Aus dem Institut für Signalverarbeitung und Prozessrechentechnik

der Medizinischen Universität zu Lübeck

Direktor: Prof. Dr.-Ing. Til Aach

Feature extraction from realistically simulated and recorded multisite neuronal signals

Diplomarbeit
im Rahmen des Informatik-Hauptstudiums

Vorgelegt von

Frau Kerstin M. L. Menne

Ausgegeben von

Herrn Prof. Dr.-Ing. Til Aach
Institut für Signalverarbeitung und Prozessrechentechnik

Betreut von

Herrn Dr. rer. nat. Ulrich G. Hofmann
Institut für Signalverarbeitung und Prozessrechentechnik

Lübeck, Januar 2002

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Lübeck, den 30.01.2002

Kerstin M. L. Menne

Stilles Reifen

Alles fügt sich und erfüllt sich,
musst es nur erwarten können
und dem Werden deines Glückes
Jahr und Felder reichlich gönnen.

Bis du eines Tages jenen
reifen Duft der Körner spürest
und dich aufmachst und die Ernte
in die tiefen Speicher führest.

Christian Morgenstern

Acknowledgments

Finishing the last chapter of this diploma thesis also means terminating an important chapter of my life, namely my years of study. Therefore I want to use the opportunity to thank both, people who supported me during this work and people who accompanied my studies. Some belong to both groups.

At first, I would like to thank my advisor Ulrich Hofmann for his contagious enthusiasm that brought me into the field of computational neuroscience and made me start with this project.

This project is rooted in a summer school about computational neuroscience in Trieste, Italy, and a stay at the Theoretical Neurobiology unit of the University of Antwerp, Belgium. I am grateful to Reinoud Maex from the University of Antwerp who advised me in Trieste and Antwerp. Without him my simulations would be kind of chaotic.

I am also indebted to Erik de Schutter for inviting my to his institute in Belgium.

Many thanks to Professor Til Aach for the friendly integration within the Institute for Signal Processing at the Medical University of Lübeck.

I express my gratitude to Andre Folkers and Thomas Malina for their technical support and the good collaboration. Thomas generated the 3D visualizations of my models. Andre provided the basis for my peak-to-peak thresholding algorithm and the highpass filter for the simulated data.

Without the following people my years of study would not have been what they were: Carsten Albrecht, Amir Madany Mamlouk, Jan Hendrik Sauselin and Birgit Schweda. Thank you.

I am grateful to Dirk for his understanding and his encouragement when needed.

And last but even most I want to thank my parents Josef and Susanne Menne for their support in each respect during all the years and for always believing in me.

Contents

List of Figures

Chapter 1

Warm-up

1.1 What you are expecting and why

You might wonder to read a computer scientist's diploma thesis dealing with computational neuroscience, while a lot of people are talking about the next processor generation, UMTS, data mining or parallelizations. But I think it is not all that surprising. Computer science is the science of information processing. Computational Neuroscience tries to explain the brain by means of theoretical models. And what else than information processing does your brain do? As you might know, information is always represented by some kind of signal. Be it the red traffic light that says “You have to stop now” or be it the stream of zeroes and ones encoding a highly complicated algorithm. Concerning information processing of the brain, at the moment we are happy if we can determine the signals, but we are far away from getting their message. For me, that is a big challenge. Of course, at the end of this work, you will not know, how the brain works. But hopefully you will have a better understanding of how one currently tries to solve this question.

This project consists of three main parts. The first one is a biologically realistic simulation of a tiny part of the brain. Models that mimick anatomical reality are frequently used in computational neuroscience. They are tuned according to known anatomy and electrophysiology and each bit of experimental data that is somehow available. Thus the experiment supports the modeler. But it is also the other way round. Once a model seems to be good, that means it shows some behaviour that can also be found by measurements in the brain, you can start to play with the model. By changing its parameters, it is possible to make predictions how the real brain might work under changed conditions. The experimenter gets a clue what he or she should look for next. But I must admit, I myself have some doubts concerning the reliability of such models. There are too many unknown parameters. I would not dare to claim that the brain absolutely certainly works like my model predicts. But probably no one does.

The process of setting-up my simulation is explained in Chapter 2. Before, in the second part of this introductory chapter, you will find the explanation of a lot of biological terms in order to facilitate your understanding of the simulation.

Chapter 3 illustrates some outcomes of the model. I do not intend to explain the brain according to my model. My motivation is of a different kind. I want to simulate data that is as realistic as possible. That means, compared to experimental data recorded in the brain, there should be almost no difference. In Chapter 3 I will show that the model approaches reality in some aspects.

The biological reality of the simulated data is important for the second and third part of this project, namely feature extraction from neural signals and the evaluation of spike detection methods. As you can imagine, huge amounts of data are produced by recording brain signals. It is almost impossible to analyze the data by hand. Efficient algorithms are needed. Spike detection algorithms are one type. They are supposed to filter interesting events, so-called *spikes*, out of the raw data as explained in Chapter 5. Different existent approaches to solve the problem of spike detection will be listed there as well.

Spike detection and feature extraction are highly correlated problems. Known features about spikes can influence spike detection methods and conversely, the more spikes are detected, the more features can be derived. Feature extraction can also be useful to answer the question “Where am I?” in the brain, since spikes originating from different *neurons* of different brain regions show diverse features. Typical spike features will be introduced on the basis of simulated and experimental data in Chapter 4, before switching to the problem of spike detection. And no worries: *Spikes* and *neurons* get explained in the next section as promised above.

But what exactly is the link to the simulated data? There are different approaches to solve the problem of spike detection, but no one knows which one is best and should be the gold standard, although spike detection is a crucial step in neurosignal data analysis. Each following step is based on this one. I will evaluate the quality of various spike detection methods on the basis of simulated and experimental data. As far as I know, no one else tested spike detection methods on simulated data like mine. Biologically realistic simulated data has the advantage that it is controllable and not too artificial at the same time. Experimental data is always subjected to interpretations concerning contained spiketrains and therefore does not allow such reliable comparisons between the returned spiketrains of the algorithms and manually detected spikes. The last chapter will announce the best spike detection method according to my evaluation.

1.2 Biological background

If we approach the *central nervous system* of vertebrates on a macroscopic level, it consists of *brain* and *spinal cord.* The *brain* in turn comprises a lot of different regions like *visual cortex*, *olfactory bulb* or *hippocampus.* The simulation is going to mimick a tiny part of hippocampus, see below. Regions can be distinguished by function and circuitry, but also by cell density and the occuring types of neurons. *Neurons* or *nerve cells* are the elementary units of the brain on a microscopic level. They can be categorized according to their cell shape and their pattern of connectivity. Figure 1.1 shows a so called *pyramidal cell* with its prominent pyramidal shaped cell body and its apical and basal dendrites.

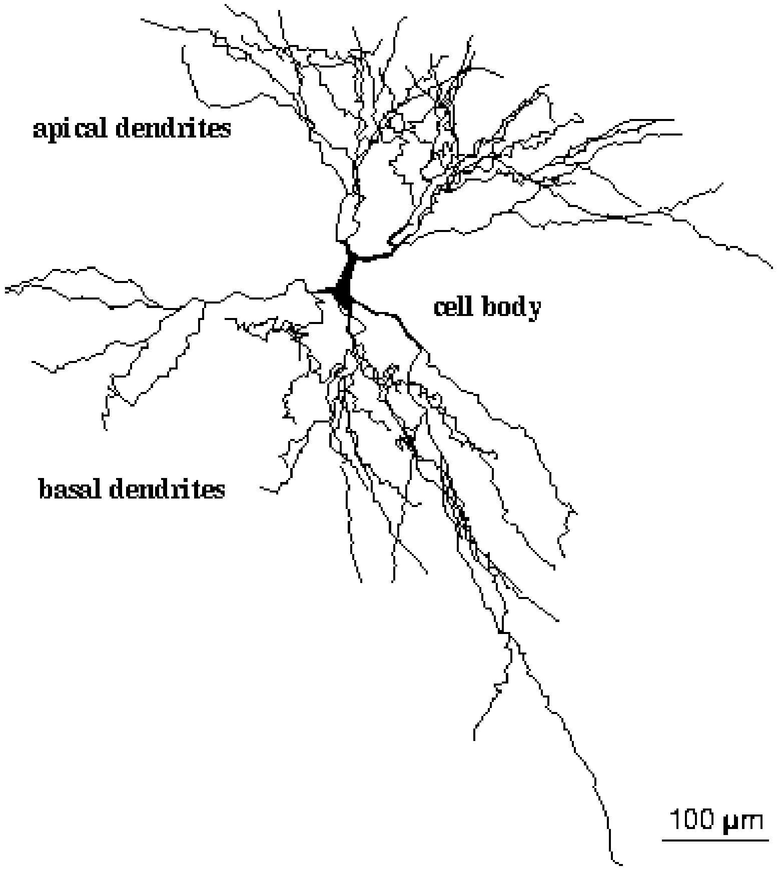

Figure 1.1: Projection of a real pyramidal cell. The picture is taken from the homepage of the Duke/Southampton Archive of Neuronal Morphology at http://www.cns.soton.ac.uk/~jchad/cellArchive/cellArchive.html. The inscription is added, the axon is not clearly visible in this picture.

The fundamental organization of each neuron is always the same: It consists of a

cell body, namely the *soma*, and branching processes, the *dendrites*. The *axon* or *nerve fiber* is a special branching process that is important for the propagation of information. Neurons with rather long axons contacting brain regions more far away are called *principal*, *relay* or *projection neurons*. *Interneuron* or *local neuron* is the name for neurons responsible for local communication [31].
How can you fancy the communication among nerve cells? Signaling in the brain is first of all of electrical nature. Like all cell types, neurons are surrounded by a membrane that separates intracellular from extracellular space. Intracellular recordings reveal, that the inside is more negatively charged compared to the outside. In case of a cell at rest, that means a cell without input, this potential difference is called *resting membrane potential*. Values range from -55 mV to -100 mV [7]. The resting membrane potential is attributed to the different ion concentrations of the aqueous solutions inside and outside the neuron. Important are high concentrations of sodium (Na^+) and chloride ions (Cl^-) extracellularly versus high concentrations of potassium ions (K^+) and impermeant anions intracellularly. Figure 1.2 illustrates this fact. The membrane is permeable for potassium ions, that means K^+ can traverse. Because of the concentration gradient, K^+ indeed leaves the cell. That leads to an accumulation of positive charges on the outside of the membrane. The arising electrical gradient influences the potassium ions in the opposite direction: It makes them stay inside the neuron. The two forces are in balance in the case of a cell at rest. By means of the Nernst equation (1.1)

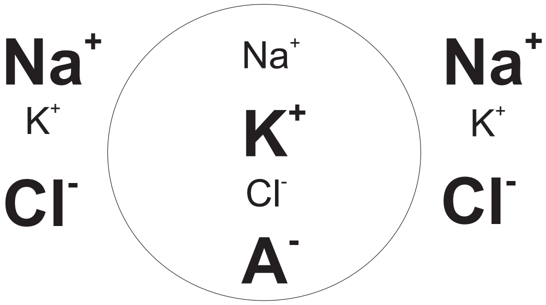

Figure 1.2: Different ion concentration inside and outside a neuron. Bold writing means high ion concentration, normal writing means low ion concentration. $\mathbf{A}^-$ denotes intracellular anions. The neuron is represented by the sphere. Compare for example with [7, page 68].

$$E_{K^+} = -58 \cdot \log \frac{K_i^+}{K_o^+} \tag{1.1}$$

one can calculate at which potential this is going to occur. K_i^+ denotes the

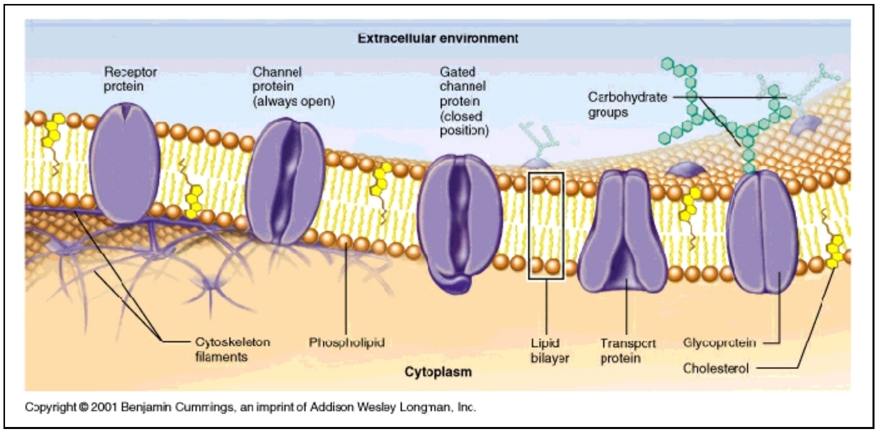
Extracellular environment
Receptor protein
Channel protein (always open)
Gated channel protein (closed position)
Carbohydrate groups
Cytoskeleton filaments
Phospholipid
Cytoplasm
Lipid bilayer
Transport protein
Glycoprotein
Cholesterol
Copyright © 2001 Benjamin Cummings, an imprint of Addison Wesley Longman, Inc.

Figure 1.3: Membrane architecture. The picture is taken from http://cellbio.utmb.edu/cellbio/membrane_intro.htm#Architecture.

potassium ion concentration inside the cell, K_o^+ that one outside. The resulting potential is called *potassium equilibrium potential*.

However, potassium equilibrium potential and resting membrane potential usually are not identical. This is due to the fact that the membrane is also slightly permeable for Na^+ and Cl^-. Chloride entering the cell supports the potential difference, sodium entering the cell weakens it by neutralizing negative charges, thus releasing K^+ from the outer surface of the membrane. These sodium and potassium currents are called *leakage currents*. Together with K^+ and Cl^- equilibrium potentials they are responsible for the resting membrane potential. Equilibrium potentials can be calculated for each ion type and add up to a total, summed potential. There is no net current across the membrane of a cell at rest.

Besides having a resting potential, cells at rest are not all that interesting. But they can get excited and exciting. Depending on the type of neuron and its location it is excited for example by visual or touch stimuli or by other neurons. But before explaining the information propagation in the brain, three sentences about membranes shall be said. As depicted in Figure 1.3, a membrane is a lipid bilayer with embedded proteins. Some of the proteins traversing the bilayer function as channels. They are selective for certain ions and their opening can be regulated by the current membrane potential and/or by the binding of ligands on the outer surface of the membrane.

Now let us assume that a neuron somehow got stimulated and that information now has to be propagated to different neurons via the branches of the axon. The stimulation leads to a depolarization at the beginning of the axon, the so called *axon hillock* next to the soma. *Depolarization* means that the potential difference between inside and outside becomes more positive. If and only if the depolarization reaches a certain threshold, a change of the sign of the potential takes place. This event is called *action potential*, *nerve impulse* or *spike*. Again, our already well-known ion currents are responsible for this event. The depolarization changes the permeability of sodium and potassium channels by altering the conformations of the respective channel proteins. Na^+ permeability increases first. Therefore sodium enters the cell according to the concentration gradient. The Na^+ influx depolarizes the cell even more and a potential reversal emerges. K^+ permeability rises delayed. The potassium outflux leads to another potential reversal and even to a *hyperpolarization*: The membrane potential becomes more negative than at rest for a short time. The general shape of an action potential together with a qualitative description of sodium and potassium channel opening times is illustrated in Figure 1.4. A spike usually has a duration of 1 to 2 ms. Within approximately the following millisecond, the axon cannot produce another action potential. This time is called *absolute refractory period*. There is also a *relative refractory period*. During this time, action potentials have markedly smaller amplitudes than the first one. The intensity of a stimulus is encoded in the frequency of action potentials. The absolute refractory period allows for a maximum of 500 spikes per second.

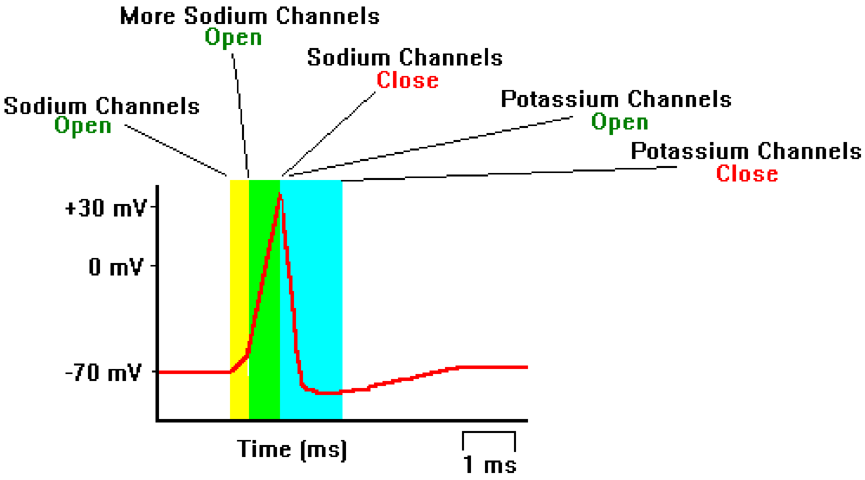

Figure 1.4: General shape of an action potential. The Neuroscience for Kids webpage about action potentials at http://faculty.washington.edu/chudler/ap.html provides this illustrative picture.

Once an action potential arose, it travels along the axon by ionic currents depolarizing neighbouring regions, starting the above described process anew (Figure 1.5).

But at the end of the axon, a different mechanism is required. Neurons do not

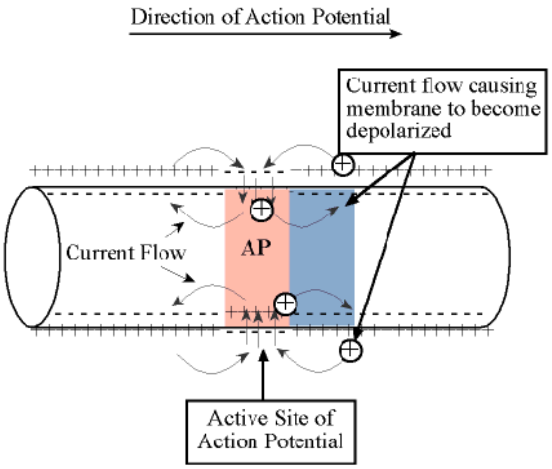
Direction of Action Potential
Current flow causing membrane to become depolarized
Current Flow
AP
Active Site of Action Potential

Figure 1.5: Propagation of an action potential along an axon. On the inside of the axon, positive charges move away from the active site of the action potential. On the outside of the axon, positive charges move towards the active site. More positve charges on the inside and less positve charges on the outside at the membrane region next to the active site cause a depolarization. This fact is explained and illustrated at http://bio.bio.rpi.edu/HB/Universal%20Files/Lectures/L40NerveAP/NEWAP/Proppg1.html.

immediately stick together, but there are tiny clefts of approximately 20 nm between them (Figure 1.6). These clefts are called *synaptic clefts*. The complex consisting of the axon terminal of a *presynaptic* neuron, the synaptic cleft and the target dendritic or somatic membrane region of a *postsynaptic* neuron is called *synapse*, see Figure 1.7. At synapses, electrical signaling is converted into chemical signaling. This chemical signaling turns information propagation in the brain into a one-way street. If an action potential reaches the end of the presynaptic axon, it gives rise to the release of certain chemical compounds, the *neurotransmitters*, from the *synaptic vesicles*. *ACh* (Acetylcholine), *glutamate* and *GABA* (γ-aminobutyric acid) for example are neurotransmitters that are found in a broad range of species. They pass the synaptic cleft in approximately 1 ms of time. On the surface of the postsynaptic membrane, there are receptor molecules where the neurotransmitters can bind. *AMPA* (α-amino-3-hydroxyl-5-methyl-4-isoxazolepropionic acid) and *NMDA* (N-methyl-D-aspartate) receptors are receptors for glutamate. *GABA_A* and *GABA_B* receptors are receptors for GABA. The binding of a neurotransmitter leads to conformation changes of postsynaptic ion channels.

Synapses are divided into *excitatory* and *inhibitory synapses*. At excitatory synapses, the released neurotransmitter, for example ACh or glutamate, depo-

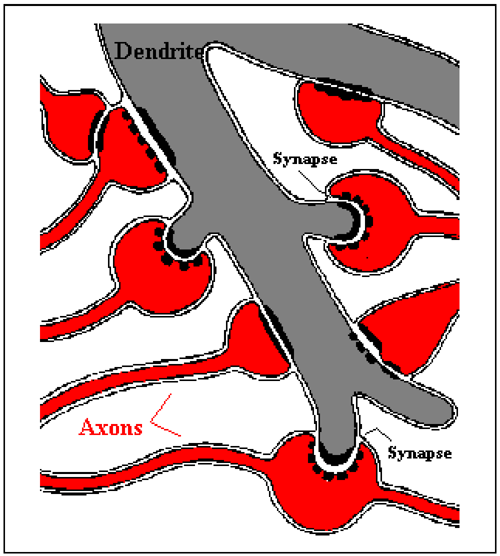

Figure 1.6: Axo-dendritic synapses: Axon terminals of presynaptic neurons contact the dendrites of another cell. From http://faculty.washington.edu/chudler/synapse.html.

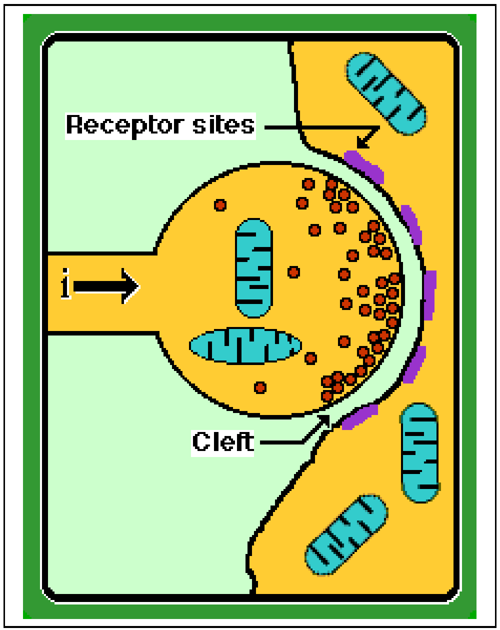

Figure 1.7: More detailed view of the synaptic cleft. The small circles represent synaptic vesicles containing neurotransmitters. The ovals are supposed to be mitochondria, the energy reservoirs of a cell. From http://gwis2.circ.gwu.edu/∼atkins/Neuroweb/synapse.html.

larizes the postsynaptic cell and thus facilitates the initiation of a new action potential. The binding at AMPA receptors influences Na^+ and K^+ channels and elicits fast *EPSPs* (excitatory postsynaptic potentials). The docking at NMDA receptors causes slower rising and decaying EPSPs. In the inhibitory case, neurotransmitters like GABA hyperpolarize the postsynaptic cell. Thus the initiation of a postsynaptic action potential is made more difficult. The opening of Cl^- channels is the consequence of occupied GABA_A receptors. The binding of GABA at GABA_B receptors leads via a different mechanism to an increase in K^+ conductance. Later on I will refer to cells that release neurotransmitters with excitatory effect as *excitatory neurons* and to cells that release inhibitory neurotransmitters as *inhibitory neurons*, respectively. One neuron usually releases either excitatory or inhibitory neurotransmitters, but gets both, inhibitory and excitatory input. In a process called *integration*, a cell adds up all these synaptic inputs onto its dendrites and its cell body from up to 10,000 different neurons in a nonlinear and complex way. If an action potential is resulting, it is in turn propagated in a divergent manner to hundreds of neurons.

At the end of this overview you shall get some information about the brain region that will be simulated: The hippocampus. The hippocampus owes its name to its seahorse-like shape."Hippo" is the Greek word for horse, and "kampos" is the Greek word for sea monster [31]. The hippocampus is part of the *limbic system*. The limbic system is a network of interconnected structures responsible for a broad range of behaviours like emotions, motivation and memory. The constituent parts of the limbic system are shown in Figure 1.8. The hippocampus attracts a lot of attention because of its role in learning and memory, especially long-term memory. As is evident from Figure 1.9, the hippocampus is divided up into different regions. CA1 and CA3 are mentioned. The distinction is made according to size and appearance of the neurons and different cell connections. The simulation will represent a part of CA3. The CA3 region gets input from other brain regions like for example the *dentate gyrus*. Pyramidal cells are the projection neurons of the hippocampus. They "project" their output onto e.g. CA1. In addition to pyramidal neurons there are also a lot of inhibitory interneurons. Both types of neurons will be modeled. Some more detailed information about the hippocampus is included in the next chapter, where parallels are drawn between simulation and anatomical reality.

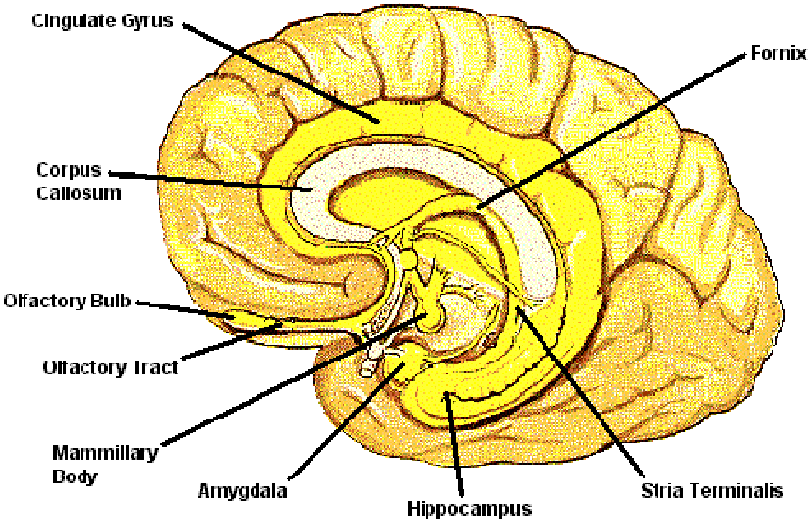

Figure 1.8: Localization of the hippocampus within the limbic system. From http://psych.athabascau.ca/html/Psych289/Biotutorials/8/part1.html.

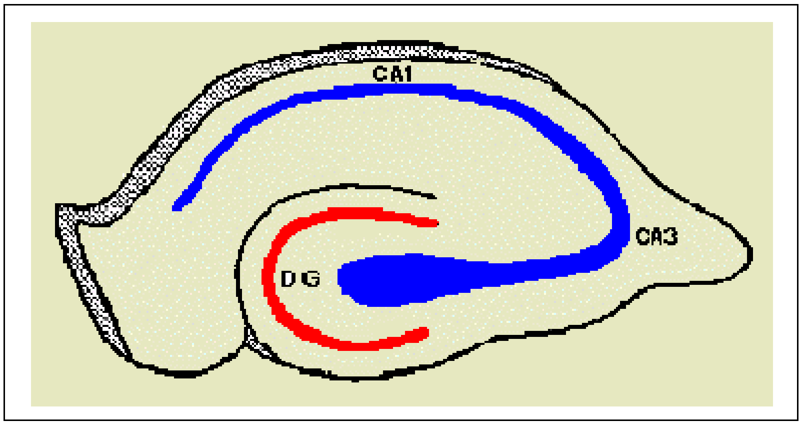

Figure 1.9: Sketchy drawing of hippocampus and dentate gyrus. From http://www.wjh.harvard.edu/~wyble/hippo.html.

Chapter 2

A biologically realistic network simulation . . .

My simulation is implemented in the freely available neural simulator GENESIS version 2.2 [5]. GENESIS is the acronym for "GEneral NEural SImulation System". GENESIS is a script language based on C that provides a relatively convenient way to set up a simulation, once one is familiar with it. But the name GENESIS, reminding of the bible, misleads: Working with GENESIS is not always heaven. At least GENESIS offers a lot of predefined objects and functions. I frequently will refer to them in this chapter when introducing the single cell models for pyramidal cells and inhibitory interneurons first and the cell connections establishing the network afterwards. But even the best simulation environment could not return good results without elaborated models.

2.1 Pyramidal cell model

You saw a real CA3 pyramidal cell in Figure 1.1 from the last chapter. The question arises, how this complicated looking structure can be modeled. I chose to represent it by a compartmental model. That means that the real cell is modeled by a bunch of small elements, the compartments that are supposed to be equipotential. The CA3 pyramidal cell model I employ is developed by Roger D. Traub [35] and was originally realized in FORTRAN. The model is based on experimental data and later on got implemented in GENESIS by Patricio Huerta (MIT) and Pulin Sampat (Brandeis University). Traub describes a model consisting of 69 compartments. The GENESIS implementation involves 66 compartments: 1 spherical soma compartment, 63 cylindrical dendritic compartments, 1 axon initial segment and 1 axon compartment only instead of 4 as suggested by Traub. Compared to the realization by Huerta and Sampat, I modified the 3D-coordinates of the compartments while keeping their sizes. The resulting compartmental model is shown in Figure 2.1. It is obvious, that the more compartments one employs to set up the model, the more realistic the simulation will be. But on the other hand, the more compartments one employs, the more ressources in terms of memory and execution time the simulation will need. 66 compartments seem to be a good trade-off. They allow for a realistic cell behaviour and especially for a realistic electrical field around the cell. The electrical field will be important later on. At the same time the model size enables

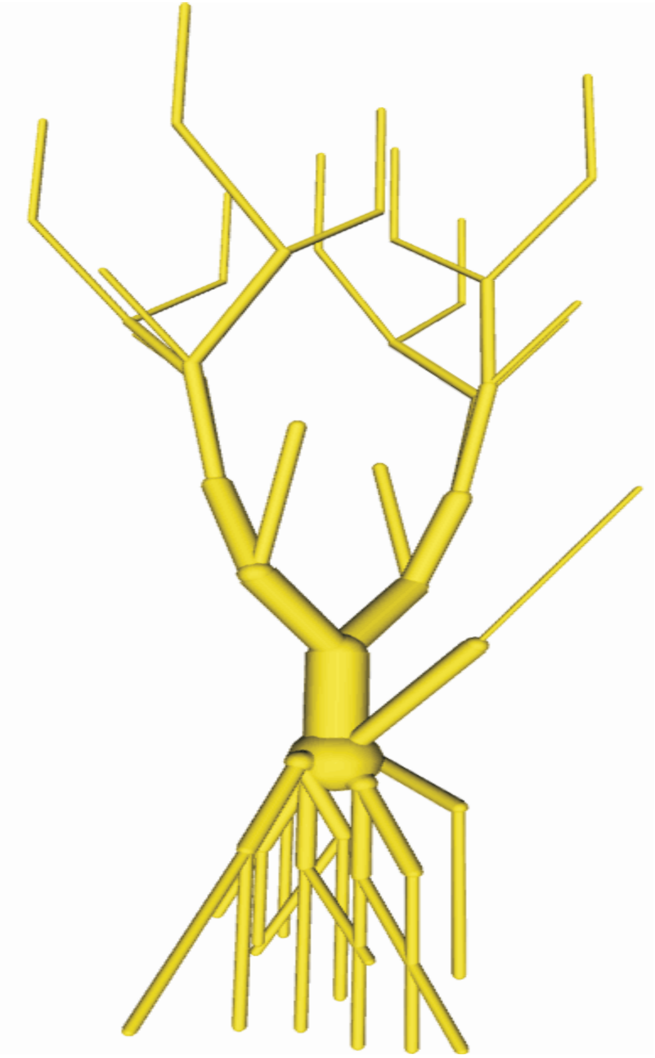

Figure 2.1: Compartmental pyramidal cell model. The process that originates in the spherical soma and branches to the right represents the axon.

the set-up of networks of cells without extending the execution time to weeks. Compartments can be implemented in GENESIS by the `compartment` object. Actually, a compartment is represented by an equivalent electrical circuit [5]. The equivalent electrical circuit is presented in Figure 2.2. The dashed lines separate adjacent compartments from each other. $\mathrm{V_m}$, $\mathrm{V_m}$' and $\mathrm{V_m}$" denote the transmembrane potentials of the respective compartments. The compartments are connected via the axial resistances $\mathrm{R_a}$ and $\mathrm{R_a}$'. $\mathrm{C_m}$ stands for the membrane capacitance. In the last chapter you read about $\mathrm{Na^+}$ and $\mathrm{K^+}$ leakage currents. Instead of a leakage conductance $\mathrm{G_{leak}}$ you find a membrane resistance $\mathrm{R_m}$ incorporated in the circuit, where $\mathrm{G_{leak}} = 1/\mathrm{R_m}$. $\mathrm{E_m}$ is the associated equilibrium potential represented by a battery. An ion current is translated into a conductance G_k and an equilibrium potential E_k. G_k can be thought of as summarizing the effects of a large number of ion channels selective for the respective ion. All parameters needed for the equivalent electrical circuit are given by Traub [35], except for $\mathrm{V_m}$, $\mathrm{V_m}$' and $\mathrm{V_m}$" that have to be calculated. The equivalent electrical circuit is very helpful in this respect. There exist well-known formula for the calculation of voltages and currents in electrical circuits. We now can calculate changes in membrane potential V_m for one compartment by means of a differential equation (2.1):

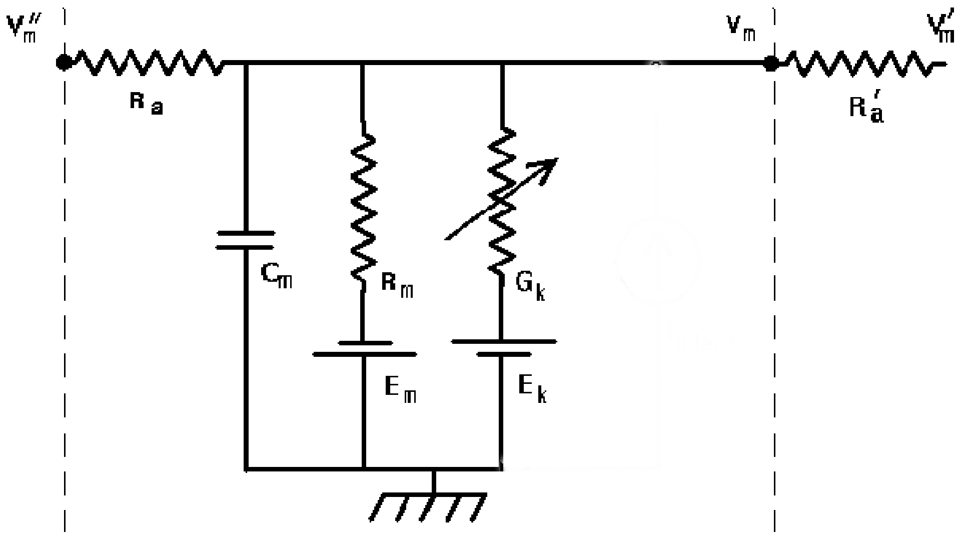

Figure 2.2: Equivalent electrical circuit of a neuronal compartment [5, modified].

$$C_m \frac{dV_m}{dt} = \frac{(E_m - V_m)}{R_m} + \sum_k ((E_k - V_m)G_k)$$

$$+\frac{(V'_m - V_m)}{R'_a} + \frac{(V''_m - V_m)}{R_a}. \tag{2.1}$$

If the `compartment` object is employed, the differential equation is automatically appropriately set up. Or to be more precise, a difference equation is set up that is numerically solved when GENESIS is launched (Section 2.3). A model neuron comprising lots of compartments can be defined in the so-called `cell descriptor file`. This file contains the diameters and the 3D-coordinates of the compartments relative to each other. Channels together with their conductances can be assigned to the compartments. The `readcell` function understands the format of the cell descriptor file and automatically initializes the system of differential equations or difference equations, respectively, representing the model. Logically, the set-up of each GENESIS simulation equals that of an UNIX directory tree (Figure 2.3). The names of possible cell-arrays constitute the first level below the root. The individual cells belonging to the different cell arrays can be found on the second level. On the third level there are different compartments, and channels are installed as children of the compartments. Channels and compartments exchange information about membrane potential and channel conductances that are needed for the calculations of the other object. The user does not have to

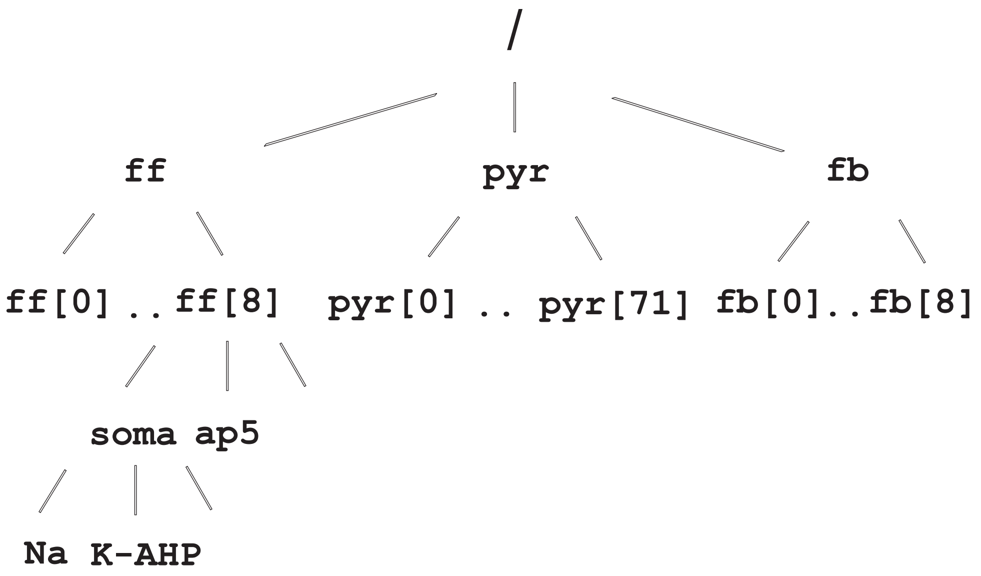

Figure 2.3: Logical organization of a GENESIS simulation. Of course, each of the cells has children compartments with channels. This graphic mirrors the set-up of the network to be established later on.

worry about the transfer into the difference equations. Objects, message passing and information hiding reveal that GENESIS follows an object oriented philosophy.

There are six types of active ion channels incorporated in the model: Channels for Na^+ and Ca^{2+} and four different conductances for K^+. *Active* means that the channels are voltage and/or ligand dependent. As outlined in the last chapter, Na^+ conductances play an important role during the depolarization of the membrane potential. Ca^{2+} currents have, among other things, the same function. The four different K^+ currents are delayed rectifier K_{DR}, slow afterhyperpolarization K_{AHP}, transient K_A and voltage and Ca^{2+}-dependent K_C current. All of the potassium currents have in common that they are responsible for repolarization and hyperpolarization. They are different in respect of how exactly they influcence for example the spike frequency during a burst. And even more obvious, they are different in respect of the channel kinetics. That means that the potassium channels show different pattern of activation and inactivation. Activation and inactivation can be transient or slow and can be dependent on different membrane potentials and additionally on ligands.

Channels are modeled by means of differential equations following the Hodgkin-Huxley (HH) Model. The HH model allows to calculate the ion currents that are contained as second addend on the right hand side of equation (2.1). The interested reader may be referred to the literature [5, page 36–38]. All parameters like conductance densities, equilibrium potentials and activation and inactivation rates that are needed for the channel equations are taken from Traub [35]. GENESIS provides a lot of different objects for the implementation of ion channels. In this case, `tabchannel` and `vdepchannel` objects are employed. The channels are assigned to axon, soma and dendritic compartments according to [35]. As Traub outlines, a high Na^+ conductance near the soma and a high Ca^{2+} conductance at the apical dendrites are crucial for the induction of different firing behaviours. In total, each pyramidal cell has 317 conductive channels, including synaptically acitvated channels that will be outlined in Section 2.3. Of course, a model consisting of 66 compartments and 317 conductive channels can only be an approximation of a real pyramidal cell. Real pyramidal cells have more complex branching dendritic trees and a bigger variety and amount of conductive channels. In the simulation, there is only one representative of a channel type per compartment accumulating the influence of hundreds of real channels. But none the less the model reflects some characteristics of real pyramidal cells: There are branching dendritic trees, there is a variety of ion channels incorporated and not uniformly distributed and channel kinetics, equilibrium potentials, conductances and resistances are oriented on experimental data.

But the set-up of a model only does not make it a good one, the model also has to behave reasonable. A biologically realistic network simulation requires first of all realistically working subunits, namely the individual cells. The CA3 pyramidal cell model as described above indeed exhibits some frequently observed behaviour. If different depolarizing injection currents are applied to single pyramidal cells experimentally, the cells change their firing pattern from low frequency bursting towards high frequency spiking, depending on the strength of depolarization [35]. This behaviour is replicated in the model by simulating depolarizing injection currents to the soma. A steady injection current of 0.23 nA elicits bursts. Spike doublets are the response to a 0.6 nA incjection current. And finally we get high frequency spikes by means of a 0.9 nA injection current. The three firing patterns are compared in Figure 2.4. All graphs throughout this work are prepared with the help of MATLAB (The MathWorks, Inc.).

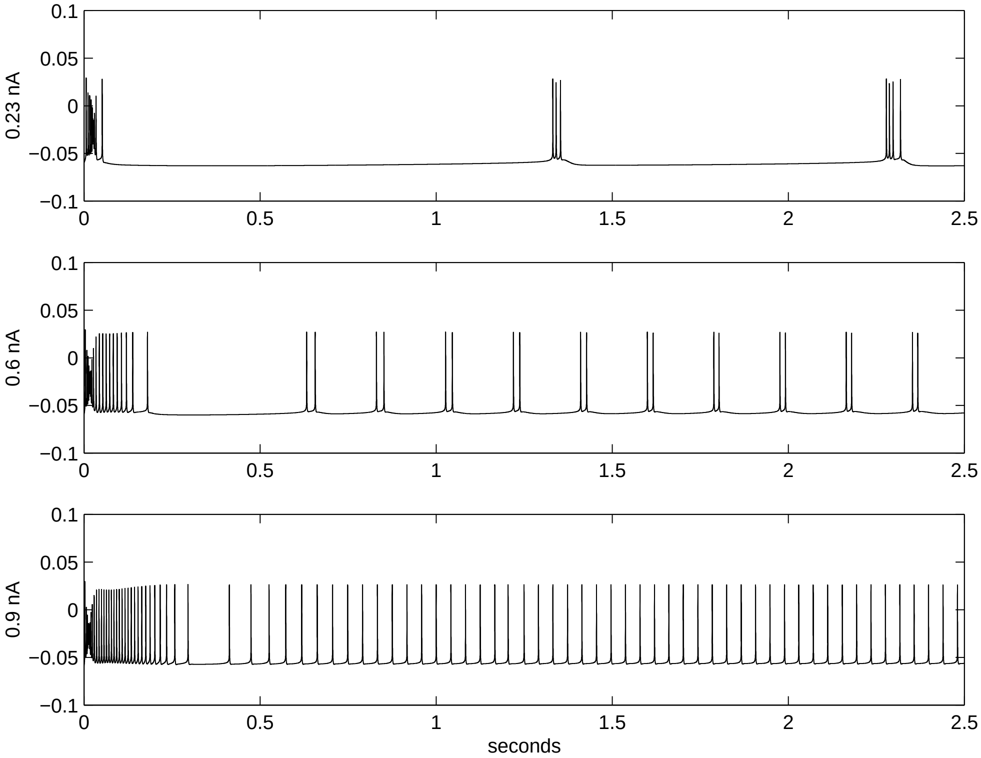

Figure 2.4: Transition from bursting to spiking behaviour of a single pyramidal cell depending on the strength of depolarization. The depolarizing currents are given on the left of each graph. Data is gained from the simulation of intracellular recordings in the soma of the model neuron. The signals are given in volt.

2.2 Interneuron model

You already learned in Section 1.2 that the hippocampus is not only made up of pyramidal cells but of interneurons with inhibitory effects as well. Before studying the interplay between these two groups of neurons, the interneuron model shall be introduced first.

The interneuron model is designed by Traub [36] as well. A GENESIS implementation of this compartmental model was recently made available online (http://www.genesis-sim.org/BABEL/) by Elliot D. Menschik [20]. Nevertheless I translated the model into GENESIS independently from that implementation. This seems a straight forward task, but nonetheless there are several subtle differences between my implementation and that by Menschik.

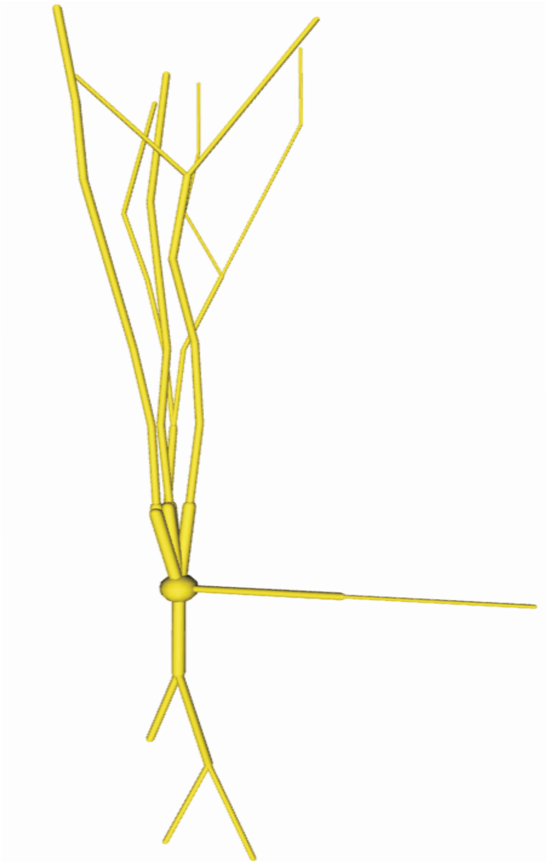

Figure 2.5: Compartmental interneuron model. The process that originates in the spherical soma and branches to the right represents the axon.

Traub assembles the interneuron model out of 51 compartments. Menschik retains this assembly, whereas I employ only 1 axon compartment instead of 4 again.

In addition to the axon compartment there are 47 others: 1 spherical soma compartment, 45 cylindrical dendritic compartments of different parameters and 1 axon initial segement. The sizes of the compartments are predifined [36], but except for a picture of the model there are no 3D-coordinates given. Therefore the two different GENESIS implementations show a varying cell shape. Mine is illustrated in Figure 2.5. The study of the influence of the actual morphology is work beyond this thesis.

Pyramidal cell model and interneuron model do not differ a lot. The two cell models show a different cell geometry and additionally can be distinguished according to the conductance density distributions. Na and K_{DR} conductance densities are higher in the dendrites of the interneuron model than in the dendrites of the pyramidal cell model. Ca and K_C conductance densities are lower. The high dendritic concentration of Na channels is important to get electrically excitable dendrites. The interneuron model comprises 214 conductive channels, including the synaptically activated. The channel kinetics, membrane resistance, membrane capacitance and equilibrium potentials are the same as in the case of the pyramidal cell model with two exceptions. First, the equilibrium potential for potassium currents equals -85 mV instead of -75 mV. And second, the somatic submembrane Ca^{2+} concentration is modelled to decay much faster than in the case of the pyramidal cell. Menschik applies the faster decay in each compartment. He also does not employ different activation and inactivation functions for Na and K_{DR} conductances for the axon compartments compared to the dendritic compartments. Traub makes this distinction between axonal and dendritic compartments in the case of the pyramidal cell model and I do the same for the interneuron model.

At the end of this section the single cell behaviour of the interneuron model shall be investigated. Many hippocampal interneurons are characterized by a repetitive firing of short-latency action potientials with short afterhyperpolarizations [36]. Frequencies over 100 Hz are not unusual. Traub [36] demonstrates the reaction of the model neuron implemented in FORTRAN to different steady somatic injection currents. Almost identical results can be achieved with the GENESIS implementation and are presented in Figure 2.6. Figure 2.7 illustrates the dependence of the number of spikes per second on the injection current. It is interesting, that the interneurons are intrinsically active: Even without injection currents or any alternative input source they elicit 28 spikes per second.

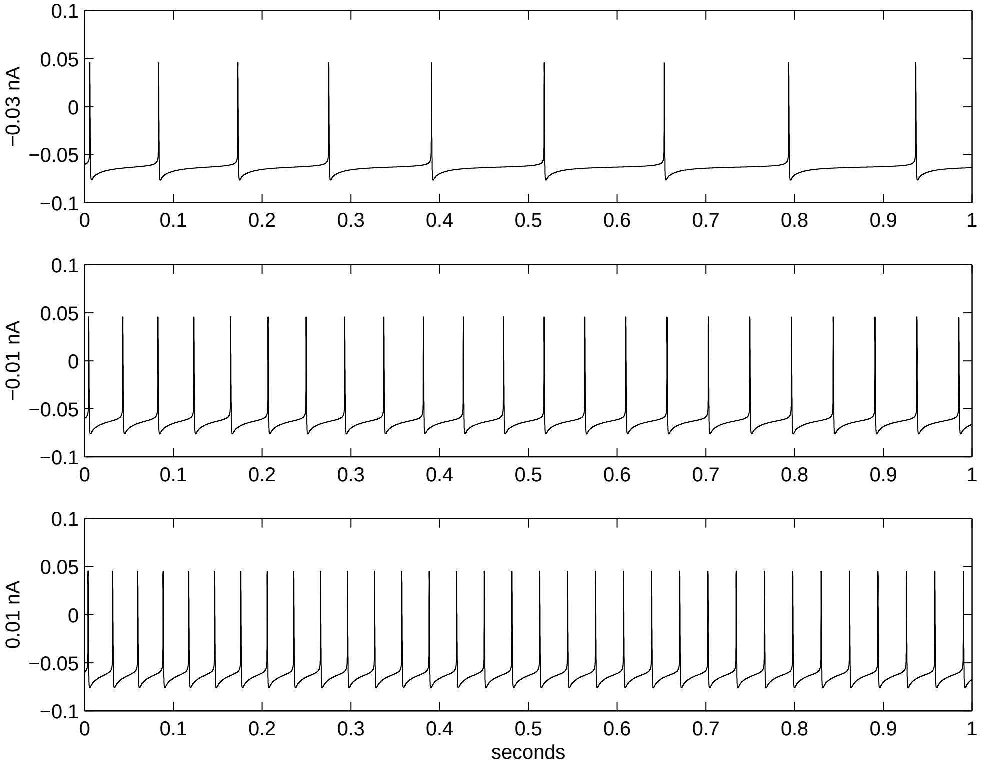

Figure 2.6: Repetitive firing of action potentials of a single interneuron as response to different somatic holding currents. The injection currents are given on the left of each graph. Data is gained from the simulation of intracellular recordings in the soma of the model neuron. The signals are given in volt.

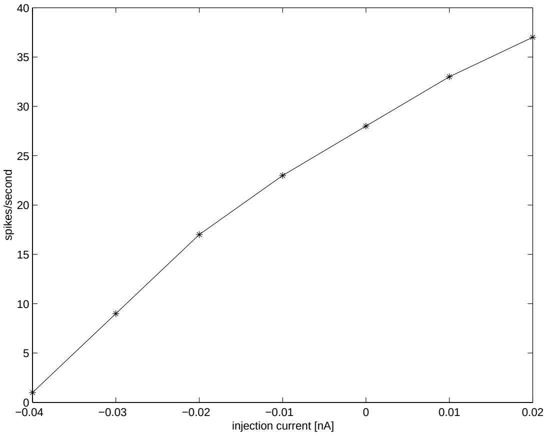

Figure 2.7: Dependence of interneuron spike frequency on injection current.

2.3 Network circuitry

Two individual cell models and even hundreds of copies of them do not constitue a network unless they are connected with each other and thus can interact, i.e. the really complicated part of each network simulation starts now. The first step in the set-up of the network is indeed to make copies of the single cell models. That is done with the help of the `create_array` GENESIS command, that has to be given the name of the original cell, the desired number of copies and their spacing. My network has a modest size. There are 72 pyramidal cells arranged in a 6 times 12 twodimensional array and 18 interspersed interneurons. Actually there are two interneuron populations, each comprising 9 individual cells in a 3 times 3 array. Each interneuron is derived from the same model (compare Section 2.2). The only difference is to be found in the input and output connections, see below. The pyramidal cells are rather densely packed with a cell-to-cell distance of 10 μm +/- 3 μm in x- and y-direction. Interneurons of one population are 20 +/- 3 μm in x-direction and 40 +/- 3 μm in y-direction apart from each other. They alternate in y-direction as can be seen in Figure 2.8. The z-coordinates are randomized in a broader range of -50 to +50 μm in order to avoid having

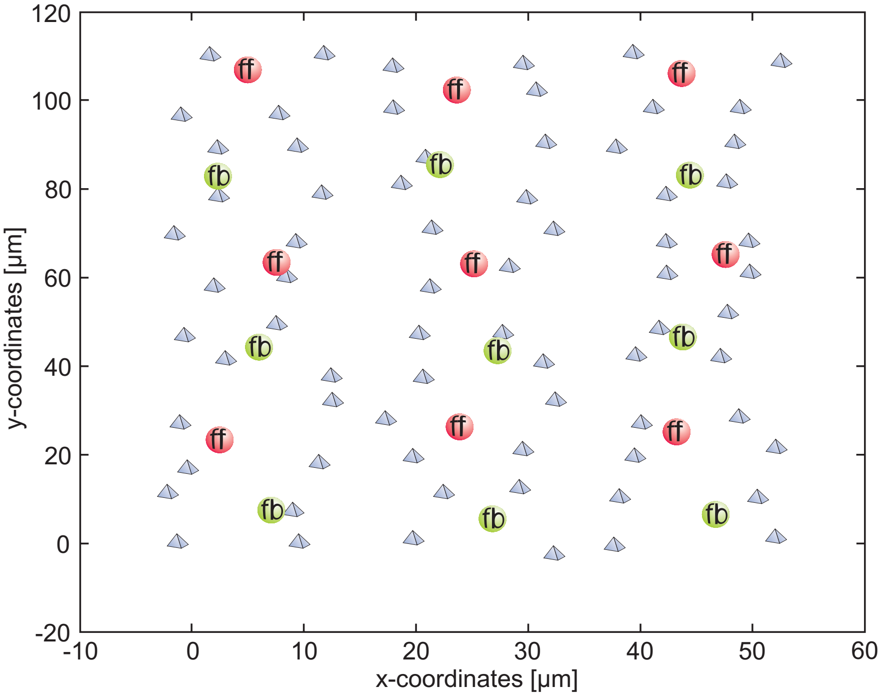

Figure 2.8: Arrangement of cells within the network. The coordinates of the somas serve as reference. Numbering of the constituent cells of a cell array starts in the lower left corner.

all cells lie in one plane. Figure 2.9 visualizes somatic z-coordinates by different grey-scale colors. Obviously it is not natural to have cell clones. Real cells do not all have the same size and the same branching dendritic trees. But the design of 90 individual cell models is work under progress beyond the scope of this thesis. The steps I took to get more variability is to randomly rotate the cells 0 to 180 degrees clockwise about the z-axis and to randomize the resting membrane potential between -65 and -60 mV. A rotation about x- or y-axis would not make a lot of sense, since it is known that the dendritic trees of the pyramidal cells are orientated perpendicularly to the pyramidal cell layer of CA3 [31, page 421].

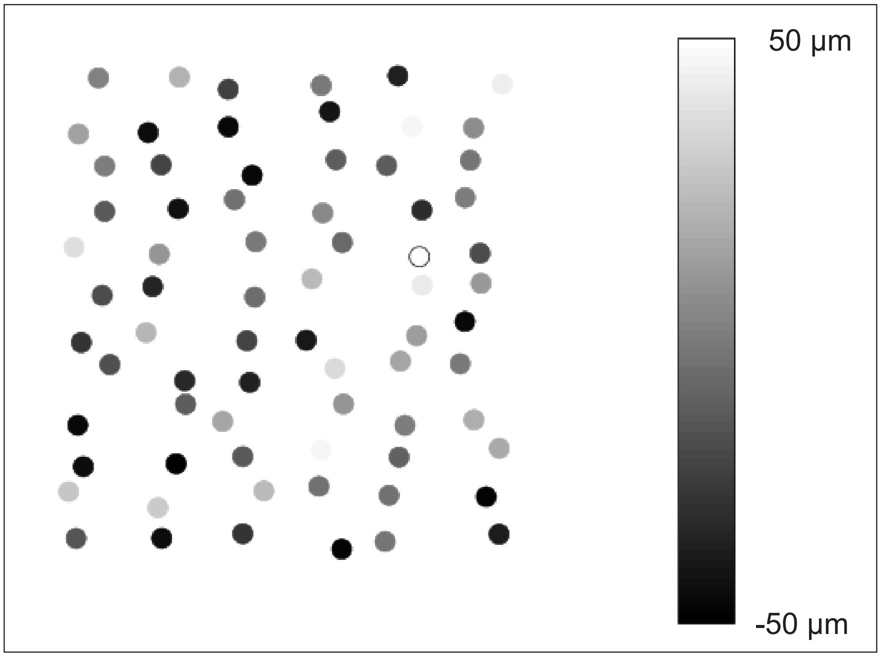

Figure 2.9: Different somatic z-coordinates.

The ratio between interneurons and pyramidal cells equals 1:4 in my network simulation. Traub and Menschik also investigated network simulations based on the pyramidal cell and interneuron models. The interneuron-to-pyramidal cell ratios that can be found in their works are 1:5 [34], 1:8 [33] and 1:0.9 [20], respectively. I did not follow the network simulation described by Traub in [33], because this simulation is intended to mimick a part of the CA1 region of the hippocampus by means of the CA3 models, which discredits it for my use. In addition the CA1 network comprises a total of 3456 cells. That is not needed for my purposes. As I already outlined in the introductory chapter, my intention is to gain realistically looking data, especially from simulated extracellular recordings. Menschik [20] and Traub [34] have more special interests. Traub wants to study the influence of a certain chemical on the firing behaviour and Menschik even looks for a model of Alzheimer's disease. Menschik's network comprises 136 neurons [20] and that of Traub 152 [34]. I am interested in keeping the simulation smaller and as easy as possible. Menschik's network shows a rather complicated circuitry, because mutual inhibitory connections among interneurons are incorporated. It is known that these connections exist anatomically, but they are not at all costs needed in a simulation, unless one wants to elicit a spiking behaviour in a certain frequency range. My network circuitry is oriented towards that of Traub [34] which also works fine without mutual inhibitory connections. For the sake of elegance and simplicity, I incorporated 2 different interneuron populations only instead of 5 to represent the variety of inhibitory input to the pyramidal cells. Additionally, I have less sites of inhibitory input onto the pyramidal cells as well. Each of Traub's interneurons is derived from the same model [36], and is modeled as feedback interneuron. *Feedback* interneurons get their excitatory input from pyramidal cells and make inhibitory contacts back to them. Only one of my interneuron populations is modeled as feedback. The feedback interneurons are excited by pyramidal cells at their apical dendrites. Soma and proximal dendritic compartments carrying GABA_A receptors are in turn the target sites of the pyramidal cell for the inhibitory input of the feedback interneurons. *Proximal*, contrary to *distal*, denotes dendritic compartments close to the soma. Feedforward interneurons contact the pyramidal cell at the distal apical dendrites, where GABA_B receptors are located. *Feedforward* means that the interneuron feeds inhibition forward onto the pyramidal cell after being excited by some source which is not a pyramidal cell, but probably also excites pyramidal cells. In reality, *Schaffer collaterals* or *perforant path fibers* from the *enthorinal cortex* can be the input source for feedforward interneurons. In my simulation, the feedforward interneurons receive random excitatory input via AMPA channels at the middle portion of their apical dendrites. For the realization of random afferent excitatory input see below.

Menschik's pyramidal cells get inhibited via GABA_A receptors only in the perisomatic region and at the axon initial segment. There is no inhibition at apical dendritic sites. But as a fact, CA3 pyramidal cells get inhibitory inputs from so-called *bistratified cells* onto their apical and basal dendrites. Inhibition in the perisomatic region originates from *basket cells*. *Chandelier cells* contact the axon initial segment [31].

Before continuing with the other cell contacts, you should remember your probably just acquired knowledge about synapses, the means of communication among cells (Section 1.2). When you read about cell contacts in this section, synaptic contacts are always meant implicitely. The most important feature about synapses is, that the presynaptic neuron releases a neurotransmitter that docks

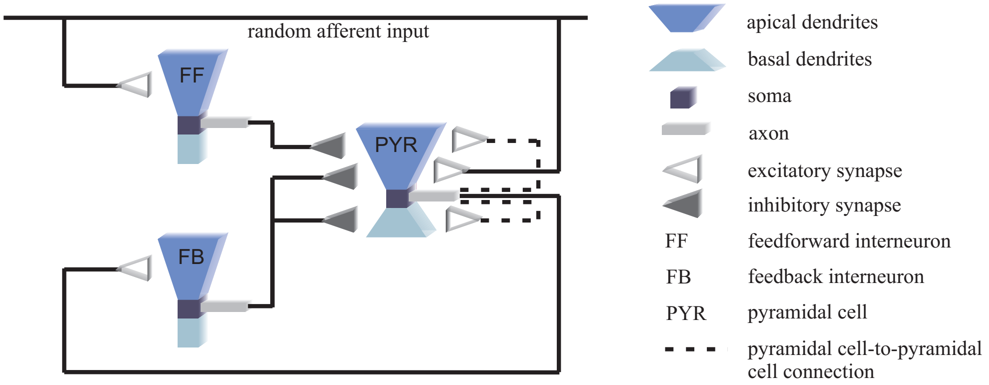

Figure 2.10: Network circuitry. Compared to the network ciruitry published in [18], [19], pyramidal cells do not get inhibitory inputs on their basal dendrites from feedforward interneurons anymore and in turn do not excite feedback interneurons at basal dendrites.

onto postsynaptic receptors, thus inducing a depolarization (excitatory effect) or a hyperpolarization (inhibitory effect) in the postsynaptic neuron. Glutamate is the most prominent neurotransmitter at excitatory synapses. AMPA and NMDA are the corresponding receptors. GABA_A and GABA_B receptors are the docking stations for GABA, the inhibitory neurotransmitter. I employed AMPA, NMDA, GABA_A and GABA_B receptor models according to [9], [21], [32] that were originally implemented in GENESIS for a granule cell model developed at the Theoretical Neurobiology unit of the University of Antwerp. In GENESIS, the presynaptic site of a synapse is represented by `spikegen` objects, the synaptically activated channels by `synchan` elements.

Now let us continue with the cell connections. Perforant path fibers are not exclusively an input source for interneurons, but for pyramidal cells as well. The latter are in addition heavily excited by *mossy fibers*, the axons of the *granule cells* situated in the *dentate gyrus*. Accordingly, pyramidal cells get two types of random input in the simulation. First, AMPA and NMDA channels react on spikes that are pretended to occur 5 times per second. This is achieved by setting the `frequency` field of the respective `synchan` objects to 5. Second, random afferent input is simulated in the perisomatic region, where in realitiy mossy fibers contact pyramidal cells. For this purpose, I installed AMPA and NMDA channels in the perisomatic region, additionally to those suggested by Traub [34]. Traub excites his cells by current pulses.

The random input can be realized in GENESIS by the `randomspike` object. It sends spike-messages to the compartments of cells it is connected to at a rate chosen by the user and always taking into account the set refractory period. Instead of incorporating one `randomspike` element only, I created two 4 times 4 twodimensional arrays of `randomspike` objects. These arrays can be thought of as lying next to each other above the pyramidal cell-interneuron-array. The seeds for the random number generators for the arrays are different. One of these afferent input arrays excites pyramidal cells 0 to 35, the other one excites pyramidal cells 36 to 71, taking into account that neighbouring pyramidal cells usually get the same excitatory input. The firing rates of the different `randomspike` objects are randomized and vary from 35 to 45 spikes per second.

However, there are even more so-called *associational contacts* among the pyramidal cells. These are *recurrent* excitatory contacts, meaning that pyramidal cells excite each other. Recurrent contacts are made in the distal apical and basal dendrites, according [34]. Projections from CA3 to CA1 are not modeled. My complete network circuitry can be recapitulated at a glance from Figure 2.10.

Another difference to the circuitries of Menschik and Traub is, that both simulate hyperpolarizing current injections into the soma of interneurons to suppress spontaneous firing (compare Figure 2.7). I do without these injection currents, because they can hardly be found in the brain. I think it would be interesting to investigate, if Menschik and Traub could achieve the same results as they present in their papers when incorporating the simplifications I introduced.

The positive feedback among the pyramidal cells and the intrinsic properties of the pyramidal cells make CA3 unstable in a sense that it is prone to epileptiform activity: Spontaneous and synchronous rhythmic firing. Therefore inhibition of the pyramidal cells is very crucial. To find the right excitation/inhibition ratio that prevents epileptic bursts while at the same time does not prevent firing behaviour at all, is the most demanding task in setting up the network simulation. Because of the above mentioned differences between my simulation and those by Menschik and Traub, I could hardly orient myself on information they deliver, if at all, about the number of synaptic connections and their strengthes. There are basically two ways to influence the effect of cellular connections in a simulation. Either one installs a lot of connections with less weight or one installs less connections but increases their weight. By synaptic "strength" or "weight" I mean the conductance resulting from activation of the receptor. A lot of connections yield for a variety of input patterns and less synchrony, but are more expensive in terms of simulation time because of the numerous occurences of messages. The contrary is true for less connections but higher weight to achieve approximately the same synaptic effect that has to cope with the reduced number of inputs compared to 10,000 synapses per neuron in biological systems [31]. There were approximately 500 parameters in total, but I changed "only" (Appendix A) 40 of them representing number of connections and synaptic conductances. Instead of completely investigating this huge parameterspace (which may become necessary for reproducing real biology), I chose the parameters arbitrarily to get a network behaviour where the pyramidal cells and interneurons are ideally not

synchronously spiking or bursting. Shifted times of activation are desirable for the recording of extracellular potentials as will be seen in Chapter 5.
The establishment of cellular connections itself is not difficult in GENESIS. The `volumeconnect` function connects groups of elements by adding SPIKE messages between source and destination objects that can be specified in wildcarded lists. Refractory periods are installed on the presynaptic site. Since the network is rather small, the connections are potentially all-to-all. "Potentially" because they are established randomly with a probability p calculated as follows:

$$\begin{aligned} p &= \frac{n\_conn \cdot n\_post}{n\_post \cdot n\_pre \cdot n\_comps} \\ &= \frac{n\_conn}{n\_pre \cdot n\_comps} \end{aligned} \tag{2.2}$$

The parameter *n_conn* denotes the number of desired connections to be installed between source and target population, for example between feedforward interneurons and pyramidal cells. *n_post* and *n_pre* contain the amount of postsynaptic and presynaptic cells, respectively. In the case of feedforward interneurons and pyramidal cells we have $n\_pre = 9$ and $n\_post = 72$. *n_comps* equals the number of compartments of the postsynaptic neuron where connections shall be made. In the case of feedforward interneuron to pyramidal cell connections, 16 compartments of the pyramidal cell are possibly involved. Thus if we want to have e.g. 10 synaptic contacts from feedforward interneurons onto each pyramidal cell, we pass $p = \frac{10}{9 \cdot 16} = 0.07$ to the `volumeconnect` function. Uniformly distributed synaptic weights are implemented with the help of the `volumeweight` function. The `volumedelay` function allows for taking into account axonal delays in the propagation of action potentials. Thus different cell distances have their effects. Connections between `randomspike` objects and individual cells are realized in much the same way as cell-to-cell connections. Further details of the implementation can be found on the accompanying CD.
One more aspect of the simulation shall be discussed in this section: The integration step size. As you know, what hides behind the simulation scripts is a system of coupled first-order ordinary differential equations that has to be solved by integration. To be more precise, the system of differential equations is translated into a system of difference equations that are numerically solved at discrete time intervals. Therefore two things have to be chosen: An integration method and an appropriate step size that ideally allows for a great accuracy and at the same time for a fast execution. GENESIS offers a variety of implicit and explicit integration methods [5]. Explicit methods shall be explained on the basis of the Forward Euler Method, where the integration of some $y(t)$ is approximated by

$$y(t + \Delta t) = y(t) + f(t)\Delta t. \tag{2.3}$$

That is equivalent to taking into account only the first derivative in a Taylor series expansion, if

$$\frac{dy}{dt} = f(t). \tag{2.4}$$

In case of equation 2.1 describing the potential change for one compartment:

$$\begin{aligned} f(t) &= \frac{(E_m - V_m(t))}{R_m} + \sum_k((E_k - V_m(t))G_k) \\ &\quad + \frac{(V'_m(t) - V_m(t))}{R'_a} + \frac{(V''_m(t) - V_m(t))}{R_a}. \end{aligned} \tag{2.5}$$

(Note, that $V'_m(t)$ and $V''_m(t)$ do not denote derivatives, Figure 2.2.) New values are calculated explicitly on the basis of functions of the old values. Explicit integration methods require small time steps in order to avoid numerical instabilities. I employed the more complicated Exponential Euler Method (the interested reader may be referred to [5]) with a time step of 2.5e-6 as Huerta and Sampat had done for the pyramidal cell model. Soon I switched to the implicit Crank-Nicholson method. This change of integration method made some extra-work since so-called `hsolve` objects had to be incorporated in the simulation. The acronym "hsolve" is derived from Hines solver. Hines [12] developed a method for the implicit solution of equations describing branching tree-like structures without closed loops [5]. The approach makes use of the fact, that the matrix representing the coupled sets of equations is a sparse tridiagonal matrix. It can be solved by Gaussian elimination. The tridiagonal form arises from the form of equation 2.1, see also 2.5: The right hand side of the equation involves the unknown membrane potential V_m and that of the two adjacent compartments (V'_m, V''_m) only.

It is true that implicite methods are more complex, but at the same time they are more stable for bigger time steps. Employing the Crank-Nicholson method, I could apply a time step of 2.5e-5. Implicit methods are characterized by the fact that new values are calculated on the basis of functions of the new values that are yet to be determined. In the case of the Crank-Nicholson method that means

$$y(t + \Delta t) = y(t) + (f(t) + f(t + \Delta t))\Delta t/2. \tag{2.6}$$

The most thrilling question in this context is: How much time does a simulation run need? The simulation of 1 second of network behaviour, including the extracellular recordings that will be introduced in the next section, still occupies a SGI with a 800 MHz PentiumIII Xeon processor for more than 1.5 hours, even without graphical outputs. Data is stored in ascii-files and is visualized with the help of Matlab. Figure 2.11 shows exemplary intracellular potentials of two pyramidal cells and interneurons. Compared to the firing behaviour of the interneuron single cell model as presented in Section 2.2, individual interneurons incorporated in the network show a similar pattern of activity. Both types of interneurons are spiking regularly at frequencies of 10 to 30 Hz (fb) and 70 to 80 Hz (ff). Pyramidal cell firing shows, as wanted, an interesting mixture of bursts, spike doublets and single spikes.

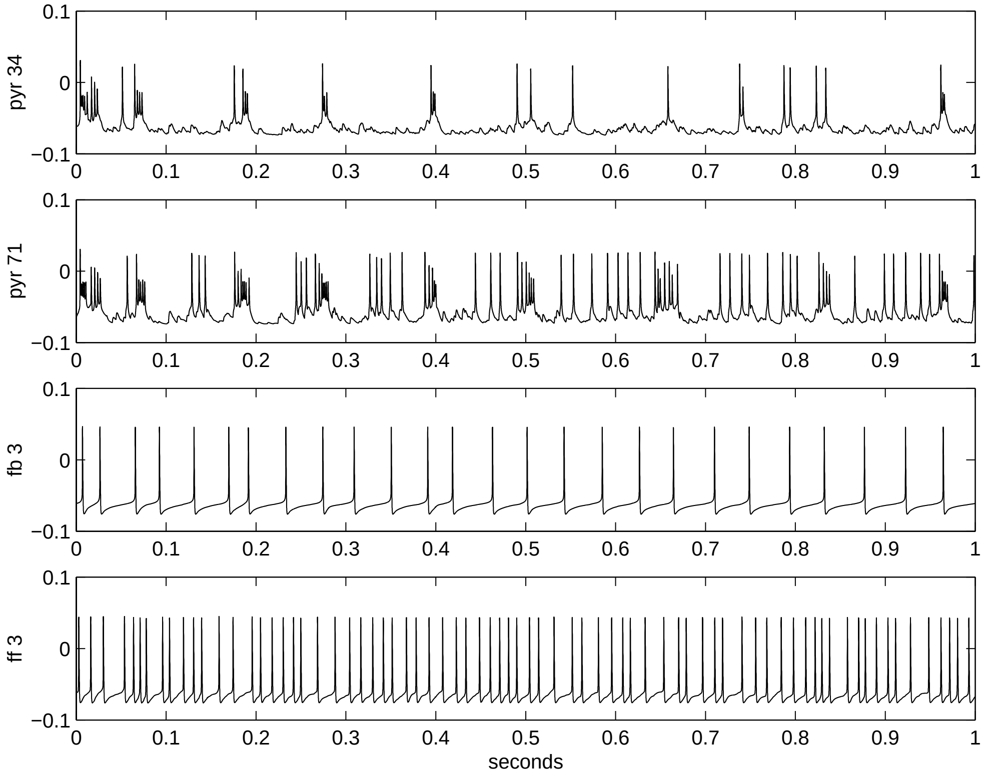

Figure 2.11: Exemplary output of two pyramidal cells and interneurons. The names of the cells are given on the left of each graph. The somatic membrane potentials are given in volt.

2.4 Extracellular recordings

Extracellular recordings are performed in experiments by inserting a single-site or multiple-site recording probe more or less blindly into the brain. Single-site recording probes have one recording electrode whereas multiple-site recording probes can have up to 128 channels, thus considerably augmenting the amount of data gained from one experiment. You can fancy a probe sticking in the extracellular space among neurons and recording extracellular potential changes induced by single cells or groups of cells. The activity of groups of cells is represented by low frequency *field potentials*. What makes extracellular recordings interesting as substitute for intracellular recordings is the fact, that membrane potential changes of many individual neurons are reflected in the extracellular data. A multi-site recording probe can acquire signals containing spike trains from hundreds of cells [38] in the vicinity of the probe without injuring any of them. Chances and problems resulting from this fact will be discussed in more detail in Chapter 5. At this point, the simulation of extracellular recordings within the previously described network shall be explained.

A single-site recording probe gets simulated in GENESIS with the help of the `efield` object. The `efield` object is the implementation of an equation by Nunez [25] for the calculation of extracellular field potentials:

$$F = \frac{1}{4 \cdot \pi \cdot s} \sum_{i=1}^{n} \frac{I_i}{r_i}. \tag{2.7}$$

Transmembrane currents I_i of n compartments are added with respect to their distance r_i from the "electrode". The parameter s denotes extracellular conductivity. If you recall Chapter 1.2, spikes travel along the axon by means of ion currents. In general, if the membrane potential is not uniformly distributed, there will be current flow within the membrane. The current paths have to be completed extracellularly. Regions where current exits the cell are called *sources*, regions where current enters the cell are called *sinks* [13]. The current field can be associated with an extracellular potential field, since the resistance, regarded as being ohmic, of the extracellular fluid is unequal zero. The current is proportional to the potential gradient at any point. That makes the envolvement of the compartment-electrode-distance necessary. Instead of a resistance, there is the finite conductance s contained in equation (2.7). In GENESIS, the fraction $\frac{1}{4 \cdot \pi \cdot s}$ is replaced by an adjustable scale factor that allows for a reasonable range of values. No capacitance and a homogeneous resistivity are assumed for the extracellular fluid.

A linear arrangement of several `efield` objects makes up a multi-site recording probe which can be placed at an arbitrary but well-defined position within the network and "records" the desired data. The individual recording sites of one probe have a distance of 12.5 μm. The spacing equals 30 to 50 μm in the case of a real probe.

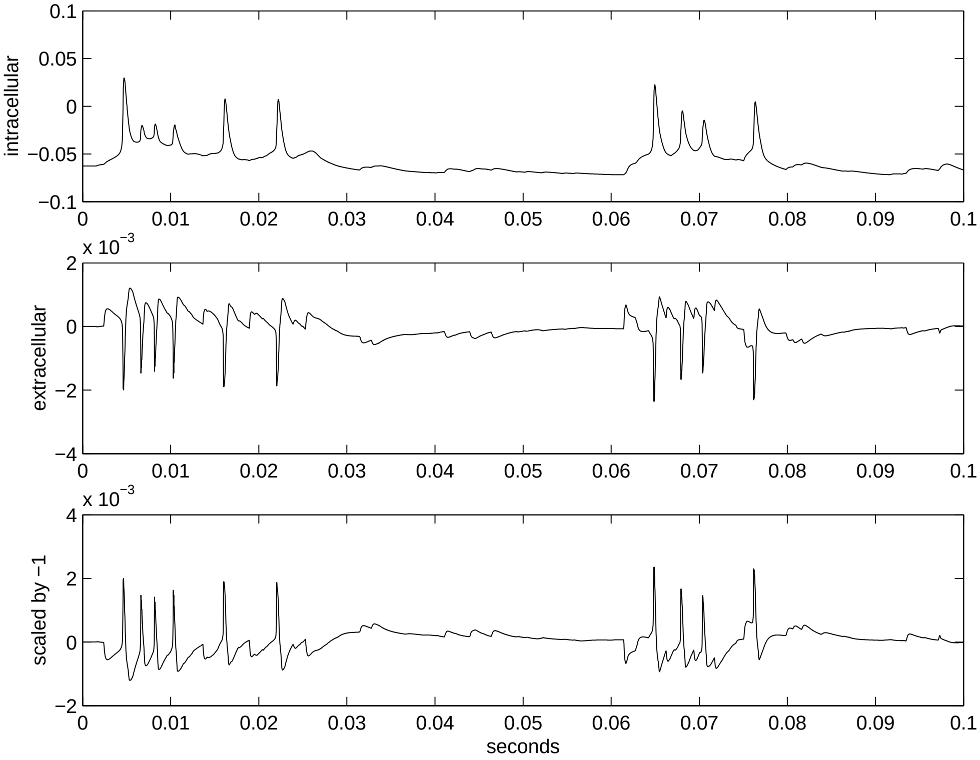

Figure 2.12: Comparison of intracellular versus extracellular spikes. Intracellular spike shapes (top) are gained from saving the membrane potential V_m of the soma of pyramidal cell 40 in an ascii-file. Extracellular spike shapes (middle) are the result of the calculations of an efield object exclusively recording from pyramidal cell 40 at a distance of approximately 4 μm. Such a recording probe can be simulated by passing messages from pyramidal cell 40 only. I scaled the extracellular data by -1 for later use for the test of spike sorting algorithms (bottom). Potentials are given in volt.

Above it is mentioned, that extracellular recording data contains spike trains, that means consecutive spikes of individual neurons in the vicinity of the electrode. However, these are not identical to the spikes you would record intracellularly with an electrode sticking directly in the nerve cell. Extracellular spikes are more or less a scaled and shifted mirror image of the intracellular spikes. A depolarization of a membrane region caused by Na^+ influx is extracellularly manifested as negative potential. A hyperpolarization caused by outward K^+ currents leads to a measurable positive potential extracellulary. The above explanations are of course simplified. In order to support an intuitive understanding, simulated intracellular and extracellular spikes are compared against each other in Figure

2.12.
In Figure 2.13, two screenshots of a movie taken from http://lnc.usc.edu/∼holt/papers/thesis/j4_potential_movie.html provide a more global view on the extracellular potential field during depolarization and hyperpolarization.

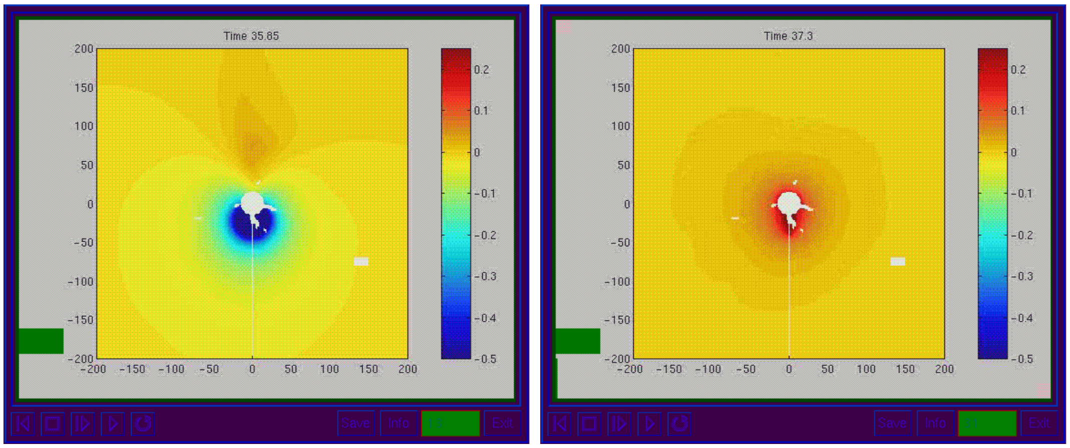

Figure 2.13: Extracellular potential fields during depolarization and hyperpolarization. The computed potential in one plane passing through the cell body of a pyramdal cell model [17] is shown. White areas correspond to intracellular regions. An added artificial cell axon is descending from the cell body. Distance scales are in μm. As described in the text, during depolarizations (left) you see extracellular negativities and during hyperpolarizations (right) you have extracellular positivities.

If you have a look at these pictures, you can imagine that a rather complicated multi-compartmental model is advantageous for the generation of realistic field potentials. All compartments of a multi-compartmental model have their own transmembrane currents inducing a complex extracellular potential field. The cell geometry is also important with respect to possible cancellations of fields resulting from linear and algebraic addition of the action currents of a number of neurons. For illustrative purposes, Figure 2.14 compares simulated extracellular recordings taking into account all compartments of the models against a recording where only the somatic transmembrane currents are regarded. The latter is kind of artificial data that can only be simulated by respective message passing but not experimentally recorded. The trace on top presents an example of the desired multi-unit simulated extracellular recording data. *Multi-unit* means, that the data originates from many neurons or units in the vicinity of the recording point, instead of one neuron only (Figure 2.12). Obviously, the soma makes the main contribution to the extracellularly recorded signal. If you have a closer look at the traces, however, you realize that some low amplitude spikes contained in the bottom trace are missing in the trace on top, probably due to cancellations resulting from currents of other model compartments.

Finally, Figure 2.15 compares intracellular potential data against the extracellulary recorded you already know from Figure 2.14, top. The extracellular signal (bottom), already scaled by -1, contains low frequency field potentials at times of synchronous activity of neurons in the vicinity (top and middle). Spikes are directly reflected in the extracellular signal. More interesting features of the simulated extracellular recordings will be given in the next chapter.

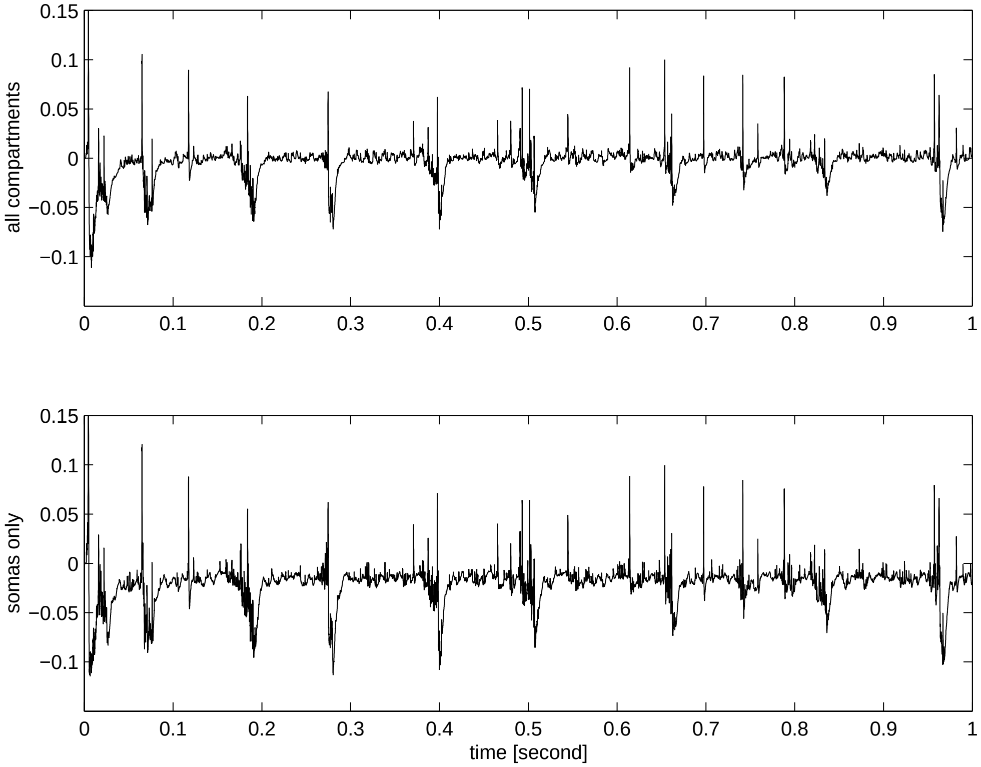

Figure 2.14: Cancellations of field potentials. The electrode is approximately 4 μm away from pyramidal cell 40. The signals are given in volt.

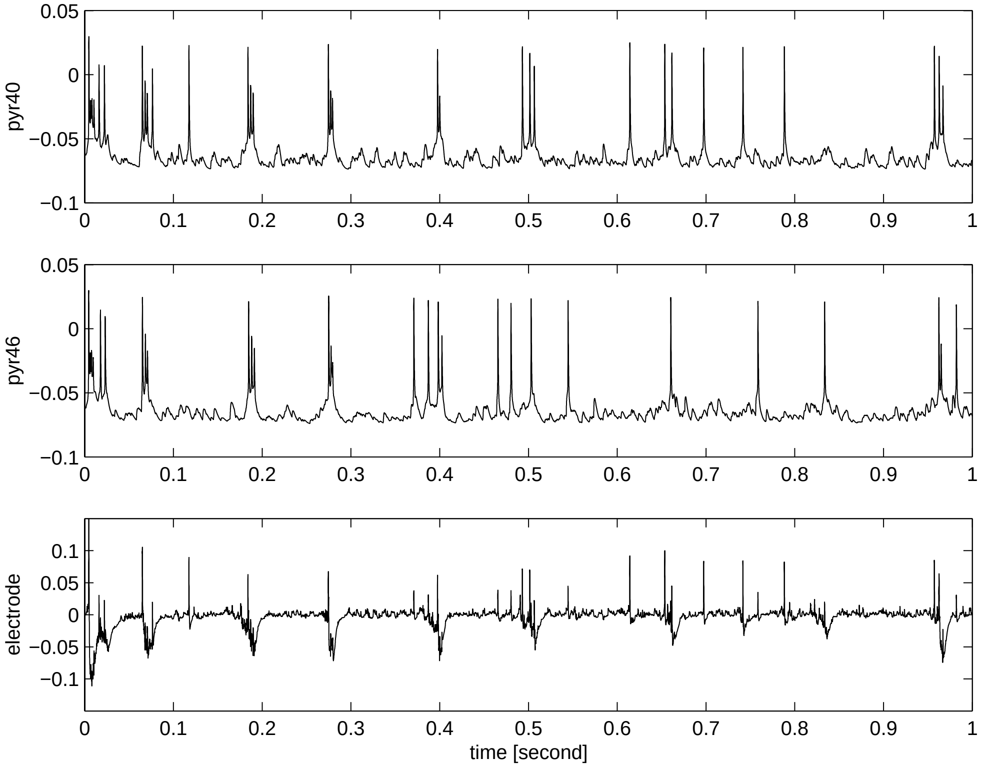

Figure 2.15: Multi-unit extracellular potential data. The electrode is approximately 4 μm away from pyramidal cell 40, at the height of the soma. The signals are given in volt.

Chapter 3

. . . and what we learn from it

The ultimate goal of my simulation is to produce biologically realistic extracellular potential data. The process of simulating data of this kind was described in the last chapter. In this chapter we will have a closer look at this data and answer two questions. First, how has the data to be prepared for the test of spike detection algorithms. And second, which features do the simulated data show compared to experimental data? But before, the network behaviour will be investigated and some remarks about the simulation in general will be given.

3.1 Network behaviour

At the end of Section 2.3 the reader was presented some exemplary intracellular recordings of pyramidal cells and feedback and feedforward interneurons. But how does the overall network behaviour look like? In the same Section 2.3, I outlined that my motivation was to produce a network behaviour, where cells ideally do not spike synchronously, at least not all the time. Is this claim satisfied? The change of activity across the pyramidal cell array can be visualized by means of a movie. The pyramidal cells are represented by a 3D-model and, in accordance with the simulation set-up, arranged in a 6 times 12 array with the help of the Visualization Toolkit VTK. Different intracellular membrane potentials are encoded by different colours. The somatic membrane potential determines the colour of the cell. Two screenshots from the movie are presented in Figures 3.1 and 3.2. Figure 3.1 shows a network state where all pyramidal cells burst synchronously. In Figure 3.2 only single pyramidal cells are active (dark colours). Much more impressive is the movie itself on the accompanying CD.
It is interesting that the pyramidal cells elicit both, simultaneous bursts as well as isolated single spikes. This fact is also illustrated in Figure 3.3(a), depicting only the times of spike initiation of some pyramidal cells and interneurons. I attribute

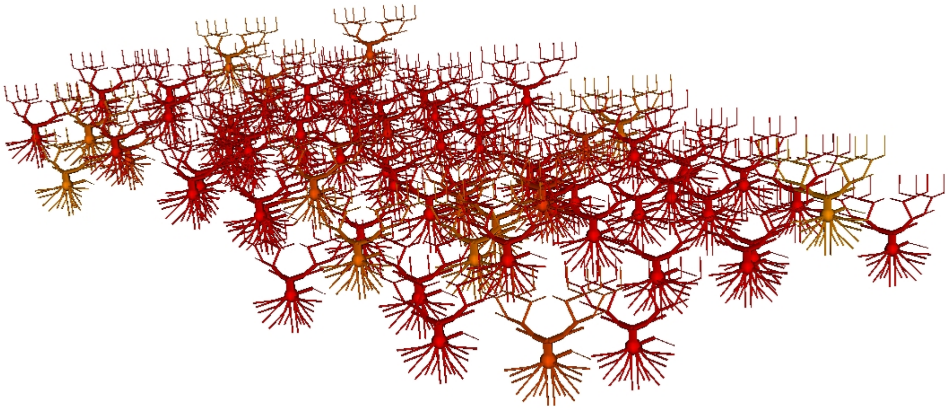

Figure 3.1: Synchronous bursts. Despite the deceiving look, pyramidal cells do not lie in one plane, recall Section 2.3.

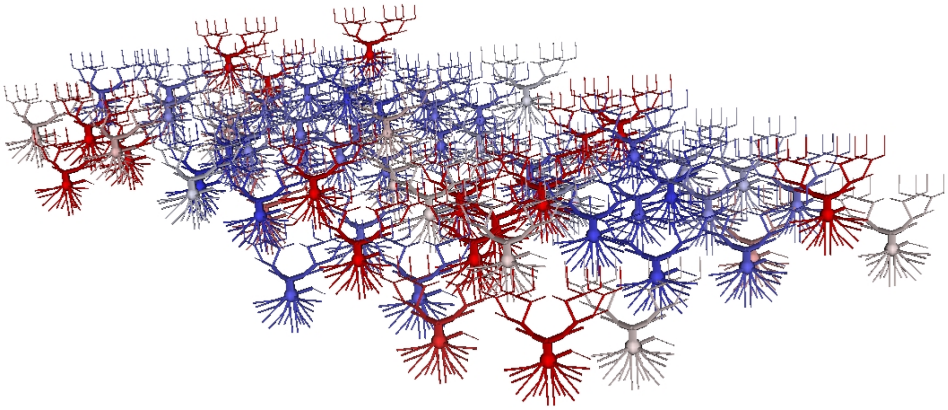

Figure 3.2: Isolated spikes.

the synchronous bursts to the strong recurrent excitatory connections among the pyramidal cells. If you have a closer look at the synchronous bursts, however, you discover, that the bursts of individual pyramidal cells are not absolutely congruent, but the constituent spikes occur shifted, sometimes overlapping in time (Figure 3.3(b)), thus paying tribute to different random inputs and different resting potentials.

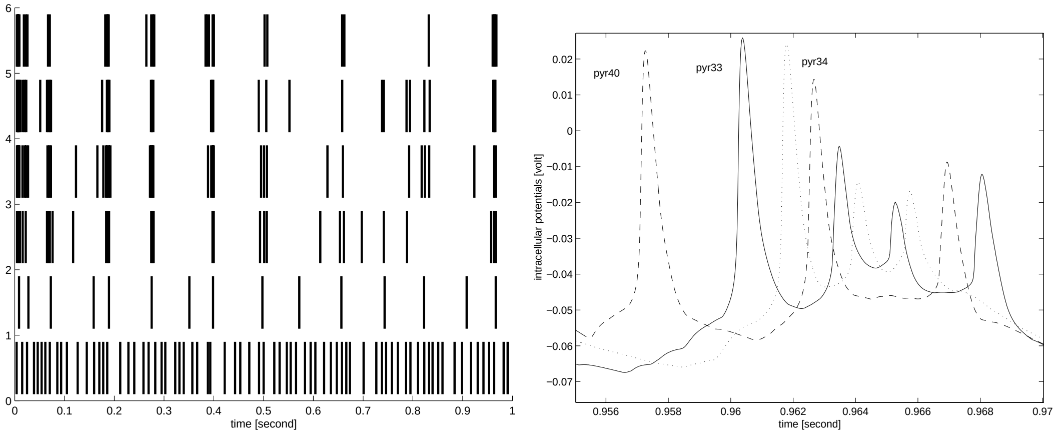

Figure 3.3: Comparison of cell activity. The spike trains on the left originate from neighbouring pyramidal cells 33, 34, 39, 40, feedback interneuron 5 and feedforward interneuron 5 (top to bottom). The times of spike initiation can be saved by means of the `spikehistory` object in GENESIS. The graph on the right provides a detailed view of a synchronous burst of pyramidal cells 33, 34 and 40.

Figure 3.3(a) also illustrates that the activity of pyramidal cells and interneurons is clearly distinguishable, compare Figure 2.11. The interneurons spike regularly, as far as sufficient random input allows for regularity.

Before switching to the investigation of the extracellular potential data, I want to give some qualitative statements about the simulation in general. The simulation results presented in this work are the outcome of a lengthy process. On my way I saw everything from pyramidal cells that are not spiking at all to pyramidal cells that are active for a few milliseconds only to pyramidal cells spiking at a frequency of 400 Hz. It is beyond the scope of this work, but of course it would be interesting to study which parameter settings probably cause which behaviour. However, having approximately 40 important parameters, it is not easy to keep track. Another problem in addition to the number of parameters is, that the system of differential equations is very sensitive to small parameter changes. It is unclear yet, whether unexpected side-effects originate in the solution of the mathematical system or in GENESIS itself. One example may illustrate this: Once I observed a decrease in spike frequency of feedforward interneurons while trying to reduce the spike frequency of feedback interneurons, although no change to the feedforward input appeared and interneurons are in no way directly connected with each other. Concerning the activity of pyramidal cells, I realized that a weak afferent input, restricted to some pyramidal cells only, could not elicit activity in any of the pyramidal cells, even with strong recurrent connections among the pyramidal cells in place. Another impression is, that the influence of feedback interneurons contacting the perisomatic region of pyramidal cells is bigger than that of feedforward interneurons contacting apical dendrites. The strength of the feedback inhibiton was sometimes responsible for activity or no activity of a pyramidal cell.

3.2 Postsimulating steps

The simulated extracellular potential data is not yet suitable as test data. Compared to experimental data, there is not enough noise contained in the data and a filtering process is missing. Therefore some ”postsimulating” steps have to be carried out. The first one is the application of a highpass filter to copy the process experimentalists do while recording spikes. Typical cut-off frequencies are 300-600 Hz. As you know from Section 2.4, low-frequency field potentials, originating from the synaptic activity of groups of cells, are contained within the extracellularly recorded data. These slow waves are problematic with respect to spike detection. At this point a short anticipation of one spike detection method will throw light on why there is a problem with field potentials. Figure 3.5 shows extracellular potential data recorded by one electrode in the ultimate vicinity of the soma of pyramidal cell 40 (Figure 3.4).

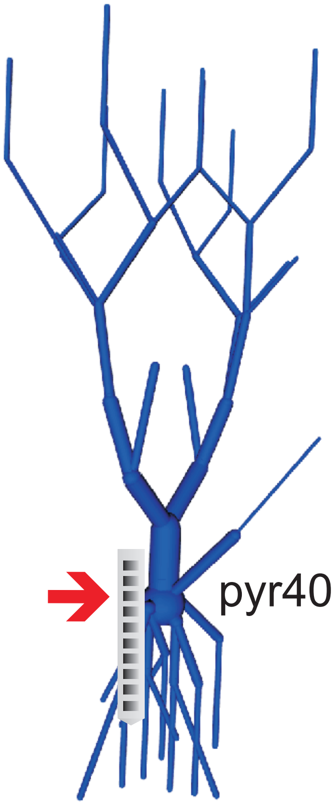

Figure 3.4: Visualization of recording probe relative to pyramidal cell 40. The arrowhead points to the recording site of which data is shown.

Now imagine that you want to detect spike events by means of a threshold. Each time a certain threshold is crossed, you assume to record a spike. But the problem with the unfiltered data is, that spikes are on top of the field potentials. A threshold of e.g. 30 mv would miss spikes hidden in the negativities of the field potential with amplitudes of up to -50 mV. A highpass eliminates the low frequency field potentials from the signal. In this case, I applied a Kaiser window with a cut-off frequency of 500 Hz. In one straight line neatly ordered spikes are the result (Figure 3.5).

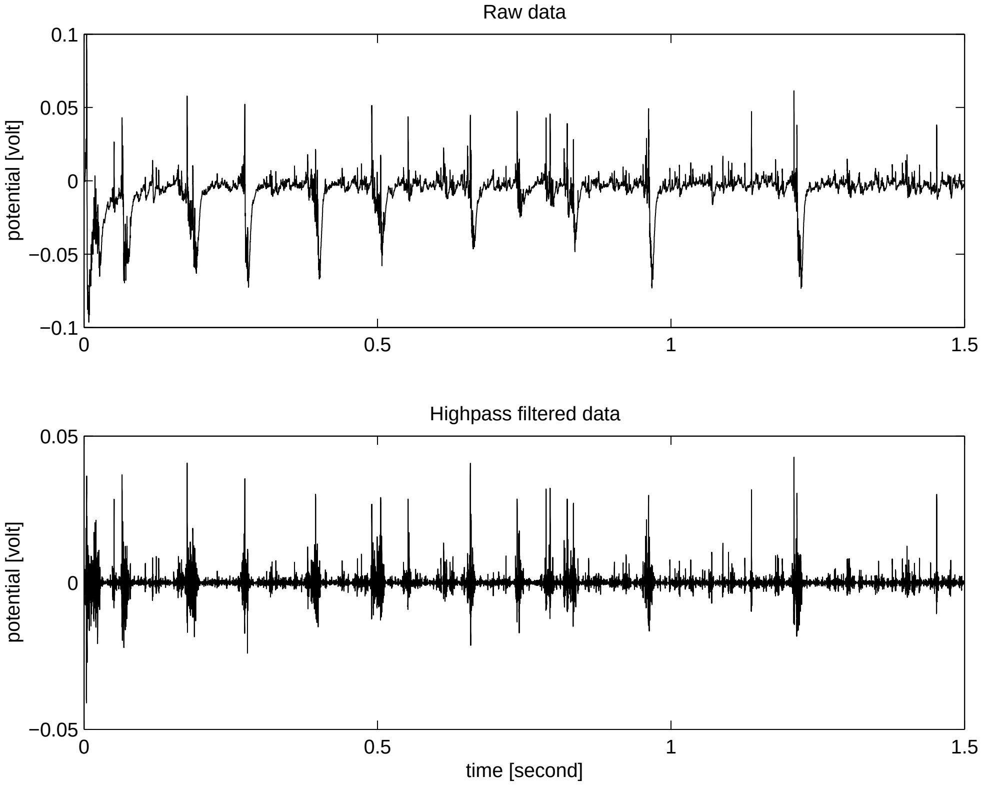

Figure 3.5: Effect of highpass filter.

The addition of noise is the second postsimulating step. You might argue that the signal already looks noisy. Besides some clearly identifiable spikes there is also some low-level activity contained in the simulated signal. This activity results from neurons farther away from the electrode and indeed is a source of noise in extracellular recordings. But in GENESIS I did not find a way to simulate thermal noise caused by the recording equipment, a second very important source of noise in experiments. The instrumentation noise is usually characterized as being white and additive:

$$r(m) = s(m) + n(m). \tag{3.1}$$

$r(m)$ is the recorded signal at the discrete time step m, consisting of neuronal signals $s(m)$ and noise $n(m)$. In order to introduce some kind of thermal noise, I added normal distributed random numbers with mean 0 and standard deviation 1 to the simulated signal with the help of MATLAB. The addition of noise at this process step has the advantage that different signal-to-noise ratios (SNRs) can be achieved by scaling the random numbers appropriately. There exist different definitions for SNR, which is a measure of the quality of the signal on top of a noise background. One possibility is to calculate the SNR as the ratio between the power of the signal and the power of the noise, calculated within windows of the same size containing either spike or noise signals only:

$$\mathsf{SNR} = \frac{\sum_{m=1}^{l} s(m)^2}{\sum_{m=1}^{l} n(m)^2}. \tag{3.2}$$

l denotes the number of sample points within the respective windows. $n(m)$ refers to the noise and $s(m)$ to neural signals at the discrete time step m. Another approach is to compare peak-to-peak amplitudes of the spikes against the noise rms (root mean square) value, which is an upper bound on the noise power. The peak-to-peak amplitude denotes the potential difference between the positive peak and the negative peak of an extracellularly recorded spike. The rms value is calculated as:

$$\mathsf{rms}(n) = \sqrt{\frac{\sum_{m=1}^{l} n(m)^2}{l}}. \tag{3.3}$$

Again, $n(m)$ denotes the noise vector, and l is the length of a window containing noise only. While studying the literature, I realized, that in most publications the SNR is given only qualitatively. Letelier [15] explicitly employs the latter described method. I also settled on this approach, according to which the signal shown in Figure 3.6 has a SNR of 23:1. This SNR of 23:1 is achieved taking all spikes detected with a threshold of 10 mV into account. The choice of threshold indirectly influences the SNR, since spikes with different peak-to-peak amplitudes can be detected dependent on the threshold. Just for comparison, the "power method" (3.2) would even result in a SNR of 45:1. Noisy simulated test data will be listed together with experimental test data in Section 5.3.

Amplitude quantization errors can be considered as a third source of noise. In experiments, quantization errors arise from A/D conversion, discretization and, not difficult to guess, quantization. In the case of my simulation, discrete signals

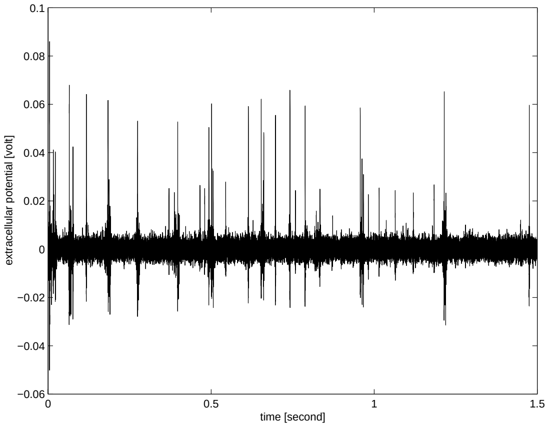

Figure 3.6: Noisy signal.

are the only possible way for calculations. There is no quantization step in the simulation, unless one wants to argue that there is not really an unfinite number of floating point numbers provided by a computer. But the integration step size causes something like a quantization error. Figure 3.7 shows the upper portion of positive peaks contained in a short sequence of the somatic membrane potential of feedforward interneuron 3.

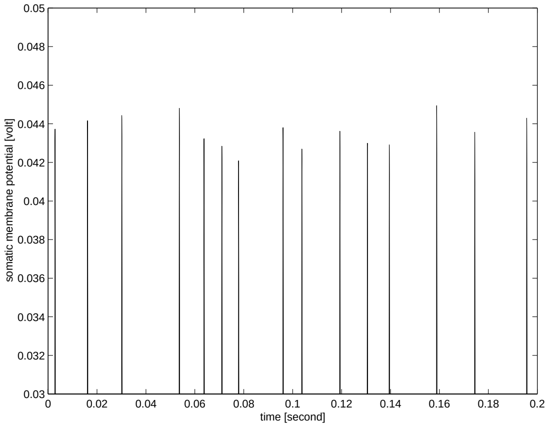

Figure 3.7: Quantization error.

One would expect the spikes to have all the same amplitude, but obviously there are small differences. Of course, real neurons probably do not spike 100 percent indentically each time, which might also lead to amplitude differences. But none the less different spike amplitudes seem to be evoked by the integration step

size in the simulation. While testing different integration step sizes, I observed changes of spike amplitudes relative to each other all the time.
Finally, a comparison of simulated data versus experimental data reveals the similarity (Figure 3.8), except for a scale factor. By choosing a different scale factor for the `efield` objects, recall Chapter 2.4, this difference could be removed.

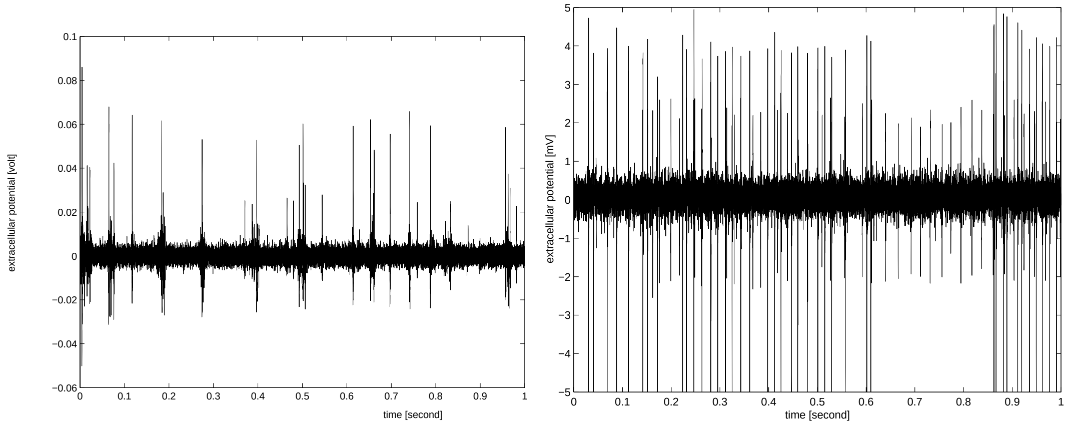

Figure 3.8: Comparison of simulated versus experimental data. The simulated data is the same as in Figure 3.6, restricted to 1 second. The experimental data contains 23.810 samples per second, whereas mine contains 40.000 samples.

Experimental data shown results from a tetrode recording in the *pallidum*, also called *globus pallidus*, of dystonia patients. *Dystonia* is a malfunction in the control of movements. A *tetrode* is a special multi-site recording probe with 4 recording points, that are not of necessity arranged linearly. Shown is the data of one channel. The data is kindly provided by Thomas RECORDING GmbH.
It has to be mentioned, that there exist some more simulation approaches to generate controllable data for test purposes. One frequently applied method is to extract spikes and chunks of noise from extracellular recordings, to distribute the spikes randomly in time and to add the noise sample [2], [27], [40]. The disadvantage of this method is quite obvious. The randomly distributed spikes do not reflect any biologically realistic network behaviour and even no biologically realistic single cell behaviour. The same applies to an approach suggested by Bergman [3]. He provides a simulation environment, where the user can generate spikes through the combination of two principal components, select some noise and define some more parameters for spike frequencies and cell interactions. This set-up provides a lot of opportunities, but still is rather artificial. Nakatani [22] employs a peripheral nerve model in order to generate synthetic data. He adds

a Gaussian noise process to the simulated signal. This method is comparable to mine, but lacks the presence of different interacting neurons within an elaborated network cicuitry, the first step to finally compare an artificial brain to its biological counterpart.

3.3 Amplitude decay alongside a multi-site recording probe

My simulated data is superior to that generated by one of the methods listed at the end of the last section because of three more reasons. First, the alternative approaches produce single-site recording data only. The simulation of multiple-site recording data, if possible, would require a lot of additional efforts. The second advantage results from the first, namely the possibility to study different direction characteristics of the electrodes. This topic will be dealt with in the next section. The third advantage is also a consequence of the first one, it is the clear amplitude decay alongside a multi-site recording probe. Even if they could generate reasonable multiple-site recording data, it is doubtful, whether the other approaches could elicit these features, seen in experimental multi-site recordings as well.

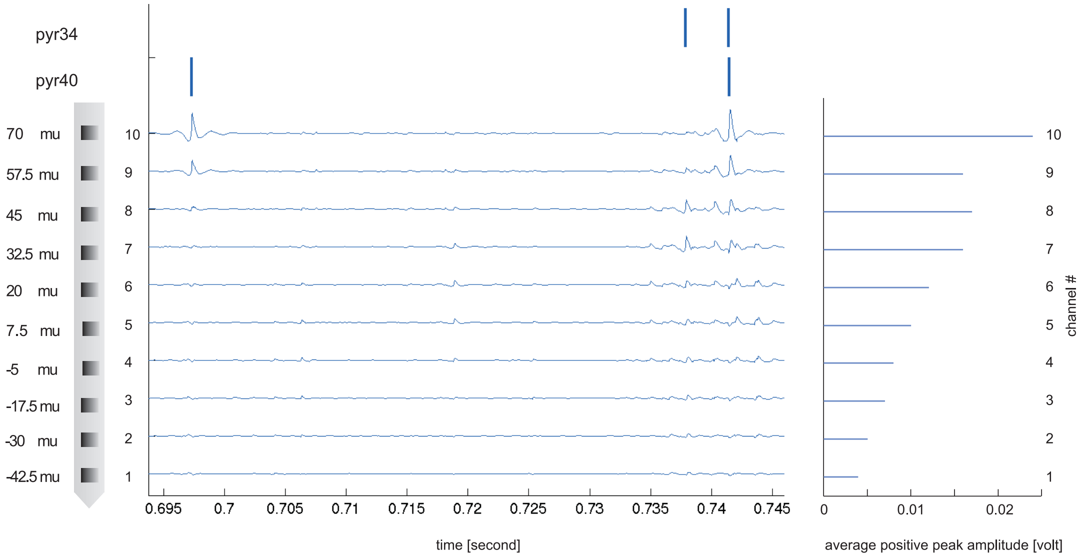

Figure 3.9: Amplitude decay alongside a multi-site recording probe. Spikes of pyramidal cells 34 and 40 are given as reference. The calculations of the average spike amplitudes are based on 90 to 120 threshold detected spikes in each case.

The spike amplitude decay along the site array, approximately with the square of the distance, is a feature of real muti-site recordings. A high amplitude of a spike on one channel/site declines further up and down a linear multi-site recording probe. Figure 3.9 depicts three examples on the basis of simulated highpass filtered multi-site recording data without added noise. It is clearly visible, that spikes are trackable in the upper four channels only, the highest amplitude not always being found in one and the same channel, see average positive peak amplitudes given on the right. The latter is due to the contribution of cells that are located at different z-coordinates. As the spike trains reveal, high amplitude spikes in the first two channels (z=70 μm and z=57.5 μm) originate from pyramidal cell 40, whereas high amplitude spikes in the third and fourth channel (z=45 μm and z= 32.5 μm) originate from pyramidal cell 34. If spikes are trackable in more than two channels, they result from different cells at different levels. The somas of pyramidal cells 34 and 40 are situated at z=12 μm and z=40 μm, respectively, and have a diameter of 30 μm. The axon tends upwards. Pyramidal cell 40 is 4 μm away from the multi-site recording probe, pyramidal cell 34 is 8.1 μm away from the probe. If you remember the position of the probe relative to pyramidal cell 40 (Figure 3.4), it becomes evident that single cell activity is best seen extracellularly at recording sites close to the somas and axon initial segments of cells. The soma is the largest compartment of the models and the axon initial segment next to it is an origin of spike activity. Both factors may contribute to the high spike amplitudes [18]. Recall also Section 2.4, where it was outlined, that the somatic transmembrane currents make the main contribution to the extracellularly recorded signal.

Since spikes are trackable in two channels only and since the simulated recording points have a distance of 12.5 μm, that means that the individual electrodes do not have a big horizon. By "horizon" I mean a cell-electrode distance. Spikes originating from cells within the horizon are clearly identifiable in extracellularly recorded signals. Depending on the experimental set-up, real spikes can be tracked for many tens of microns [16], see below. Therefore the decrease in amplitude seems to be too fast further up and down the simulated linear array. The way too rapid decay of amplitude as well can be found, when the simulated multi-site recording probe is moved away from the cells. I investigated this by generating the highpass filtered signal of the electrode at z=70 μm at different distances from pyramidal cell 40. 17 spikes were detected with a threshold of 0.04 mV at a distance of 4 μm. These spikes were tracked throughout the recordings at increasing distances. The average positive peak amplitude of the 17 spikes was calculated at each distance. The result (Figure 3.10) confirms the above made observation of a rapid decay. At a cell-electrode distance of 15 μm, the spike amplitudes lose already more than 2/3 of their height. Therefore, the more distant pyramidal cell 40 is, the more difficult is the detection of all 17 spikes. While evaluating the data at different distances, I also realized, that the time of detection was on average 2/10 msec delayed at a distance of 41 μm compared to a distance of 4 μm.

The small horizon of the recording sites made it necessary to increase the cell density in the simulated network. In the first implementations [18], the pyramidal cells had a distance of 35 to 45 μm from each other in x- and y-direction.

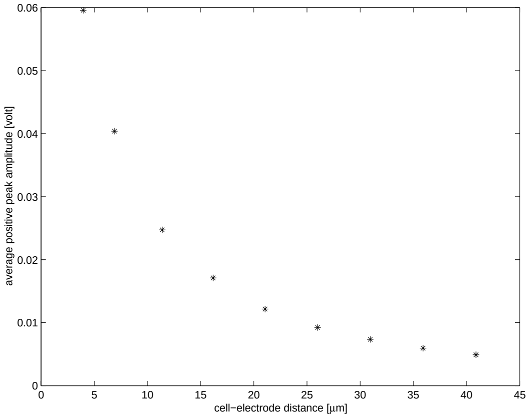

Figure 3.10: Influence of cell-electrode distance on spike amplitude.

Now, the cell distances amount to 7 to 13 μm, compare Section 2.3, in order to prevent recording of equal amplitude spikes from neurons in the closest vicinity of the electrode only [18]. The coordinates of interneurons and input elements are adjusted accordingly.

In addition to the statement of Lewicki [16] that spikes can be tracked for many tens of microns, I found more precise information concerning the horizon of electrodes in articles by Abeles [1] and Gray [11]. Abeles describes a recording by means of a mircroelectrode with a bare metal tip of about 12 μm length in the primary auditory cortex of cats. He claims to record from cells within a radius of 20 μm, this radius not being much bigger than mine in the simulation. Gray, however, specifies for tetrode (impedance of 0.5 to 1 MΩ at 1 kHz) recordings in the neocortex, area 17 of cats, a cell-electrode distance of 65 μm as the distance at which a 90% reduction in voltage occurs. He outlines, that this result agrees with theoretical estimates by Rall concerning the decay of extracellular spike voltage [28]. We therefore conclude, that the small horizon of the simulated electrodes costs doubt on the suitability of the Nunez equation (2.7) for the calculation of fast extracellular potentials and requires further simulation and experimental work to be done in the future.

3.4 Direction characteristic

The question arises, if an electrode sticking in the brain records from cells all around in a 360° angle, or if the electrode has a certain direction characteristic, that means if it records from a smaller opening angle only. The experimental answer depends on the employed type of recording device. A single-site recording probe with the recording site situated at the tip, will record from cells all around with no particular direction preference. But in the case of a multi-site recording probe, where the individual electrodes are fixed on an insulating carrier, this seems rather unlikely.

The Nunez equation

$$F = \frac{1}{4 \cdot \pi \cdot s} \sum_{i=1}^{n} \frac{I_i}{r_i}. \tag{3.4}$$

which is the basis for the simulation of single site recording points in GENESIS (Chapter 2.4) supports the 360° case. But it is possible numerically to realize different direction characteristics by taking into account only the transmembrane currents of cells that lie within a certain sector as seen from the probe (Figure 3.11). An opening angle of 90° proves to be too small. The recorded signal contains almost exclusively spikes of the same amplitude, that means that the opening angle does not allow for the recording from cells farther away that would contribute spikes of different amplitudes. Compared to experimental data (Figure 3.8) this seems unrealistic. An opening angle of 120° allows for different amplitude heights (Figure 3.12). All types of simulated extracellular data, which you were presented in previous sections, result from electrodes with a direction characteristic of 120°.

The discrete structure of the cell array and the small recording horizon of the electrodes do not allow for an investigation of a finer gradation of the angles (Figure 3.11).

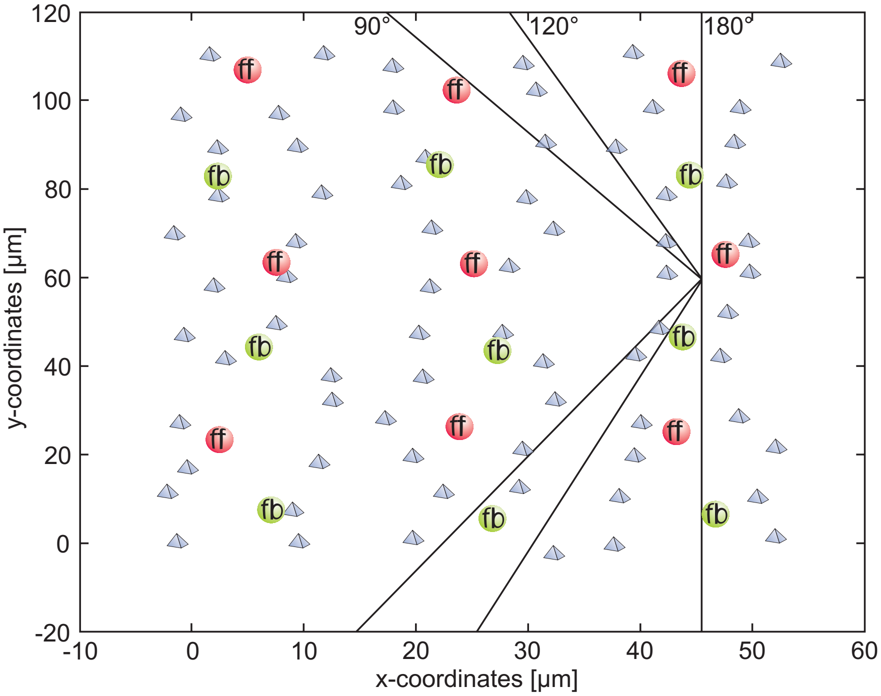

Figure 3.11: Projection of different simulated sectors to be recorded from. All cells on the left of the bordering lines contribute to the respective recorded signal.

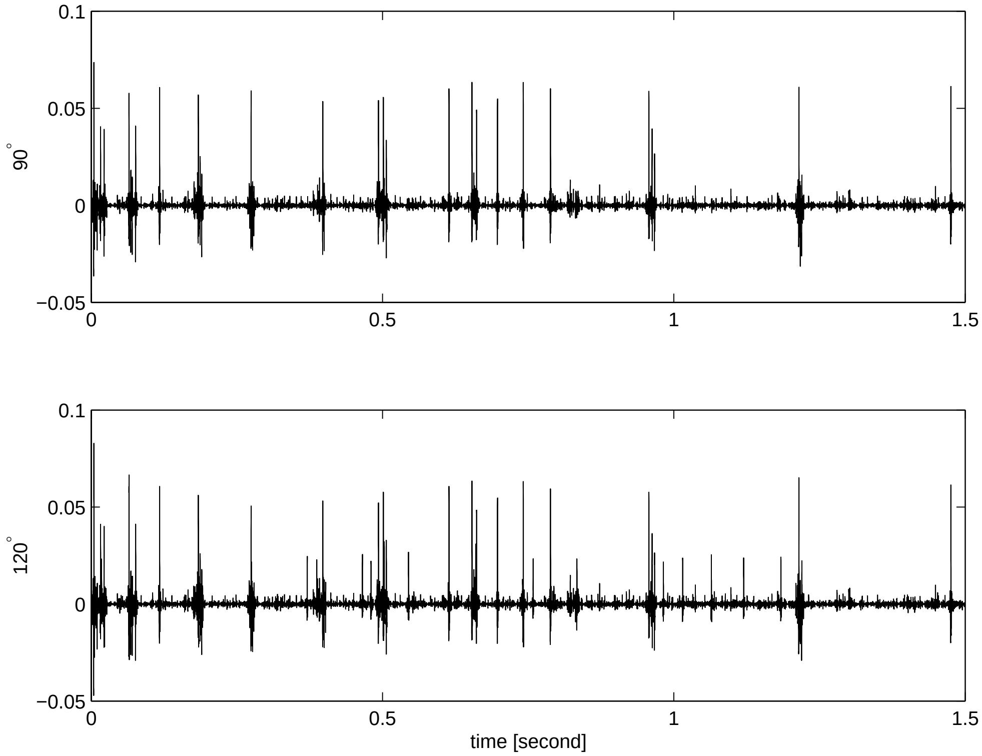

Figure 3.12: Influence of direction characteristic on extracellularly recorded signal.

Chapter 4

Spike features

The shapes of spikes originating from an individual neuron and even from a distinct cell type, are almost congruent. Spikes coming from different cell types are usually not alike concerning their shape. In experiments, the kind of recording electrode and the distance between recording site and neuron also influence the spike shape. But how can these differences be described? Spike characteristics that are commonly used for separation are compiled in the first section of this chapter. In the second section, some of the features will be evaluated for experimental and simulated data.

4.1 Distinguishing marks of spikes

Spike characteristics that allow for a differentiation among spikes are interesting for several reasons. Extracellularly recorded data usually contains the activity of more than one neuron (Section 2.4). In order to get valuable information out of this data, it is a prerequisite, that spikes can be assigned to the neurons or units that elicited them. This process is called *spike sorting* and follows that of spike detection, see next chapter. Spike sorting and sometimes spike detection methods as well make use of certain spike features to separate spikes and to distinguish spikes from noise, respectively. Spike characteristics are important in the framework of neuro-navigation as well. During brain surgeries, it is crucial for the surgeon to know the exact position of his instruments with respect to the patients' brain. As outlined in Section 1.2, different brain regions consist of different cell types eliciting typical spike shapes. The synchronous evaluation of data gained from a recording probe proceeded into the brain can therefore give information about the current position on the basis of detected spike shapes. Currently, the most frequently employed spike feature in neuro-navigation, is, however, none of an individual spike. Rather, the average firing rate, that means the number of

spikes contained in 1 second of the recorded data of one channel, is investigated [3]. Either all spikes detected in 1 second are added up or spikes assigned to individual neurons are separately taken into account. The latter approach obviously requires more efforts, since spike sorting, based on other features, has to be carried out beforehand. If we invest this additional effort, we can take advantage of two more features: Interspike interval histograms can be calculated for the activity of an individual neuron. The interspike interval histogram reveals the distribution of the lengths of time intervals between consecutive spike events. Intervals of less than 1 ms are a hint, that spikes originating from different cells got erronously assigned to one neuron (recall the refractory period of 1 ms). The other feature is the interaction among neurons [3]. It is doubtful, however, whether the study of this feature is reasonable performed during a surgery. Interactions with other neurons are not found in a straightforward way. The investigation of conduction velocities or conduction latencies, respectively, might be more easy, at least off-line. *Conduction velocity* denotes the speed at which an action potential travels along the axon. The estimation of the conduction velocity requires electrodes at two different positions next to the axon. Different signal-to-noise ratios are another global characteristic of extracellular recordings [3].
Spike features characterizing an individual spike shape are even more complex. Figure 4.1 illustrates some of them.

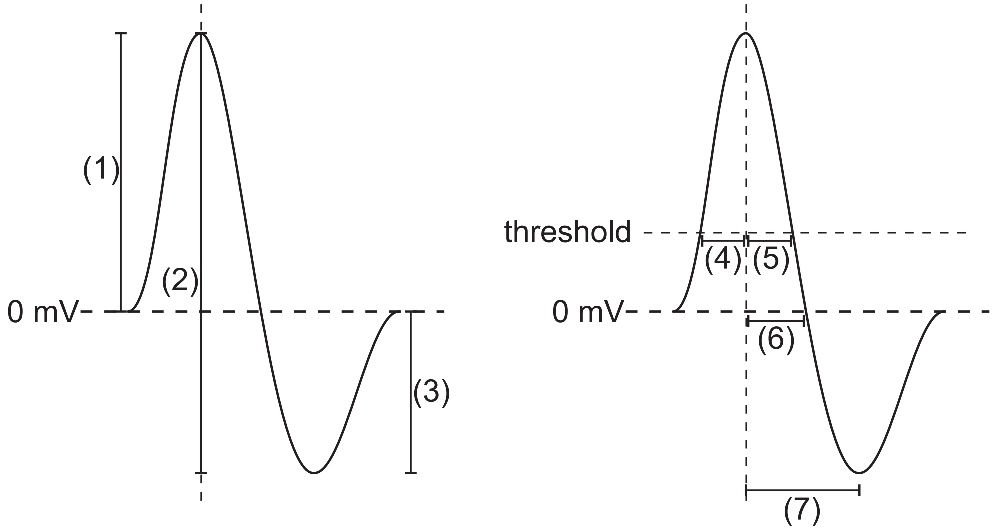

Figure 4.1: Spike features.

The most prominent feature of a spike is its positive peak amplitude (1). In addition, the negative peak amplitude (3) and the peak-to-peak amplitude (ppa) can be measured (2). The estimation of potential values is, however, not restricted to the peaks. An arbitrary number of times can be defined at which the amplitude is to be measured. And the other way round, the lengths of certain time intervals provide another set of spike properties. If the spikes are assumed to be detected by a positive threshold, there are the times between threshold crossing and positive peak (4), between positive peak and recrossing of the threshold (5), between positive peak and next zero crossing (6), between positive and negative peak (7) or the complete spike width. Related features are the slope of the initial sharp rise and the slope of the sharp drop from positive to negative peak. The sharpness, calculated as the second derivative of the signal, can be investigated as well. Equivalent measures can also be applied, if the negative peak preceeds the positve one and if a negative threshold is employed. Schmidt [30] cites a paper by O'Connell [26], where the total spike area and the spike rms value are, in addition to the peak-to-peak amplitude, employed for spike separation. According to O'Connell, the rms value proved to be the most effective feature.

In addition to the analysis of a signal as it was recorded, some more elaborated features can also be derived from a transformed signal. The discrete Fourier transformation (DFT) is one possibility:

$$s(m) \circ\!\!-\!\!\bullet\, S(k) = \sum_{m=0}^{M-1} s(m) e^{-j2\pi\frac{km}{M}}, 0 \leq m, k < M. \tag{4.1}$$

$s(m)$ denotes a discrete signal of length M, for example a spike waveform. Abeles [1] outlines, however, that the spectral analysis of nerve impulses is inconvenient. The problem is, that spikes are narrowly concentrated in the time domain and therefore broadly spread in the frequency domain. As cited by Schmidt [30] at last, the first two Fourier coefficients $S(k)$ in the reconstruction of a waveform cannot provide for a good separation. The coefficients of the orthogonal principal components or Karhunen-Loeve functions are more efficient features concerning the separation of different spike shapes. The principal components are calculated as the eigenvectors of the correlation matrix R of averaged spike waveforms $\mathbf{w}_i$, aligned for example in the spike maximum:

$$R = \sum_i \mathbf{w}_i \mathbf{w}_i^T. \tag{4.2}$$

$\mathbf{w}_i$ denotes a column vector, $\mathbf{w}_i^T$ a row vector. Sometimes the waveforms $\mathbf{w}_i$ are normalized before calculating R:

$$\mathbf{w}'_i = \frac{\mathbf{w}_i}{\parallel \mathbf{w}_i \parallel} \tag{4.3}$$

where $\parallel \cdot \parallel$ denotes a norm. The eigenvectors can be ordered according to their associated eigenvalues. A linear combination

$$\mathbf{y} = l_1 \mathbf{p}_1 + l_2 \mathbf{p}_2 \tag{4.4}$$

of the first two principal components $\mathbf{p}_1$, $\mathbf{p}_2$, that means of the eigenvectors with the highest eigenvalues, approximates a waveform and accounts for more than 90% of the energy of an extracellulary recorded nerve spike [1]. The scaling factors l_1, l_2, calculated as

$$l_j = \mathbf{p}_j^T \mathbf{w}_i, j = 1, 2 \tag{4.5}$$

represent the features. If the waveforms $\mathbf{w}_i$ were normalized, the coefficients l_1, l_2 are more sensitive to small amplitude spikes [39]. Alternatively, principal components may be calculated as eigenvectors of the covariance matrix [16], see next section.

The above mentioned spike characteristics can be combined arbitrarily for the sake of the best possible separation. Some comments about the effectiveness of Fourier coefficients, spike area and ppa were already cited. What about the other features? The positive peak amplitude only, although very obvious, is in most cases not sufficient for the separation task, since usually spikes of nearly equal amplitude but different shapes are obtained [1]. Wheeler and Heetderks [39] state, that amplitude together with conduction velocity are appropriate features in relatively noise free recordings. Spike width, peak-to-zero-crossing time and peak-to-peak time do not seem to be significant [39]. For rather noisy signals, Wheeler and Heetderks [39] propose the pricipal components as the feature of choice. Separation capabilities have obviously to be regarded as functions of noise [6].

Some of the features will be evaluated on simulated and experimental data in the next section. Within the framework of this work, it is only intended to list spike features, not to denote their separation power, which is beyond the scope. The latter would first of all require a measure for "separation power". Wheeler and Heetderks [39] for example define separation matrices. There exist several algorithms for spike sorting, combined with a variety of clustering algorithms. Spike sorting on the basis of any of the above mentioned features can be subsumed as spike sorting on the basis of a reduced feature set, see also [8]. Alternatively, the whole waveform can serve as template in template matching algorithms [3]. Spike sorting on the basis of wavelet transforms [15], with the help of neural networks [6] or by means of optimal filters [10] is also tried. The employment of multi-channel data instead of single-channel data opens the door to yet another class of spike sorting algorithms [4], [11], [14], [27], [29], for example based on correlation techniques. This list of spike sorting methods does not claim to be complete. The interested reader may be referred to Lewicki [16], who provides a rather recent and extensive comparison of spike sorting methods.

When evaluating spike sorting algorithms, besides the separation capability, two more aspects should be investigated: The ease of implementation and the possi-

bility of real-time execution. The latter might be especially important for neuronavigation.
This section shall be terminated with a listing of all discussed spike features in the order of their appearance in the text:

- Average firing rate
- Interspike interval histogram
- Interaction among neurons
- Conduction volocity
- SNR
- Positive peak amplitude
- negative peak amplitude
- ppa
- time between threshold crossing and positive peak
- time between positive peak and recrossing the threshold
- time between positive peak and next zero crossing
- peak-to-peak time
- spike width
- slope of initial sharp rise
- slope of sharp drop from positive to negative peak
- sharpness
- spike area
- spike rms value
- Fourier coefficients
- PCA coefficients

4.2 Characteristics of simulated and experimental spikes

The study of spike features of simulated and experimental data is especially interesting, if the respective data contains distinguishable spike waveforms. A superposition of threshold-detected waveforms, aligned at their positive peaks, reveals at a glance, if different shapes are contained (Figure 4.2). The first threshold crossing would be an alternative alignment point.

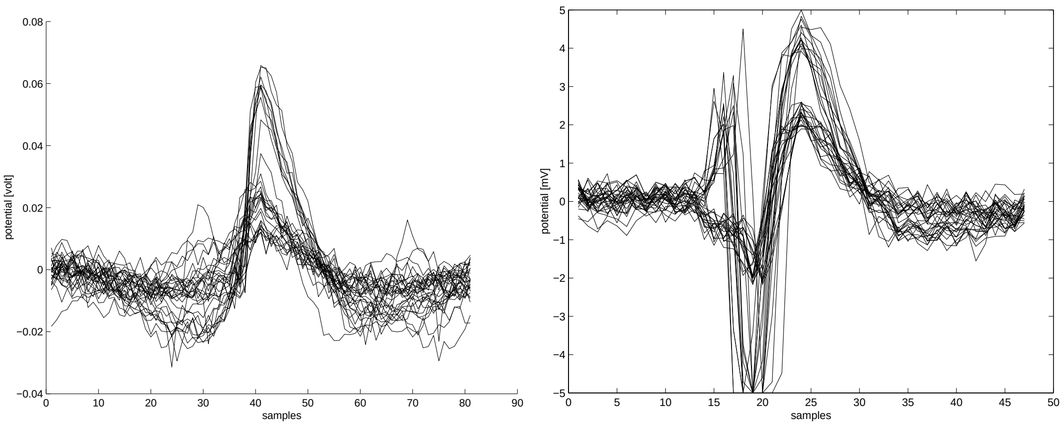

Figure 4.2: Superposition of spike waveforms. Simulated spikes on the left, real spikes on the right. In both cases, 30 spikes are superimposed.

Simulated spikes got detected with a threshold of 10 mV, real spikes with a threshold of 1 mV. One msec preceeding the positive peak and one msec following the positive peak are extracted for each detected spike. Since simulated and experimental data are sampled at 40 kHz and 23 kHz, respectively, spike waveforms consist of a different number of samples. Simulated and experimental data correspond to that shown at the end of Section 3.2. The experimental data in Figure 4.2 contain obviously two different waveforms. In the case of the simulated data, it is difficult to say, if there are two or three different waveforms included. There is an almost continuous transition of low amplitude spike waveforms. Waveform representatives are separately shown in Figure 4.3 for simulated and experimental data.

The study of spike features throws light on the question about the number of different waveforms contained in the simulated data. One second of experimental data and 1.5 seconds of simulated data are investigated. These stretches contain almost the same number of spikes, detected with the above mentioned thresholds:

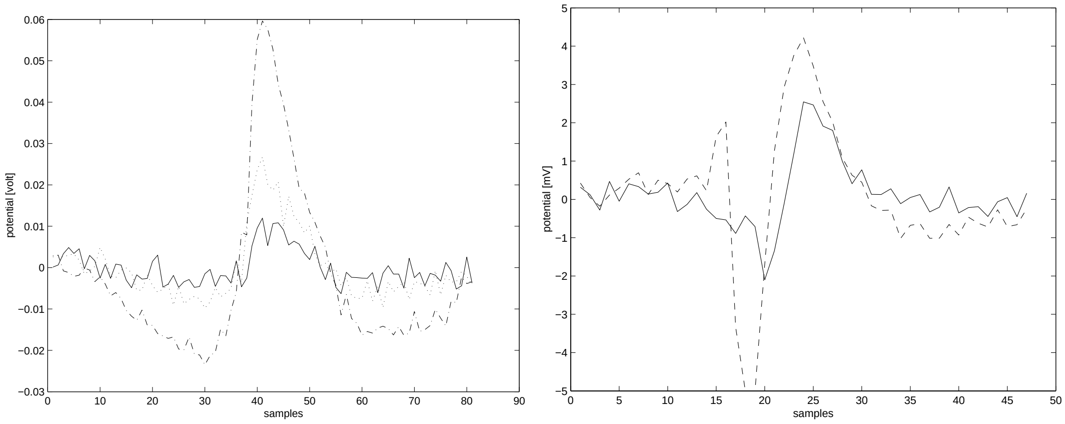

Figure 4.3: Waveform representatives. Simulated spikes on the left, real spikes on the right.

77 in the case of experimental data and 75 in the case of simulated data. The firing frequencies lie at 77 spikes/second for the experiment and approximately 50 spikes/second for the simulation. The general feature of firing rates appears mainly useful for the distinction of different brain regions, see last section. Figure 4.2 gives a hint, that in this case the investigation of the obviously different positive peak amplitudes might lead to further insight. The distribution of positive peak amplitudes is visualized by means of histograms in Figure 4.4.

The positve peak amplitudes of the real spikes are assigned to 2 clearly distinguishable clusters with peaks at 2.25 mV and 4 mV, respectively. The histogram for the simulated spikes supports the suspicion that there are three different waveforms contained in the data. There are peaks at 15 mV, 25 mV and 60 mV, respectively. Note once again the different scaling factor for experimental and simulated data.

What about the distribution of the negative peak amplitudes (Figure 4.5)?

In the case of the real data, the histogram for the distribution of the negative peak amplitudes confirms the hypothesis of the presence of 2 different spike waveforms. The clusters are even more concentrated than in the case of the positive peak amplitude distribution. Despite of the smaller bins, the histogram for the simulated spikes shows 2 clearly identifiable peaks only instead of 3, one at approximately -22.5 mV and the other one at -10 mV.

It is almost redundant to additionally investigate the peak-to-peak amplitudes (Figure 4.6). As is expected, the histogram for the experimental data shows peaks at 4 mV and 9 mV. Although I must say, I would have expected a more obvious peak at 4.25 mV. In the case of simulated data, peaks sit at 20 mV, 35

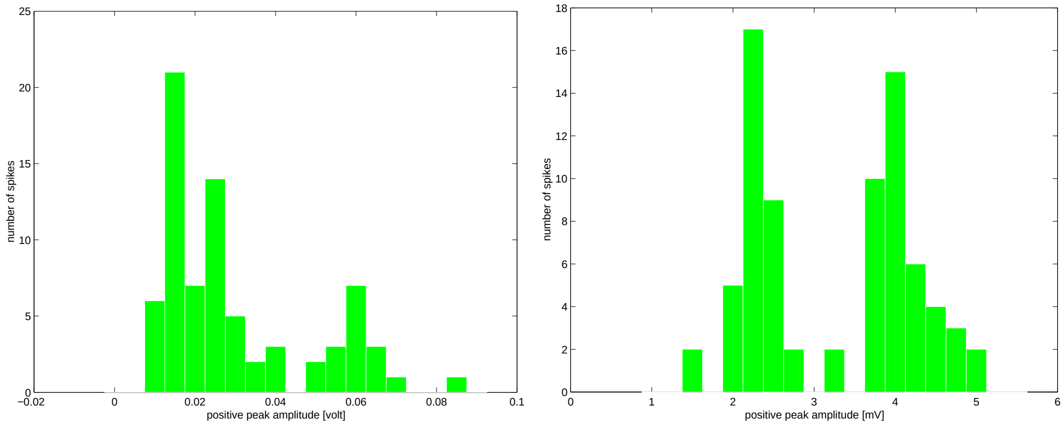

Figure 4.4: Positive peak amplitude distributions. Simulated case on the left, real case on the right. In both histograms, spike amplitudes are distributed among 19 bins of width 5 mV in the simulated case and 0.25 mV in the real case.

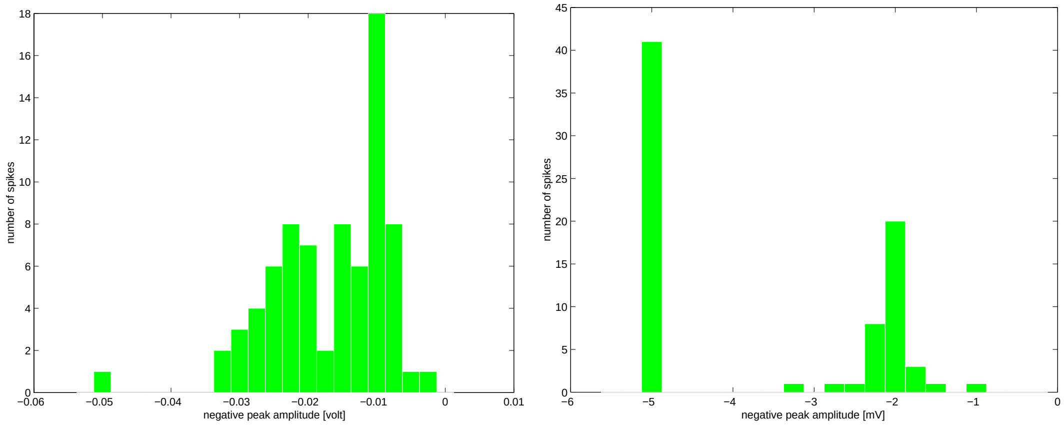

Figure 4.5: Negative peak amplitude distributions. Simulated case on the left, real case on the right. In both histograms, spike amplitudes are distributed among 22 bins of width 2.5 mV in the simulated case and 0.25 mV in the real case.

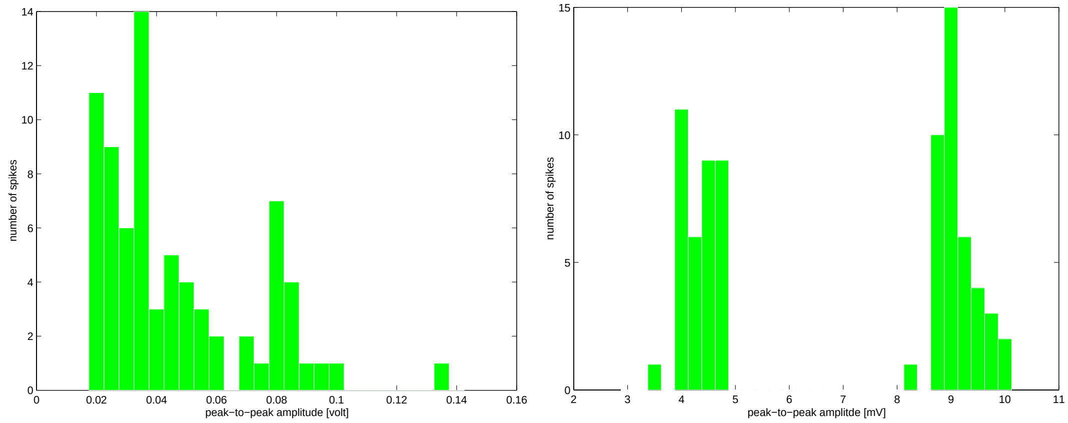

Figure 4.6: Peak-to-peak amplitude distributions. Simulated case on the left, real case on the right.

mV and 80 mV. The peaks in the ppa distribution histograms seem to be less concentrated than in the previous histograms. That is probably, because the ppa includes more information about a waveform. Therefore, the variety within the waveform classes becomes more evident.

The time span between positive peak and negative peak will be the only duration feature investigated in this section (Figure 4.7).

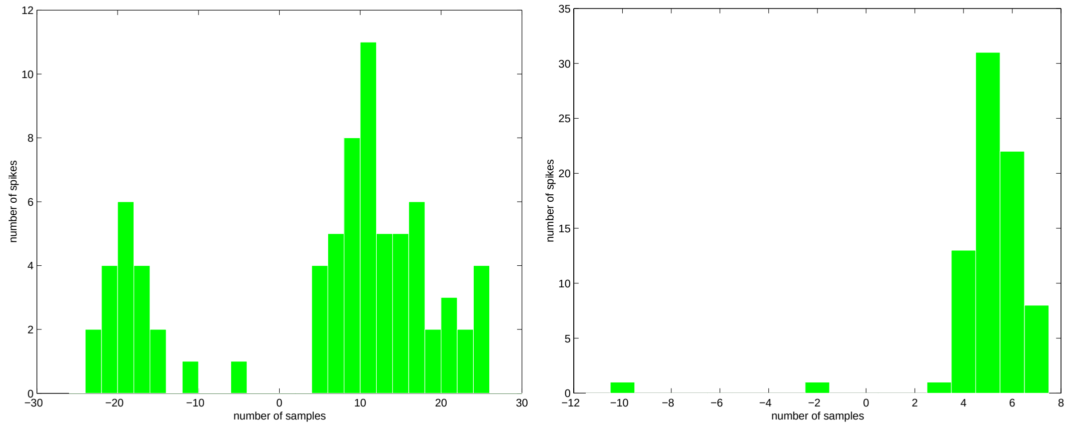

Figure 4.7: Peak-to-peak time distributions. Simulated case on the left, real case on the right.

Except for some outlier, probably resulting from detection errors, almost each real spike has a peak-to-peak time of 4 to 6 samples, corresponding to 0.17 msec and 0.26 msec. This histogram would not make you suspect the presence of two different waveforms in the data. The peak-to-peak times of the simulated spikes are more broadly spread. Negative numbers denote, that the positive peak preceeds the negative one. Two clusters with peaks at -21 samples (0.525 msec) and 11 samples (0.275 msec) are identifiable. There is no obvious hint on the presence of three different waveforms.

Summarizing, for the presented data sets, the positive peak amplitude feature seems to be the most suitable feature of those investigated for spike separation. If the amplitudes have almost the same hight, that might be completely different. None the less, at least for illustrative purposes the principal components shall be calculated.

The Statistics Toolbox of Matlab includes the `princomp` function, that calculates the principal components on the basis of the covariance matrix of the data:

$$[\mathsf{PC}, \mathsf{SCORE}, \mathsf{latent}, \mathsf{tsquare}] = \mathsf{princomp}(X). \tag{4.6}$$

The spike waveforms constitute the rows of input matrix X. The principal components, that means the eigenvectors of the covariance matrix, are returned column by column in matrix PC. Figure 4.8 presents the first three principal components for simulated and experimental spikes.

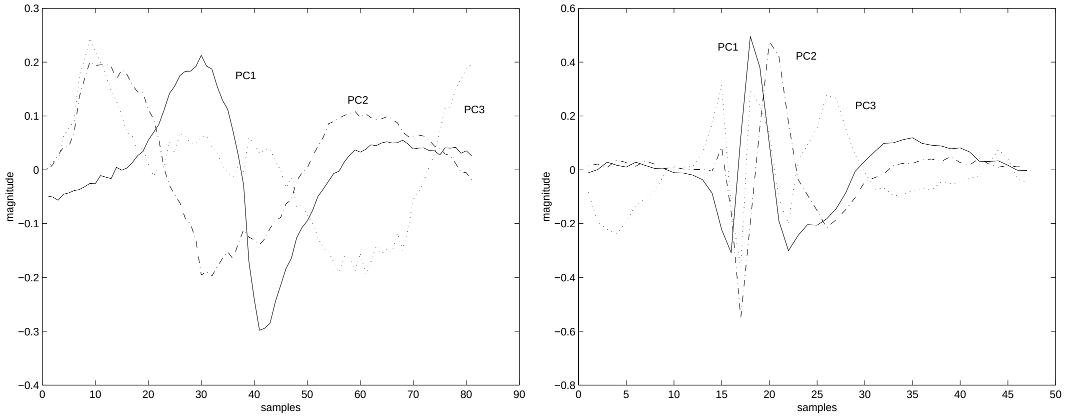

Figure 4.8: Principal components. Simulated case on the left, real case on the right.

The first two principal components have spike-like shapes. They capture directions of largest variation in the data space. The third component is the first one representig variability due to background noise [16]. The principal components get ordered according to their eigenvalues, see below. The coefficients or scale factors l_j, needed for the representation of each spike (Equations 4.4 and 4.5), are contained in the rows of matrix SCORE. In each row, the coefficient of the first principal component can be found in the first column, that of the second component in the second column and so on. The vector latent comprises the eigenvalues λ_j of the covariance matrix of X, representing the variance of the columns of SCORE. The vector tsquare holds Hotelling's T2 statistic for each waveform, which is a measure of the multivariate distance of each waveform from the center of the data set (see Matlab Reference Guide). Figure 4.9 illustrates, that the first two eigenvectors indeed are most significant. Shown is the standard deviation of the coefficients in the direction of each component.

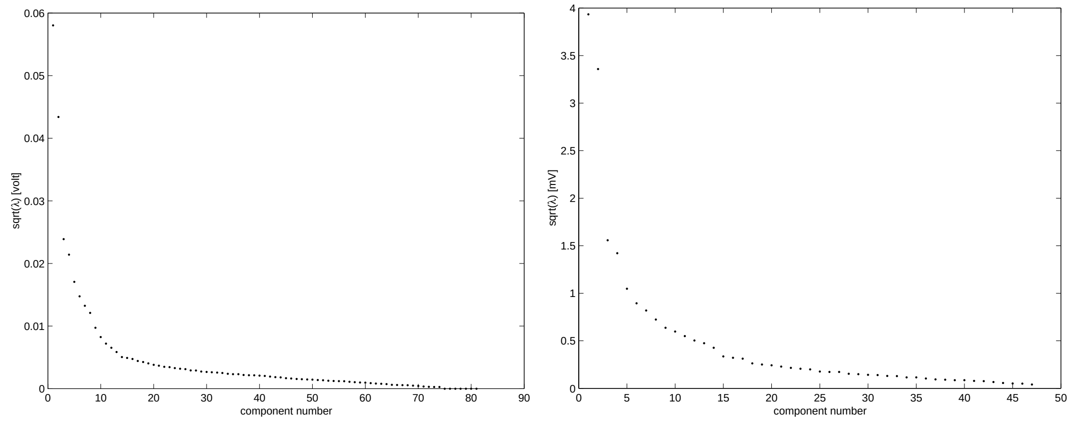

Figure 4.9: Standard deviation of PC coefficients. Simulated case on the left, real case on the right.

Spike separation is carried out on the basis of the first two coefficients, plotted against each other (Figure 4.10). In the case of the experimental data, there are two clearly separable clusters. Spikes with assigned coefficients from the cluster on the left would be regarded as belonging to one waveform class. An equivalent statement holds for the second cluster. Unfortunately, the scatter plot in the case of the simulated data does not allow for an easy distinction of clusters. The question, if there are two or three different waveforms contained in the simulated data has to remain unsolved, unless one takes into account the positive peak amplitude. The PCA method, recommended by Wheeler and Heetderks [39], did not provide non-ambigious results.

In the next chapter, spike features will be used in the framework of spike detection in experimental and especially in simulated data sets.

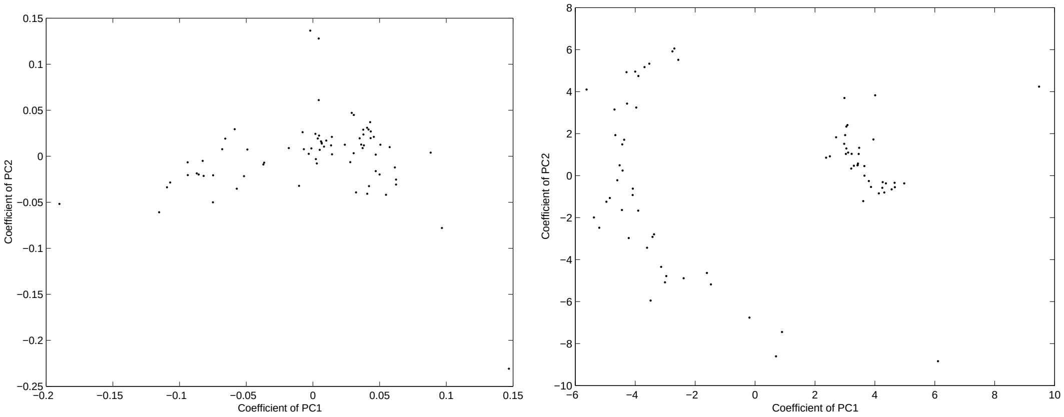

Figure 4.10: Scatter plot of the coefficients of the first two principal components. Simulated case on the left, real case on the right.

Chapter 5

Spike Detection

You probably gained a fuzzy idea about spike detection from the preceeding chapters. You learned what a spike is and that spikes of numerous neurons are reflected in extracellularly recorded data. It was mentioned, that spike detection preceeds the process of spike sorting. Spike sorting denotes the assignment of extracellularly recorded spikes to individual nerve cells, also called "units", according to their shape, once the spikes are detected. Spike detection means, that the points in time have to be found, when there is spike activity, distinguishable from background noise. So far, neither the problems in spike detection have been discussed, nor alternatives to the thresholding method have been listed and evaluated. I will make good for this in this chapter.

5.1 Problems in spike detection

A human expert with some experience is able to decide visually very fast, if waveforms in the extracellular data have to be regarded as spikes or not. But it is absolutely impossible to make a human expert analyse the data of 128 channels, probably recorded for hours with high speed. Algorithms have to be found that can take over this task. A short sequence of experimental data (Figure 5.1) reveals, however, that the task of spike detection is not an easy one.
First of all, there are action potentials of different amplitudes and shapes contained in the data. They originate from different neurons and they all have to be detected by the same method. Second, a sometimes significant amount of background noise makes the task even more difficult. For a more precise description of the noise, recall Section 3.2. Most methods concentrate on coping with spike shapes and background noise, but many of them do not even try to manage the third problem, that of overlapping spikes. The leftmost spike in Figure 5.1 would be denoted as partially overlapping. Figure 5.2 shows almost completely

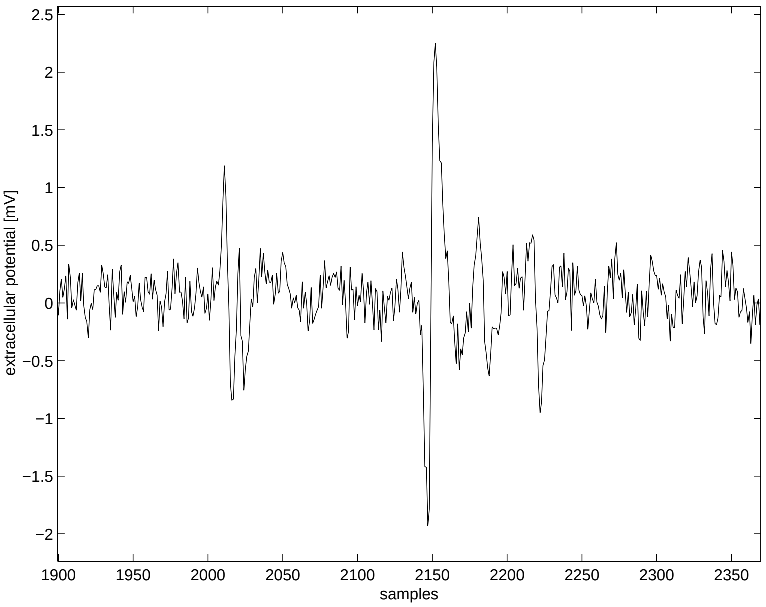

Figure 5.1: Basic problems in spike detection.

overlapping spikes in a stretch of highpass filtered simulated data. On top, spike trains of pyramidal cells 46 and 40, contributing to the extracellularly recorded signal, are presented for comparison. Obviously, the time resolution of individual spike events originating intracellularly is bad extracellularly. Ideally, intracellular spikes would be reflected 1:1 in the extracellular data. Because of the activity of many neurons, however, quite differently shaped overlapping spikes arise.
Figure 5.3 illustrates another problem I did not find to be discussed in the literature: Some intracellular spikes are not at all expressed extracellularly, probably due to cancellations of extracellular potentials by the summed activity of many neurons. Of course, for spike detection algorithms that is not really a problem; there is just nothing to detect. But statements about firing frequencies of individual neurons or the interplay of neurons become less trustworthy. This aspect indeed can hardly be investigated and discussed on the basis of experimental data, since the intracellular acitivity of many neurons is usually not monitored.

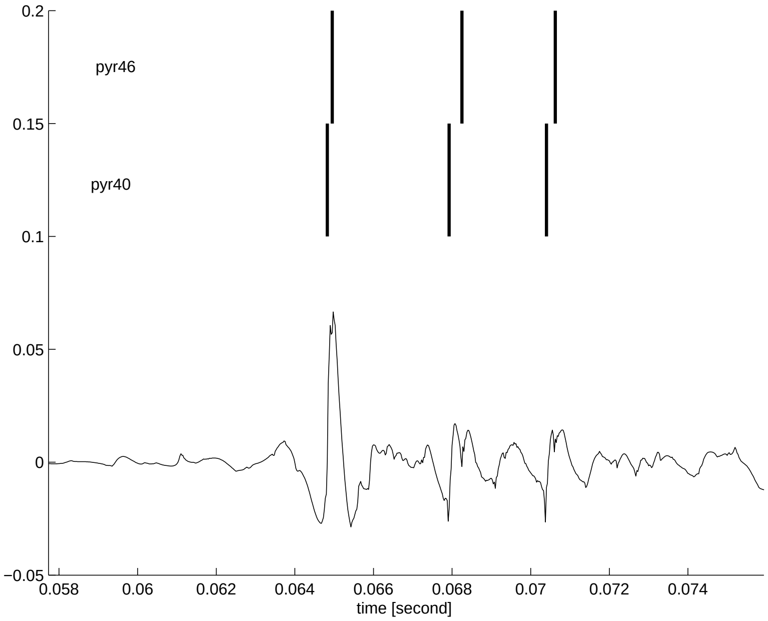

Figure 5.2: Poor resolution of overlapping spike events extracellularly.

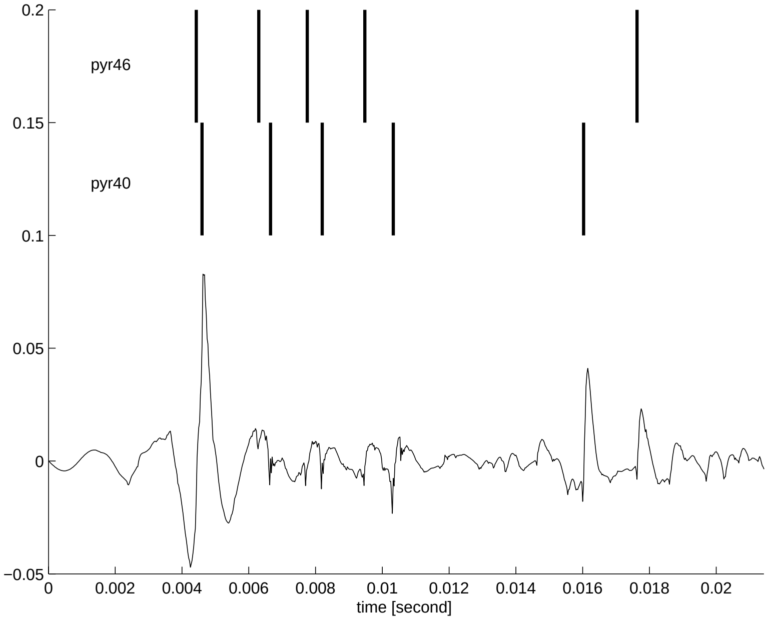

Figure 5.3: Missing extracellular equivalents.

5.2 Spike detection methods

Which spike detection methods are found in the literature? Only practical oriented software-based methods working on digitized data of one channel will be briefly described. That means the special case of dense linear or tetrode arrays are left for further investigations. Spike detection algorithms applied to data recorded with one of these devices can take advantage of the fact, that spikes coming from an individual cell can be tracked on more than one channel. Since a lot of methods aresuitable for both, spike detection and spike sorting, you will retrieve a lot of approaches mentioned in the last chapter in the framework of spike sorting. The method of threshold detection is the most frequently cited, and I already made use of it in previous chapters. You might receive the impression, that a voltage threshold is a straightforward and simple method. It takes advantage of the positive or negative peak amplitude. Each time a positive or negative voltage threshold is crossed, the presence of a spike is assumed. But the devil can be found in the details. How is the threshold to be chosen? How many consecutive samples have to be above threshold to make the assumption of the presence of a spike reliable? And how many samples shall be skipped after the detection of a spike in order to prevent the manifold detection of one and the same spike? These parameters are dependent on the indvidual data set to be analyzed. A suitable threshold can be chosen manually by the user or is calculated on the basis of the noise rms value. Three to five times the noise rms value is a frequently applied positive voltage threshold. The number of consecutive samples above threshold helps to distinguish high frequency noise from spikes. Even if one noise sample point should be above, the following one likely is below threshold. The next section and the tables in Appendix B will reveal, how different parameter settings influence spike detection results.

In addition to positive or negative spike amplitude, some more spike features as discussed in the last chapter can be helpful in the process of spike detection. A threshold for the ppa can be applied exclusively or in addition to a positive or negative voltage threshold. A limitation of allowed peak-to-peak time might further improve the distinction between spike and noise. Sometimes the diverse thresholds are not applied to the raw data, but to the first or second derivative of the signal. Alternatively, the peak above threshold of the signal energy estimated within a shifted fixed sized window (equation 5.1) can be detected. The energy $e(m)$ of the signal $r(m)$ within a window of length k is calculated as

$$e(m) = \sum_{i=m}^{m+k} r(m)^2. \tag{5.1}$$

Denoising with the help of the discrete wavelet transformation (DWT) is another signal modification, that increments the performance of e.g. thresholding methods.

The second most applied spike detection approach, after that of thresholding, is template matching. Templates, in this case spike waveforms, have to be extracted from the first stretch of the recorded data. Either the extracted waveforms themselves serve as template or averaged waveforms make up the template. The template can be adjusted, for example on the basis of the most recently detected spikes, to improve detection performance. One template is responsible for the detection of one characteristic waveform. Congruence can for example be expressed in terms of the Euclidean distance. Detection by templates is one possibility to handle the problem of overlapping spikes. Since the constituent spikes are supposed to add linearly, one rather expensive approach is to apply templates that are the sum of individual templates. All possible combinations have to be built and against against the overlapping spikes found in the data, thus making online detection impossible.

Another method is the detection with the help of principal components, the eigenvectors of the correlation matrix, recall last chapter. Since the first two principal components $\mathbf{p}_1$, $\mathbf{p}_2$, in which the signal energy exceeds the noise energy, represent approximately 90% of the energy of a spike, it seems to be useful to compute an energy $e^{\mathsf{PC}}(m)$, where

$$e^{\mathsf{PC}}(m) = l_1^2 + l_2^2 = [\sum_{i=m}^{m+k} \mathbf{p}_1(m)\mathbf{r}(m)]^2 + [\sum_{i=m}^{m+k} \mathbf{p}_2(m)\mathbf{r}(m)]^2 \tag{5.2}$$

and to apply a threshold to this kind of signal [24, page 66].

The above descriptions outline only the basic ideas of the diverse approaches. More or less subtle differences are hidden in individual implementations. But the question remains, which spike detection method is best? In a book edited by Nicolelis [24], Wheeler cites an empirical investigation that found threshold detection performing better than any alternative method. Most performance evaluations found in the literature, however, concentrate on the problem of spike sorting. Since spike sorting and spike detection methods are related to each other, two recent results shall be mentioned. In [24], a reduced feature set approach is favoured, followed by template matching and the employment of principal components. Lewicki's order [16] looks only slightly different: Template matching, principal components, manually chosen feature sets and optimal filters. The problem is, however, that either author carried out performance comparisons on his special data set. There doesn't exist something like a general agreed benchmark, that means a data set on which each approach is tested. I suggest that a benchmark should in any case comprise biologically realistic simulated data like mine. It provides for control of precise spike trains and still realistic features.

5.3 Performance evaluation

Now let us satisfy your curiosity: Which method is best according to my data sets? The performance of the following methods will be evaluated: Positive voltage threshold (pt), positive voltage threshold plus ppa threshold (pt+ppa), positive voltage threshold plus ppa threshold plus peak-to-peak time limitation (pt+ppa+pp-time), ppa threshold only (ppa-window) and positive threshold applied to the second derivative of the signal (sec_dev+pt), to the energy of the signal (energy+pt) or to the signal denoised with the help of the DWT (dwt-denoising+pt). The respective algorithms, implemented in Matlab, can be found on the accompanying CD.
Figure 5.4 presents experimental and simulated test data at a glance. For the experimental data, I calculated SNRs of 25:1 and 9:1, respectively, on the basis of peak-to-peak amplitudes and the noise rms-value, recall Section 3.2. The noise rms-value is calculated on stretches where no spikes were detected. The simulated data has a SNR of 21:1.

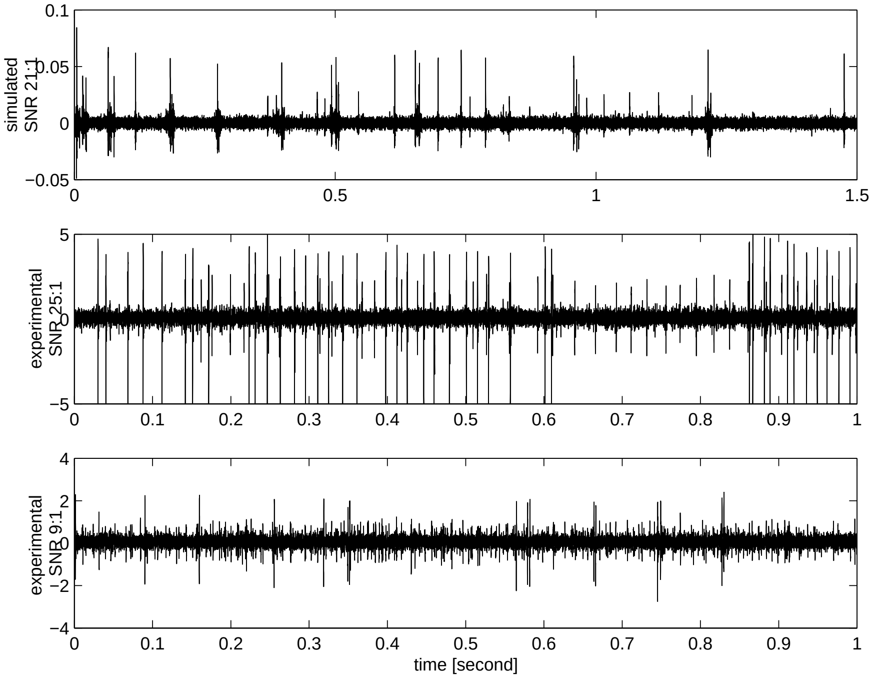

Figure 5.4: Test data.

In order to be able to evaluate the performance of different methods, the performance has somehow to be measured. For this purpose, typically the number of correctly detected spikes, the number of false positives and that of false negatives are estimated. *False negatives* are all those spikes that are missed by a method. *False positives* are any background events wrongly accepted as spikes. Ideally both numbers should be small. These numbers cannot be calculated of course without the knowledge about the spikes really contained in the data. In the case of the simulated data, there is no doubt at which moments there are action potentials. GENESIS provides the `spikehistory` object for saving the points in time at which a neuron generates an action potential. Each pyramidal cell and interneuron in the network has its own assigned `spikehistory` object. Intracellular spikes only get a time stamp this way, but in a split milisecond the nerve impuls should also be visible in the extracellular data. I write "should be", because we saw in Section 5.1 that some intracellular spikes are not reflected extracellularly, thus contributing to the number of false negatives, when points in time of extracellularly detected spikes are compared against points in time of intracellular spikes. Since the horizon of simulated electrodes proved to be small, only three neurons are markedly contributing to the simulated test signal: Pyramidal cells 40, 46 and 27. Extracellular spikes coming from pyramidal cells 40 and 46 have spike amplitudes clearly above the noise level. The spikes originating from pyramidal cell 27 are almost hidden in the noise. Detection results will be compared against spike times of pyramidal cells 40 and 46 only (70 spikes in total), and additionally against the spike times of all three pyramidal cells (112 spikes in total).

In the case of experimental data, I had to detect the spikes manually. While doing so I experienced myself what I mentioned initially: Experimental data is always subjected to interpretations concerning contained spiketrains. However, since I tested all methods on the same interpretation, it should not be all that important, whether I detected absolutely correct or not. Nobody could judge this with certainty and even a large number of experts wouldn't come to 100% agreement. Also spikes hidden in the noise level are regarded. In some performance evaluations they are intentionally skipped. The experimental data with a SNR of 25:1 contains 95 spikes, those with a SNR of 9:1 180 spikes. I saved the moment of the positive peak of each spike and made the algorithms return the same. Thus a direct comparison is possible.

Performance results are given in numbers of spikes in Figures 5.6, 5.8, 5.9. The figures present the best results for each method, achieved with a distinct parameter setting. Classification results for different parameter settings can be found in the tables of Appendix B. It is not really obvious, which result returned by one of the methods should be judged the best one. I decided to concentrate on a high number of correctly detected spikes and a low number of false negatives, the number of false positives being the least important criterion. But that does not mean, that the number of false positives is negligible. A high number of false positives obscures statements about firing frequencies and cell interactions.

The influence of some parameters, first of all threshold values, shall be outlined at this point. In general, low thresholds, be it the positive voltage threshold or the ppa threshold, lead to a small number of false negatives (fn, Figure 5.5) , but to a high number of false positives (fp, Figure 5.5). The contrary is true for high threshold values. Figure 5.5 illustrates this fact for a positive voltage threshold applied to the experimental data with a SNR of 9:1, containing 180 spikes. Obviously one has to decide if it is more important to get a low number of false positives, that means background events wrongly detected as spikes, or a low number of false negatives, contained spikes not detected. Both has to be payed for with a high number of events in the other error class. Even a threshold of approximately 0.44 mV leading to error classes of the same size, would not provide really satisfactory results. In the example, 23% of the manually detected spikes are missed by a positive voltage threshold of 0.44 mV.

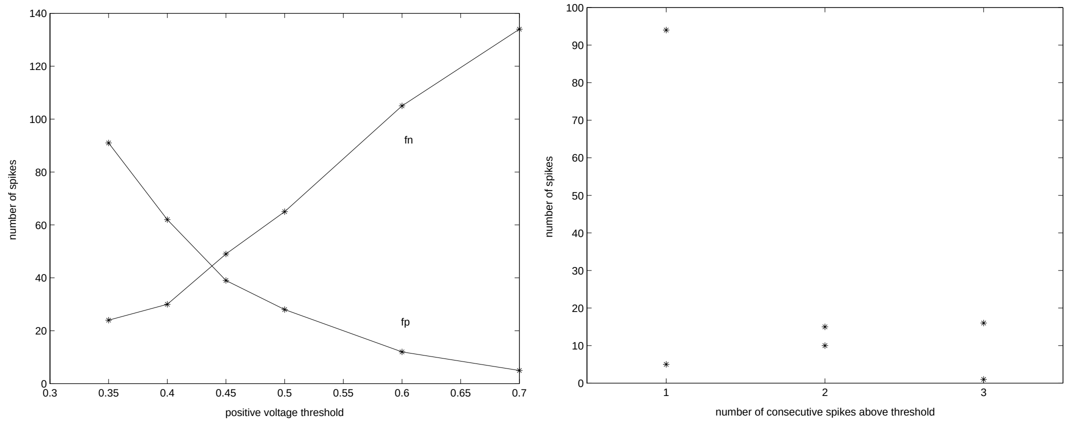

Figure 5.5: Influence of parameters in positive voltage threshold approach. Influence of the threshold on the left. Influence of the number of consecutive samples that have to be above threshold on the right.

Another important parameter concerning the positive voltage threshold approach is the number of consecutive samples that have to be above threshold (Figure 5.5). The graph is based on the experimental data with a SNR of 25:1. The more consecutive samples are asked to be above threshold, the more false positives are rejected. The number of false positives is reduced by 90%, if two consecutive samples have to be above threshold instead of one only, while the number of false negatives is only slightly increased. This is explainable by the fact that the noise has a high frequency. If one noise sample point should be above, the following one likely is below threshold. On the other hand, nerve spikes have a width of a few samples, of which more than one can be expected to lie above threshold.

Only few high amplitude spikes are that narrow as to be missed by a demand for two consecutive samples above threshold. Some low amplitude spikes, hardly crossing the threshold, might rather be missed.

Now back to the results presented in Figures 5.6, 5.8, 5.9. The results for the experimental data make clear, that the detection methods work best under high SNRs. In the case of the experimental data with a SNR of 25:1, even the simple positive voltage threshold performs comparatively well. Because of the high signal to noise ratio, spikes can be separated from noise quite effectively, thus leading to a low number of false positives. However, 17% of the manually detected spikes are missed. Additional investigation of ppa and pp-time cannot reduce this percentage, they are checked after the study of the positive threshold. By taking into account the pp-time, the one false positive event can be eliminated, obviously having a width exceeding the preset limitation. Taking only the ppa into account results in a high number of false positives. The method I called "ppa-window" works as follows [2]: A window comprising a fixed number of samples is moved along the data. The maximum and minimum values within this window are looked for each time and the difference maximum-minimum is calculated. Let me call this value max-min value. If this value crosses the preset ppa threshold, the presence of a spike is assumed. The problem seems to be, that not the global peaks are found, thus leading to small max-min values. The ppa threshold has to be chosen very low, although we know from Section 4.2, that the spikes have real ppas of several mV. The low threshold allows for the detection of a huge number of false positives. When thresholding the second derivative, the energy or the denoised signal, another problem arises. The detected peaks cannot directly be assigned to the manually detected peaks of the original signal, since the peaks of the modified signal occur shifted. Therefore one has to allow for a certain discrepancy when carrying out the comparison. Thresholding of the second derivate or the denoised signal does not improve detection results. Thresholding of the energy adds a reasonable number of false positives.

The positive threshold combined with the ppa threshold performs best in the case of the experimental data with a SNR of 9:1. "Best" in this case still means, however, that 30% of all detected spikes are false positives and 20% of all manually detected spikes are false negatives. Especially the high number of false positives may be contributed to the low SNR. Some noise amplitudes are of approximately the same height as spike amplitudes, thus being detected together with them. The investigation of the positive peak amplitude only does not perform much worse. Additionally taking into account the ppa-time rejects some false positives, but unfortunately also some formerly correctly detected spikes. Thresholding the second derivative of the signal delivers the smallest number of false positives, but also a high number of false negatives. Maybe some words should be said about the differentiation algorithm. Since differentiators amplify high frequencies, they are

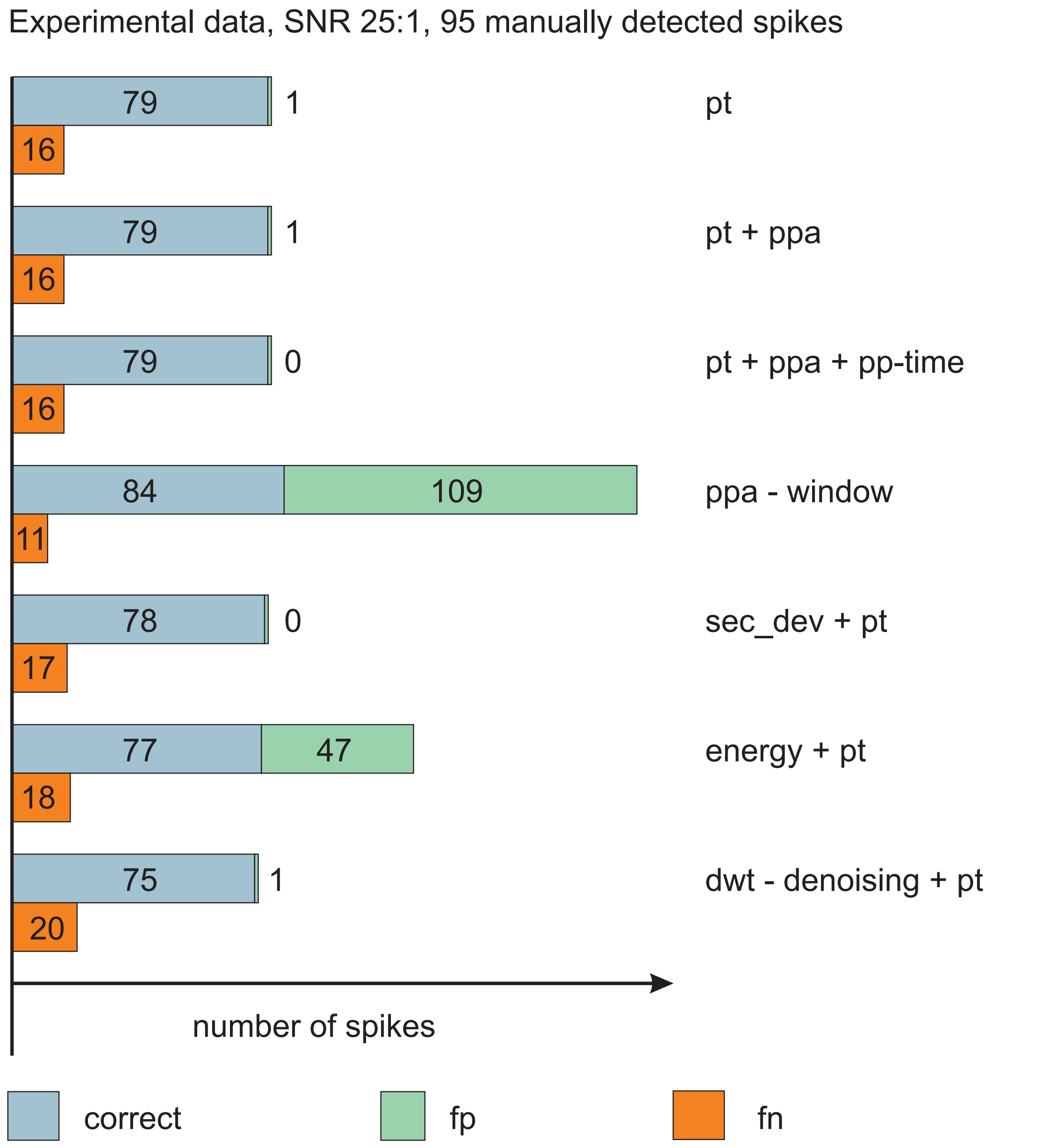

Figure 5.6: Results for experimental data with SNR of 25:1.

rather unsuitabel for the analysis of neuronal signals containing high frequency noise. Therefore I applied a low-pass differentiation algorithm [37], instead of a full-band differentiation. The low-pass differentiation filter of fourth order has the coefficients

$$[1, 2, 1, -2, -4, -2, 1, 2, 1], \tag{5.3}$$

scaled by $\frac{1}{36 \cdot T^2}$, thus realizing the operation

$$\begin{aligned} x^{(2)}(m) \quad = \quad & \frac{1}{36T^2}[x(m-4) + 2 \cdot x(m-3) + x(m-2) - 2 \cdot x(m-1) \\ & -4 \cdot x(m) - 2 \cdot x(m+1) + x(m+2) + 2 \cdot x(m+3) + x(m+4)] \end{aligned} \tag{5.4}$$

T denotes the sampling interval.

Astonishingly, the denoising impairs the detection performance. Probably different parameters have to be chosen, either for the detection or for the denoising itself. For denoising I applied the Matlab function `wden`, which performs automatic 1-D denoising using wavelets, with the following parameter settings:

$$\mathsf{denoisedsignal} = \mathrm{wden}(\mathsf{signal}, '\mathsf{minimaxi}', '\mathsf{h}', '\mathsf{mln}', 4, '\mathsf{db4}'). \tag{5.5}$$

That means that the decomposition of the input signal is performed up to level 4 employing the Daubechie 4 wavelet. The wavelet coefficients get clipped hard ('h') according to the minimax threshold selection rule. The thresholds are rescaled using level-dependent estimation of level noise. Thresholding of the energy and the ppa-window method again perform worst.

Figure 5.9 illustrating the results for the simulated data shows four bars assigned to each method. The two bars on top result from comparisons with the spike times of pyramidal cells 40 and 46 only, the two bars on bottom from comparisons with the spike times of pyramidal cells 40, 46 and 27. Obviously the low amplitude spikes coming from pyramidal cell 27 are difficult to detect. In the case of the pt method e.g., 24% of the 70 spikes originating from pyramidal cells 40 and 46 are missed, compared to 45% of 112 spikes originating from pyramidal cells 40, 46 and 27. This might also be a result of the sometimes synchronous activity. As outlined in Section 5.1, extracellular time resolution of individual spikes during synchronous activity is bad. An overlapping spike can easily be detected as one spike event only. Again, the positive voltage threshold belongs to one of the best methods, beaten only by the DWT. Despite of the bad performance of the DWT-denoising in the case of the experimental data, the good performance in this case is not unexpected. Denoising by means of the DWT is especially effective if the noise is white, and if you recall Section 3.2, the artificial noise added to the simulated data was of this kind. Thresholding of the second derivative or the

energy of the signal performs especially poor on this kind of data. It is unclear, why the performance on the simulated data with a SNR of 21:1 is much worse than that on the experimental data with a SNR of 25:1. Even by trying different parameter settings (B) I did not achieve better results. The high number of false positives is also astonishing with respect to one false positive only in the experimental data with a SNR of 25:1.
Since the positive voltage threshold proved to be among the best performing methods on simulated data, this method will be investigated for two lower SNRs, taking advantage of the fact, that with the simulated data different SNRs can easily be achieved by scaling the added noise respectively. In Figure 5.7 the number of correctly detected spikes and of false positives and false negatives are compared against each other for SNRs of 21:1 (see above), 11:1 and 6:1. Spikes of pyramidal cells 40 and 46 only are taken into accout. It becomes very evident how a bad SNR influences performance results. In the case of a SNR of 11:1, the same number of correctly detected spikes as in the case of a SNR of 21:1 has to be paid for with 13 more false positives. In the case of a SNR of 6:1, the thresholding method obviously does not make sense any more. The number of false positives is by far dominating. Summarizing, according to the presented results, the statement

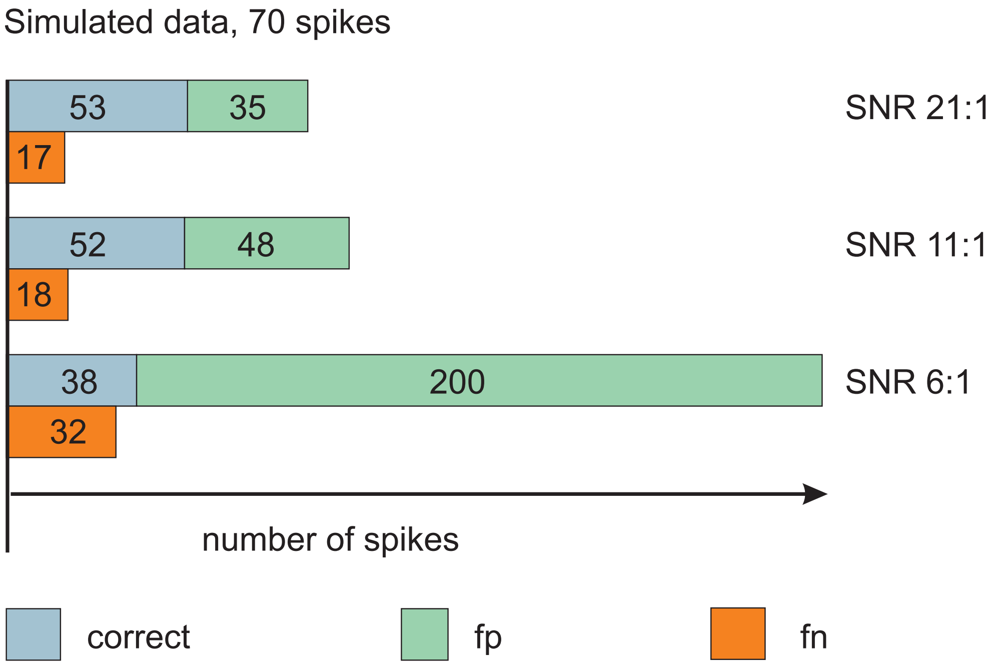

Figure 5.7: Results of positive threshold method for simulated data with different SNRs.

can be supported,that threshold detection performs best compared to alternative methods, but only for high SNRs. However, the evaluation of approaches like template matching or employment of principal components is dedicated to future work as well as the performance investigation on overlapping spikes. It will be interesting to look for a method that performs well even in the case of bad SNRs.

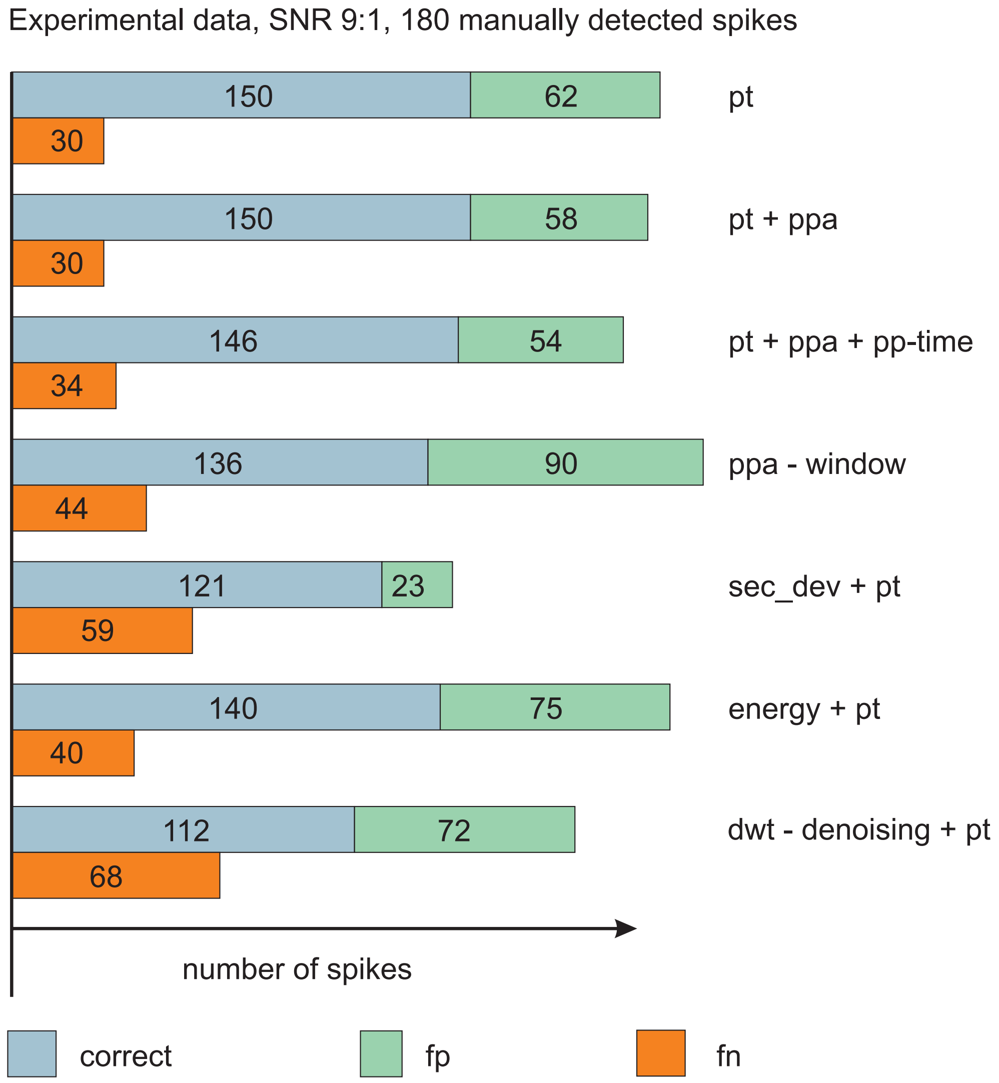

Figure 5.8: Results for experimental data with SNR of 9:1.

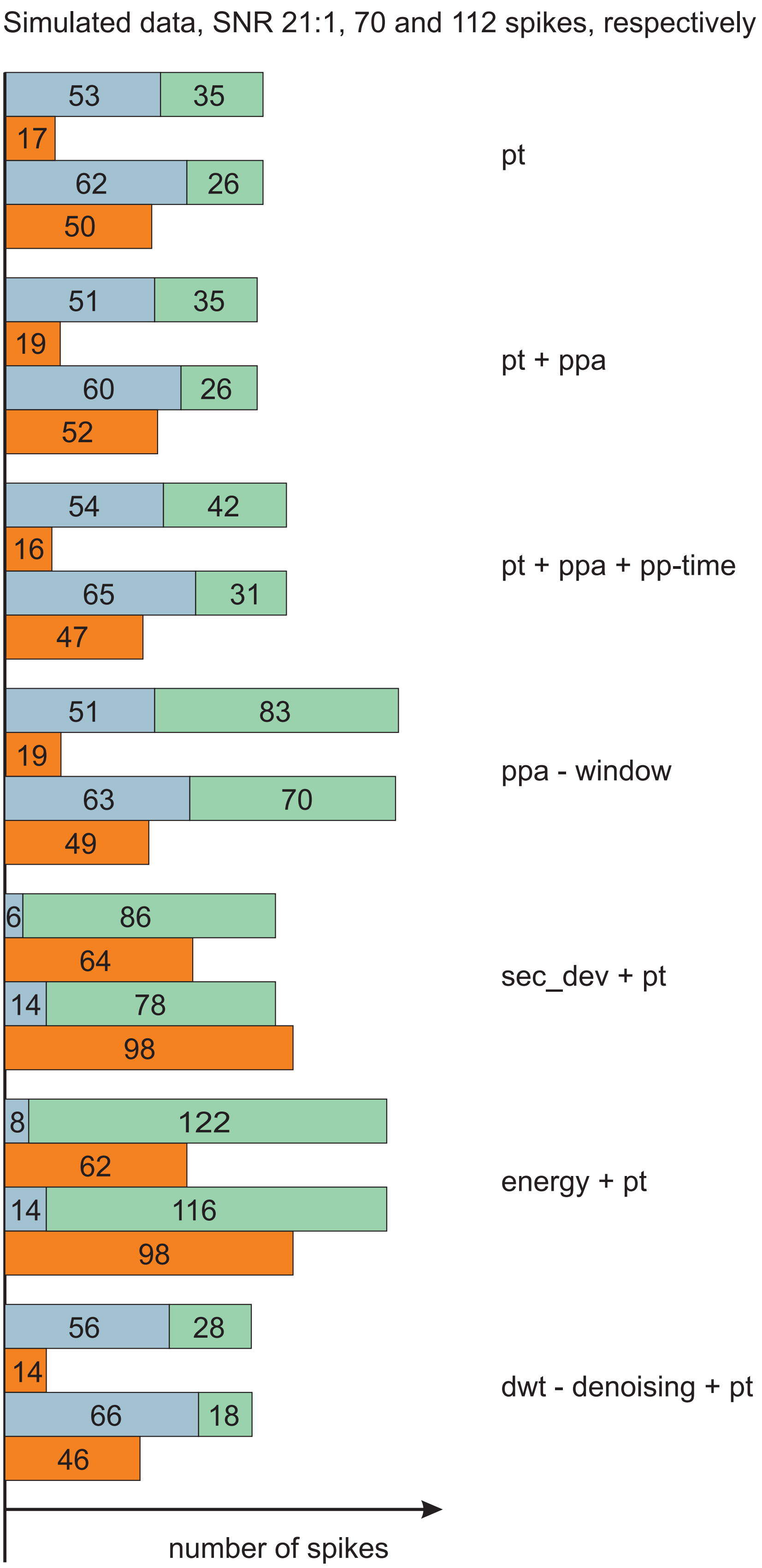

Figure 5.9: Results for simulated data with SNR of 21:1.

Chapter 6

Cool-down

6.1 Summary

The main goal of this work is the simulation of biologically realistic extracellular potential data for the evaluation of various spike analysis methods. For this purpose a biologically realistic network simulation is set up that mimicks a tiny part of hippocampus. The network, implemented in the freely available neural simulator GENESIS, consists of 72 pyramidal cells and 18 interspersed interneurons. The models are oriented on experimental data. The cells interact via simulated synapses. The overall network behaviour ranges from synchronous bursts to single spikes of individual cells. The simulation of the recording of extracellular potential data is simulated on the basis of an equation by Nunez [25] for the calculation of extracellular field potentials. The simulated recording data has several advantages compared to other controllable synthetic data commonly applied for test purposes: Multisite recordings can be simulated and the data resulting show some features of experimental data, for example spike amplitude decay alongside a linear recording array. However, a severe difference between experiment and simulation can be found in the horizon of the simulated electrodes: The simulated electrodes record only from neurons within a distance of 15 μm instead of up to 65 μm as real electrodes do. After highpass filtering and the additon of white noise, the simulated extracellular recording data serves as test data for spike detection algorithms. Many spike features are listed and explained on the basis of simulated and experimental data. Some of the features get employed in spike detection algorithms. For the first time, different spike detection methods, including thresholding of the positive peak amplitude or peak-to-peak amplitude and thresholding of the signal denoised by means of the DWT and thresholding of the energy and second derivative of the signal, are successfully tested on biologically realistic simulated data. Tests on experimental data are carried out additionally. The results support an earlier finding, that a simple positive voltage threshold frequently performs better than alternative more elaborated methods, but only for high SNRs.

6.2 To do

I did the first steps concerning the evaluation of spike detection methods on biologically realistic simulated data. On the basis of them, a lot of things can be worked out. Regarding the simulation, it would be desirable to have neurons that elicit more differently shaped action potential waveforms. There should especially be an obvious difference between projection neurons and interneurons. This could be achieved by either applying different cell models or by modifying the channel kinetics within the current models. Especially potassium channel kinetics have a strong influence on spike waveforms. The small horizon of the simulated electrodes also deserves further study. It has to be investigated, whether the Nunez equation is really suitable for the simulation of extracellular potentials or if it should be substituted by another equation to be found. Once the simulated electrodes have a bigger horizon, it might be interesting to increase the network size. This might in turn lead to a demand for parallelization or for simplified models that are still able to elicit complex firing pattern.

Concerning the evaluation of spike detection methods, the test of some approaches is still pending, especially for multichannel data. One should not stop with spike detection, however, but should continue with the performance evaluation of spike sorting methods on one and the same data set comprising, among other signals, biologically realistic simulated data. Reliable gold standards have to be established. Or are you, confronted with non-optimal separation results, satisfied by agreeing with Wheeler [24, page 74]:"Thus it is of great practical importance to remain flexible in the use of the results of spike sorting"?

Appendix A

Comic

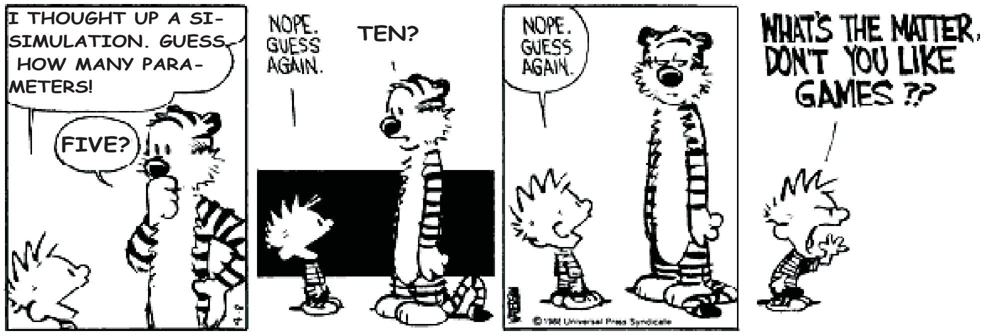
I THOUGHT UP A SI-SIMULATION. GUESS HOW MANY PARA-METERS!
FIVE?
NOPE. GUESS AGAIN.
TEN?
NOPE. GUESS AGAIN.
WHAT'S THE MATTER, DON'T YOU LIKE GAMES ?!?
©1988 Universal Press Syndicate

Figure A.1: Don't you like games? For comic with original text see below.

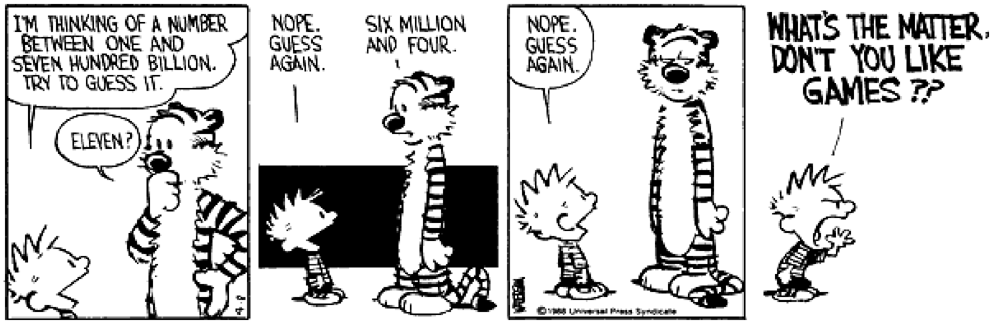
I'M THINKING OF A NUMBER BETWEEN ONE AND SEVEN HUNDRED BILLION. TRY TO GUESS IT.
ELEVEN?
NOPE. GUESS AGAIN.
SIX MILLION AND FOUR.
NOPE. GUESS AGAIN.
WHAT'S THE MATTER, DON'T YOU LIKE GAMES ?!?
©1988 Universal Press Syndicate

Figure A.2: Calvin and Hobbes comic with original text by Bill Watterson. Taken from http://www.ucomics.com/calvinandhobbes/.

Appendix B

Parameter settings and performance results

Methods and parameters		# detected spikes	# correct	# fp	# fn
pt					
threshold	# samples above threshold				
0.80	1	184	90	94	5
0.80	2	90	80	10	15
0.80	3	80	79	1	16
0.90	3	79	78	1	17
0.75	3	80	79	1	16
0.70	3	80	79	1	16
0.60	3	85	81	4	14
pt + ppa					
threshold pt	threshold ppa				
0.80	1.60	80	79	1	16
0.90	1.80	79	78	1	17
0.70	1.40	80	79	1	16
0.60	1.20	84	81	3	14
pt + ppa + pp-time					
threshold pt	maximum width				
0.8	7	79	79	0	16

ppa-window					
threshold	window				
ppa	length				
1.40	0.30	227	76	151	19
1.40	0.50	233	88	145	7
1.60	0.25	158	49	109	46
1.60	0.40	201	78	123	17
1.60	0.50	194	81	113	14
1.60	0.60	196	81	115	14
1.60	0.70	196	81	115	14
1.60	1.00	193	84	109	11
1.60	1.50	188	80	108	15
2.00	0.50	157	64	93	31
4.00	1.00	111	60	51	35
sec_dev + pt					
threshold	# samples				
pt	above threshold				
0.30	3	44	44	0	51
0.30	2	74	74	0	21
0.20	2	82	79	3	16
0.25	2	78	78	0	17
energy + pt					
threshold	window				
pt	length				
50	1.00	82	41	41	54
20	1.00	150	76	74	19
20	0.75	142	76	66	19
20	0.50	124	77	47	18
dwt-denoising + pt					
threshold	# samples				
pt	above threshold				
1.00	3	76	75	1	20
0.50	3	99	82	17	13

Parameters in the case of experimental data with a SNR of 25:1 (95 spikes). Thresholds are given in mV, the pp-time in numbers of samples, window lengthes in msec.

Methods and parameters		# detected spikes	# correct	# fp	# fn
pt					
threshold	# samples above threshold				
0.35	3	247	156	91	24
0.40	3	212	150	62	30
0.45	3	170	131	39	49
0.50	3	143	115	28	65
0.60	3	87	75	12	105
0.70	3	51	46	5	134
pt + ppa					
threshold pt	threshold ppa				
0.40	0.80	277	161	116	19
0.45	0.90	234	159	75	21
0.40	0.80	208	150	58	30
0.45	0.90	167	131	36	49
0.60	1.20	85	75	10	105
0.70	1.40	51	46	5	134
pt + ppa + pp-time					
threshold pt	maximum width				
0.40	6	122	96	26	84
0.40	7	159	119	40	61
0.40	8	178	131	47	49
0.45	8	145	114	31	66
0.40	9	192	141	51	39
0.40	10	200	146	54	34
0.40	11	200	146	54	34
ppa-window					
threshold ppa	window length				
1.0	1.0	354	113	240	67
1.2	1.0	245	135	109	45
1.2	0.5	226	136	89	44
1.2	0.4	214	132	82	48
1.3	0.5	197	127	69	53

sec_dev + pt					
threshold pt	# samples above threshold				
0.07	3	144	121	23	59
energy + pt					
threshold pt	window length				
3	1.00	292	141	151	39
3	0.75	215	140	74	40
3	0.50	151	114	37	66
dwt-denoising + pt					
threshold pt	# samples above threshold				
0.40	3	107	74	33	106
0.30	3	184	112	72	68

Parameters in the case of experimental data with a SNR of 9:1 (180 spikes). Thresholds are given in mV, the pp-time in numbers of samples, window lengthes in msec. In the case of pt+ppa, there are two times the same parameter settings. The above two results are achieved, if 2 consecutive sample points are asked to be above threshold. The other results are achieved, if 3 consecutive sample points are asked to be above threshold.

Methods and parameters		# detected spikes	# correct	# fp	# fn
pt					
threshold	# samples above threshold				
0.01	3	67	49	18	21
		67	55	12	57
0.008	3	88	53	35	17
		88	62	26	50
0.008	2	105	54	51	16
		105	66	39	46
pt + ppa					
threshold pt	threshold ppa				
0.008	0.016	86	51	35	19
		86	60	26	52
0.007	0.014	103	53	50	17
		103	66	37	46
0.007	0.014	122	54	68	16
		122	70	52	42
pt + ppa + pp-time					
threshold pt	maximum width				
0.007	25	102	53	49	17
		102	66	36	46
0.007	25	122	54	68	16
		122	69	53	43
0.008	25	96	54	42	16
		96	65	31	47
ppa-window					
threshold ppa	window length				
0.020	0.5	134	51	82	19
		134	63	70	49
0.016	0.5	193	53	138	17
		193	78	113	34
0.016	1.0	190	54	134	16
		190	75	113	37

sec_dev + pt					
threshold pt	# samples above threshold				
0.001	2	6	86	64	
		14	78	98	
energy + pt					
threshold pt	window length				
0.002	1.00	8	122	62	
		14	116	98	
0.002	0.75	3	114	67	
		11	106	101	
dwt-denoising + pt					
threshold pt	# samples above threshold				
0.010	2	67	50	17	20
		67	57	10	55
0.008	2	84	56	27	14
		84	66	17	46
0.007	2	105	57	47	13
		105	70	34	42

Parameters in the case of simulated data with a SNR of 21:1. Thresholds are given in volt, the pp-time in numbers of samples, window lengthes in msec. Lines, in which the parameters are given, refer to comparisons with pyramidal cells 40 and 46 (70 spikes) only. The others to comparisons with pyramidal cells 40, 46 and 27 (112 spikes).

Bibliography

[1] Moshe Abeles and Moise H. Goldstein. Multispike train analysis. *Proceedings of the IEEE*, 65:762–773, 1977.

[2] Amir F. Atiya. Recognition of multiunit neural signals. *IEEE Transactions on Biomedical Engineering*, 39(7):723–729, 1992.

[3] Hagai Bergman and Mahlon R. DeLong. A personal computer-based spike detector and sorter: implementation and evaluation. *Journal of Neuroscience Methods*, 41:187–197, 1992.

[4] Steven M. Bierer and David J. Anderson. Multi-channel spike detection and sorting using an array processing technique. *Neurocomputing*, 26–27:947–956, 1999.

[5] James M. Bower and David Beeman. *The Book of GENESIS*. Springer-Verlag, second edition, 1998. http://www.genesis-sim.org/GENESIS/.

[6] Rishi Chandra and Lance M. Optican. Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network. *IEEE Transactions on Biomedical Engineering*, 44:403–412, 1997.

[7] Karl Daumer and Renata Hainz. *Verhaltensbiologie: Ethologie, Kybernetik und Neurophysiologie.* Bayerischer Schulbuch-Verlag, sixth edition, 1991.

[8] Gregory J. Dinning and Arthur C. Sanderson. Real-time classification of multiunit neural signals using reduced feature sets. *IEEE Transactions on Biomedical Engineering*, 28:804–811, 1981.

[9] F. Gabbiani, J. Midtgaard, et al. Synaptic integration in a model of cerebellar granule cells. *Journal of Neurophysiology*, 72(2):999–1009, 1994.

[10] Shai N. Gozani and John P. Miller. Optimal discrimination and classification of neuronal action potential waveforms from multiunit, multichannel record-

ings using software-based linear filters. *IEEE Transactions on Biomedical Engineering*, 41:358–372, 1994.

[11] Charles M. Gray, Pedro E. Maldonado, et al. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *Journal of Neuroscience Methods*, 63:43–54, 1995.

[12] M. Hines. Efficient computation of branched nerve equations. *Int. J. Bio-Med. Comp.*, 15:69–76, 1984.

[13] J.I. Hubbard, R. Llinas, and D.M.J. Quastel. *Electrophysiological Analysis of Synaptic Transmission. Monographs of the Physiological Society, 19.* London, 1969.

[14] Hidekazu Kaneko, Shinya S. Suzuki, et al. Multineuronal spike classification based on multisite electrode recording, whole-waveform analysis, and hierarchical clustering. *IEEE Transactions on Biomedical Engineering*, 46:280–290, 1999.

[15] Juan Carlos Letelier and Pamela P. Weber. Spike sorting based on discrete wavelet transform coefficients. *Journal of Neuroscience Methods*, 101:93–106, 2000.

[16] Michael S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computational Neural Systems*, 9:R53–R78, 1998.

[17] Z. F. Mainen and T. J. Sejnowski. Influence of dendritic structure on firing pattern in model neocortical neurons. *Nature*, 382:363–366, 1996.

[18] Kerstin M. L. Menne, Andre Folkers, Thomas Malina, Reinoud Maex, and Ulrich G. Hofmann. Test of spike sorting algorithms on the basis of simulated network data. In *Neurocomputing, Proceedings of the CNS 2001, Monterey.* to be published.

[19] Kerstin M. L. Menne, Reinoud Maex, and Ulrich G. Hofmann. Extracellular potential data generated in a network simulation. In *The Neurosciences at the Turn of the Century, Proceedings of the 4th Meeting of the German Neuroscience Society 2001*, volume 1, page 602. Georg Thieme Verlag, 2001.

[20] Elliot D. Menschik and Leif H. Finkel. Neuromodulatory control of hippocampal function: Towards a model of alzheimer's disease. *Artificial Intelligence in Medicine*, 13(1-2):99–121, 1998.

[21] R. Miles. Variation in strength of inhibitory synapses in the CA3 region of guinea-pig hippocampus in vitro. *Journal of Physiology-London*, 431:659–676, 1990.

[22] H. Nakatani, T. Watanabe, and N. Hoshimiya. Detection of nerve action potentials under low signal-to-noise ratio condition. *IEEE Transactions on Biomedical Engineering*, 48(8):845–849, 2001.

[23] John G. Nicholls, A. Robert Martin, and Bruce G. Wallace. *From Neuron to Brain.* Sinauer Associates, Inc., third edition, 1992.

[24] Miguel A.L. Nicolelis. *Methods for Neural Ensemble Recordings.* CRC Press LLC, 1999.

[25] P.L. Nunez. *Electric Fields of the Brain: The Neurophysics of EEG.* Oxford University Press, Oxford, first edition, 1981.

[26] R. J. O'Connell, W.A. Kocsis, and R.L. Schoenfeld. Minicomputer identification and time of nerve impulses mixed in a single recording channel. *Proceedings of the IEEE*, 61:1615–1621, 1973.

[27] Karim G. Oweiss and David J. Anderson. Noise reduction in multichannel neural recordings using a new array wavelet denoising algorithm. *Neurocomputing*, 38-40:1687–1693, 2001.

[28] W. Rall. Electrophysiology of a dendritic neuron model. *Biophys. J.*, 2:145–167, 1962.

[29] Sergei P. Rebrik, Brian D. Wright, et al. Cross channel correlations in tetrode recordings: implications for spike-sorting. *Neurocomputing*, 26–27:1033–1038, 1999.

[30] Edward M. Schmidt. Computer separation of multi-unit neuroelectric data: a review. *Journal of Neuroscience Methods*, 12:95–111, 1984.

[31] Gordon M. Shepherd. *The Synaptic Organization of the Brain.* Oxford University Press, fourth edition, 1998.

[32] H. C. Suarez, C. Koch, et al. Modeling direction selectivity of simple cells in striate visual cortex within the framework of the canonical microcircuit. *Journal of Neuroscience*, 15(10):6700–19, 1995.

[33] R.D. Traub, Miles A. Whittington, et al. On the mechanism of the $\gamma \to \beta$ frequency shift in neuronal oscillations induced in rat hippocampal slices by tetanic stimulation. *The Journal of Neuroscience*, 19(3):1088–1105, 1999.

[34] Roger D. Traub, Simon B. Colling, and John G.R. Jefferys. Cellular mechanisms of 4-aminopyridine-induced synchronized after-discharges in the rat hippocampal slice. *Journal of Physiology*, 489.1:127–140, 1995.

[35] Roger D. Traub, John G. R. Jefferys, et al. A branching dendritic model of a rodent CA3 pyramidal neuron. *Journal of Physiology*, 481.1:79–95, 1994.

[36] Roger D. Traub and Richard Miles. Pyramidal cell-to-inhibitory cell spike transduction explicable by active dendritic conductances in inhibitory cell. *Journal of Computational Neuroscience*, 2:291–298, 1995.

[37] Shiro Usui and Itzhak Amidror. Digital low-pass differentiation for biological signal processing. *IEEE Transactions on Biomedical Engineering*, 29:686–693, 1982.

[38] S. D. Van Hooser, U. G. Hofmann, et al. Relationship between field potentials and spike activity in rat S1: multi-site cortical recordings and simulations. *Neurocomputing*, 32–33:591–596, 2000.

[39] Bruce C. Wheeler and William J. Heetderks. A comparison of techniques for classification of multiple neural signals. *IEEE Transactions on Biomedical Engineering*, 29:752–759, 1982.

[40] George Zouridakis and David C. Tam. Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering. *Computer Methods and Programs in Biomedicine*, 61:91–98, 2000.