of neurons. Another very important assumption for the ICA algorithm is that sources are independent, which is not true in neural recordings, since the output of one neuron clearly depends on the outputs of the others (neural networks). Even though, by using ICA it is possible to obtain very interesting results in spike sorting, the number of neurons that can be extracted in this way is usually very limited and the accuracy highly depends on the spatial distributions of the neurons with respect to the electrodes.

More recent techniques naturally combine feature extraction and ICA approaches, obtaining more accurate spike sorting *e.g.* [Takahashi 2003a] and [Takahashi 2003b]. The algorithm we will present in this chapter is in spirit similar to such approaches, but we will explore some particular properties of the neural signal in order to obtain much more robust and generally more accurate results.

One common difficulty in spike sorting is that often, the activities of different neurons are merged into the same cluster. The sorting accuracy depends mainly on a spatial distribution of the neurons around the electrode, what gives a set of physical parameters which we can not control. Merging the activities of several neurons in a spike sorting may lead to serious misinterpretation, especially when the sorting result is used for understanding the neural code.

Generally analyzing the problem of usefulness of spike sorting, we can say that missing some spikes (spikes that are not detected or that are left unsorted) is much less important mistake than the sorting into a wrong cluster

(spikes assigned to the neuron which did not fire them). Thus, in the trade-off between the number of missed spikes and the number of false spikes it is highly preferable to have more missed spikes. See [Ilan 2008] for detailed argumentation of this statement.

In this chapter we present a new algorithm for spike sorting in multi-channel extracellular recordings. We use a deflation technique to improve the performance of ICA: after we isolate the firing instants of a single neuron, we remove them from the original recording and repeat the procedure until the algorithm becomes unable to isolate any more neurons or until a prescribed number of neurons is obtained. Moreover, within each iteration we implement, in an internal loop, another iterative algorithm for removing the noise and all the spikes that are not coming from the neuron that appears as the closest to the electrode. Within each iteration of the internal loop we apply the sorting algorithm described in the previous chapter. At the end of the algorithm, we sort the overlapped spikes. To do this we take all the possible pairs of the isolated neurons and apply ICA only on the segments of the signal while one of the neurons in the pair was active. This algorithm has been presented in the shorter form in [Tiganj 2011a].

To elaborate use of the algorithm presented in the previous chapter as a module for the multi-channel sorting algorithm presented here, we point out that automatic and stable performance are the essential properties we want to obtain. Moreover, the method described in the previous chapter gives a good measure of a distance between clusters. Since each element belongs to the same cluster as its nearest neighbor, a distance between two clusters is well given as a distance between the two closest elements that belong to two different clusters.

Each step of the algorithm is analyzed in section 4.3. Before, in the following section we discuss on general possibility of applying ICA in neural recordings and in section 4.2 we describe real and artificial neural recordings we are going to use to present the algorithm. We present the results in section 4.4 and give some final discussion in section 4.5.

4.1 ICA in neural recordings

In this section we will analyze possibility of applying ICA for spike sorting. ICA relies on a statement that, having a linear mixture of $S > 1$ mutually independent non-Gaussian sources (in our case neurons) recorded with $L \geq S$ recording sites (channels), the individual activity of all the sources can be reconstructed by minimizing the mutual information between the recording sites. For a great tutorial on ICA see [Hyvärinen 2000].

If the activity of the sources is given with a vector $s = [s_1, s_2, ..., s_S]^T$ and the recorded activity is $l = [l_1, l_2, ..., l_L]^T$ then their linear mixture can be expressed as $l = As$, where $A_{L\times S}$ is called the mixing matrix. We want to find $W_{S\times L}$, such that $s = Wl$. Since we do not know either A or s the problem is called BSS problem.

Development of ICA algorithms is based on the central limit theorem which says that the distribution of a sum of independent, non-Gaussian distributed variables tends to be more Gaussian than the distribution of each variable separately. So the distribution of the signal from any of the recording sites will be more Gaussian than the distribution of any of the sources. ICA finds such transformation, $\tilde{W}$ (estimate of W), that makes the distribution of the resulting signal as different as possible from the Gaussian distribution. As a measure of Gaussianity, kurtosis or negentropy are commonly used. Ideally, sources reconstructed using $\tilde{W}$, $\tilde{s} = \tilde{W}l$, will be as independent as possible.

Mixtures of neural activity do not have the properties generally assumed by ICA algorithms. However, ICA can still be applied on such mixtures, but the separation will have a limited accuracy. We will discuss below the influence of some properties of neural mixtures on this accuracy.

4.1.1 Linearity

An action potential is generated in the body of a neuron and travels through the axon carrying an information to other neurons. The medium between a neuron and an electrode is not isotropic, since it is filled by other neurons. Thus, we can say that the recorded neural signal is not a linear mixture of individual neural activity. However, for the neurons that are close to the electrode, which are the subject of our interest, this nonlinearity is generally not very large.

Since ICA is a linear transformation it can not separate the sources from a nonlinear mixture. Perfect reconstruction of the activity of individual neurons is therefore not possible by the application of ICA. However, $\tilde{W}$ is still a transformation matrix that transforms the data in such way that a mutual information between the recording sites is minimized. So even though the estimation is not perfect it still leads to a signal from which the separation of individual neurons will be easier than it would be from the original recording. We remind here that we are not interested in reconstruction of the exact waveforms of neural activity, but only in separation of the firing instants. The result that we are interested in is binary - *e.g.* 1 for the signal sample when a neuron fires an action potential and 0 for all the other samples. Thus, any transformation that makes the separation of activities of different neurons easier can be very useful.

Moreover, recent research in the field of BSS addresses development of nonlinear ICA, see [Comon 2010], [Jutten 2004] and their references for more details. In the future these approaches could be used instead of the classical ICA to improve the sorting results.

4.1.2 Independency

Clearly, the activity of one neuron is not fully independent from that of other neurons. In neural tissues, each neuron is directly connected through synapses with thousands of other neurons. So the relation between two single neurons, even though it definitely exists, is not that strong and for most of situations we can say that the activities of two single neurons are almost independent. Moreover, individual neural activities are generally more independent than the mixtures recorded with extracellular recording sites placed next to each other. Thus applying of ICA should result with the signal in which the activities of individual neurons are easier to identify than from the original recording.

4.1.3 More sources than sensors

There are only a few specific situations in neural recordings where the number of neurons (sources) is similar to the number of recording sites (channels, sensors). The cockroach escape system [Levi 2000] is one example of such situation. In a vast majority of cases the number of neurons is thousands of times larger than the number of recording sites. This is obviously a very difficult problem for ICA.

Luckily, neurons which distance from the electrode is much larger than the mutual distance between the recording sites appear in the recording almost as a single equivalent source. This is because the amplitudes of the spikes from distant neurons are very low and practically the same on all the recording sites. The noise that is introduced by the recording process itself also appears as almost the same on all the recording sites. Thus, majority of the neurons, together with the noise, will be considered all together as only one source. Due to its high energy this source usually appears as a separated source in the output of ICA.

However, around the electrodes there are still many neurons left (much more than the recording sites) and we want to find individual activity of as many of them as possible. Performance of ICA on such data is highly dependent on the spatial distribution of the neurons with respect to the recording sites. For example, when a single neuron is much closer to the electrode than all the other neurons, most of its activity will be projected on a single independent component (IC). If there are many neurons on similar distances

from the electrode then spike sorting is likely to be difficult, however still less difficult than it would be without applying ICA (if the spike sorting was done directly on the recorded signal).

4.2 Description of the neural recordings that will be used in the simulations

In this section we will describe the simulated and the real signal that we will use for the demonstration of the proposed algorithm in section 4.3 and for quantifying the results in section 4.4. With the simulated recordings we can validate the sorting accuracy for all the separated neurons, while with the real recordings we can validate the accuracy only for a single neuron - the one which was simultaneously recorded with the intracellular electrode.

4.2.1 Simulated neural recordings

To describe the proposed algorithm in an illustrative way we will use the simulated extracellular neural recording, in order to have the precise knowledge on which neuron fired which spikes. In this way we are able to clearly demonstrate the performance of the proposed algorithm. Also, using such signal, we are able to precisely measure the sorting accuracy and the sorting mistake and thus compare the proposed approach with the other approaches.

To create the recording we first generate spatial positions for 1000 artificial neurons. We distribute the neurons randomly in a 3D space, respecting some constrains about their minimal and maximal mutual distances. We place four virtual recordings sites on four different locations around the middle of the space. Distance between two neighboring sites is set to be equal to an average distance between two neighboring neurons. The activity of each neuron is obtained using 32 spike templates which are created from the real extracellular recordings (for the description of the real extracellular recordings see the following subsection). We use the intracellular recordings to localize the spikes fired by the same neuron. Then we extract those spikes and average them. Each template is created by averaging at least 100 spikes. The length of each template is 60 samples what corresponds to $4ms$ when reported to the sampling frequency of the real data. We randomly associate each of the 1000 neurons with one of the 32 templates (we assume that a single neuron will fire action potentials of the same shape during the whole recording time). For each neuron we generate a Poisson distributed firing times (time instants when action potentials are fired by a particular neuron) with the randomly generated probability of firing, between 0.0004 and 0.001.

To make the simulated signal more realistic, we have to simulate the sampling effects. In neural recordings continuous time signal is sampled with a sampling period T_s. Thus extracted spikes will be mutually shifted from 0 to T_s seconds. To illustrate better this effect, we can imagine that a neural recording consists only of the activity of a single neuron, which always fires the spikes of the same shape. When recorded these spikes will actually look mutually different, because they will be sampled at different time instances with respect to the beginning of each spike. We simulated this effect by upsampling the spike templates for 100 times. Then we associate the spike templates with the corresponding time locations in the signal. Finally we downsample the signal for 100 times to have similar sampling effect as in the real recordings.

We set that the amplitude of the recorded spikes decays linearly with the euclidian distance between a neuron and an electrode. First column of figure 4.2 shows an example of the simulated recording obtained in this way.

4.2.2 Real neural recording

A drawback of the validation using the simulated signal lays in the complexity of a real neural mixture, which is very hard to reproduce artificially. Real recordings, depending on the tissue in which the recording is taken, consist of superposed activity of up to millions of neurons. The neurons interact through the large number of dendrites and axons, what makes the spatial configuration extremely complex. Thus, the simulated signal is often not a truthful counterpart of the real signal.

Apart from the simulated signal we examen the proposed algorithm using the real neural recordings. We use the simultaneous intra- extra-cellular recordings from a rat hippocampal area CA1 done by Henze et al., described in [Henze 2000], that have been used in the last two chapters as well. When the extracellular electrode is close to the intracellular, the activity of the neuron recorded intracellularly will be also visible in the extracellular recording. Of course, with the extracellular electrode we pick the activities of a vast number of neurons, so to separate the activity of individual neurons we have to do spike sorting. Figure 4.1 shows a part of such simultaneous recording. The four top plots show the extracellular recording with tetrode and the bottom plot shows the corresponding intracellular recording. The action potentials fired by the neuron recorded intracellularly are clearly visible on all four extracellularly recorded channels. This indicates that the recording sites are located very close to each other.

On the plots of the extracellular channels we can also see the activity of other neurons that are around the extracellular electrode. Notice that many spikes from the other neurons have amplitudes similar to those of the spikes

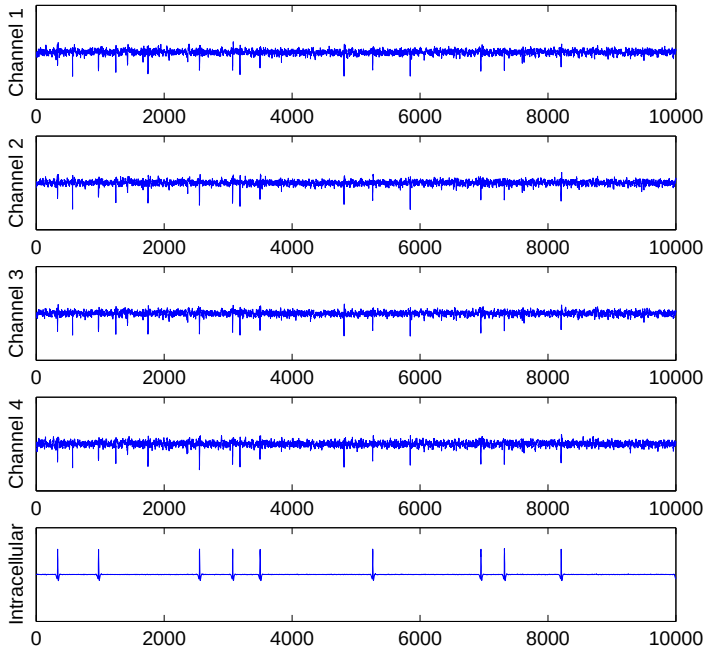


Figure 4.1: Simultaneous intra- extra-cellular recording. The four recording sites are very close, so the action potentials recorded with the intracellular electrode are clearly visible on all four extracellularly recorded channels

that come from the neuron recorded intracellularly. Thus, the task of spike sorting is obviously not trivial in this case.

4.3 The algorithm description

In this section we describe the proposed algorithm for multi-channel spike sorting. The algorithm iteratively removes the activities of distant neurons from the original recording as well as the activities of neurons close to the electrode, once they are separated from the mixture. By doing so we reduce the number of sources, which brings the application of ICA in a more comfortable setting.

Apart from the recorded signal, to use the algorithm we only need to provide a lower bound for the firing rate, call it G, and the spike detection threshold level. Minimal firing rate is usually easy to obtain since, generally, we are not interested in separating the activities of neurons that fired only a few spikes. The choice of the threshold level, on the other side, is not very critical, since we will simply neglect all the spikes that belong to the cluster which has the lowest average peak-to-peak amplitude of the spikes. This will

make the algorithm more reliable, since the spikes with the lowest amplitudes usually come from many different neurons, which activity is only partially detected.

We assume that a four-channel recording is given, but the algorithm is the same for any number of channels. The four recorded signals are labeled as $E1$, $E2$, $E3$ and $E4$. An example of a part of the simulated four-channel recording is shown on figure 4.2 and of the four-channel real recording on figure 4.3 (spikes from the neuron recorded intracellularly are marked with the red starts).

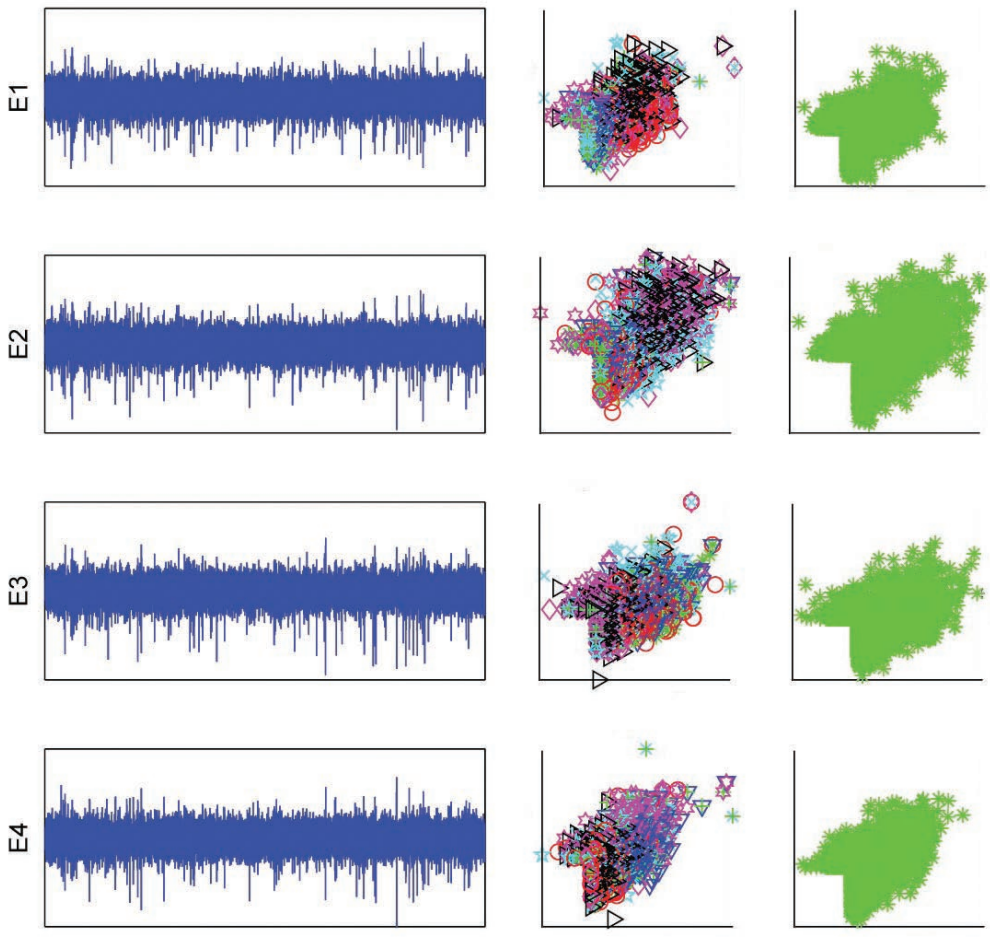


Figure 4.2: 50000 samples from the simulated four-channel extracellular recording (first column). 2D feature vector space with the extracted spikes - positive peak amplitude vs negative peak amplitude (second column). Results of sorting algorithm from the previous chapter are given in the third column (only one cluster was detected for each channel).

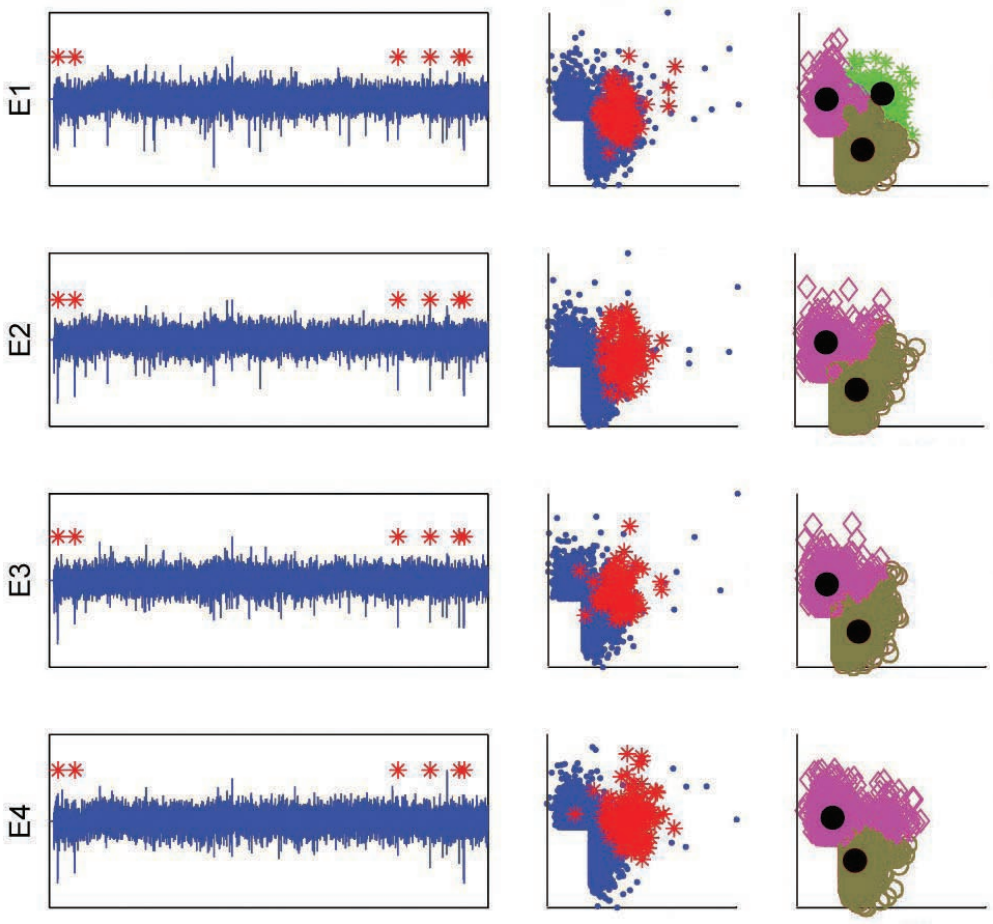


Figure 4.3: 50000 samples from the real four-channel extracellular recording (first column). 2D feature vector space with the extracted spikes - positive peak amplitude vs negative peak amplitude (second column). Spikes that come from the neuron recorded intracellularly are labeled with the red stars, while the rest of the spikes are labeled with the blue dots. Results of sorting algorithm from the previous chapter are given in the third column, where the black dots marks are the detected centers of the clusters.

Only to demonstrate that spike sorting is very difficult for such recording we apply directly the method we described in the previous chapter. First, we detect the spikes using the spike detection method presented in the chapter two. We project the spikes in a 2D vector space by keeping only two features per spike: the positive and the negative peak amplitudes. Note that one could also use *e.g.* the two first principal components. In the particular case, better (generally more reliable) results were obtained when the positive and the negative peak amplitude were used as the features, instead of the

commonly applied PCA. The plots of the features are shown next to the corresponding signals, in the second column on figures 4.2 and 4.3. In the third columns on figures 4.2 and 4.3 we give the outputs of the algorithm from the previous chapter. The spikes that are left unsorted by the algorithm are automatically eliminated and not plotted. Note that the colors/symbols chosen for the representation of the clusters for the plots in the second column are not related to the one in the third column. It is evident that from such feature vector space it is not possible to do an accurate spike sorting.

We will now demonstrate the performance of the proposed algorithm using step-by-step description. The flowchart that describes the proposed algorithm is shown on figure 4.4.

• `Step 1:` **ICA.** We process the input signal with the classical fast-ICA algorithm [Hyvärinen 1999] and obtain $IC1$, $IC2$, $IC3$ and $IC4$ as the outputs (figures 4.5 and 4.6). The second and the third column on the both figures are again shown only to give an illustration of what the results would be if the feature extraction based sorting method was directly applied on the outputs of the ICA algorithm. The difference between the clusters is more obvious than on figures 4.2 and 4.3.

• `Step 2:` **Remove everything but spikes from** $E1$**,** $E2$**,** $E3$ **and** $E4$**. Set: k=false.** Before detecting the spikes we first want to identify which of the ICs is the most likely to contain the activity of the neurons closest to the electrode. The criteria we set to determine this IC is spike dynamics. As a measure of the spike dynamics we use the average difference between the positive and the negative peak. Since spikes naturally contain a large peak, such difference should be a good measure of the dynamics. We first find all the spikes from all four ICs using the method described in the second chapter. Then, we calculate the corresponding spike dynamics and consider as actual spikes only those detected from the IC with the largest spike dynamics. Next, from all the 4 input recordings we remove (set to zero) all but the actual spikes. We assume that the removed parts of the signal correspond only to the noise and the cumulative activity of the neurons that are distant from the electrode. The outputs are called $E1^*$, $E2^*$, $E3^*$ and $E4^*$.

• `Step 3:` **ICA.** We apply ICA on $E1^*$, $E2^*$, $E3^*$ and $E4^*$. The resulting ICs, called $IC1^*$, $IC2^*$, $IC3^*$ and $IC4^*$, are shown on figure 4.7 for the simulated signal and on figure 4.8 for the real signal. To demonstrate the results as before, we plot the spikes in the greedy (second column) and output of the feature extraction based sorting algorithm (third column). As the number of the sources is now lower, the ICA algorithm performed a little better and a little bigger difference between the clusters is visible, in comparison with the results from figures 4.5 and 4.6.

• `Step 4:` **Find which** IC **is the most suitable for the further**

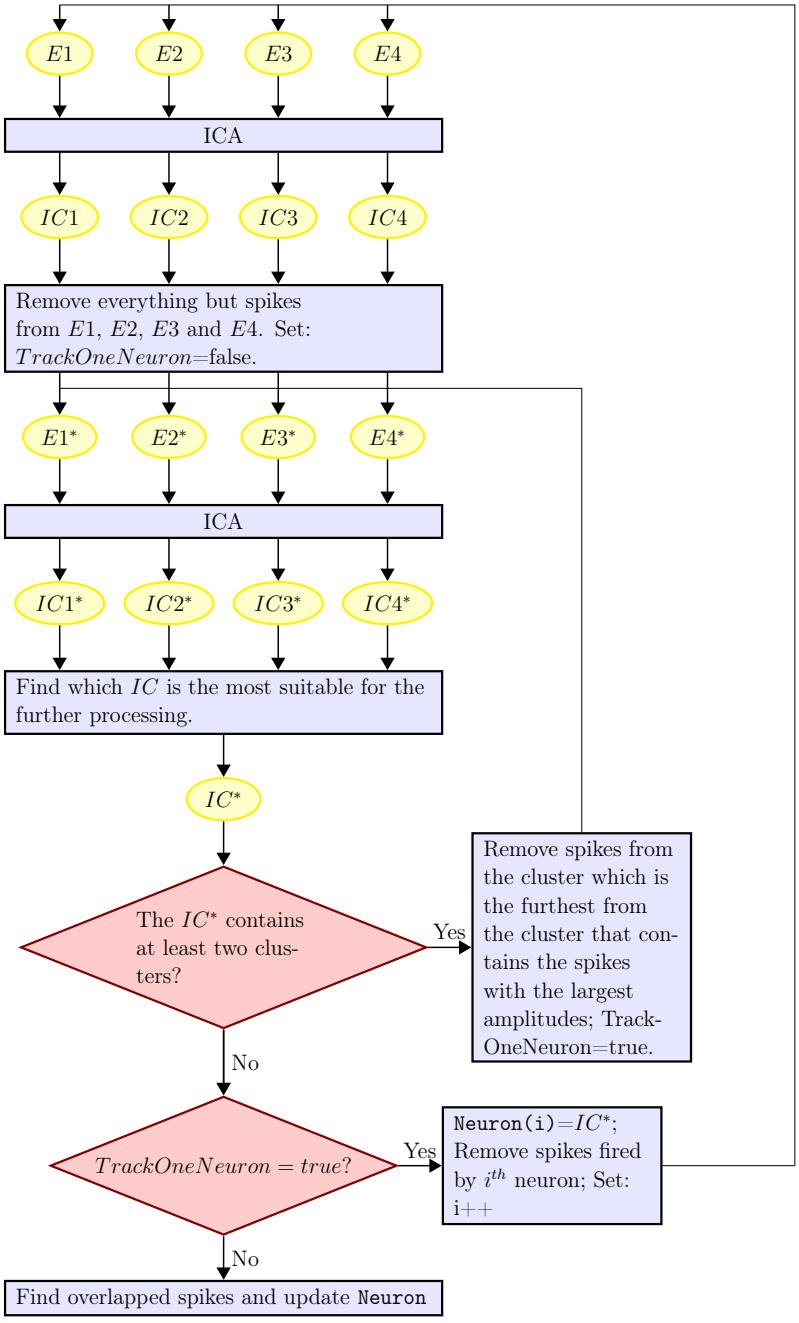


Figure 4.4: Flowchart of the proposed multi-channel spike sorting algorithm.

processing. In the further steps we will need only one IC. Thus in this step we want to chose the one which contains well preserved spike waveforms. It is likely that a good choice would be the one for which the detected spikes have the largest dynamics. We calculate the spike dynamics in the same way as in the `step 2`. For the further processing we chose the IC which has the largest

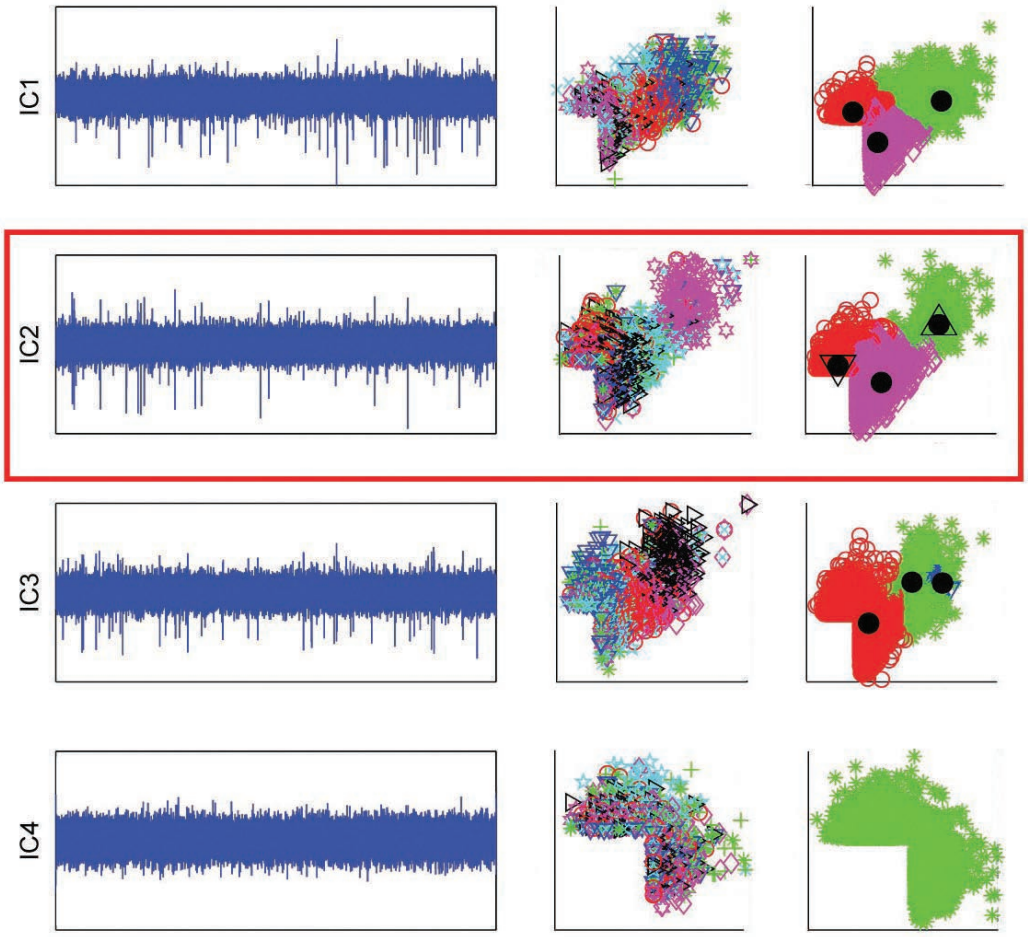


Figure 4.5: Result of applying ICA on the recording shown of figure 4.2. The four ICs are shown in the first column (50000 samples). 2D feature vector space with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from the previous chapter on the vector space from the second column. The red rectangle marks the IC that contains the spikes of the largest dynamics.

value of the dynamics and call it IC^*.

- *`Condition 1:`* **The IC^* contains at least two clusters?**
- `If` *`Condition 1`* `is true:` **Remove spikes from the cluster which is the furthest from the cluster that contains the spikes with the largest amplitudes; TrackOneNeuron=true.** In this step we have to apply the algorithm proposed in the previous chapter, this time it is not only for demonstration, but to find which spikes we can remove from the signal. For

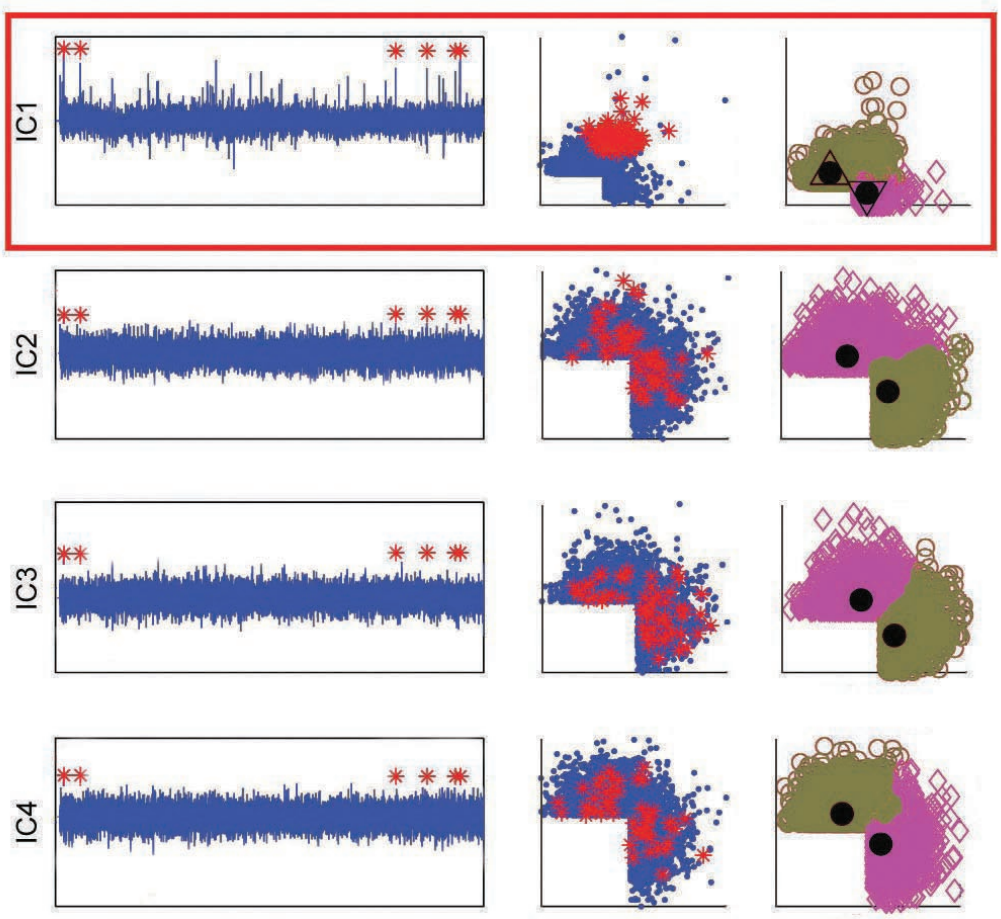


Figure 4.6: Result of applying ICA on the recording shown of figure 4.3. The four ICs are shown in the first column (50000 samples). 2D feature vector space with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from the previous chapter on the vector space from the second column. The red rectangle marks the IC that contains the spikes of the largest dynamics.

each cluster we compute the average of peak-to-peak amplitudes from the first G spikes. The spikes that have been classified in the cluster furthest from the one with the largest average peak-to-peak amplitude are now removed from $E1^*$, $E2^*$, $E3^*$ and $E4^*$. Next, the algorithm goes back to the `step 3`. We show on figures 4.9 and 4.10 the outputs after applying ICA on the new $E1^*$, $E2^*$, $E3^*$ and $E4^*$ for the simulated and the real recording respectively. From the simulated recording (figure 4.9) we have detected only one cluster on each of four IC. We use the same criterion as before to chose which IC we should

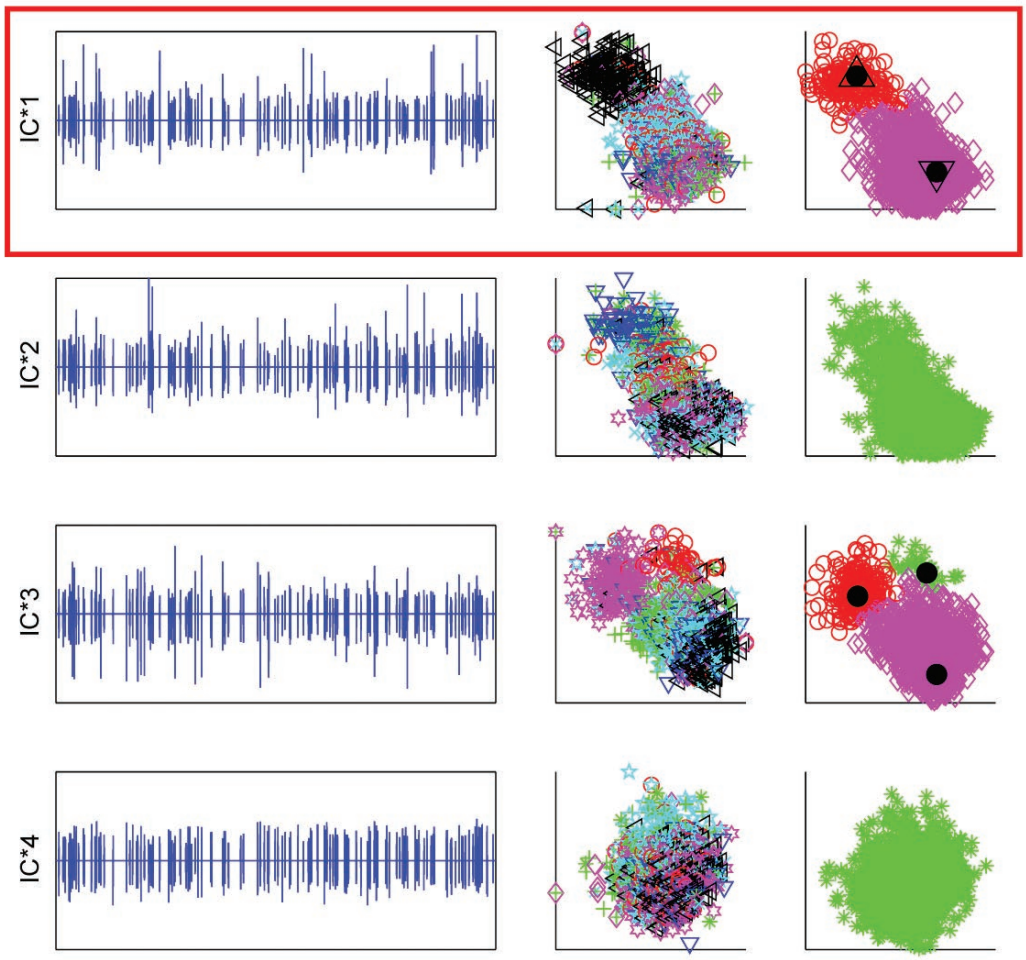


Figure 4.7: Result of applying ICA on the simulated recording after all but spikes was removed from the signal (set to zero) - 50000 samples are displayed. The four *IC*s are shown in the first column. 2D feature vector spaces with the spikes extracted from the *IC*s are in the second column. The plots in the third column display the results of applying the algorithm from the previous chapter on the vector space from the second column. IC that has been chosen for the further processing is marked with a red rectangle. The black triangle, with the tip pointing up, marks the center of the cluster that contains the spikes with the largest average peak-to-peak amplitude. Opposite, the black triangle with the tip pointing down marks the center of the cluster that contains the spikes with the smallest average peak-to-peak amplitude.

keep. Since the *condition 1* is now `false` so the algorithm will continue to the *condition 2*. Notice that almost all the remaining spikes are fired by the

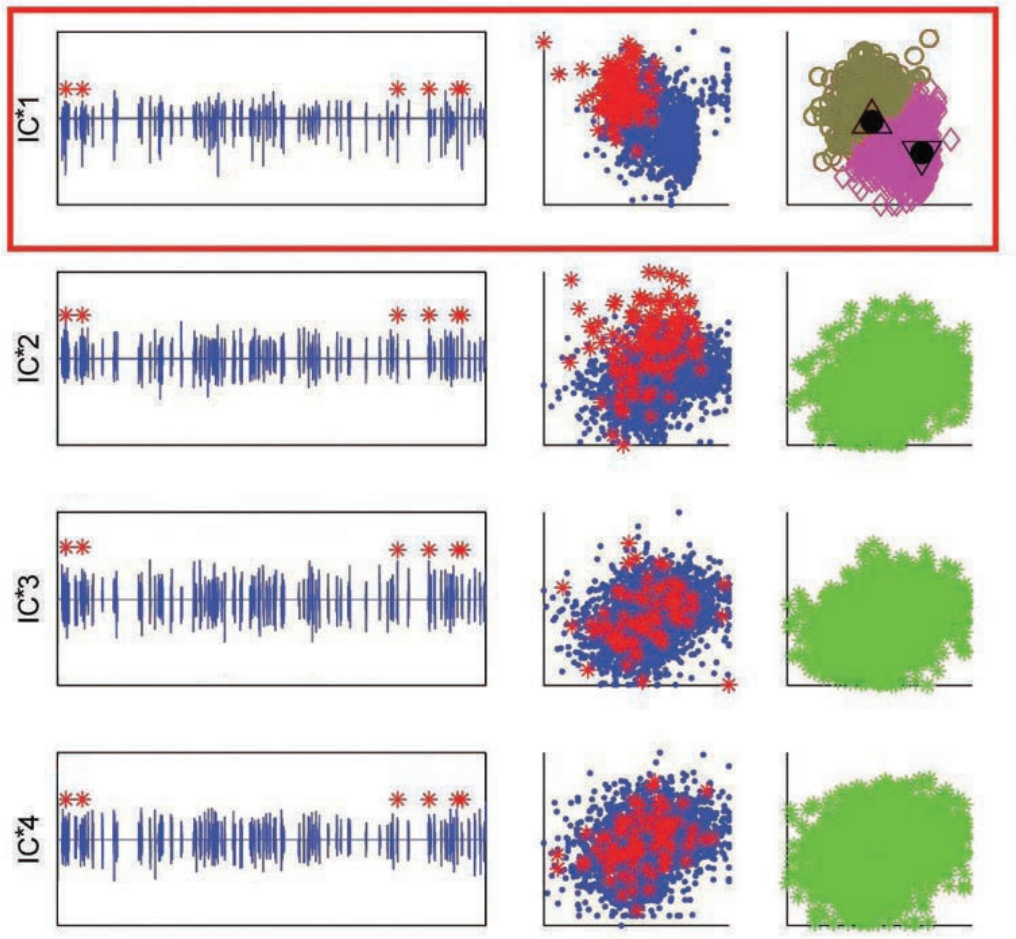


Figure 4.8: Result of applying ICA on the real recording after all but spikes was removed from the signal (set to zero) - 50000 samples are displayed. The four ICs are shown in the first column. 2D feature vector spaces with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from the previous chapter on the vector space from the second column. All the labels are the same as for figure 4.7

same neuron. From the real recording (figure 4.10) we have detected more than one cluster on $IC1^*$ and $IC2^*$; we keep $IC1^*$. The cluster that will be removed is the one labeled with a black triangle whose tip is pointing down. After that cluster is removed, spikes from the neuron recorded intracellularly will be almost the only one remaining in the signal. Only a few spikes from the intracellularly recorded neuron were removed. We set in this step the flag $TrackOneNeuron$, in order to track if the algorithm converged just after a

new neuron is separated, what would indicate generally that no more neurons can be separated from the given recording.

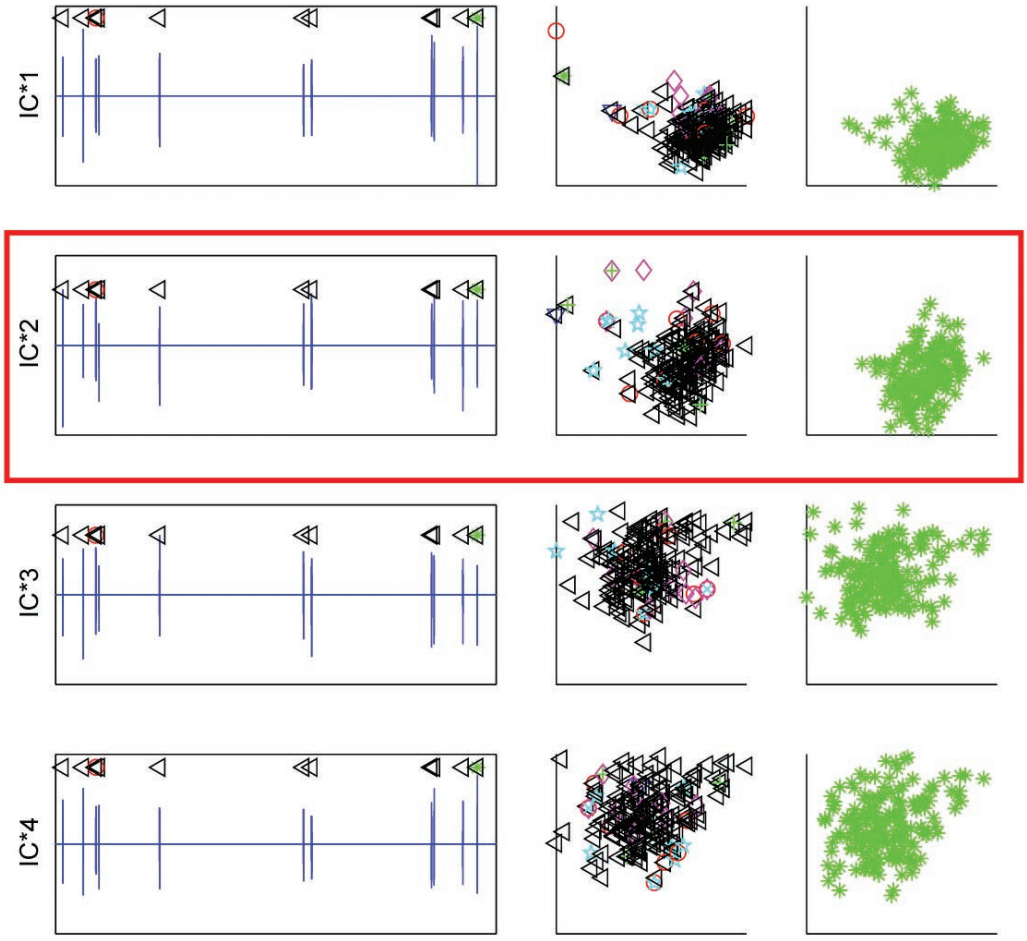


Figure 4.9: Result of applying ICA on the simulated recording, after removing (setting to zero) of the low amplitude spikes (see figure 4.7) from $E1^*$, $E2^*$, $E3^*$ and $E4^*$ (50000 samples). The four ICs are shown in the first column. 2D feature vector spaces with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from the previous chapter on the vector spaces from the second column. All the labels are the same as for figure 4.7.

• `If` *`Condition 1`* `is false:` *`Condition 2:`* **TrackOneNeuron** = `true`?

• `If` *`Condition 2`* `is true:` **Neuron(i)**=IC^***; Remove the spikes fired by the** i^{th} **neuron; Set:** $i++$. If $TrackOneNeuron$ is `true` that

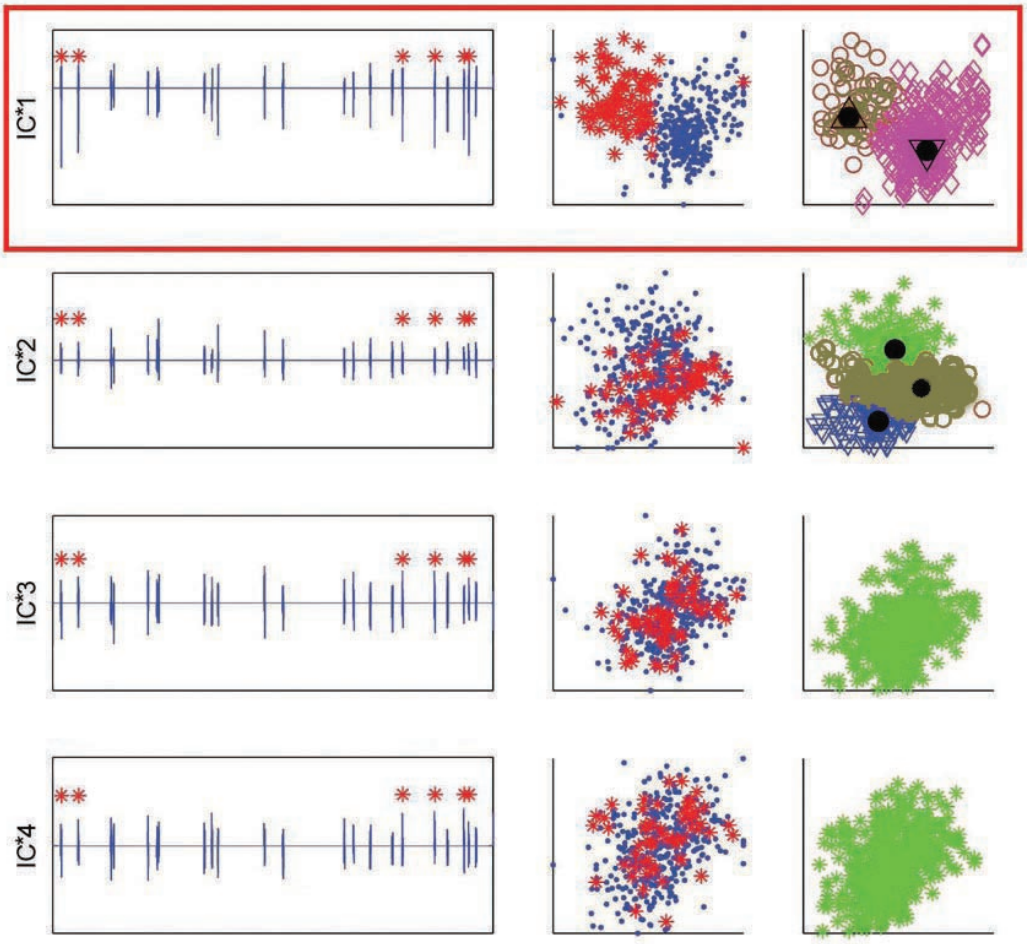


Figure 4.10: Result of applying ICA on the real recording, after removing (setting to zero) of the low amplitude spikes (see figure 4.8) from $E1^*$, $E2^*$, $E3^*$ and $E4^*$ (50000 samples). The four ICs are shown in the first column. 2D feature vector spaces with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from the previous chapter on the vector spaces from the second column. All the labels are the same as for figure 4.7.

means that we have detected more than one cluster in the previous iteration so now, since only one cluster is left (the *condition 1* was `false`), we can conclude that the remaining spikes are coming from the same neuron. This is the i^{th} separated neuron. To continue our iterative algorithm we set to zero all the samples from the recording (from $E1$, $E2$, $E3$ and $E4$) when any of the already detected neurons (`Neuron(1)`, `Neuron(2)`,...,`Neuron(i)`) was

active and go back to `step 1`. By doing this we removed an important high amplitude sources.

• `If` *`Condition 2`* `is false:` **Find the overlapped spikes and update** `Neuron`. This is the last step of the proposed algorithm. We address now a very important problem in spike sorting: simultaneous firing. The proposed deflation based algorithm is obviously not able to detect simultaneous firing, since once a spike has been sorted it is removed from the signal for the following iterations. Thus, if two or more neurons fired at approximately the same time, the spike will be assigned only to one of these neurons. To find the simultaneously fired spikes we analyze the activity of the sorted neurons two by two, for all possible pairs of the detected neurons. When analyzing the activity of each pair, we keep in the recording only the segments when one of the neurons in the pair fired a spike. Each such recording, call it $\tilde{s}_{i,j}$, where $i = 1, ..., S-1$ and $j = i+1, ..., S$ can be expressed[1] as:

$$\tilde{s}_{i,j} = s_i + s_j + sgn(|s_i| + |s_j|) \sum_{k=1, k\neq i,j}^{S} s_k,$$

where S is the number of sorted neurons. It is obvious that most of the energy in any $\tilde{s}_{i,j}$ comes from s_i and s_j. Thus, applying ICA on $\tilde{s}_{i,j}$ leads to the separation of s_i and s_j: one IC will represent mostly s_i, another one s_j and the remaining ICs will represent $sgn(|s_i| + |s_j|) \sum_{k=1,k\neq i,j}^{S} s_k$. Since the activity of s_i and s_j will be separated, any spikes fired simultaneously by these two neurons should be visible on both of the ICs that represent the activity of these neurons. We demonstrate this by an example given on figure 4.11. In the first two rows of the both figures we plot part of the activity of two simulated neurons that fired three spikes simultaneously. The next four plots show four channels of the simulated recording. The activity of the two neurons is visible, but hard to recognize, especially to differentiate. We remove from the four channels everything but segments when any of the two neurons was active and apply ICA on such sparse signal. The result of ICA is shown on four bottom plots on the same figure. Since this is the last step of the proposed algorithm, we already have the estimation of the activity of all the isolated neurons. Thus, it is easy to find, from the result of ICA, that in this example $IC3$ represents the activity of *Neuron 1* and that $IC1$ represents the activity of *Neuron 2*. All the overlapped spikes are visible on both $IC3$ and $IC1$. They are now very easy to extract and to update the detected activity of the isolated neurons.

[1]For notational simplicity, we drop the time argument and write s_k instead of $s_k(t)$ for the recorded activity of neuron $\#k$. Note that $s_k(t)$ is a sparse signal: We have $s_k(t) = 0$ except for intervals $[\tau, \tau + T]$ when an action potential is emitted from time τ. The length T of the interval corresponds to the duration of an action potential.

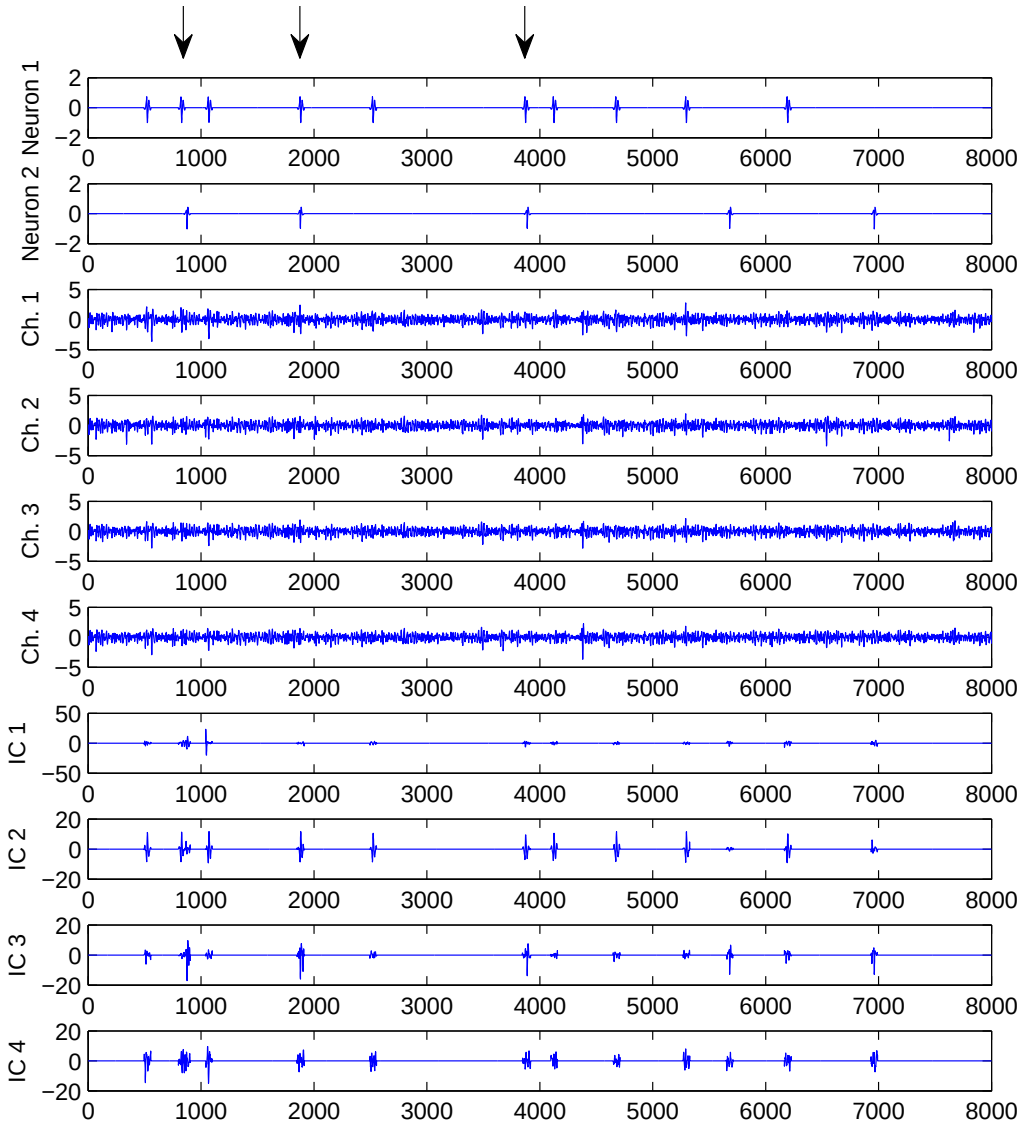


Figure 4.11: The first two plots show part of the activity of the two simulated neurons. The following four plots show the part of the four-channel simulated recording. The four plots at the bottom show the results of applying ICA on the four channels. Before applying ICA the recording was modified by setting to zero all but the segments where one of the two neurons was active. It appears that the activity of *Neuron 1* is represented by *IC3* and the activity of *Neuron 2* is represented by *IC1*. The three arrows on the top of the figure point out the overlapped spikes from the two neurons. All of the three overlapped spikes are successfully sorted.

4.4 Results

We apply the algorithm described in the previous section on 5 real recordings and on 10 simulated recordings. The results are given in tables 4.1 and 4.2. There are four key-operations in the proposed algorithm: 1) removing of the noise (*Noise rem.*); 2) removing of the spikes from the clusters furthest from the one that contains the spikes with highest amplitudes (*Clus. rem.*); 3) removing of the spikes from already separated neurons (*Sorted rem*); 4) detection of the overlapped spikes (*Over. det.*). We analyze contribution of each of these steps by simply comparing the proposed algorithm to algorithms which perform none, only one or more of these steps. Namely, we compare the proposed algorithm with the following:

1. Only the spike sorting algorithm, without performing ICA (*Spike sor.*).
2. ICA and spike sorting
3. ICA, spike sorting and removal of the spikes from already separated neurons.
4. As 3, but with removal of everything but spikes (`step 2` of the algorithm).
5. As 4, but with removal of clusters that generally contain spikes small in the amplitude (clusters which are furthest from the one that contains the spikes which show the largest average peak-to-peak amplitude).

The proposed algorithm is actually an extension of the algorithm number 5, with a sequence for the detection of the overlapped spikes.

We express the results in terms of sorting accuracy ($SA(\%)$) - the percentage of accurately sorted spikes and sorting mistake ($SM(\%)$) - the percentage of missed spikes.

4.4.1 Results of the spike sorting on the real signal

For the real recordings we can quantify the results only for one neuron per recording, the one which was simultaneously recorded intracellularly. Thus SA and SM are very simple to calculate: $SA(\%) = 100 * C/(F + C)$ and $SM(\%) = 100 * (T - C)/T$. Here C is the largest number of the spikes from the neuron recorded intracellularly, which are sorted into the same cluster (correct detections). F is the number of spikes from other neurons which are wrongly placed into the same cluster for which C is calculated (false spikes). T is the total number of spikes fired by the intracellularly recorded neuron.

In table 4.1 and on figure 4.12 we give SA and SM for each of the real recordings. Names of the recordings are kept from [Henze 2000]. Durations of the recordings were from 60 to 120 seconds with the sampling frequency of 10kHz for some recordings and 20kHz for others. The total number of the detected spikes per recording was from 800 to 2000, while the number of spikes fired by the intracellularly recorded neuron was from 200 to 500.

The results suggest that the proposed approach improves significantly the sorting results in comparison with the basic spike sorting algorithm and the ICA algorithm. As it is expected, the proposed algorithm generally decreases the number of falsely sorted spikes and increases the number of spikes that are left unsorted (missed spikes). As we said in the introduction to this chapter, in the trade-off between the number of missed spikes and the number of false spikes it is generally highly preferable to have more missed spikes.

When the intracellularly recorded neuron is sorted first (recordings `d561102`, `d533101` and `d533102`) there is no benefits from the removal of the spikes from already sorted neurons (results in the third and the fourth column of table 4.1 are the same in such cases). Also, for those recordings contribution from detection of the overlapped spikes is very small, since most of the overlapped spikes were initially correctly assigned to the intracellularly recorded neuron (results in the last two columns of table 4.1 are the same/similar in such cases).

4.4.2 Results of the spike sorting on the simulated signal

On the other side, using simulated recordings we can quantify the results for more than one neuron and also we can easily perform several simulations and average their results. For each simulated recording we set the maximum number of clusters (sorted neurons) to 6. However, in some situations (especially for simpler methods like the basic spike sorting and combination of the basic spike sorting and ICA) less than 6 neurons have been found.

Here we do a computation of SA and SM in a bit different way than for the real signal. We compute the average $\bar{SA}$ and the average $\bar{SM}$ over 10 simulations for each j^{th} sorted neuron, where j goes from 1 to 6, which is the maximum number of the sorted neurons:

$$\bar{SA}_j = \frac{\sum_{i=1}^{N_j} SA_{i,j}}{N_j}, \bar{SM}_j = \frac{\sum_{i=1}^{N_j} SM_{i,j}}{N_j}$$

where N_j is the number of iterations in which at least j neurons were detected. We give also a standard deviation (sd) of $\bar{SA}$ and $\bar{SM}$ in order to quantify the robustness of each algorithm.

Table 4.1: Comparison of different spike sorting algorithms, in terms of Sorting Accuracy (SA) and Sorting Mistake (SM), on the real recordings. We start with the basic feature extraction algorithm (*Feat. extr.*) described in the previous chapter, its results are given in the first column; In the second column we show results for the combination of *Feat. extr.* and ICA; In the third column are the results of combination of *Feat. extr.*, ICA and removal of the spikes from already isolated neurons (*Sorted rem.*); The results in the forth column are obtained by expanding the algorithm from the third column in such way that after the applying ICA, everything but the detected spikes are removed from the recordings (*Noise rem.*); For the results in the fifth column, the algorithm from the fourth column is expanded by removal of clusters that generally contain spikes that are small in the amplitude (*Clus. rem.*); Finally the results in the sixth column are obtained by adding a sequence for detection of the overlapped spikes (*Over. det.*). The column *Recording* contains the labels of each recording.

Recording	Feat. extr.		Feat. extr. ICA		Feat. extr. ICA Sorted rem.		Feat. extr. ICA Sorted rem. Noise rem.		Feat. extr. ICA Sorted rem. Noise rem. Clus. rem.		Feat. extr. ICA Sorted rem. Noise rem. Clus. rem. Over. det.	
	SA %	*SM* %	*SA* %	*SM* %	*SA* %	*SM* %	*SA* %	*SM* %	*SA* %	*SM* %	*SA* %	*SM* %
d561102	79	46	83	7	83	7	90	8	93	8	93	8
d561103	32	3	48	5	52	15	83	17	89	20	89	14
d561104	21	2	33	4	49	12	75	13	83	16	84	11
d533101	73	67	64	26	64	26	78	28	80	27	80	27
d533102	64	52	80	13	80	13	88	15	92	18	92	17

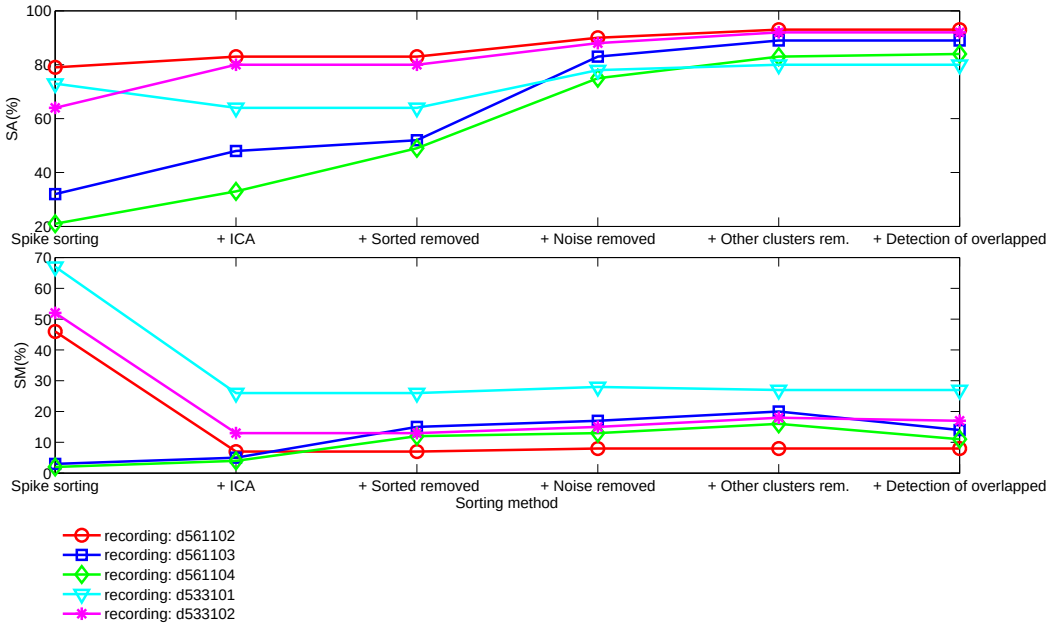


Figure 4.12: Comparison of different spike sorting algorithms for 5 real recordings, in terms of Sorting Accuracy (SA) and Sorting Mistake (SM).

The results are given in table 4.2 and on figure 4.13. Even though the results significantly vary from neuron to neuron (spikes from the neurons close to the electrode are, of course, sorted much more accurately that from the more distant ones), we can say again that overall each step of the proposed algorithm leads to significant improvement of the final results.

An important property of the proposed algorithm that increasing of SA is paid by increasing of SM as well, is now even more evident than on the figure 4.12. However, the last step (detection of overlapped spikes) decreases the SM, especially as j increases, since for the larger j (more distant neurons) many spikes were not detected because they were overlapped by the spikes from the closest neurons (which correspond to the lower j).

The standard deviation is larger for the neurons further from the electrode and tends to decrease as a more complex algorithm is used. The standard deviation illustrates the stability of the results, *e.g.* sometimes the sorting results are good even for the distant neurons, but sometimes they are quite bad for the distant neurons, this leads to larger standard deviation. On the other side the results for the neurons very close to the electrode are always relatively good, thus the standard deviation of the results is lower.

Table 4.2: Comparison of different spike sorting algorithms in terms of average sorting accuracy ($\bar{SA}$), average sorting mistake ($\bar{SM}$) and their standard deviations (sd) for first 6 sorted clusters. Labels *Feat. extr.*, *ICA*, *Sorted rem.*, *Noise rem.*, *Clus. rem.*, *Over. det.* stands for the same algorithms as in table 4.1. Label j stands for the cluster index. N is the number of trials, out of 10, in which at least j clusters were detected.

	Feat. extr.			Feat. extr. ICA			Feat. extr. ICA Sorted rem.			Feat. extr. ICA Sorted rem. Noise rem.			Feat. extr. ICA Sorted rem. Noise rem. Clus. rem.			Feat. extr. ICA Sorted rem. Noise rem. Clus. rem. Over. det.		
j	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N
1	18	15	10	68	12	10	76	10	10	85	8	10	88	6	10	88	6	10
2	45	7	2	62	13	10	74	11	10	84	8	10	86	7	10	86	7	10
3	0	0	0	54	14	5	69	13	10	78	9	10	81	7	10	82	7	10
4	0	0	0	44	16	2	60	15	10	72	11	10	78	9	10	79	9	10
5	0	0	0	0	0	0	54	16	9	68	14	10	74	11	10	75	11	10
6	0	0	0	0	0	0	39	17	7	65	15	9	72	11	9	73	11	10

	$\bar{SM}$ %	sd %	N	$\bar{SM}$ %	sd %	N	$\bar{SM}$ %	sd %	N	$\bar{SM}$ %	sd %	N	$\bar{SM}$ %	sd %	N	$\bar{SM}$ %	sd %	N
1	3	5	10	7	9	10	9	7	10	10	7	10	13	6	10	12	6	10
2	20	7	2	10	10	10	14	8	10	16	7	10	19	6	10	16	6	10
3	0	0	0	15	18	5	21	11	10	22	10	10	25	8	10	20	8	10
4	0	0	0	18	12	2	23	12	10	24	12	10	28	11	10	24	10	10
5	0	0	0	0	0	0	26	16	9	28	15	10	29	11	10	26	11	10
6	0	0	0	0	0	0	30	18	7	33	17	9	34	12	9	30	12	9

4.5 Discussion

In this chapter we proposed a new algorithm for spike sorting from multichannel recordings. We use iterative application of ICA and deflation to sort the spikes. Activity on the recording sites is mutually highly dependant, while the activity of single neurons is mutually much less dependant. This naturally invokes application of ICA to separate the neural mixture.

A condition that the number of sensors should be equal or greater than the number of sources appears as the mayor problem and limitation to application of ICA on neural recordings. The proposed algorithm uses deflation to eliminate (or at least to reduce the contribution of) some sources from the recording. In particular, we remove: 1) everything but spikes, 2) spikes which have already been sorted and 3) spikes which belong to the clusters far from the one which contains the spikes of the largest average peak-to-peak amplitude.

Such deflation results by inability of the algorithm to detect spikes fired by different neurons at the approximately same time - simultaneous firing. Only one spike per each simultaneous firing can be detected. The solution to this problem is implemented as the last step of the proposed algorithm. We take two of the separated neurons and keep in the recording only the segments while one of these two neurons was active. Then we apply ICA on such signal. Since we have only two high-energy sources in the signal, ICA will, in most of situation, separate very well the activity of the two neurons, including the

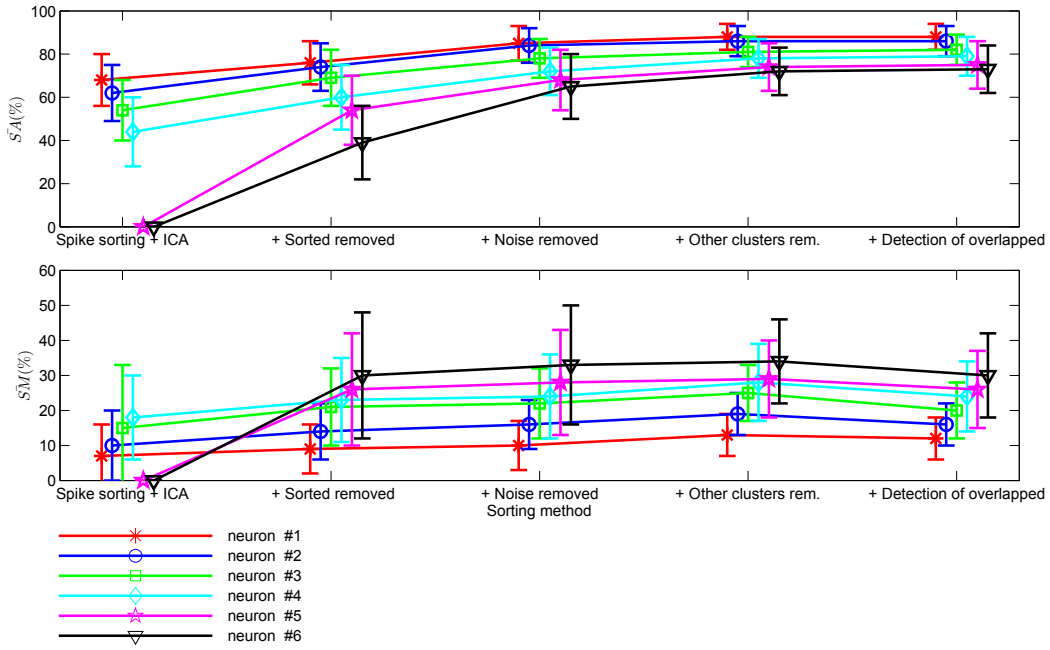


Figure 4.13: Comparison of different spike sorting algorithms in terms of average sorting accuracy ($\bar{SA}$) and average sorting mistake ($\bar{SM}$) for the first 6 sorted neurons. The length of the perpendicular lines corresponds with the standard deviation of $\bar{SA}$ or $\bar{SM}$ for the particular algorithm and the particular neuron. The neurons are labeled with respect to their minimum distance to the electrodes.

simultaneously fired spikes. We repeat this procedure for all the possible pairs of the separated neurons.

The results given in tables 4.1 and 4.2 suggest that each of the performed steps of the algorithm improves the final sorting results. Overall, the algorithm significantly increases the number of correctly sorted spikes and increases the number of missed spikes for a relatively small amount. ICA alone was not sufficient to obtain reliable results. Using the proposed algorithm led to generally solid and stable results. However, if a recording is very noisy the proposed algorithm will not be able to detect any cluster.

As spike features we used simply positive and negative peak amplitude. The proposed algorithm is compatible with practically any feature extraction method. Methods that can make the difference between the spikes fired by different neurons more evident will of course lead to better performance of the proposed algorithm, in terms of the number of sorted neurons and the sorting accuracy. Nevertheless, no matter how good the chosen feature extraction technique is, the proposed algorithm should always improve the final sorting

results when they are compared to the results obtained with the basic ICA algorithm. Finally we can say that the obtained results suggest that the proposed algorithm gives better sorting results as compared to ICA.

Chapter 5

Discussion on information coding in nervous system

Contents

The concern of this chapter is how information is coded in a nervous system. Even though we know that action potentials play a crucial role in the information transmission, it is still unclear which other biological mechanisms participate in this process and which code is used. Evidence suggest that the neural code is very extensive and includes several coding schemes. In this chapter several of these schemes are analyzed, the state-of-the-art is given. In particular a discrete nature of the neural code is closely discussed.

The neural code can be characterized through space and time dimensions. Space dimension relies on the fact that information is processed simultaneously by a vast number of neurons, the main information processing units. It has been indicated that a single stimulus parameter is usually not coded by a single neuron, but however by a large number (population) of neurons. This dimension makes the nervous system very robust and generally stable. Knowing that deaths of individual neurons are very common, lowering the importance of a single neuron by the space dimension seems to be extremely important. On the other side, time dimension, through temporal precision on different scales, makes the information processing more efficient. Later in this chapter we will analyze different coding schemes and we will see that they usually rely more on one of these two dimensions.

We first present the so-called *rate code* (section 5.1) where the information is conveyed by the number of action potentials per some time window, regardless of the precise timing within the window. Second, *population code* (section 5.2) where several neurons are coding the same information together. Then, *latency code* (section 5.3), where the information is conveyed in the latency of the first spike after the stimulus. *Interspike-interval* code (section 5.4) conveys the information in duration of the interval between two spikes. In section 5.5 a *phase code* is described. In this code the information is conveyed using the phase of the local field potentials. Finally, in section 5.6, we discussed whether the nature of the neural code could be discrete, more precisely whether the code could consist of symbols that come from a finite alphabet. We particulary analyzed pulse-position modulation in the context of neural coding.

The reader should notice that this chapter is not a debate on which coding scheme is actually used by neurons, but more an attempt to point out the possibility that several coding schemes might coexist, especially respecting the large variety among neurons and among different parts of the nervous system.

5.1 Rate code

Rate code refers to a type of code where the information is conveyed by a number of action potentials per some time window, regardless of the precise timing within the window. Several decades ago it has been indicated that a number of spikes fired by a neuron depends on stimuli intensity [Adrian 1954].

A classical example is the stretch receptor in a muscle spindle. The number of spikes emitted by the postsynaptic neuron increases with the force applied to the muscle [Adrian 1954]. Another example is the touch receptor of a leech. The stronger the touch stimulus is, the more spikes occur during the stimulation period of $500ms$ [Kandel 2000]. It is generally observed that, when the same stimulus is applied, sensory neurons have trial-to-trial variability in spike timing and number. It is assumed that temporal summation reduces this variability [Shadlen 1998].

When the rate code was first observed, the problem of coding seemed to be solved. However, very fast stimuli processing by particular parts of nervous system indicated that the whole information can not be coded only by the number of spikes.

5.2 Population coding

Knowing that a nervous system is able to obtain very fast responses to stimuli led to an idea that simultaneous activity of populations of neurons (spatial dimension) can also code an information - this is known as *population code*.

In tissues like cortex, it is generally very difficult to explicitly analyze possibility of population coding, since each neuron is connected with a large number of others. Explicit analysis is usually done on smaller elements of a nervous system, particulary parts that have short stimulus reaction time. It is assumed that in such parts population code will be much more important than the rate code. A typical examples of such, fast processing systems, are escape mechanisms.

An escape mechanism of a cockroach specie *periplaneta americana* has been particulary well studied. These cockroaches respond to approach of their natural enemy, toad *bufo marinus*, by turning and then running away. They detect the predator's approach by sensing wind gust caused by predator's movement. Wind-receptive hairs on a pair of abdominal antenna-like organs, called cerci, are displaced by the wind, thereby activating the underlying sensory cells. These sensory cells excite the interneurons in the last abdominal ganglion. The interneurons are subdivided into a ventral and a dorsal group. The ventral group contains 4 bilateral pairs of cells. At least three of these are known to be involved in specifying the direction of the initial escape turn [Comer 1993]. In contrast, the dorsal neurons are not specifically known to be involved in this process. Axons of the ventral neurons excite thoracic interneurons that, in turn, excite leg motor neurons responsible for the escape behavior. It has been shown that the numbers of spikes from the left vs the number of spikes from the right ventral neurons is a dominant cue determining whether the animal turns to the left or to the right [Liebenthal 1994], [Levi 2000].

Very similar results are obtained when a monkey's arm movement was studied by recording neurons from the motor region of the cortex
[Georgopoulos 1986]. Each neuron in charge of an arm movement coding "attracts" the movement of the arm, from any other direction, toward its own preferred direction.

Apart from these explicit observations of population code, there is an another perspective on a population based coding. This perspective is implicit, and raises from a fact that responses on repeated stimuli of a single neuron typically exhibit time uncertainty (jitter). This raises a hypothesis that such responses are made more precise by averaging on the population level. However, such hypothesis is hard to validate due to a large number of neurons which would be involved in coding of simple information.

Population code thus shows a great ability of generating a response to a stimulus rapidly, however it requires finely tuned mechanism that includes a large number of simultaneously active neurons.

5.3 Latency code

Several different experiments indicated that in some situations the temporal structure of a spike train carries additional information that can not be obtained from the spike count. This is generally called temporal coding.

Probably the simplest way of proofing that temporal structure carries information is by showing that different stimuli can result with the same number of spikes, but with different temporal structure (see [Rieke 1997] for a large number of examples).

One type of temporal encoding is the *latency code*, in which information is carried by the latency of the first spike with respect to the time of stimulus occurrence.

Existence of such type of code in rat's cortex was for example discussed in [Panzeri 2001], where spike trains which code location of an external stimulus were investigated. It was found that major part of the information about the stimuli was given by the latency of the first spike.

It is generally noticed that stronger stimulus provokes shorter spike latency [Reich 2001], [Richmond 1997].

Another example that shows correlation of stimulus properties with latency of the first spike is given in [Gollisch 2008]. Spike trains were recorded simultaneously from many ganglion cells in the isolated salamander retina. The stimuli were a uniform gray field followed by appearance of a square-wave grating. Eight different shifted versions of the grating were used in a pseudo-random sequence. Each response was characterized by two numbers: the latency of the first spike after stimulus onset and the total spike count in the response. The results obtained in that study are shown on figure 5.1. They strongly suggest that, for some types of cells, most of information is coded by the first spike latency. Although this code is simple and metabolically efficient, in order to decode it, it is necessary to have a precise information on stimulus onset time - an external reference. If the new retinal image was initiated by an eye movement, then the brain does know the onset time, but it is unclear whether this motor information gets distributed to visual centers.

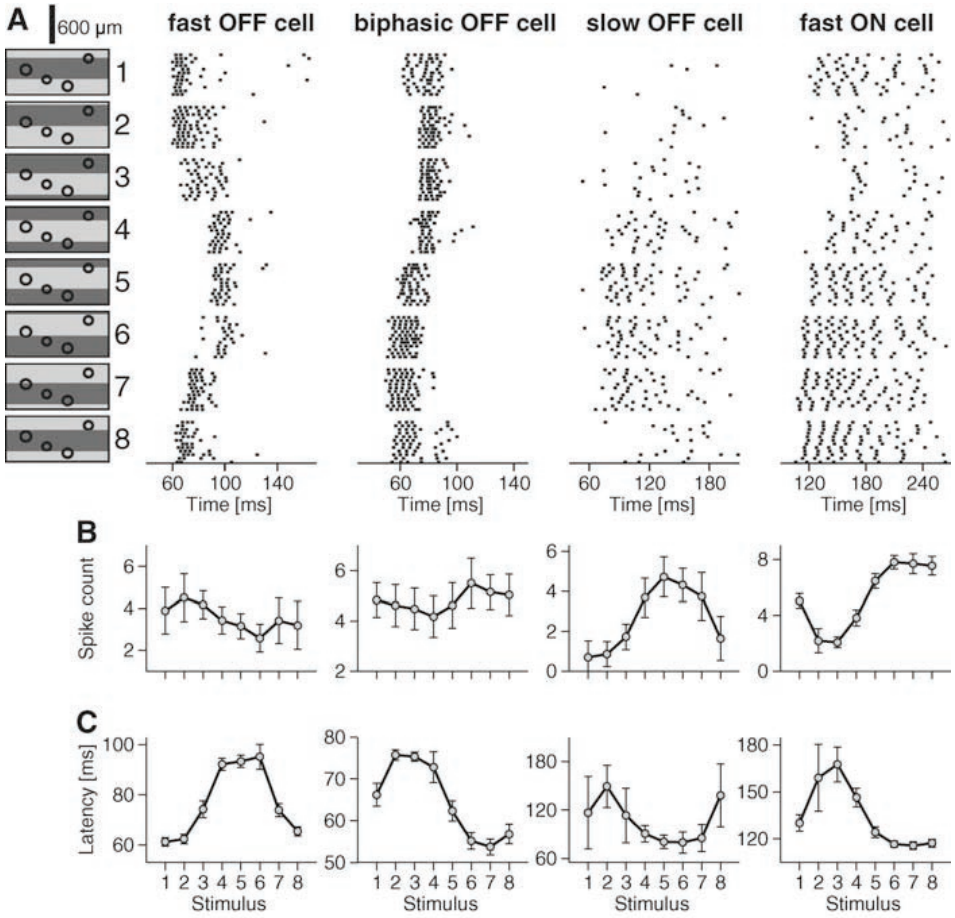


Figure 5.1: Ganglion cell responses to flashed gratings with different spatial phases. (A) Raster plots of spike responses from four types of ganglion cells (*Fast OFF*, *biphasic OFF*, *Slow OFF*, *Fast ON*) to several $150ms$ presentations of each of eight gratings. Time is measured from stimulus onset. (Left) Schematic drawings of the eight stimuli with different spatial phases. The circles show 1-SD contour lines of the spatial receptive fields of the four cells, correspondingly from left to right, in relation to the stimuli. (B) Tuning curves of the elicited spike count. The error bars show the standard deviation across trials with the same stimulus. (C) Tuning curves of the first-spike latency. *Fast OFF* and *biphasic OFF* cells typically showed strong tuning in the latency and only mild tuning in spike count. *Slow OFF* and *ON* cells, on the other hand, displayed good tuning in the spike count and often did not respond with spikes to all stimuli. From [Gollisch 2008].

5.4 Interspike-interval code

Instead of relying on an external reference, the neural code could rely on an internal one. Such internal reference could be defined, for example, by duration of time intervals between two spikes, either from the same from different neurons. This type of code is commonly refereed as *interspike-interval* code.

An interesting example that might indicate the existence of such code can be (as well as for the latency code) found in [Gollisch 2008]. For many pairs of ganglion cells, the difference between first spike times was strongly tuned by the presented stimuli. Each cell's latency underwent some trial-to-trial variation, but these fluctuations were often positively correlated in cells recorded simultaneously; when cell 1 fired earlier than usual, cell 2 tended to do the same. As a result, the latency difference fluctuated less than expected from observing the individual cell's latencies. This is demonstrated on figure 5.2. The stimuli are well represented by latencies, but also by differences between the two latencies.

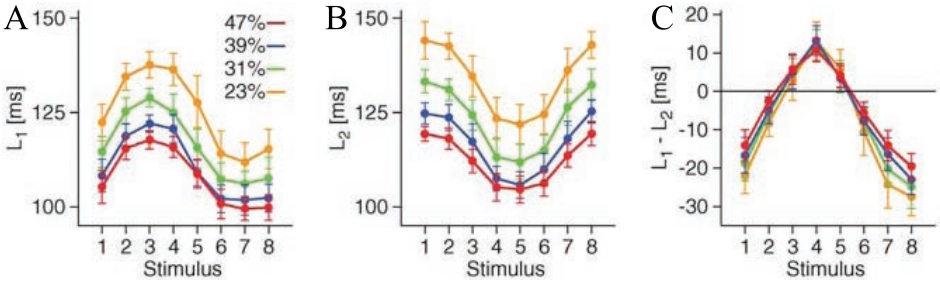


Figure 5.2: Contrast-invariant encoding by pairs of ganglion cells. (A) Latency tuning curves for a fast OFF cell whose responses were recorded for flashed gratings at different contrast levels. (B) Latency tuning curves of a second, simultaneously recorded fast OFF cell. (C) Tuning curves of the latency difference for the two neurons. From [Gollisch 2008].

5.5 Phase code

The phase code is another example of a code that relies on internal references. For the phase code these references are extracellular voltage oscillations.

5.5.1 Local field potentials

In chapters two, three and four we were talking about spike detection and sorting from extracellular neural recordings. We were concentrated on frequency band from 300Hz to 5kHz, since most of the energy from spikes fired

by the neurons close to the electrode appears to be in this band. Now we will analyze part of the spectrum below 500Hz, where some interesting time and spatial coherence can be found. When recording from *e.g.* cortex (where density of neurons is very high), approximately sinusoidal shaped, oscillations in the band from 0.05Hz to 500Hz are found. The oscillations in the band from 1Hz to 300Hz usually have the largest amplitude [Buzsáki 2004]. These oscillations are called *local field potentials* (LFP). The LFP spectrum is standardly divided into sub-bands:

- $<$**1.5Hz**: *Slow*
- **1.5-4Hz**: *Delta*
- **4-10Hz**: *Theta*
- **10-30Hz**: *Beta*
- **30-80Hz**: *Gamma*
- **80-200Hz**: *Fast*
- $>$**200Hz**: *Ultra fast*

It is interesting to point out that what EEG records is actually a projection of these oscillations on EEG electrodes, filtered by the skull.

To discuss the origin of LFP we remark that the extracellular space is non-homogenous, but generally a well conductive medium. Neural activity significantly affects this extracellular medium; we can generally identify three processes which are part of activity of any neuron and which induce change of potential outside the cell that decays with the distance from the current source:

- receiving spikes through synaptic inputs,
- spike integration and
- generation of spikes as response to stimuli.

This change of potential varies in time and thanks to the feedback effect, which will be discussed later in this section, it causes relatively stable oscillations in the extracellular medium.

The spatial coherence of these oscillations varies depending on the part of the cortex where the oscillations are recorded and the oscillations' frequency. It is generally observed that the oscillations of lower frequencies sometimes cover larger spatial area than the higher frequency oscillations

[Montemurro 2008]. This is explained by the non-homogenous structure of the extracellular medium which has frequency characteristics of a low pass filter [Brette 2011].

It is important to mention that, since a large number of neurons is necessary to create these oscillations, LFP can occur only in a large neural assemblies, such as cortex. In many other parts of a nervous system, for example escape mechanisms [Levi 2000], LFP do not appear, since such systems consist of only a few neurons.

5.5.2 Phase-locked firing

An important question regarding LFP is whether they actually influence the neural activity or whether they are only its consequence. In the sequel we will discuss this question.

First, let us mention a theoretical study, described in [Anastassiou 2010], where influence of LFP on neurons has been investigated. The study concludes that the oscillations have very significant effect on spike timings. This makes the phenomena of LFP very interesting from the perspective of neural coding.

Several experimental studies reported high correlation between phase of the ongoing oscillations and the spike timings. Two recent studies [Kayser 2009] and [Montemurro 2008] will be described here in order to illustrate properties of the phase code.

In [Kayser 2009] an extracellular activity in auditory cortex of alert monkeys was recorded. The acoustic stimuli were presented during the recordings. Figure 5.3 shows the results that examine particulary LFP in $8 - 12$Hz band. Each cycle of LFP is divided into four parts, marked with different colors so the correlation between the phase and spike timings could be easily tracked. Several interesting conclusions can be drawn from these results:

- LFP phase remains relatively stable and constant with respect to the stimulus occurrence, over all 30 trials (figure 5.3B).
- Activity of neuron given on figure 5.3C is noticeably phase-locked.
- In some situations (three of them are pointed out on figure 5.3D) the phase code conveys an information unavailable by rate code.

The authors in [Kayser 2009] concluded that the low frequency LFP (< 30Hz, what includes delta, theta and beta bands) can be combined with spiking activity to form a highly informative code. The frequencies higher than 40Hz did not add new information to the one already contained in the firing rates. A question whether more information is coded by energy or by phase of LFP

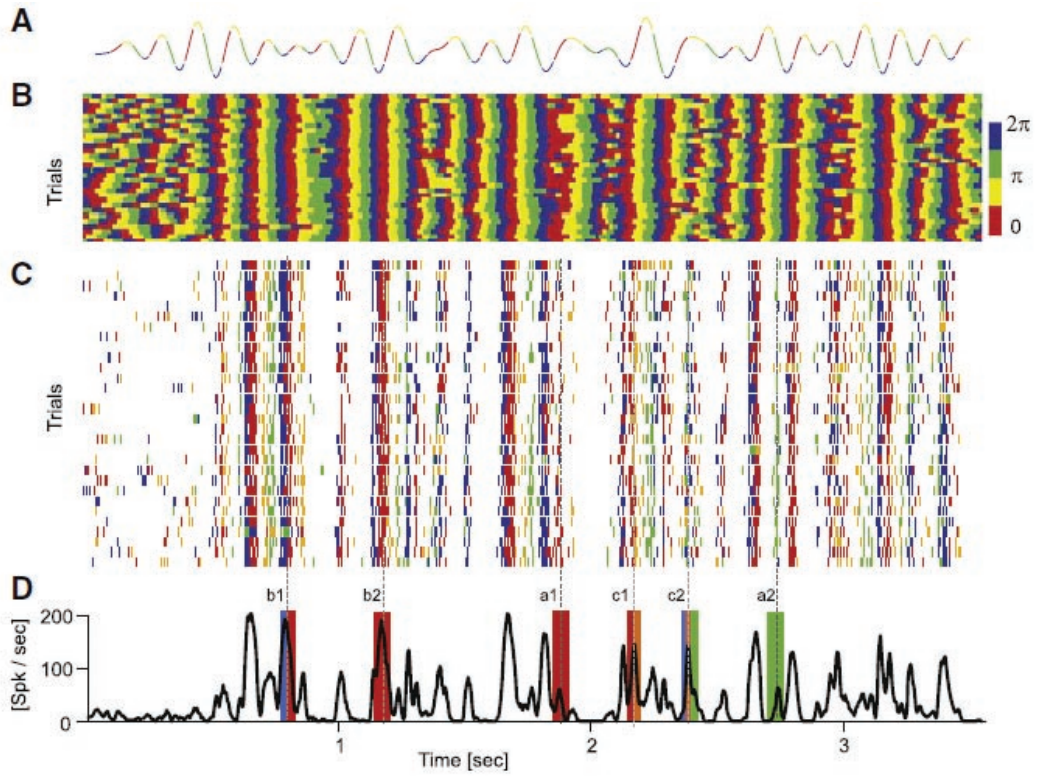


Figure 5.3: (A) Low-frequency ($8 - 12$Hz) LFP during a single trial. The phase angle is color coded. (B) Binned LFP phase during 30 subsequent trials. (C) Spike raster plot for one unit recorded during the same trials as in (B). Individual spikes are color coded according to the LFP phase at the time of spike. (D) Mean firing rate at this site. The colored bars indicate the typical phase at each peak. Pairs of peaks with similar height but different phase angle are marked by colored lines. The two instances in each pair (e.g., a1, a2) can be distinguished based on LFP phase but not based on the firing rate. From [Kayser 2009].

was also investigated in [Kayser 2009]. Their results, shown on figure 5.4, suggest that the phase code is much more informative. Measuring response when a noise is added to the stimuli led the authors to conclusion that the phase code is more robust than the rate code.

A recent study, presented in [Montemurro 2008], addresses similar questions as [Kayser 2009]. It is based on extracellular recordings from the primary visual cortex of anaesthetized macaques while binocularly presenting a color movie. The obtained results (figure 5.5) lead to the quite similar conclusions as in [Kayser 2009]. The authors of [Montemurro 2008] found that the phase

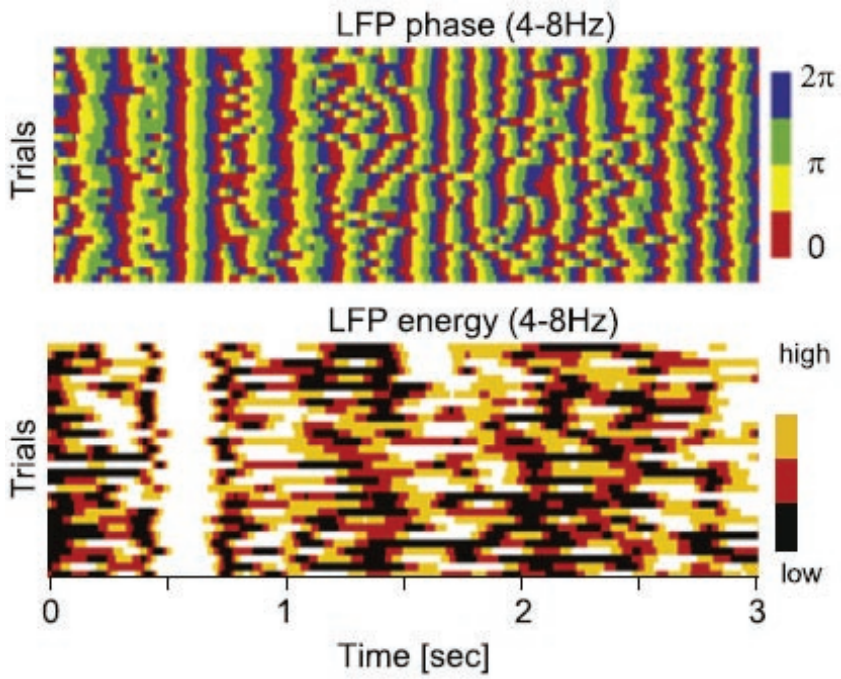


Figure 5.4: Examples of low-frequency $(4-8\text{Hz})$ LFP phase and energy for 30 subsequent trials. Both signals are binned into four bins. From [Kayser 2009].

code conveyed 54% additional information beyond that conveyed by rate code. Using circular variance they concluded that high-frequency LFP (> 50Hz) are very unreliable and can not efficiently code information about the stimuli. On the other side, LFP with frequencies below 12Hz were very reliable. Also, the firing rates were highly correlated with the LFP reliability. Generally, higher firing rates meant more reliable LFP.

5.6 Finite alphabet code

For the finite alphabet code we consider that the firing activity can be represented by a text written using symbols (characters) which are drawn from a finite alphabet. In this section we will first observe one common realization of such code in communication systems: *pulse-position modulation* (PPM). Later, we will try to investigate whether it is possible that finite alphabet code, particulary the PPM, exists in neural communication.

PPM is well known as a robust digital modulation. The information is coded in the position of a pulse fired by the transmitter, with respect to some time reference (clock). The clock is available to both transmitter and receiver. The transmitter can fire only at time instants nT_s, where $T_s > 0$

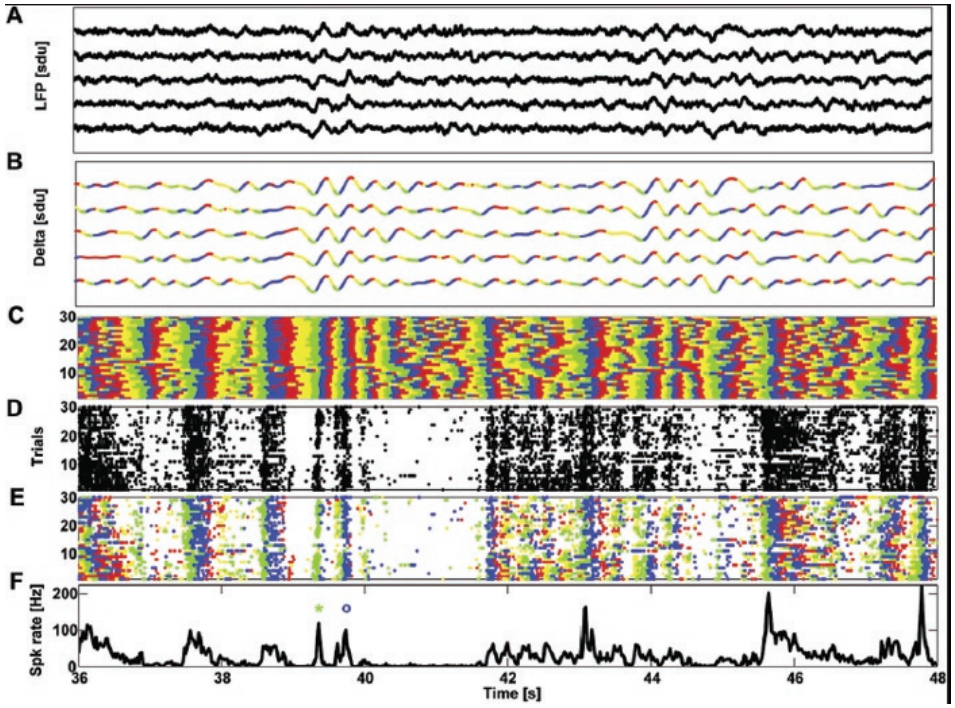


Figure 5.5: Illustration of the time course of the LFP phase and of the spikes. (A) LFP traces from five presentations of a 12s long movie segment. Traces were displaced on the vertical axis so that they could be distinguishable. (B) Time courses of the 1−4Hz (delta band) band-passed LFP to five presentations of the same 12s long movie extract as in (A). Traces were again displaced on the vertical axis. The color of the line at each time denotes to which of the four phase quadrants the instantaneous LFP phase belongs to. (C) Time course of the phases of the $1-4$Hz LFP over 30 repetitions of the movie extract. Phase values are again color coded into quadrants (D) Raster plot of spike times (indicated by dots) resulting from 30 repeated presentations of the selected 12s movie extract. (E) Raster plot of the same spike times as in (D) but with the dots representing the spikes color coded according to the $1-4$Hz LFP phase quadrant at which they were emitted. These colored spike times illustrate the phase code, whereas the colorless spike times in (D) illustrate the rate code. (F) Spike rate, averaged over all 30 trials and computed in 4ms long sliding time bins, during the 12s movie extract. The green star and the blue circle indicate movie points that elicit similar spike rate responses but different and reliable phase values. These two movie points can be much better discriminated from each other by consideration of the phase at which spikes were emitted rather than just the counting of spikes. From [Montemurro 2008].

is given by the particular code protocol available to the transmitter and the receiver and n is integer, such that $n < N$, with N also given by the code protocol. NT_s in fact defines duration of one symbol. A sample pulse train is shown on figure 5.6, where three different symbols, called "A", "B" and "C" are represented with different pulse timings. Because of the time jitter

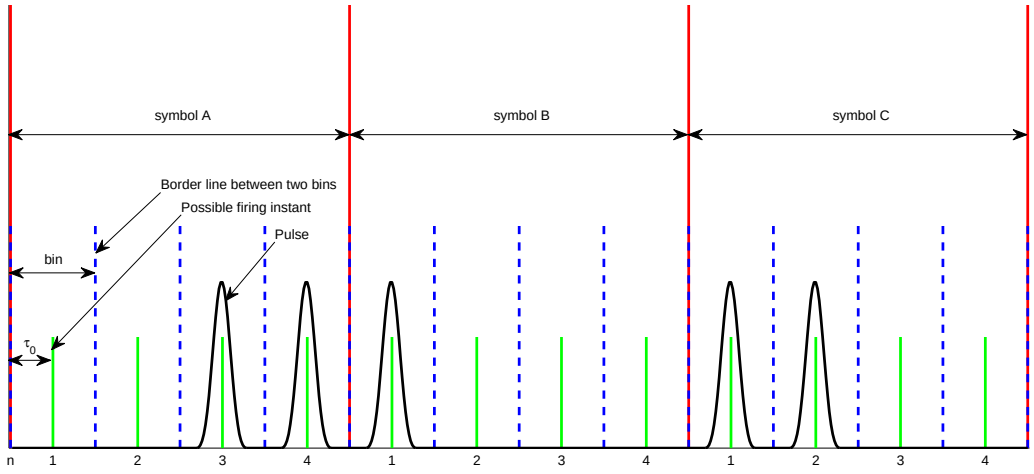


Figure 5.6: A principle of a pulse-position modulation exhibited through demonstration of coding of three different symbols.

caused by the noise in the communication channel, a pulse sent at nT_s will be received at $nT_s + \tau, \tau \in \mathcal{R}$. The code is arranged in such way that if $|\tau| < \tau_0$ the transmission is successful, otherwise the received information will not be the same as the one that was sent; here $\tau_0 \in \mathcal{R}^+$ is given by the code protocol. On figure 5.7 a possible response to the pulse train shown on the top plot is given. Despite presence of the jitter, symbols "A", "B" and "C" are correctly transmitted since $|\tau| < \tau_0$ for each pulse. In PPM, τ usually has zero mean, bell shaped distribution. The standard deviation of that distribution should be used to make optimal choice of τ_0. τ_0 is typically chosen such that probability of having $|\tau| > \tau_0$ is very low; knowing that the distribution is bell shaped, this implies that around the border of two bins the PDF is very close to zero. Since time window that corresponds to a symbol is divided into bins, it is obvious that the number of different symbols is finite *i.e.* it belongs to a finite alphabet.

5.6.1 Neural communication and finite alphabet code

Motivation for considering possibility of such type of coding scheme in neural communication raises from a common experimental observation: several

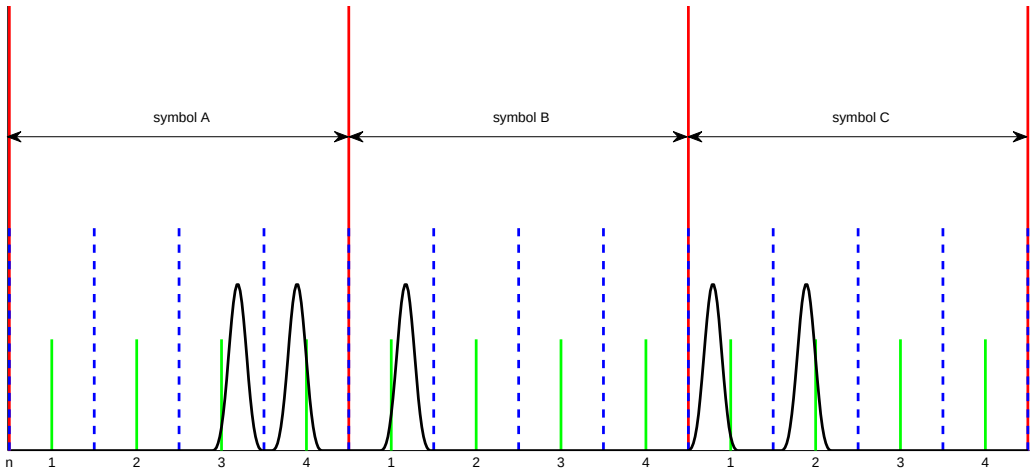


Figure 5.7: A message from figure 5.6 perturbed by the jitter caused by the communication channel. Received pulse times are not the same as the one on figure 5.6, but the reconstruction of all the symbols is still correct, since the jitter, in the particular example, was below of the half of the bin width.

studies have shown that neural responses are time locked to some level; also a jitter is widely present in those responses.

To elaborate existence of such time-locked responses we will first mention a typical results of an experimental paradigm that includes repeated stimulation. We refer to the study described in [Montemurro 2007] where a single neuron response to repeated stimulation of rat's vibrissa was recorded. The results of the study, shown on figure 5.8, clearly demonstrate the time-locking. For example, plot B on figure 5.8 shows activity of a single neuron across 100 trials: the neuron frequently fires at the same time. Similar time-locking is visible on plot C as well. The authors also calculated trial-to-trial variability in spike timing (jitter) for single neurons. First, they divided the $15s$ time course of the stimulus in 15000 $1ms$ time bins and averaged across trials to compute the firing rate in each bin. Firing episodes corresponded to local peaks in the firing rate. For each such peak, greater than some threshold they extracted all spikes fired within $2ms$ of the time of the peak and computed spike times relative to that. The results for two neurons, a bell shaped curves, similar to Gaussian distribution are shown on figure 5.9. Low standard deviation ($0.35ms$ and $0.36ms$ for the two results) indicates very high time precision.

Knowing this, let us now discuss the possible nature of the neural code - whether it is continuous or discrete. Say that when triggered with a stimulus, call it A, an observed neuron fires a spike train described with spiking

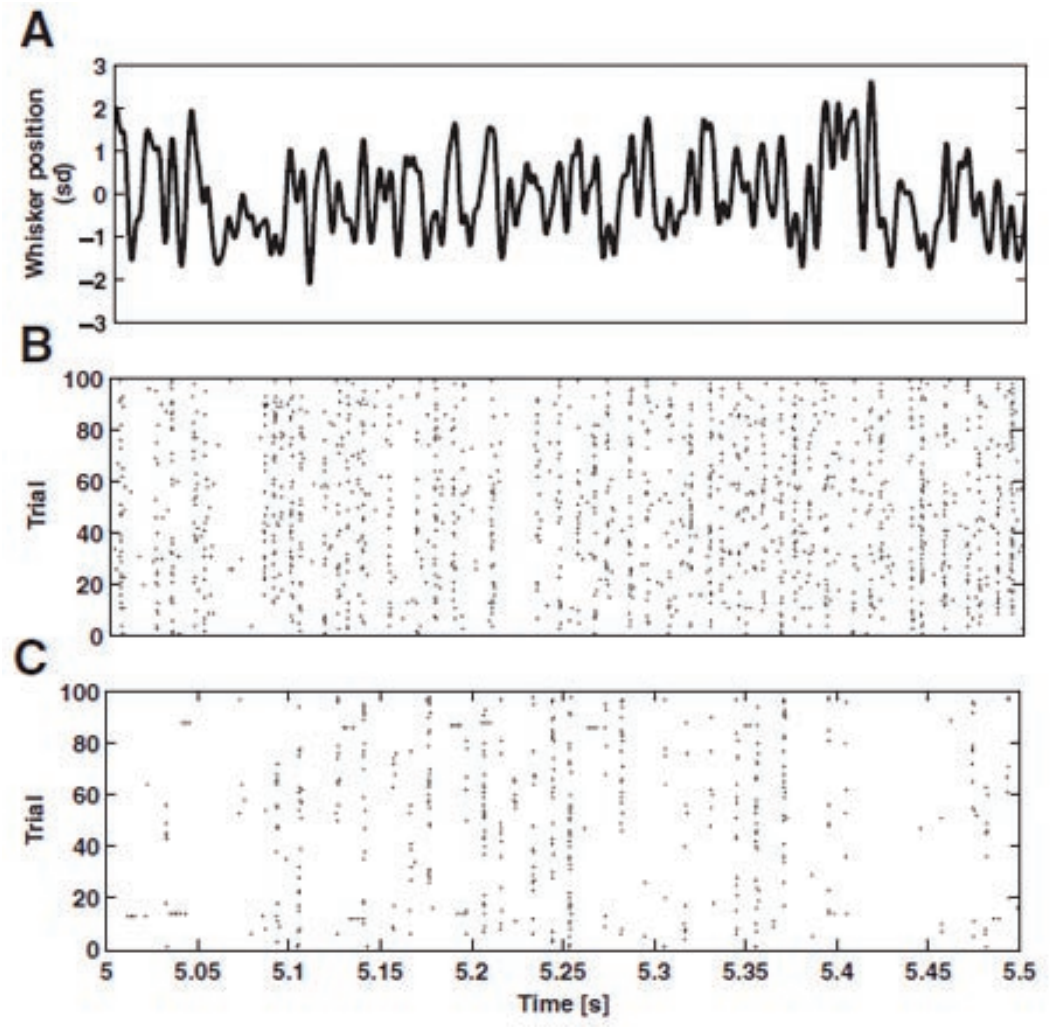


Figure 5.8: Responses of neurons to white noise vibrissa stimulation. A: vibrissa position as a function of time in units of stimulus SD ($1SD = 70\mu m$). B: spikes fired by neuron 1 in response to 100 repetitions of the stimulus shown in A. C: response of neuron 2 to the same stimulus. From [Montemurro 2007].

instants $t_1, t_2, ..., t_N$, where N is the number of spikes fired as a response to the stimulus. Let us assume that the spike train codes some property of the stimulus. We raise a question whether the neural code is such that the spike train $t_1 - \delta, t_2, ..., t_N$, where $t_1 > \delta$ (for causality purpose) and $\delta \neq 0$, necessary carries different information than the spike train $t_1, t_2, ..., t_N$.

If the answer to this question is yes, then, taking into account the observed jitter in the neural response to the repeated stimuli, we can say that it is not possible to deduce the original message (the stimulus property that is coded

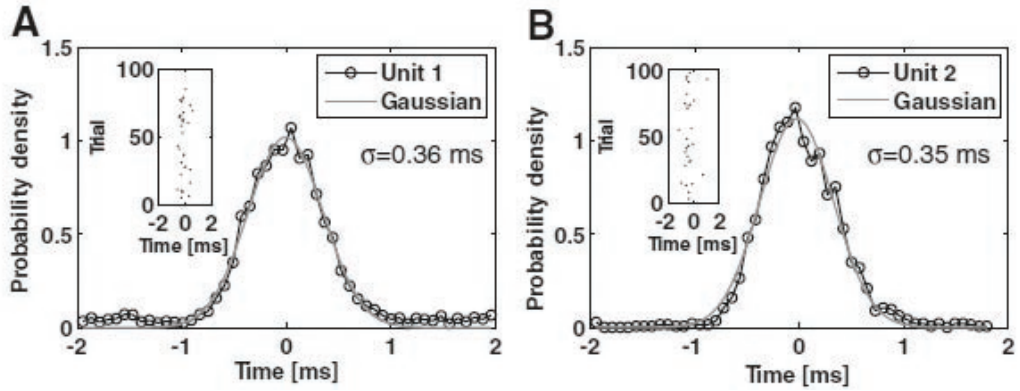


Figure 5.9: A: probability density of trial-to-trial differences in spike time together with the fit to a Gaussian distribution. B: corresponding results for unit 2. In both panels σ is the standard deviation of the fitted distribution. From [Montemurro 2007].

by that neuron) from the observed, single neuron, response. For the further discussion, we denote such case as *continuous time coding*.

In the opposite case, if it is possible that two different spike trains: $t_1, t_2, ..., t_N$ and $t_1 - \delta, t_2, ..., t_N$ carry the same information, then, despite the jitter, the message can be decoded even when only a single neuron is observed. We denote such general scheme as *discrete neural code*.

Let us first discuss a bit further the continuous time coding. Even though such coding makes it impossible to recover the original message from a single neuron, if the same stimulus property is coded by a large number of neurons it can be possible to understand the message. To elaborate this let us say that population of M neurons codes the same message and that each of the neurons responds with the N spikes that can be described with the spike trains: $t_j - \delta_{i,j}$, where $i = 1, 2, ...M$, $j = 1, 2...N$ and $t_j > \delta_{i,j} \forall i$, such that for any j, $\delta_{i,j}$ is zero mean. Then for any j a histogram created from the time occurrences $t_j - \delta_{i,j}$ of all j^{th} spikes from all M neurons will have a peak at t_j when $\lim_{N\to\infty}$. This is true if $\delta_{i,j}$ is uncorrelated over all M neurons for any j (*i.e.* $\{\delta_i\}$, $i = 1, 2, ...$ is an uncorrelated sequence). Whether this is the case is still an open question. Even though it seems that the large complexity of neural circuits will results that jitter δ_i is uncorrelated, some studies suggest that in some parts of the nervous system the jitter could be correlated. For instance, in the study described in [Gollisch 2008], that we have already mentioned in description of latency and interspike interval code, it is observed that trial-to-trial variations of the spike latencies were positively correlated among different neurons. On the other side, in [Michael 2002] a binary coding scheme was demonstrated in

a rat auditory cortex. The authors concluded that in the particular case each neuron responds to a stimulus with only one spike and that the time jitter was eliminated on the population level.

We discuss now more closely about the continuous time coding on a level of a single neuron. We concluded that, due to a jitter and under assumption of continuous time coding, two different messages can not be distinguished on a single neuron level. However, depending on the nature of the message that is being coded, we can possibly talk about the coding error. For example, if the message has the form of a number (especially real number) which represents some stimulus property (*e.g.* sound frequency) then it is possible that two similar spike trains, in terms of spike times t_j, encode similar frequency. In such case a smaller jitter would lead to a smaller mistake in the coding of the message and a single neuron activity itself could code a message with some limited accuracy.

In discrete time code, when $t_1, t_2, ..., t_N$ and $t_1 - \delta, t_2, ..., t_N$ can carry the same information, we assume that response to a stimulus is divided into some non-overlapped bins, such that spike occurrence anywhere within the bin carries the same information. This brings us to the finite alphabet PPM coding scheme.

In order for the finite alphabet code to exist within a nervous system it is necessary to have some clock that defines the bins. To make the decoding process possible such clock should have some spatial coherence *i.e.* it should be available to both, the neuron that codes the message and the neuron that has to decode it. Also, an another clock that defines symbol length should also exist.

5.6.2 Finite alphabet code based on LFP

An interesting candidate for such clock are the LFP. Just to give some general idea how such a clock could separate symbols and bins we show on figure 5.10 a possible relation between clock cycles and symbols and bins durations.

An important question, that naturally raises when potential role of LFP is discussed, is how fine the phase resolution is in a nervous systems, *e.g.* if one postsynaptic stimulus is received by a neuron at phase φ_1 and another at phase φ_2 of the same cycle of local oscillations, what is the minimal value of $|\varphi_1 - \varphi_2|$ the neuron can detect? In other words, what is the phase precision of the nervous system?

This question has been addressed in [Montemurro 2008] and it was found that to extract information from the phase of firing, it is necessary to measure phases with a precision of approximately $\pi/2$. The exact precision depends on several factors that influence LFP reliability such as, as mentioned in the

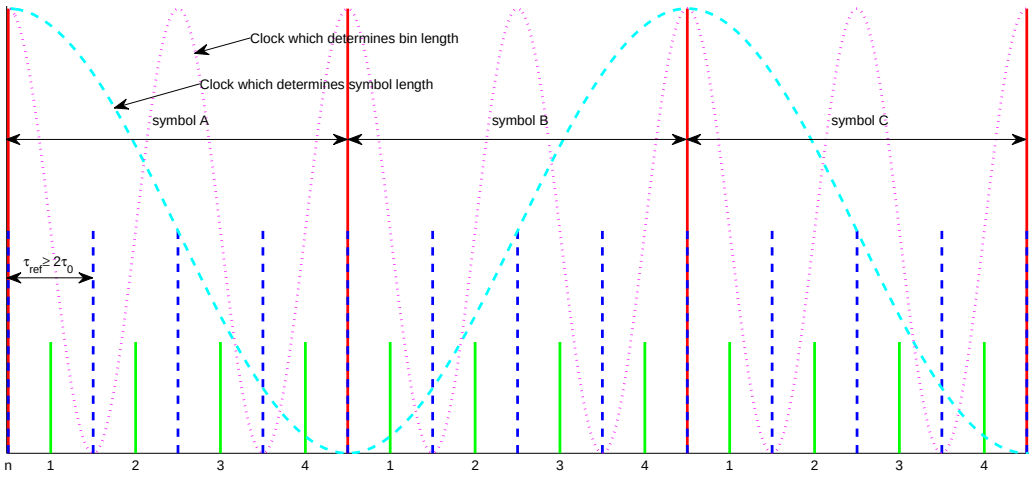


Figure 5.10: A possible relation between clock cycles and symbols and bins durations. τ_{ref} stand for the refractory period; the bin width should be at least τ_{ref}.

previous section, firing rate and LFP frequency.

Another very interesting study, described in [Jensen 2000], has also addressed this question. The principal goal of the study was to investigate a possibility of the phase coding in a rat hippocampal *place cells* (cells that exhibit a high firing rate whenever an animal is in a specific location in an environment that corresponds to the cell's *place field*). The data from the first half of the recordings was used to find correlations between the rat's position and the firing phase of individual cells. For the second half, these correlations were used to reconstruct position from the observed spikes. The average distance between the reconstructed position and the actual position was used as a quantitative measure of the efficacy of different decoding schemes. As in the studies described in the previous section, it was shown that the phase code conveys more information than the rate code. The authors particulary investigated the number of bins in which one oscillation cycle is divided (phase precision). They found that dividing into 6 or 7 phase bins leads to significantly better reconstruction results (reconstruction of position of the rat was more accurate) than when 1, 2, 3 or 9 bins were used.

The conclusions from [Jensen 2000] were exploited by one of the authors of that study to propose the finite alphabet coding scheme [Lisman 2005]. Such scheme is based on the assumption that the observed discrete cycles are actually modulated by higher frequency oscillations (gamma band). The studies discussed in the previous section did not show high coherence of gamma oscillations in visual [Montemurro 2008] and auditory [Kayser 2009] cortex.

However, some contrary results are reported for dentate region of hippocampus - coherence of gamma oscillations were very high over long distances. If these results are correct, in some parts of the cortex gamma oscillations might be one of the clocks; the other, lower frequency clock could come from theta band oscillations, which seems to have a generally good spatial coherence. Figure 5.11 shows gamma-modulated firing of rat pyramidal cells during different states. The author used the results to propose a discrete coding scheme for long-term memory of the place cells (figure 5.12).

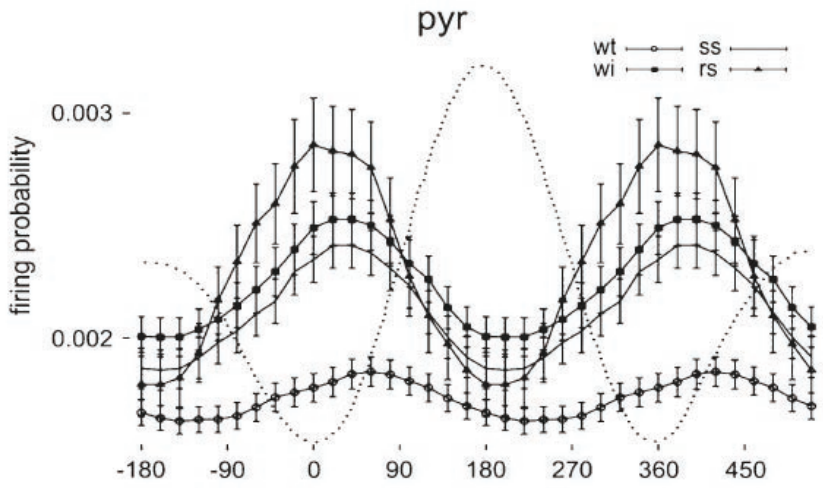


Figure 5.11: Gamma-modulated firing of rat pyramidal cell during different states (wt, waking theta; wi, waking immobility; ss, slow wave sleep; rs, REM Sleep). Dashed line is gamma LFP waveform. The "waking" data are taken at a time when the theta phase precession was occurring, thus demonstrating that both theta and gamma oscillations influence the timing of firing. From [Lisman 2008] where it is modified from [Senior 2008].

If the neural code resembles pulse-position modulation scheme used in communication systems then firing of more than one spike within the same bin should be impossible. This would imply that the width of the bins in neural communication should be lower or equal to the refractory period. Since the duration of a spike and the corresponding absolute refractory period is about $1 - 2ms$, a clock of $500 - 1000$Hz should exist to define such bins. However, up to now neuroscientist have not found a coherent clock of such frequency.

Considering pulse-position modulation, if T_s is fixed then it is relatively easy to check whether such modulation has been used. Even if we do not a priori know T_s (which is the case in neural coding problem), we can test different values for T_s and observe whether we will find a single which is a multiple of all the firing instants. Of course, to find such T_s, if it exists, it

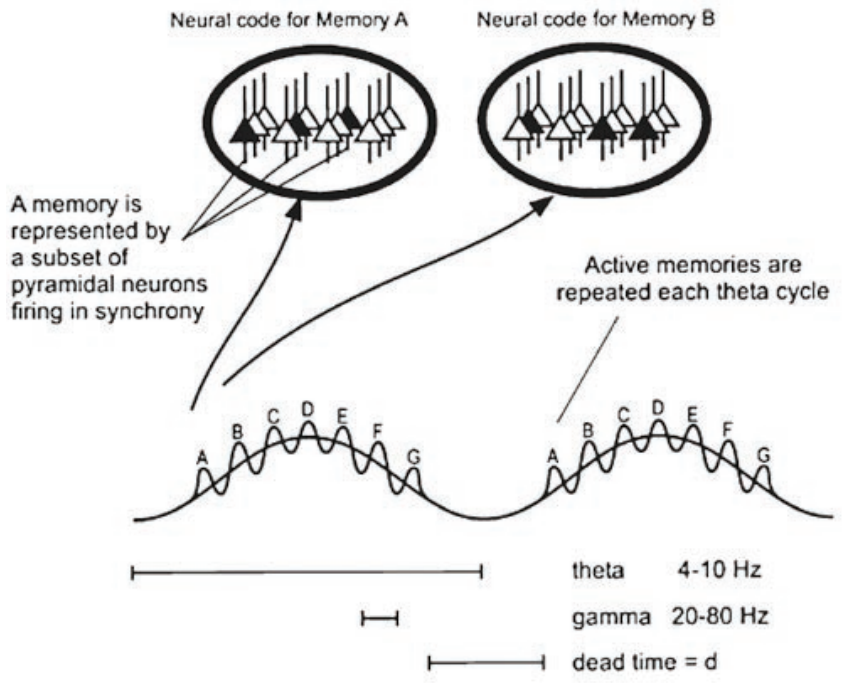


Figure 5.12: Scheme of Theta-Gamma Discrete Phase Code. Two theta cycles are shown, each of which contains 7 gamma cycles. Different items (A-G) are represented by activity in different gamma cycles. The ovals above show the network activity that represents memory A during the first gamma cycle; note that a spatial code of active cells (black) is used to encode the memory. During the next gamma cycle, different cells in the same network are active, thereby encoding memory B. This entire pattern can repeat on the next theta cycle. Note that in this schematic, the theta modulation of gamma amplitude is not shown. From [Lisman 2008].

is necessary to have access directly to the transmitter. In neural recordings, this means to record directly from a single neuron. This is feasible using *e.g.* a technique called *patch clamp*. Results obtained with such recordings do not suggest existence of fixed T_s. However, the pulse-position modulation is still possible if T_s varies in some way known to both, transmitter and receiver.

At the end of this section we can not draw any conclusion whether pulse-position modulation, used in communication systems, is used in neural networks as well. Moreover, we can not say whether the neural code is continuous or discrete. Even though we know a lot about neural coding, these elementary properties of the code are still mysteries. It seems likely that these questions will not be fully answered soon. Even finding a substantial amount of evidences for one coding scheme in one brain region does not mean that in some

other regions neural coding does not rely on a completely different scheme: the difference might be that large so continuous and discrete coding scheme could even coexist, *e.g.* one for sensory processing and the other for memory.

5.7 Discussion

Knowing that synaptic transmission, a necessary part of neural communication, is very noisy, and yet the neural systems perform extremely efficient, gives an important perspective on neural code. Regardless how the information is coded, the code is for sure a very robust one.

The main observation that can be made after the analysis in this chapter is that different neurons in different parts of the nervous system seem to rely on different coding schemes. The proposed schemes, in most of cases, do not exclude each other. For example, it is possible that some neuron codes majority of information by LFP phase-locked timing instants, but still some part of information can be given by the number of spikes. This same-neuron-different-codes scheme is demonstrated *e.g.* in [Knutsen 2008] and [Richmond 1997]. It seems that multiple neural codes might operate simultaneously at different temporal scales, where each carries complementary information [Panzeri 2010]. We even state that it is possible that the nature of the neural code can be both, continuous and discrete, depending on the brain region but also on the time scale. For example, pulse-position modulation can carry information in both, precise position of a spike in discretized time and in number of spikes per some time unit (rate code).

To analyze different coding schemes, responses of sensory neurons are usually observed. When recording from sensory neurons it is possible to conduct an experiment where the stimuli properties can be controlled, *e.g.* precise timing and amplitude. Thus, a correlation between some particular parameters of the spike train and the stimuli can be analyzed (*e.g.* delay of the first spike, inter-spike intervals histogram). In this way it is possible to see which of the parameters represent the stimuli in the most accurate and informative way. However, this raises a question whether some cortex regions or particular types of neurons might be discriminated in such type of research.

Existence of LFP and correlation of their phase with spike timings gives a new perspective on neural coding. It might be possible that LFP play a role of a clock in neural communication. Such clock might tick on several different frequencies at the same time, giving different possibilities of time multiplexed code. Having a clock opens a possibility of existence of discrete time coding schemes, such as pulse-position modulation. Motivation for considering discrete code also lies in the time-locked neural responses. These responses

exhibit a jitter (even though the jitter is usually relatively small), thus discrete time code might exist to increase the robustness of such code.

Advances in the research of neural coding generally have a large influence in many different areas of science. From philosophical point of view, where better understanding of neural coding is important for better understanding of the consciousness and what we actually are to more practical aspects such as neural prothesis and brain stimulation which were discussed in the introduction of the thesis.

Let us now discuss one research field, strongly related with the neural coding problem: artificial intelligence (AI). We particulary address so called *strong* AI which stands for AI that matches or exceeds human intelligence. Often considered as futuristic, but still widely debated, this field is particulary interesting in the context of this chapter, because some hypothesis discussed here could have a strong implications in that field. Let us first describe briefly some basic ideas in *Strong* AI field and then we will relate them with discussions given in this chapter. *Strong* AI is based on idea of emulating neural communication using computers. It is often assumed that by simulating more neurons than there is in the human brain it would be possible to exceed human intelligence. Some discussions went so far that they try to guess (using so called *Moore's law*[1]) when this will be possible; the time when such intelligence will be developed, if ever, is usually refereed as technical singularity. As an example of research towards *strong AI* direction, we mention *The blue brain project*, which started in 2005, and attempts to create a synthetic brain by reverse-engineering the mammalian brain. In 2007 the end of the first phase: *Modeling and simulation of the first rat cortical column* has been reported. Modeling of a brain, or a part of a brain usually means computational modeling of neural networks: neurons and connections between them. For example in the first phase of *The blue brain project* approximately 10000 neurons had been modeled according to their real biological structure in rat neocortical columns [Markram 2006]. Even in its developing phase, such project is extremely beneficial for understanding of cortical dynamics. However, to make such a model that is a precise functional replica of the brain (or some of its part) it might be necessary to address the influence of the extracellular medium on neural communication. As we saw in this chapter, several studies suggest that changes of electrical potential in the extracellular medium (mostly caused by the neural communication itself) can affect the communication and in some situations seem to be a part of the code (*e.g.* firing with respect to the phase of the LFP cycle). If it is true that neurons

[1] A trend that the number of the transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.

convey information through both, action potentials and extracellular potentials then neglecting the extracellular potentials obviously leads to a limited representation of the real neural networks. In that case, regarding the *strong* AI, modeling of a large neural networks with a number of neurons similar to the one in human brain will not necessarily lead to similar performance in terms of *e.g.* intelligence. To make artificial neural networks which are real replica of the biological ones it might be necessary to model the extracellular environment as well. This drastically complicates the modeling process, since neurons are usually modeled using relatively simple adequate models (see e.g. [Carnevale 2006] for a description of a commonly used software NEURON which is based on the famous *HodgkinHuxley* neural model). To model the extracellular environment with some precision which is enough for making functional replicas, some different approaches might be needed. For example using finite element methods where the biological tissue would be divided into very small portions and for each portion the change of potential along discrete time could be calculated. A different example would be using some adequate schematic for extracellular space. Finding an optimal way to realistically simulate biological neural networks, in case that the neural code depends on extracellular potentials, remains a subject for future research. A very important step in that research would be a better understanding of the neural code, what would help to realize what kind of a model could be a satisfactory representation of the extracellular dynamics.

Among the one that has been analyzed here, there are some other ongoing debates and generally new perspectives on problem of neural coding and modeling of the neural activity. In [Cessac 2008] the authors discuss how fine time resolution is necessary to adequately model neural behavior. Commonly neural recordings use sampling frequency of 10kHz which is considered to be enough to represent the spectral properties of spikes. Using continuous time calculations the authors in [Cessac 2008] analyze effects of time discretization on neural models. The main issue was not how fine the resolution should be, but what are the effects of discretization that are not taken into account in discrete neural models. They identify situations where small disturbances in spike times may critically change the circuit dynamics.

We conclude this chapter by observing that complete neural coding scheme is still unknown even though some of its principles seem to be understood. However, in the last few years more questions than answers has been raised. Complexity and variety of the neural language continue to astonish us, especially since we know that all these principles are integrated in such a profound performance, as the one the nervous system exhibits.

Chapter 6
Conclusion

Several important and challenging problems in neural signal processing and computational neuroscience are discussed and tackled in this thesis. Despite a large amount of scientific effort, particulary during this and the last century, a nervous system still holds many mysteries. An elementary motivation behind the research done within this thesis was a better understanding of how a nervous system conveys and process the information. The goal of the thesis was to give a small, but recognizable contribution, by proposing some algorithms that could help in processing of neural signals and by raising some questions about the nature of the neural code.

From the anatomical point of view, our knowledge about the nervous system is quite advanced. Still, due to the extreme complexity of the most interesting part of the nervous system - the brain, we are still not able to understand how the sensory information is being processed and how it leads to such proficient outputs. What is generally noticed by observing the nervous system is that whatever code is used to represent the information it must be very robust, since occasional dying of individual neurons as well as noisy inputs do not perturb significantly the information transmission.

To summarize our anatomical knowledge about the nervous system into just a very few words, we can say that we know that the principal information processing units are neurons (in human brain around 100 billion of them) that are mutually connected (in the human cortex each neuron is usually connected with about 10000 others) and exchange information. Neurons are not unique, contrary they can have significantly different anatomical structure. Majority of information is transmitted by using action potentials (spikes). Connections between neurons are called synapses and they have an ability to transfer the ion current from one neuron to another. Moreover, the synapses multiply their inputs with a certain weight coefficient. These weight coefficients are functions of the past responses of the system on a more global level. Such way of changing the weight coefficient between the synapses is what permits the neural networks to learn.

To understand the communication between neurons it is important to analyze the individual neural activity. Among many recording techniques, in this

thesis we mostly addressed extracellular neural recordings. These are particulary interesting since they can be obtained with multi-site electrodes and on moving animals.

We observe the problem of understanding an information transmitted within a nervous system as a three step problem. As the first step we consider spike detection, the second step spike sorting and the third decoding. This three step approach describes well the structure of the thesis.

After the introduction in chapter one, in chapter two, we attacked the spike detection problem. The problem is seen as a change point detection problem. The spike time occurrences appear (as irregularities) explicitly in the distributional derivatives of the neural signal. Using operational calculus, which provides a convenient framework to handle such distributional derivatives, we characterize the time occurrence of a spike by an explicit formula. Such method brings a new perspective on spike detection problem since introducing explicit derivatives gives a whole new mathematical framework to deal with spikes. For a large class of different spike models we can obtain an explicit characterization of spike locations, but due to the noise such explicit characterization leads to errors. We thus implement several decision functions which lead to more robust results. Since the obtained mathematical results can be seen as a whole framework, depending on particular change point detection problem (*e.g.* desired trade-of between the number of the false alarms and the correct detections - detection sensitivity), one can try to realize the most suitable detector. Very simple implementation of the proposed algorithm, through classical FIR filters, makes its application in online spike detection systems very easy, suggesting possible commercial implementation in neural prosthetics or some applications that will be developed in the future.

Following the spike detection, we address the spike sorting problem; first, in chapter three, by proposing an algorithm for spike sorting from a single-site recordings and after, in chapter four, an algorithm for spike sorting from a multi-site recordings, that uses the algorithm from the chapter three. The proposed algorithm for spike sorting from single-site recordings is based on properties of distribution of the extracted features from the detected spikes. The algorithm is simple and yet adequate when difference between spikes from different neurons is relatively large. The proposed multi-site sorting algorithm uses an iterative ICA and deflation of what we consider as undesirable parts of the signal. The proposed algorithm obtained much better results than when only ICA was applied. In general our results suggested that ICA has a great potential for neural spike sorting, even though all formally required conditions on the signal for successful ICA application were not fulfilled.

There are some other open questions in field of spike sorting that we did not address here. For instance, techniques for spike feature extraction are

under constant development. These play a very important role in the process of spike sorting. Also we did not particularly address problem of electrode drifts, we assumed that the electrode setup was stable.

Spike sorting algorithms, including the ones proposed here (especially the iterative ICA based algorithm), generally can not perform in real time. This is the major problem from perspective of clinical applications, which generally require online processing. On the other side, research applications are usually compatible with the offline processing.

Finally in chapter five, we discussed the neural coding problem itself. We raised a question on the nature of the neural code, whether it is continuous or discrete. The case that it is discrete was particulary investigated. The finite alphabet coding scheme was presented and related with some existing examples from the literature. Such coding scheme is linked with a well known pulse-position modulation. Notice that we do not give an answer whether the neural code is either continuous or discrete, but state that it is an open question and simply discuss the discrete case by analyzing its potential structure and the implications of such coding scheme. It seems that the nervous system is using several different coding schemes. The nature of the neural code could be continuous in some parts (*e.g.* sensory processing) and discrete in some other (*e.g.* memory). Moreover, continuous and discrete coding schemes might coexists in the same parts of the nervous system *i.e.* the nature of the neural code might be dual.

Overall, we can say that the thesis bring together some advanced mathematical tools, signal processing techniques and communication theory in order to attack several different problems in neuroscience.

Bibliography

[Adamos 2008] D. A. Adamos, E. K. Kosmidis and G. Theophilidis. *Performance evaluation of PCA-based spike sorting algorithms.* Computer Methods and Programs in Biomedicine, vol. 91, pages 232–244, September 2008. 72

[Adrian 1954] E. D. Adrian. *The basis of sensation; some recent studies of olfaction.* British medical journal, vol. 1, no. 4857, pages 287–290, February 1954. 124

[Aksay 2001] E. Aksay, G. Gamkrelidze, H. S. Seung, R. Baker and D. W. Tank. *In vivo intracellular recording and perturbation of persistent activity in a neural integrator.* Nature Neuroscience, no. 2, pages 184–93, 2001. 39

[Anastassiou 2010] C. A. Anastassiou, S. M. Montgomery, M. Barahona, G. Buzsáki and C. Koch. *The Effect of Spatially Inhomogeneous Extracellular Electric Fields on Neurons.* The Journal of Neuroscience, vol. 30, no. 5, pages 1925–1936, February 2010. 130

[Bankman 1993] Isaac N. Bankman, Kenneth O. Johnson and Wolfger Schneider. *Optimal Detection, Classification, and Superposition Resolution in Neural Waveform Recordings.* IEEE Transactions on Biomedical Engineering, no. 8, pages 836–841, August 1993. 60

[Basseville 1993] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice-Hall, 1993. 33

[Belkoura 2009] L. Belkoura, J.P. Richard and M. Fliess. *Parameters estimation of systems with delayed and structured entries.* Automatica, pages 1117–1125, 2009. 40

[Bla 2009] Cerebus data acquisition system - users manual. Blackrock Microsystems, 2009. 75, 95

[Brette 2011] R. Brette and A. Destexhe. Handbook of neural activity measurement. Cambridge University Press, 2011. 130

[Brown 2001] G. D. Brown, S. Yamada and T. J. Sejnowski. *Independent component analysis at the neural cocktail party.* Trends in Neuroscience, vol. 24, no. 1, pages 54–63, January 2001. 96

[Buzsáki 2004] G. Buzsáki and . Draguhn. *Neuronal oscillations in cortical networks.* Science, vol. 304, no. 5679, pages 1926–1929, June 2004. 129

[Carnevale 2006] N. T. Carnevale and M. L. Hines. The NEURON book. Cambridge University Press, 2006. 144

[Cessac 2008] B. Cessac and T. Vieville. *On dynamics of integrate-and-fire neural networks with conductance based synapses.* Frontiers in Computational Neuroscience, vol. 2, 2008. 144

[Chah 2011] E. Chah, V. Hok, A. Della-Chiesa, J. J. H. Miller, S. M. O'Mara and R. B. Reilly. *Automated spike sorting algorithmbased on Laplacian eigenmaps and k-means clustering.* Journal of neural engineering, vol. 8, no. 1, February 2011. 72, 89

[Cho 2003] S. Y. Cho, B. Y. Kim, E. H. Park, J. W. Kim, W. W. Whang, S. K. Han and H. Y. Kim. *Automatic recognition of Alzheimer's disease with single channel EEG recording.* International Conference of the IEEE Engineering in Medicine and Biology, vol. 3, pages 2655–2658, 2003. 63

[Comer 1993] C. M. Comer and J. P. Dowd. *Multisensory processing for movement: antennal and cercal mediation of escape turning in the cockroach.* In Proceedings of the workshop on "Locomotion Control in Legged Invertebrates" on Biological neural networks in invertebrate neuroethology and robotics, pages 89–112. Academic Press Professional, Inc., 1993. 125

[Comon 2010] P. Comon and C. Jutten. Handbook of blind source separation, independent component analysis and applications. Academic Press (Elsevier), 02 2010. 99

[Delescluse 2005] M. Delescluse and C. Pouzat. *Efficient spike-sorting of multi-state neurons using inter-spike intervals information.* Journal of Neuroscience Methods, no. 1, pages 16–29, 2005. 29

[Delescluse 2006] M. Delescluse and C. Pouzat. *Efficient spike-sorting of multi-state neurons using inter-spike intervals information.* Journal of neuroscience methods, vol. 150, no. 1, pages 16–29, January 2006. 30, 72

[Desobry 2005] F. Desobry, M. Davy and C. Doncarli. *An Online Kernel Change Detection Algorithm.* IEEE Transactions on Signal Processing, no. 8, pages 2961–2974, 2005. 33

[Dornhege 2007] G. Dornhege, J. R. Millán, T. Hinterberger, D. J. McFarland and L. R. Muller. Toward brain-computer interfacing. MIT Press, 2007. 63

[Eric 2003] A. N. Eric. *Glial Cell Inhibition of Neurons by Release of ATP.* Journal of Neuroscience, vol. 23, no. 5, pages 1659–1666, 2003. 3

[Fan 2006] T. H. Fan, K. C. Chang and C. B. Lee. *Bayesian Estimation of the Number of Change Points in Simple Linear Regression Models.* Communications in Statistics: Theory and Methods, no. 4, pages 689–710, 2006. 33

[Fee 1996] M. S. Fee, P. P. Mitra and D. Kleinfeld. *Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability.* Journal of Neuroscience Methods, vol. 69, no. 2, pages 175–188, November 1996. 80

[Fliess 2003] M. Fliess and H. Sira-Ramírez. *An algebraic framework for linear identification.* ESAIM: COCV, vol. 9, pages 151–168, 2003. 34, 39, 41

[Fliess 2010] M. Fliess, C. Join and M. Mboup. *Algebraic change-point detection.* Applicable Algebra in Engineering, Communication and Computing, vol. 21, pages 131–143, 2010. 34

[Fraley 1998] C. Fraley and A. E. Raftery. *How many clusters? Which clustering method? Answers via model-based cluster analysis.* The Computer Journal, vol. 41, pages 578–588, 1998. 72

[Georgopoulos 1986] A. P. Georgopoulos, A. B. Schwartz and R. E. Kettner. *Neuronal population coding of movement direction.* Science, vol. 233, no. 4771, pages 1416–1419, September 1986. 125

[Gerstner 2002] W. Gerstner and W. Kistler. Spiking neuron models. single neurons, populations, plasticity. Cambridge University Press, 2002. 30

[Ghanbari 2009] Y. Ghanbari, L. Spence and P. Papamichalis. *A graph-Laplacian-based feature extraction algorithm for neural spike sorting.* In 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 3-6 Sept. 2009, pages 3142–5, Piscataway, NJ, USA, 2009. 72

[Ghanbari 2010] Y. Ghanbari, P. Papamichalis and L. Spence. *Graph-spectrum-based neural spike features for stereotrodes and tetrodes.* In International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 598–601, March 2010. 77

[Gibson 2008] S. Gibson, J. W. Judy and D. Markovic. *Comparison of spike-sorting algorithms for future hardware implementation.* Annual Internationa Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, pages 5015–20, 2008. 71

[Gijbels 1999] I. Gijbels, P. Hall and Kneip A. *On the estimation of jump points in smooth curves.* Annals of the Institute of Statistical Mathematics, no. 2, pages 231–251, 1999. 33

[Gollisch 2008] T. Gollisch and M. Meister. *Rapid Neural Coding in the Retina with Relative Spike Latencies.* Science, vol. 319, no. 5866, pages 1108–1111, February 2008. xii, 126, 127, 128, 137

[Harris 2000] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase and G. Buzsáki. *Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements.* Journal of Neurophysiology, vol. 84, pages 401–414, Jul 2000. 60, 73

[Hastie 2003] T. Hastie, R. Tibshirani and J. H. Friedman. The Elements of Statistical Learning. Springer, 2003. 80

[Henze 2000] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris and G. Buzsaki. *Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo.* Journal of Neurophysiology, vol. 84, no. 1, pages 390–400, July 2000. 31, 52, 71, 75, 83, 85, 101, 116

[Hulata 2002] E. Hulata, R. Segev and E. Ben-Jacob. *A method for spike sorting and detection based on wavelet packets and Shannon's mutual information.* Journal of neuroscience methods, vol. 117, no. 1, pages 1–12, May 2002. 72

[Huxtable 2010] A. G Huxtable, J. D Zwicker, T. S Alvares, A. Ruangkittisakul, X. Fang, L. B. Hahn, E. Posse, K. Baker G. B.and Ballanyi and G. D. Funk. *Glia contribute to the purinergic modulation of inspiratory rhythm-generating networks.* Journal of Neuroscience, vol. 30, no. 11, pages 3947–58, 2010. 3

[Hyvärinen 1999] A. Hyvärinen. *Fast and Robust Fixed-Point Algorithms for Independent Component Analysis.* IEEE Transactions on Neural Networks, vol. 10, no. 3, pages 626–634, 1999. 105

[Hyvärinen 2000] A. Hyvärinen and E. Oja. *Independent component analysis: algorithms and applications.* Neural Networks, vol. 13, no. 4-5, pages 411–430, 06 2000. 97

[Ilan 2008] N. G. Ilan and H. J. Don. *Information theoretic bounds on neural prosthesis effectiveness: The importance of spike sorting.* In ICASSP, pages 5204–5207, 2008. 83, 97

[Iriarte 2003] J. Iriarte, E. Urrestarazu, M. Valencia, M. Alegre, A. Malanda, C. Viteri and J. Artieda. *Independent component analysis as a tool to eliminate artifacts in eeg: a quantitative study.* Neurophysiology, vol. 20, pages 249–257, 2003. 63

[Izhikevich 2006] E. M. Izhikevich. Dynamical systems in neuroscience: The geometry of excitability and bursting. The MIT Press, 2006. ix, 2, 9

[James 2003] C. J. James. *On the use of single-channels for sensing multi-source activity in biomedical signals.* 4th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Special Topic Conference on Information Technology, 2003. 63

[Jensen 2000] O. Jensen and J. E. Lisman. *Position reconstruction from an ensemble of hippocampal place cells: contribution of theta phase coding.* Journal of neurophysiology, vol. 83, no. 5, pages 2602–2609, May 2000. 139

[Johnston 1995] D. Johnston and S. M. Wu. Foundations of cellular neurophysiology. MIT press, 1995. ix, 9

[Jutten 2004] C. Jutten and J. Karhunen. *Advances in blind source separation (BSS) and independent component analysis (ICA) for nonlinear mixtures.* International Journal of Neural Systems, vol. 14, no. 5, pages 267–292, October 2004. 99

[Kandel 2000] E. R. Kandel, J. H. Schwartz and T. M. Jessell. Principles of neural science. McGraw-Hill Medical, 2000. ix, 2, 8, 124

[Kayser 2009] C. Kayser, M. A. Montemurro, N. K. Logothetis and S. Panzeri. *Spike-Phase Coding Boosts and Stabilizes Information Carried by Spatial and Temporal Spike Patterns.* Neuron, vol. 61, no. 4, pages 597–608, February 2009. xii, 130, 131, 132, 139

[Kim 2000] K. H. Kim and S. J. Kim. *Neural Spike Sorting Under Nearly 0-dB Signal-to-Noise Ratio Using Nonlinear Energy Operator and Artificial Neural-Network Classifier.* IEEE Transactions on Biomedical Engineering, vol. 47, no. 10, pages 1406–1411, Oct 2000. 33, 60

[Knutsen 2008] P. M. M. Knutsen and E. Ahissar. *Orthogonal coding of object location.* Trends in neurosciences, December 2008. 142

[Land 2002] B. Land and A. Spence. *Spike (action potential) detection and sorting.* Journal Club 25, http://www.nbb.cornell.edu/neurobio/land/PROJECTS/spikeSort/, 2002. 67

[Lavielle 2005] M. Lavielle. *Using penalized contrasts for change-point problem.* Signal Processing, vol. 85, pages 1501–1510, 2005. 33, 72

[Levi 2000] R. Levi and J. M. Camhi. *Wind direction coding in the cockroach escape response: winner does not take all.* Journal of Neuroscience, vol. 20, no. 10, pages 3814–21, 2000. 99, 125, 130

[Lewicki 1998] M. S. Lewicki. *A review of methods for spike sorting: the detection and classification of neural action potentials.* Network (Bristol, England), vol. 9, no. 4, pages 53–78, November 1998. 71, 72

[Liebenthal 1994] E. Liebenthal, O. Uhlmann and J. M. Camhi. *Critical parameters of the spike trains in a cell assembly: coding of turn direction by the giant interneurons of the cockroach.* Journal of Comparative Physiology, vol. 174, no. 3, pages 281–96, 1994. 125

[Lisman 2005] J. Lisman. *The Theta/Gamma Discrete Phase Code Occuring During the Hippocampal Phase Precession May be a More General Brain Coding Scheme.* Hippocampus, vol. 15, pages 913–922, 2005. 139

[Lisman 2008] J. Lisman and G. Buzsáki. *A neural coding scheme formed by the combined function of gamma and theta oscillations.* Schizophrenia bulletin, vol. 34, no. 5, pages 974–980, September 2008. xii, 140, 141

[Liu 2011] D.Y. Liu, O. Gibaru and W. Perruquetti. *Error analysis of Jacobi derivative estimations for noisy signals.* Numerical Algorithms, pages 53–83, 2011. 34

[Lloyd 1982] S. P. Lloyd. *Least squares quantization in PCM.* pages 129–137, 1982. 54

[Mallat 1999] S. Mallat. A wavelet tour of signal processing. Academic Press, 2nd edition, 1999. 33, 34

[Malmivuo 1995] J. Malmivuo and R. Plonsey. Bioelectromagnetism : Principles and applications of bioelectric and biomagnetic fields. Oxford University Press, 1 édition, July 1995. x, 63, 64

[Mamlouk 2005] A. Mamlouk, K. Menne, U. Hofmann and T. Martinetz. *Unsupervised spike sorting with ICA and its evaluation using GENESIS simulations.* Neurocomputing, vol. 65-66, pages 275–82, June 2005. 71

[Markram 2006] H. Markram. *The Blue Brain Project.* Nature Reviews Neuroscience, no. 2, pages 153–160, February 2006. 143

[Mathieson 2004] K. Mathieson, S. Kachiguine, C. Adams, W. Cunningham, D. Gunning, V. OŚhea, K.M. Smith, E.J. Chichilnisky, A.M. Litke, A. Sher and M. Rahman. *Large-area microelectrode arrays for recording of neural signals.* IEEE Transactions on Nuclear Science, vol. 51, no. 5, pages 2027–2031, 2004. 18

[Mboup 2008] M. Mboup. *A Volterra filter for neuronal spike detection.* Research report, INRIA, http://hal inria fr/inria-00347048/fr/, 2008. 85

[Mboup 2009a] M. Mboup. *Parameter estimation for signals described by differential equations.* Applicable Analysis, pages 29 – 52, 2009. 34, 41

[Mboup 2009b] M. Mboup, C. Join and M. Fliess. *Numerical differentiation with annihilators in noisy environment.* Numerical Algorithms, pages 439–467, 2009. 34

[Michael 2002] R. Michael, Deweese and M. Z. Anthony. *Binary coding in auditory cortex.* In In Advances in Neural Information Processing Systems. The MIT Press, 2002. 137

[Mikusiǹski 1987] J. Mikusiǹski and T.K. Boehme. Operational Calculus. PWN Varsovie & Oxford University Press, Oxford, 1987. 40

[Monderer 2002] R. S. Monderer, D. M. Harrison and S. R. Haut. *Neurofeedback and epilepsy.* Epilepsy and Behavior, vol. 3, pages 214–218, 2002. 63

[Montemurro 2007] M. A. Montemurro, S. Panzeri, M. Maravall, A. Alenda, M. R. Bale, M. Brambilla and R. S. Petersen. *Role of precise spike timing in coding of dynamic vibrissa stimuli in somatosensory thalamus.* Journal of neurophysiology, no. 4, pages 1871–1882, 2007. xii, 135, 136, 137

[Montemurro 2008] M. A. Montemurro, M. J. Rasch, Y. Murayama, N. K. Logothetis and S. Panzeri. *Phase-of-firing coding of natural visual stimuli in primary visual cortex.* Current biology, vol. 18, no. 5, pages 375–380, March 2008. xii, 130, 131, 133, 138, 139

[Nenadic 2005] Z. Nenadic and J. W. Burdick. *Spike detection using the continious wavelet transform.* IEEE Transactions in Biomededical Engineering, vol. 52, pages 74–87, 2005. 30, 34, 51, 56, 60, 67, 68

[Netter 2002] F. A. Netter, J. A. Craig and J. Perkins. Atlas of neuroanatomy and neurophysiology: Selections from the netter collection of medical illustrations special edition. Icon Custom Communications, 2002. 1

[Nolte 2002] J. Nolte. The human brain: An introduction to its functional anatomy. Mosby, 2002. ix, 4

[Novak 2009] D. Novak, J. Wild, T. Sieger and R. Jech. *Identifying number of neurons in extracellular recording.* In 4th Annual International Conference of the IEEE Engineering in Medicine and Biology Society on Neural Engineering, pages 742–745, Antalya, Turkey, 2009. 72

[Othmer 2000] S. Othmer and D. Kaiser. *Implementation of Virtual Reality in EEG Biofeedback.* Cyberpsychology and Behavior, vol. 3, pages 415–420, 2000. 63

[Panzeri 2001] S. Panzeri, R. S. Petersen, S. R. Schultz, M. Lebedev and M. E. Diamond. *The role of spike timing in the coding of stimulus location in rat somatosensory cortex.* Neuron, vol. 29, no. 3, pages 769–777, March 2001. 126

[Panzeri 2010] S. Panzeri, N. Brunel, N. K. Logothetis and C. Kayser. *Sensory neural codes using multiplexed temporal scales.* Trends in neurosciences, vol. 33, no. 3, pages 111–120, March 2010. 142

[Pouzat 2002] C. Pouzat, O. Mazor and G. Laurent. *Using noise signature to optimize spike-sorting and to assess neuronal classification quality.* Journal of Neuroscience Methods, vol. 122, no. 1, pages 43–57, Dec 2002. 72

[Pouzat 2005] C. Pouzat. *Methods and Models in Neurophysics.* Les Houches 2003 Summer School., pages 729–786, 2005. ix, 30

[Purves 2008] D. Purves. Neuroscience. Sinauer Associates, Inc., 2008. ix, 2, 11

[Quiroga 2004] R. Q. Quiroga, Z. Nadasdy and Y. B. Shaul. *Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering.* Neural Computing, vol. 16, no. 8, pages 1661–1687, August 2004. 31, 60, 72, 77, 87

[Reich 2001] D. S. Reich, F. Mechler and J. D. Victor. *Temporal Coding of Contrast in Primary Visual Cortex: When, What, and Why.* Journal of Neurophysiology, vol. 85, no. 3, pages 1039–1050, March 2001. 126

[Rezk 2008] S. Rezk, C. Join, S. El Asmi, M. Dogui and M. Hédi Bedoui. *Frequency Change-Point Detection in physiological Signals : an Algebraic Approach.* International Journal on Sciences and Techniques of Automatic control and computer engineering, no. 1, pages 456–468, 2008. 34

[Richmond 1997] B. J. Richmond, T. J. Gawne and G. X. Jin. *Neuronal codes: reading them and learning how their structure influences network organization.* Biosystems, vol. 40, no. 1-2, pages 149–157, 1997. 126, 142

[Rieke 1997] F. Rieke, D. Warland and W. Bialek. *Spikes: Exploring the Neural Code.* 1997. 29, 126

[Roa 2007] S. Roa, M. Bennewitz and S. Behnke. *Fundamental frequency estimation based on pitch-scaled harmonic filtering.* pages 397–400, 2007. 29

[Rogers 2007] C. Rogers, J. Harris, J. Principe and J. Sanchez. *A pulse-based feature extractor for spike sorting neural signals.* In 3rd Annual International Conference of the IEEE Neural Engineering, pages 490–493, Kohala Coast, HI, USA, 2007. 72

[Rose 2006] S. Rose. The future of the brain: The promise and perils of tomorrow's neuroscience. Oxford University Press, 2006. 2

[Schwartz 1998] L. Schwartz. Théorie des distributions. Hermann, 3rd edition, 1998. 37

[Sebald 2007] D. J. Sebald and A. Branner. *Automatic Spike Sorting For Real-time Applications.* In 3rd Annual International Conference of the IEEE Engineering in Neural Engineering, pages 670–674, 2007. 75

[Senior 2008] T. J. Senior, J. R. Huxter, K. Allen, J. O'Neill and J. Csicsvari. *Gamma Oscillatory Firing Reveals Distinct Populations of Pyramidal Cells in the CA1 Region of the Hippocampus.* Journal of Neuroscience, vol. 28, no. 9, pages 2274–2286, February 2008. xii, 140

[Shadlen 1998] M. N. Shadlen and W. T. Newsome. *The Variable Discharge of Cortical Neurons: Implications for Connectivity, Computation, and*

Information Coding. Journal of Neuroscience, vol. 18, no. 10, pages 3870–3896, May 1998. 124

[Shiraishi 2009] Y. Shiraishi, N. Katayama, T. Takahashi, A. Karashima and M. Nakao. *Multi-neuron action potentials recorded with tetrode are not instantaneous mixtures of single neuronal action potentials.* Annual International Conference of the IEEE Engineering in Medicine and Biology Society., pages 4019–4022, 2009. 96

[Simeral 2011] J. D. Simeral, S. P Kim, M. J. Black, J. P. Donoghue and L. R. Hochberg. *Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array.* Journal of Neural Engineering, vol. 8, no. 2, April 2011. ix, 24, 95

[Takahashi 2003a] S. Takahashi, Y. Anzai and Y. Sakurai. *A new approach to spike sorting for multi-neuronal activities recorded with a tetrode–how ICA can be practical.* Neuroscience research, vol. 46, no. 3, pages 265–272, July 2003. 71, 72, 96

[Takahashi 2003b] S. Takahashi, Y. Anzai and Y. Sakurai. *Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes.* Journal of neurophysiology, vol. 89, no. 4, pages 2245–2258, April 2003. 96

[Thomas 2000] E.A. Thomas. *A Parallel Algorithm for Simulation of Large Neural Networks.* Journal of Neuroscience Methods, 2000. 39

[Thulasidas 2004] M. Thulasidas, C. Guan, S. Ranganatha, J. K. Wu, X. Zhu and W. Xu. *Effect of Ocular Artifact Removal in Brain Computer Interface Accuracy.* 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 6, pages 4385–4388, 2004. 63

[Tiganj 2009] Z. Tiganj and M. Mboup. *Spike Detection and Sorting: Combining Algebraic Differentiations with ICA.* In Independent Component Analysis and Signal Separation, 8th International Conference, pages 475–482, Paraty, Brazil, 2009. 85

[Tiganj 2010] Z. Tiganj, M. Mboup, C. Pouzat and L. Belkoura. *An Algebraic Method for Eye Blink Artifacts Detection in Single Channel EEG Recordings.* In IFMBE Proceedings. 17th International Conference on Advances in Biomagnetism (Biomag2010), pages 175–178, 2010. 62

[Tiganj 2011a] Z. Tiganj and M. Mboup. *Deflation technique for neural spike sorting in multi-channel recordings.* IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2011. 97

[Tiganj 2011b] Z. Tiganj and M. Mboup. *A non-parametric method for automatic neural spikes clustering based on the non-uniform distribution of the data.* Journal of Neural Engineering., vol. 8, no. 6, page 066014, 2011. 74

[Vargas-Irwin 2007] C. Vargas-Irwin and J. P. Donoghue. *Automated spike sorting using density grid contour clustering and subtractive waveform decomposition.* Journal of neuroscience methods, vol. 164, no. 1, pages 1–18, August 2007. 75

[Vaswani 2007] N. Vaswani. *Additive change detection in nonlinear systems with unknown change parameters.* IEEE Transactions Signal Processing, no. 3, pages 859–872, 2007. 33

[Walls-Esquivel 2007] E. Walls-Esquivel, M. F. Vecchierini, C. Héberlé and F. Wallois. *Electroencephalography (eeg) recording techniques and artefact detection in early premature babies.* Clinical Neurophysiology, vol. 37, pages 299–309, 2007. 63

[Wood 2004] F. Wood, M. Fellows, J. Donoghue and M. Black. *Automatic spike sorting for neural decoding.* In 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Sep 1-5 2004, volume 26 I, pages 4009–4012, 2004. 72

[Xu 2007] G. Xu, J. Wang, Q. Zhang, S. Zhang and J. Zhu. *A spike detection method in EEG based on improved morphological filter.* Computers in biology and medicine, no. 11, pages 1647–1652, 2007. 33