AN IN SILICO DESCRIPTION OF AN ALZHEIMER NETWORK

Jesper Lansink Rotgerink

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE
CHAIR: APPLIED ANALYSIS AND MATHEMATICAL PHYSICS

EXAMINATION COMMITTEE
prof.dr. Stephan A. van Gils
dr. Antonios Zagaris
dr. Kerensa Broersen

UNIVERSITY OF TWENTE.

JANUARY 17, 2013

Abstract

Alzheimer's disease (AD) is the most well known form of dementia. Already in 1907 Alois Alzheimer published a first paper about the symptoms of this disease, of which we know that it afflicts one in three people above 85. By now it is broadly accepted that indeed the loss of short-term memory is among the first symptoms of AD, followed by behavioural changes and difficulties with motor activities. In the end, Alzheimer is a terminal disease, and with its huge direct impact on caregivers and financial impact on society it is an essential topic in several research fields. Additionally, by the increasing average age of current society it only becomes more urgent.

Among these fields of research, computational neuroscience can be helpful as a first step in deciphering phenomena that occur with the appearance of Alzheimer's disease. For many years after the first descriptions about AD there was no clue about a possible cause. Nowadays it is known that the appearance of the disease is mediated by a protein called Amyloid beta (Aβ). The exact role of Aβ remains unknown, though a prominent hypothesis is that it affects several presynaptic processes, thereby suppressing neuron efficiency.

We model networks containing up to 600 neurons as computational networks in which the vertices represent a neuron. The internal dynamics of the neuron are governed by the so-called Izhikevich model. The connections represent the synapses between neurons. In this way, the behaviour of neural networks can be simulated by using computational mathematics.

In this thesis a computational framework designed to map experimental data to synaptic parameters is constructed. In this way, having *in vitro* measurements of Aβ-treated neuronal networks and untreated control networks, the existence of significant differences in the synaptic properties of the two types of networks can be systematically investigated.

Taking our cues from previous research by Philip Hahn & Cameron McIntyre [2010], we attempt to use a Gauss-Newton least mean square error optimisation. The computational framework then extracts the first two statistical moments, Mean Firing Rate (MFR) and Coefficient of Variation (CV), from measurements. This Gauss-Newton method subsequently searches for synaptic properties yielding networks with comparable MFR and CV.

We conclude that this Gauss-Newton method should be handled with great care. In particular, we find that, for our particular model the algorithm either diverges or converges at a poor rate. Besides this statement and giving an overview on how varying the synaptic parameters affects the behaviour of neural networks, this thesis gives elaborate suggestions for future computational research on Alzheimer's disease.

Contents

Chapter 1

Introduction

Alzheimer's disease

An unfortunate nevertheless endlessly returning diagnosis is *Alzheimer's disease* (AD). It is the most common form of dementia and mostly afflicting elderly people. In fact, estimations are that every one in three people above the age of 85 suffers from Alzheimer, which by the increasing average age of our society could become significantly more [24]. Moreover, the financial impact of AD on the world is immense, since the payments for health-care, long-term care and hospice in the United States of America only, were estimated to be 200 billion dollars in 2012 [22]. This amounts to approximately a million times the minimum wage of an American citizen[1] or equivalently 640 dollars per person when divided over all the inhabitants of the USA[2].

The disease, of which the observable effects on the physiological state of a patient were already extensively described by Alois Alzheimer back in 1907 [1], has many forms though is always characterised by short-term memory loss at an early stage [22]. The most common early symptoms are difficulties with remembering names and recent events, as well as apathy and depression. Later stages of the disease include behavioural changes and trouble with both speaking and motor activities such as swallowing or walking. In the severe and advanced cases of AD, patients will also need help with activities of daily life, placing high responsibilities and pressure on caregivers. In the end, Alzheimer's disease is fatal, leading to the patient's death.

Like all dementias, AD is a brain disease. It is caused by irreversible damage to brain cells. The brain is the center of all activity in the body, and information is transported from and to the brain continuously. A significant part of the brain cells are *neurons* and conducting information through the complete body takes place through these neurons. We will take a brief look

---

[1]Minimum wages as published by the European Commission Eurostat on the 2nd of April 2012, http://epp.eurostat.ec.europa.eu/portal/page/portal/labour_market/earnings.
[2]US population number as published in the 2010 census http://www.census.gov.
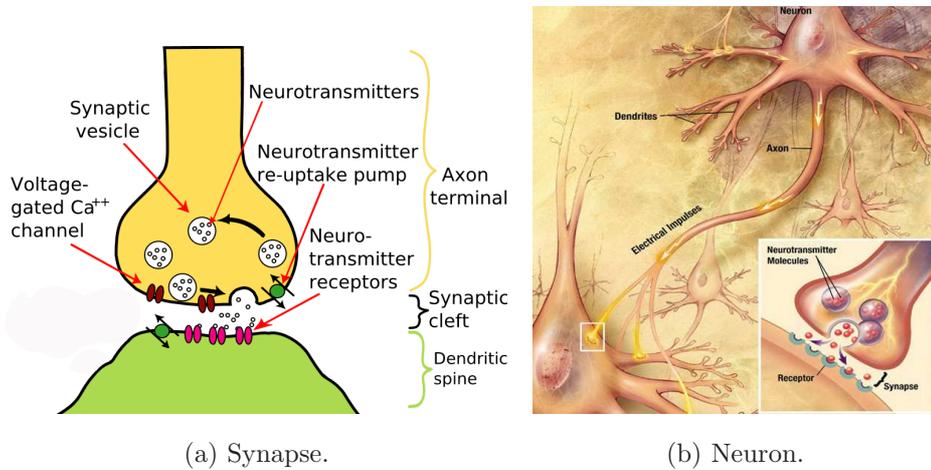
(a) Synapse.

(b) Neuron.

Figure 1.1: a) Schematic picture of all important components of the chemical synapse including pre- and postsynaptic terminals. b) Schematic picture of a neuron connecting to another neuron via its axon and synapse. Both figures are freely available at http://en.wikipedia.org/wiki/Chemical_synapse

at these neurons in the next section.

Neurons

Neurons are electrically excitable cells that transport signals by using chemicals and electrical pulses. That way information is conducted from for instance the brain to muscles. The neurons consist of a cell body, *dendrites* and an *axon*. Signals are passed on from the axon of a neuron to the dendrites of another neuron (figure 1.1b), by which the neuron's dendrites can be interpreted as feelers and the axon as the transporter to the other neuron.

The signals that are being passed on by neurons travel as electrical pulses through the dendrites and axons. Such a pulse is actually a difference in voltage between intracellular and extracellular medium. That voltage difference is propagated along the membranes of the neuron; this is then called an *action potential*.

Propagating such a voltage difference is achieved by closing and opening several ion channels [11]. Axons have a certain rest potential, at which the neuron operates more or less at dynamic equilibrium, which for most neurons is approximately -70mV. If for instance by the arrival of an action potential a certain threshold value of the voltage difference between intra- and extracellular medium is reached, the ion channels open allowing ions to flow through the membrane. These channels are called *voltage gated ion channels*, because depending on the value of the potential, they open or close. For instance, if in the axon a voltage difference of -55mV is reached,

sodium channels open, which causes an inward ion current of sodium. As a consequence the voltage inside the neuron increases even more, causing other ion channels to open at certain other threshold values. In this way the action potential is sustained and propagated.

When an action potential reaches an end of an axon, it might be passed on to another neuron. If so, the site where this happens is situated around a small gap between axon and dendrite, called the *synaptic cleft*. The brain contains both chemical and electrical synapses, yet in this thesis we only deal with the chemical ones. The synaptic cleft connects two neurons, and by connecting more and more neurons, chains of subsequent neurons might form connections between different parts of the brain. The brain itself contains billions of neurons where each neuron can be part of numerous chains. All these chains together form one big network of neurons. As a consequence the brain can be interpreted as one big *neural network*, which in turn is connected to, and can thereby communicate with neurons in for instance the spine or arm muscles. If two neurons share a synaptic cleft, we define them to be each other's neighbours and a certain chain can be considered as a path through this network.

Together with the axon (the presynaptic terminal) and the dendrite (the postsynaptic terminal) the synaptic cleft conducts the action potential to another neuron. This is a process built up out of many steps which we outline directly below.

Synaptic cleft

With the conduction of electrical pulses, synapses are essential; they form the connections between neurons and determine to which neurons a pulse can be passed on (figure 1.1a). This conductance of an action potential is mediated by several synaptic compartments and chemical substances, like the transmitter chemical called *neurotransmitter*, which stimulates *NMDA receptors* and is contained in the *synaptic vesicles*, whereas these last protein compartments are activated by calcium flowing in through *voltage dependent calcium channels*. All are explained further in this paragraph. Apart from the ion channels being common to the axon, the presynaptic terminal also contains voltage dependent calcium channels (VDCCs). By opening these the presynaptic part is able to absorb a large amount of calcium ions, which are used to activate the so-called synaptic vesicles situated in that same area. These vesicles are small compartments made out of proteins and they contain neurotransmitter, the chemical that transfers the signal through the synaptic cleft and stimulates the postsynaptic terminal. This stimulation is achieved via receptors on the postsynaptic membrane, among which are NMDA receptors. These bind the neurotransmitter and a necessary co-agonist (mostly glycine), and based upon both bindings it allows calcium (Ca^{2+}), potassium (K^+) and sodium (Na^+) ions to flow through the chan-

3

nel[3].

The process of transferring the action potential from the axon of a neuron to the dendrites of another neuron takes place following these steps:

- Upon arrival of an action potential, the VDCCs open if a certain threshold voltage is reached, causing calcium ions to flow inside [25].

- Vesicles that are docked near the presynaptic surface facing the synaptic cleft are activated by receptors on these vesicles in reaction to the influx of calcium, which makes them fuse with the cell membrane [15].

- By this fusion the neurotransmitter is released into the cleft [3].

- At the postsynaptic membrane, these neurotransmitter molecules, together with the co-agonist bind to the NMDA receptors.

- This binding causes ion channels to open, letting Ca^{2+}, K^+ and Na^+ ions to flow into the postsynaptic neuron.

- This inflow of ions is the trigger for a new action potential at the postsynaptic terminal and moreover for the whole process to start over at that postsynaptic neuron.

When as a result of this whole process an action potential is passed on to the postsynaptic neuron, we say that the neuron *fires* onto another neuron, or that a *spike* occurs.

We regard the described construction of presynaptic terminal, synaptic cleft and postsynaptic terminal as one individual synapse or connection. From the biophysical situation it is known that apart from having one connection, two neurons can actually share more individual synapses. The amount of connections between the presynaptic and postsynaptic neuron is actually a measure of *synaptic strength*, which is defined as how strong a postsynaptic neuron reacts to a presynaptic action potential. By changing this synaptic strength as response to external stimuli, the brain can carry out tasks of learning and memorizing.

Neuronal networks and memory

Although the process of forming memory is a very complex one and not every detail is known yet, several studies have led to the understanding of the overall structure and the unraveling of more and more details [13]. As mentioned before one of the earliest symptoms of AD is memory loss and thus having an understanding of this phenomenon is essential to become familiar with the processes of Alzheimer's disease.

---

[3]The interested reader is referred to [16], [23].

In principle, learning and memory involves synaptic modification. More specifically, when certain events happen frequently, the neurons are able to change their synaptic strengths and thereby their efficiency to repeat the same event. This is for instance achieved by creating more individual synapses. The events that trigger these changes basically come down to regular firing of neurons. The ability of neurons to increase their synaptic strengths as a reaction to repetitive firing patterns is linked with information storage in the brains and is called *synaptic plasticity*. When during high frequency firing the efficiency of a synapse is increased this is seen as the brain storing information: memorising or learning. This last effect is called *long-term potentiation* (LTP) [2]. Long-term potentiation can last from minutes up to years and is among others indeed realised by creating more individual synapses.

Besides long-term plasticity, there is also a short-term version [13], which mostly lasts up to a few seconds. This process modifies existing synapses and processes, by for example temporarily increasing the amount of synaptic vesicles and neurotransmitter. Next to enhancement of the synaptic strength, also a decay in efficiency is possible, called either *short-term* or *long-term depression* (STD or LTD).

Since memory is reflected in the strengths of synapses that the neurons share, it is actually a property of the connections between neurons. Therefore it is useful to look at networks of multiple neurons. These networks can mathematically be seen as graphs containing vertices and edges. The vertices represent the cell bodies of the neurons, whereas the edges represent synaptic connections between neurons. These graphs can satisfy several properties [17]. They can be completely known, in the sense that the positions of nodes and the connections between them are fixed. Biophysically, the complete structure of a neuronal network is unknown. That is the reason that we use computational networks of which a priori the positions of neurons and existence of connections is not known. Still, they can be created in such a way that the network satisfies some general conditions like density. Positions of the neurons will be determined randomly and also connections between neurons will be made with a certain probability. The synaptic strength can then be realised by placing weights on the edges, determining how well nodes can communicate. From a biophysical point of view, the chemical synapses are directional: they only fire from presynaptic to postsynaptic neuron. That is why the edges in our graphs are directional as well.

In this way, complete neuronal networks such as entire regions of the brain, can be represented using graphs. Consequently the memory will then be a property of those graphs, especially of the connections. Since the loss of short-term memory is the first symptom, the properties of those connections will be our main concern in investigating the phenomenology of Alzheimer's disease.

The cause of Alzheimer

An exact mechanism of the arising of Alzheimer's disease is yet to be found, though a strong presumption about its cause lies with the protein *Amyloid β* (Aβ) [5]. One of the arguments supporting this candidacy is the location of the gene for the Amyloid β precursor protein (APP), a protein that can form Aβ. Genes hold the information for building cells and also proteins. The location of the gene that produces APP and is thus indirectly responsible for creating Aβ, is located on chromosome 21. Patients with Down syndrome carry three instead of two of those chromosomes 21, which means that the production of Aβ is increased [8]. One of the main features of Down syndrome is that patients almost universally exhibit Alzheimer's disease at an age of 40. This thus suggests a strong link between AD and Aβ.

The hypothesis that Aβ causes AD will be brought up for discussion in this thesis. It is known that Aβ attacks the synapse, yet so far it is unclear in what ways exactly. Biophysically, there are some hypotheses about this. It is thought that Aβ dysregulates some presynaptic processes; it harms calcium channels [19], synaptic vesicles and presumably also mitochondria [26]. Mitochondria are the energy manufacturers of a living organism and in the presynaptic area they also regulate the calcium metabolism. Since calcium activates the vesicles to release their neurotransmitter into the synaptic cleft, all three are important to synaptic transmission. Thus, by harming these, Aβ in the end decreases the ability of a neuron to pass on information. This causes decrease in synaptic strength, which can ultimately lead to the effective death of a synaptic connection. Eventually, the death of several synapses might even cause the death of neurons. And exactly that is where memory loss caused by Aβ as a first symptom of Alzheimer's disease becomes noticeable.

An in silico description of an Alzheimer network

Our focus is on a mathematical approach to gaining insight in effects of Aβ on neuronal networks, in which especially the synapses are of interest. Several mathematical models describing the propagation of action potentials traveling down an axon exist. A broadly accepted mathematical model for the behaviour of the axon regarding potentials and ion currents, is the *Hodgkin-Huxley (HH) model* [7], which is a four-dimensional nonlinear system of first order ordinary differential equations. Hodgkin and Huxley determined that an axon can phenomenologically be modeled as a *passive* yet nonlinear electrical circuit, thereby using Kirchhoff's law for electrical circuits to obtain the corresponding differential equations. Here, passivity is a property of the circuit, meaning that it only contains components that consume though not produce energy, or components that are unable to gain power. In the axon, the membrane acts as a capacitor and the other compo-

nents that can be modeled by electrical components are ion channels. None of them consumes energy, so by using Kirchhoff's law Hodgkin and Huxley were able to set up equations that link the change in voltage difference over time, to currents caused by ion streams. This empirical model was originally obtained for the axon of a giant squid by a series of experiments, in Hodgkin and Huxley's summarising paper [7] by the electrical circuit. Even so, they suggested by means of the experimental results that the behaviour of the voltage gated ion channels can be represented by nonlinear electrical conductances.

A similar model, also based on Kirchhoff's law, exists for a complete neuron, including synapses. This model is accurate, though quite complex too. A computationally more efficient model is the *Izhikevich neuron*, in which several types of neuronal behaviour can be reproduced by tweaking four parameters and a possible external current [9].

Mathematics can be of great value in obtaining a first insight into the phenomena occurring when Aβ comes into play, by estimating which parameters in existing mathematical models are changed when adding Aβ. To do this, data is at our disposal originating from *cultured networks*, in which cell bodies of neurons are isolated and used to grow a new network on which measurements can be performed. This data is created with *Microelectrode Arrays* (MEA) and previous results obtained with this data are published in a biological work by Kuperstein *et al* [12]. The MEA data contains measurements of networks with and without Aβ injected. Kuperstein *et al* published results about the effects of different forms of Aβ.

The aim of our project is to develop a computational framework which maps experimental data to synaptic parameters. This can be used in investigating the existence of significant differences in the values of these parameters in the Aβ-treated and untreated control networks. A computational network is built, and similar to an article by Hahn and McIntyre [4], we try to use a least mean square error algorithm, namely Gauss-Newton, to tune synaptic paramaters in such a way that they simulate the Aβ data on one hand and the control data on the other. This fitting is done by means of statistical moments extracted from the dataset, referred to as the *Mean Firing Rate* and *Coefficient of Variation*.

Using mathematical models has the advantage that, for different configurations or types of networks one does not have to create new networks and perform new measurements. One can use a computational network in which a change in some parameters can give new results. Cultured networks take several weeks to grow before they show activity and then have a fixed cell density and other network properties. A computational network however, after the correct software has been created, can be built in the order of a second. Thus a slight change in the structure or properties of a network is made relatively quick and results in behaviour of the network might be computed in the same order of time as well. That is the reason

for *computational neuroscience* to be a big field of research; it has been used in for instance research related to Parkinson's disease. The essence of applied mathematics lies in here, since one mathematical model might be usable in several applications. In our case we are extending these fields with Alzheimer's disease.

These computational networks make for instance changing synaptic parameters and computing the corresponding network behaviour possible, from which we developped the idea that without performing new measurements, we might be able to estimate these parameters. The main contribution of this thesis is an investigation of the suitability of the Gauss-Newton optimisation procedure in the desired computational framework. Unfortunately, after analysing several components of the method, like error functions and the time horizon of simulations, this leads to conclusions that the suggested Gauss-Newton method is not the finest choice. Yet several observations throughout the thesis lead to numerous suggestions for further research regarding the in silico discription of Alzheimer's disease. Since, by the unsuccessfulness of the attempt with Gauss-Newton, the need for the described computational framework was not fulfilled we did not reach the stage of handling the data. Moreover, analysing the dataset it seemed that it was noisy, containing several bust-out electrodes. Besides that, the overall activity was weak whereas the activity of mutual electrodes was of strongly varying levels. The above, combined with the absence of an understanding how the experimental device exactly works, are several concerns that lead to not discussing the data in this thesis.

However, arguments above indicate that future development of the described computational framework, keeping in mind suggestions given in the last chapter of this thesis, can definitely be useful in the research regarding the phenomena that occur when $A\beta$ is injected into a neural network. Especially since it costs respectively little time to change the parameters and resimulate the network, compared with doing this with cultured networks, the effectiveness of the framework should stimulate its development.

Finally, as a contribution to understanding the behaviour of computational networks, we investigate the changes in both Mean Firing Rates and Coefficient of Variation, as well as single neuron behaviour in a configuration where MEAs are left out of consideration. Moreover, we attempt to use the Gauss-Newton procedure in this case as well. Since the need for the computational framework mentioned above will not be fulfilled, we also give an elaborate view on possible future improvements of this computational research into Alzheimer.

Chapter 2

Modeling

In this chapter every mathematical model that is used in this project is introduced. Moreover, notations used throughout the thesis are introduced. Below already a table of symbols with their corresponding quantity and units is presented.

Symbol	Quantity	Units
MFR	Mean firing rate	Hz
C_V	Coefficient of variation	-
V	Membrane potential	mV
I	Current	mA
E_{ion}	Reversal potential of a certain ion	mV
g_i	Conductance of synapse i	$(\text{mm}^2\ \Omega)^{-1}$
w_i	Weight of a synapse i	mm^{-1}
τ_i	Time constant of synaptic decay of synapse i	ms
N	Number of neurons	-
$p_{con}(i,j)$	Probability of existence of a connection from neuron i to neuron j	-
$d_{i,j}$	Distance between neuron i and neuron j	mm
u	Membrane recovery variable	mV
dt	Numerical timestepper	ms
T	Time horizon of simulations	s
C	Membrane capacity	mF/mm^2
$R_{i,j}$	Distance from neuron i to electrode	mm

The mathematical modeling steps should lead to the development of a computational network that simulates the behaviour of the cultured networks best. These cultured networks contain for instance neurons and synapses, thus models that represent these components need to be introduced. Accurateness of these models always comes in a tradeoff with computational efficiency. To be able to simulate these cultured networks, these

mathematical models for the networks should be introduced, as well as the behaviour of neurons and synapses. Moreover it is important that we try to mimic the way that the measurements took place in the cultured network setup. Before introducing these models in the next sections, some details about cultured networks are brought up, since this is where our data that has to be estimated originates from.

2.1   Cultured neuronal networks

As with every field of science, the basis of biological research dealing with neuronal networks is measurements. To obtain a first clue about certain phenomena taking place in the human body, one may perform experiments and analyse their outcome. Based on these outcomes hypotheses and experiments to test these can be formulated. A big issue with measurements is that some of them cannot be done on humans directly. Especially the brain is a complicated and vulnerable tissue, on which risky experiments are controversial or outright prohibited. On the other hand, the brain takes an important role in the bulk of neuronal activities, even so with neuronal diseases.

A way to by-pass this problem is to create cultured networks. Biologists are able to dissociate neurons; axons and dendrites are detached and only the cell body of the neuron is left. After dissociation, the cell bodies are plated in an environment where the neurons are maintained and can actually grow new dendrites and axons. In that way after some weeks new connections between the neurons have been made and activity via synapses takes place. By using various measurement equipments, this activity can be recorded, allowing researchers to analyze the behaviour of the neurons under given conditions. These sort of measurements are called *in vitro*, as opposed to *in vivo*; the latter takes place in an actual living organism. To create such cultured networks, mostly neurons of a rat are used, in our case brain neurons belonging to embryonal mice. The exact region of the brain that the neurons originated from was the hippocampus.

In vitro research has been a popular biological tool for decades already, and it has proven to be of great value in research of different neural properties, for instance memory [18]. In our research too, ultimately a computational framework is developed that uses *in vitro* measurements to investigate influences of Aβ to neuronal networks.

2.1.1   Computational neuronal networks

Since we attempt to simulate the behaviour of a cultured network, we need to create a computational network that resembles the cultured one as much as possible. The cultured neurons form one big network, grown without any prefixed structure, so it cannot for instance be known between which

neurons a synapse is formed. This means that we are dealing with networks in which at least positions and connetivity have to be modeled with some probability distribution. Moreover it exists in two dimensions, since the neurons are plated onto a surface. As shown in the Introduction, networks can satisfy multiple properties, thus it has to be specified which of these our group of neurons and synapses satisfy.

Graphically, the neuronal network can be represented as a graph G containing a population of vertices V and edges E; $G(V, E)$. In here the points represent neurons, and just as in the biophysical situation, neurons can connect to each other via synapses, in this case corresponding to the edges. We create networks with a finite amount of vertices, N, in which the positions are drawn from a two-dimensional uniform distribution. This is done in such a way that the overall density of the neurons is 1000 cells/mm^2, as in the article of Kuperstein *et al* [12]. Biophysically, chemical synapses fire from one neuron onto another and not the other way around, thus we model this by having directed edges, thereby creating a directed graph.

Unfortunately, because of the probabilistics in the network, not much of its exact properties are known. However, from biophysical knowledge, it has been concluded that the probability of a connection appearing should depend on the distance between two neurons. The exact dependence, or even the average number of connections that one neuron makes, are unknown though. An often used distribution for the connection probability is the Gaussian distribution and in our case we chose parameters conformed to the findings of the specialised study by Holmgren *et al* [6]. In our computational network, an edge from a certain point i to another point j is created with a probability $p_{con}(i, j)$, which depends on their distance in the following way:

$$p_{con}(i, j) = 0.02 \ \mathrm{e}^{-40\mathrm{d(i,j)}^2}, \tag{2.1.1}$$

in which $d(i, j)$ is the Euclidean distance between neuron i and j, given in mm. As a consequence the number of connections is distributed in a way quite similar to the common binomial distribution, since now its success probability $p_{con}(i, j)$ is distance-dependent instead of constant. Binomial distributions are always characterised by this success probability and the number of experiments, in this case $N(N - 1)$, the total possible number of connections. As common in studies regarding graphs, we define the meaning of neighbours in our specific graph. We say that neuron j is a neighbour of neuron i if a connection exists running from neuron i to neuron j.

Summarising we make a graph representative of our cultured network with a certain given density of neurons, represented by vertices, randomly placed in a two-dimensional domain, such that the density is 1000 neurons/mm^2. Synapses are distributed by creating edges with a distance dependent probability which has a Gaussian distribution and is fitted to previous results by Holmgren *et al*. Thereby the geometry of our network is fixed. All other

properties are cell-based; for instance the vertices satisfy a certain mathematical model and also the edges are subject to some properties, treated in the next two subsections.

2.1.2 Neurons

2.1.2.1 Izhikevich model

Each neuron is a *dynamical system*, having the membrane potential of the neuron V (in mV) and the membrane recovery variable u (in mV) as their state. Their behaviour follows a set of differential equations. As mentioned in the Introduction, in 1952 Hodgkin and Huxley already found that the axon can be modeled as a nonlinear passive electrical system, thereby describing the behaviour of the potential in time by using Kirchhoff's law. This can be expanded to the whole neuron, including synapse, which gives an accurate mathematical model. As with most models though, accuracy comes at the cost of numerical efficiency. Having in our mind the fact that we would like to run a simulation of our network numerous times within our optimisation method, a less time-consuming model is preferable.

An accurate and moreover efficient mathematical model of a neuron is the Izhikevich model. It describes the dynamics of the neuron as a two-dimensional nonlinear dynamical system. The evolution of the state variables V and u is determined via:

$$\begin{cases} \dot{V} & = & \alpha V^2 + \beta V - u + I \\ \dot{u} & = & a(bV - u), \end{cases} \tag{2.1.2}$$

subject to the following reset equations:

$$V(t^-) \geq 30\text{mV} \Rightarrow \begin{cases} V(t^+) & = & c \\ u(t^+) & = & u(t^-) + d. \end{cases} \tag{2.1.3}$$

In these differential equations, α and β are fixed constants taken to be as in Izhikevich's article [9]. Parameters a and b are dimensionless, c and d have dimension mV and I is current given in mA. All parameters are explained below.

The recovery variable u describes the activation of outward currents. As explained in the Introduction, when an action potential arrives, at first ion channels open which allows positive ions to flow in. After some time though, an outward ion current initiates causing the potential to recover to its resting value again. The variable u mimics this process.

When the membrane potential V reaches a certain threshold value given by equation (2.1.3), this causes a reset in the state of the corresponding neuron. Biophysically, this means that the neuron fires onto another neuron and thus releases its neurotransmitters according to the cycle already explained in the Introduction. Thus these reset equations model the spiking

of a neuron and based upon different Izhikevich parameters a neuron can exhibit different spiking patterns.

Symbol	Description	Typical value
a	Timescale of the recovery variable	0.02
b	Sensitivity of the recovery variable to sub-threshold fluctuations of the membrane potential V	0.2
c	Post-spiking reset value of the membrane potential	-65 mV
d	Post-spiking reset value of the recovery variable	2 mV
I	Synaptic, injected or background (noise) current	-

Table 2.1: Izhikevich parameters

2.1.2.2 Biophysical interpretation of the Izhikevich parameters

Apart from the dependent variables u and V, there are five more unknowns in (2.1.2) and (2.1.3), which are the Izhikevich parameters. Different values of these parameters, regarded as properties of the neurons themselves, highly influence their spiking patterns.

The explanation of the five parameters of Izhikevich's model is given in short in table 2.1. Typical values are given to have an idea of the order of magnitude of all parameters (I varies from values like -65mA to 80mA, thus no typical value is given).
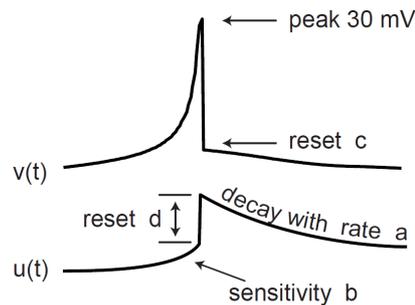

Figure 2.1: Interpretation of the Izhikevich parameters.

Parameters a and b describe the behaviour of the recovery variable, and can thereby for instance make a distinction between *resonator* or *integrator neurons*. Resonators react to a periodic input by creating a fluctuating

13

output with increasing amplitude. The reaction of an integrator is to add up all the inputs to reach higher membrane potentials, which resembles to integrating the signal. The parameter a is the timescale of the recovery mentioned above, whereas b describes the sensitivity of the recovery value with respect to the membrane potential V. When b is bigger, the membrane potential and its recovery value are coupled more strongly. The description in table 2.1 names sub-threshold fluctuations. These are fluctuations in V, in which V does not actually reach the spiking value. When b is bigger, the recovery value will follow these fluctuations more. Parameters c and d describe the reset behaviour of both variables, both given in mV. Figure 2.1[1] gives a good overview of these parameters.
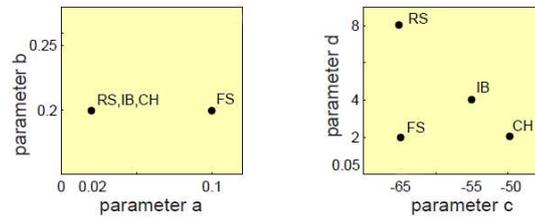
(a) Parameter planes.
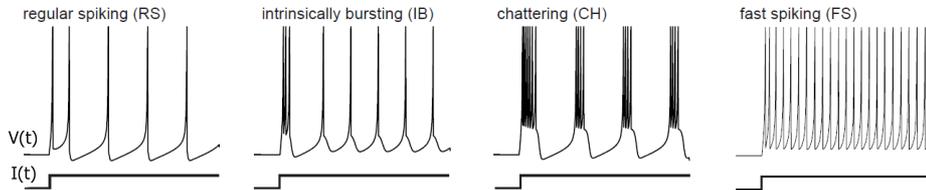
(b) Examples of spiking patterns.

Figure 2.2: b) Four examples of spiking patterns. a) Corresponding parameter values can be found in the parameter planes. NB: much more spiking patterns can be simulated by the Izhikevich neurons (see his website or articles [9] and [10]), for instance with changing b, which for these four cases is not necessary.

The last parameter is I, which can be decomposed according to:

$$I = I_0 + I_{syn},$$

in which I_{syn} is the synaptic current, caused by the firing of another neuron, and I_0 is an injected current and/or a constant background current (noise), which is biophysically also seen in neurons.

---

[1]Electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com.

Changing all these five parameters can lead to different neuronal behaviours. Four examples are shown in figure 2.2, in which the fifth parameter, the current I, is not shown though varies from 0 to 15.

In these spiking patterns an important difference occurs, and a distinction between two of those different patterns should be made. The first is a spiking pattern seen in for instance the first graph in figure 2.2. There we see spiking at an approximately constant frequency, called *regular spiking*. The third figure is far more irregular, since the spikes occur in clusters called bursts. This behaviour in which there are silent periods and periods of rapid spiking, is referred to as *bursting* or *chattering*. This distinction will play a role in describing the behaviour of the network in the next chapter.

2.1.2.3 Phase plane sketch

To have an idea about how Izhikevich's dynamical system behaves and where this behaviour comes from, we show some phase plane sketches for different parameter values. Firstly, in figure 2.3 a sketch is displayed in which the current I is equal to zero. Moreover, the other parameters are chosen as $a = 0.02$, $b = 0.2$, $c = -65$ and $d = 6$.

The phase plane sketches contain two nullclines, drawn in red and yellow and corresponding to V and u respectively. The first and second nullcline respectively satisfy:

$$\begin{aligned} u &= \alpha V^2 + \beta V + I \\ u &= bV. \end{aligned} \qquad (2.1.4)$$

The green solid line is the reset value c for the potential V, to which the V-value is set everytime it crosses $V = 30$mV (off range in the figure). All green arrows indicate the directions of the vector field and the blue lines represent some solution curves.

In the first, simplest sketch, we have two intersections of the nullclines. This implies that we have two equilibrium points, in this case one nodal sink and one saddle point. The shown orbits having initial point above the parabolic nullcline converges to the stable equilibrium point. The two other plotted trajectories move out of the window, though at $V = 30$mV the variables are reset to the values of $V = -65$mV and $u = u + 6$mV, leading the orbit after one spike to a position above the parabolic curve. Starting further below the undermost trajectory, after one reset we might still be below the parabola leading to another spike. Still, in the end the convergence holds everywhere; either we converge to the stable equilibrium point immediately, or by the choice of initial point the system spikes one or several times and approaches the nodal sink thereafter.

From equation (2.1.4) we derive that by changing parameters b and I, the shapes and position of the nullclines change. Changing c and d only affects the reset points after a spike, in which c determines the position of the vertical green line. By increasing b, the steepness of the u-nullcline
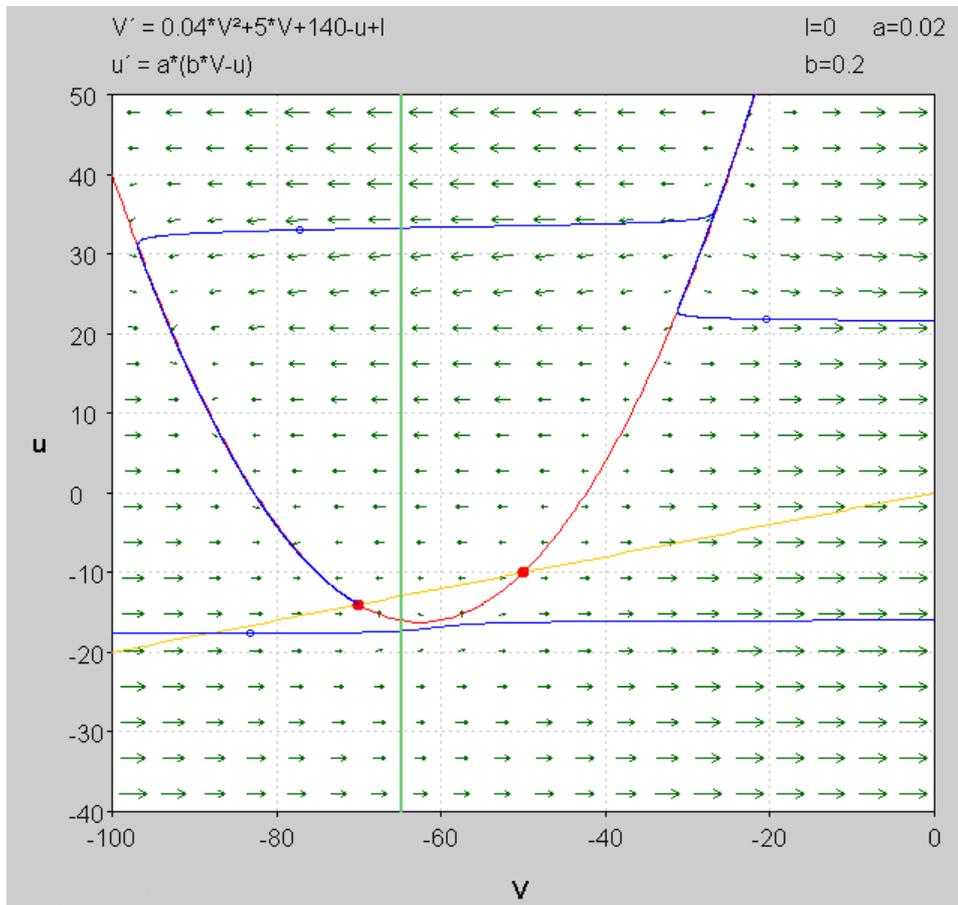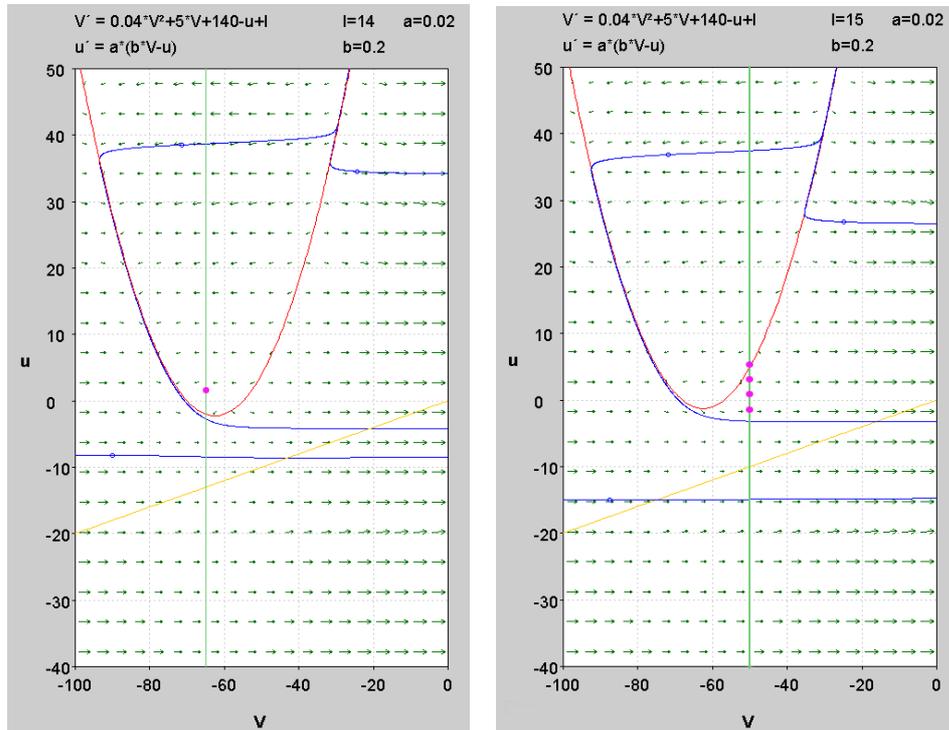
Figure 2.3: Phase plane sketch of Izhikevich dynamics with no current. Red, yellow and green solid lines correspond to respectively V-nullcline, u-nullcline and reset value for V. The blue curves display some trajectories and the red dots are equilibrium points.

increases. By setting I to positive values, the parabolic nullcline is shifted upwards. In this way, changing the parameters results in different vector fields and different trajectories.

This is why in figures 2.4a and 2.4b we obtain two different kinds of neuron behaviour, which we referred to as regular spiking and bursting. In figure 2.4a the current is set to $I = 14$mA and all other parameters are equivalent to figure 2.3. Looking at the trajectory starting above the parabolic nullcline, it crosses this parabola and then diverges in the V-direction. At $V = 30$mV we reach the reset condition and the solution curve is reset to the pink point on the green line. This again is situated above the parabolic line, repeating these events in a periodic way, creating a constant train of spikes.

(a) Regular spiking.   (b) Chattering.

Figure 2.4: Phase plane sketch of Izhikevich dynamics with positive current. Red, yellow and green solid lines correspond to respectively V-nullcline, u-nullcline and reset value for V. The blue curves display some trajectories and the pink dots on the reset line correspond to spikes. Parameters are such that we have a) regular spiking and b) chattering behaviour.

Changing I to 15mA and the reset conditions to $c = 50$mV and $d = 2$mV leads to different behaviour, of which a phase plane sketch is shown in figure 2.4b. Again, the orbits starting above the V-nullcline move to and along this parabola and then explode in the V-direction. This leads to a reset, although as can be seen by the pink dots in figure 2.4b one reset still situates the state below the red line. This is caused by both the increased current and the smaller reset increase in the u-direction. It means that again we diverge to $V = 30$mV, after which these events are repeated until the solution curve is finally positioned above the parabola. Here the whole above procedure starts over again, leading to a repetitive pattern of (here) four closely spaced spikes followed by a period characterised by the absence of spikes: bursting.

The remaining parameter a only influences the timescale in which the recovery variable u evolves. In figure 2.3 one already sees that V can be called the fast variable compared to u, since the arrows are almost all hor-

izontal, except near the *V*-nullcline. When increasing a, the influence of u to the vector field becomes bigger and the membrane potential becomes respectively less fast.

2.1.2.4 Excitatory versus inhibitory

Neurons can be subdivided into two classes; *excitatory* and *inhibitory*. When a neuron is excitatory, this means that when it fires, the potential of its neighbour increases. Thus this spike gives an input to another neuron which brings that membrane potential closer to its firing threshold.

Inhibitory neurons do exactly the opposite; when an action potential arrives at the synapse and a spike occurs, this will give a negative input to their neighbours, decreasing their membrane potential. This makes it more difficult for these neighbour neurons to reach their firing threshold.

Since we have dissociated neurons from a particular embryo and a particular region of the brain, the hippocampus, we know what kind of neurons the cultured network contains. In our specific network these are neurons that are excitatory. Based on this we can now try to find suitable parameter values for our Izhikevich model.

2.1.2.5 Our cultured neurons

To model the neurons correctly, we need values for all five Izhikevich parameters which make sure that the neurons exhibit behaviour that is comparable to our cultured hippocampal neurons.

In order to find these parameters one would like to have single neuron measurements, to see what kind of behaviour the cultured neurons show. This data is often created by *patch clamp techniques*. Here a micropipette is used, which has an extremely small tip. When it is placed on the membrane of a neuron, only one or a few ion channels are covered. This micropipette acts as an electrode and records currents in the single neuron.

The article by Kuperstein *et al* [12] shows figures of these patch clamp recordings. These figures show single spikes, regular spiking, single bursts and a more regular bursting pattern. For every neuron in our network we need to pick a set of five values for a, b, c, d and I, such that the neurons can exhibit these spiking patterns. Since apparently there is some variability in behaviour, we choose the parameters to be picked from a normal distribution $\mathcal{N}(\mu, \sigma)$, to resemble that variation in our neurons. This is justified by the *Central Limit Theorem* if we make sure that our network contains enough neurons.

It remains to choose values for the mean and variance of all five Izhikevich parameters. With all four spiking patterns mentioned above, the corresponding Izhikevich parameters to obtain this behaviour are known. For instance, for all sorts of behaviour it suffices to choose $a = 0.02$. That is

why we choose the mean of this first parameter to be 0.02, whereas σ is chosen to be relatively small and equal to 0.001. The parameter b shows a little more variability over the four spiking patterns, changing from 0.2 to 0.25, resulting in a choice for the distribution $\mathcal{N}(0.22, 0.015)$. The mean and variance of the other three variables are again chosen similar to the mean of the parameter values corresponding to the four spiking patterns. Besides that, to realise the described variability in the neuron population, the variation of both reset values c and d and the current I was chosen relatively big (up to 40% for the current). That indeed these parameters influence the separation between spiking and bursting the most, was already shown in section 2.1.2.3.

Comparance of the results with the computational network to the patch-clamp recordings, conformed that the recorded behaviour was also included in the simulated spiking patterns, in which we finally chose the following parameter values:

$$
\begin{cases}
a & \sim & \mathcal{N}(0.02, 0.001) \\
b & \sim & \mathcal{N}(0.22, 0.015) \\
c & \sim & \mathcal{N}(-58, 5) \\
d & \sim & \mathcal{N}(4.5, 1.5) \\
I & \sim & \mathcal{N}(12.5, 5).
\end{cases}
$$

Now that the neurons are all specified, we need a way to actually simulate them. This is done with software that combines both programming tools of MATLAB and C++. In C++ a custom simulation program that is specialised in simple neuronal networks, called the *Verdandi* simulation environment, is used. For timestepping we use $dt = 0.1$ms.

Just like in every system of differential equations, for having a unique solution one needs initial conditions. With our two-dimensional dynamical system this means that we should have two initial conditions for every neuron, one for the membrane potential and one for the recovery variable. To have an asynchronous firing network, we choose these initial conditions with some variability:

$$
\begin{cases}
V_0 & \sim & \mathcal{N}(-65, 5) \\
u_0 & \sim & \mathcal{N}(-12, 3).
\end{cases}
$$

With the above, the internal dynamics of the neurons are all determined. Yet as already seen in equation (2.1.2), there is an influence of the synaptic currents originating from other neurons, given by I_{syn}. Thus, to complete the modeling of the cultured network, we need to specify how the connections between vertices are established and which properties they satisfy.

2.1.3 Connections

Edges between the vertices in our graph represent synapses between two neurons. As in the biophysical situation, there does not exist a synapse between every pair of neurons. As explained before, connections exist with a certain probability $p_{con}(i,j)$ as given in equation (2.1.1). The synapses represented by the connections, satisfy a model which is an extension of the model of Hodgkin and Huxley.

The model of Hodgkin and Huxley relates the change in membrane potential V over time to the currents flowing in and out of the cell membrane. Among these currents, there are ion currents through the ion channels, injected currents and synaptic currents. The original form of the Hodgkin-Huxley equations then is:

$$\begin{cases} C\dot{V} &= I - I_K - I_{Na} - I_L \\ \dot{\mathbf{y}} &= f(V, \mathbf{y}), \end{cases} \qquad (2.1.5)$$

in which $(\dot{\cdot}) = \frac{\mathrm{d}}{\mathrm{dt}}(\cdot)$, C represents membrane capacity in $(\mathrm{mF/mm}^2)$, I_L is an Ohmic leak current in mA (by leaking Cl^- ions) and I_K and I_{Na} are ion currents of respectively potassium and sodium, modeled in the following way:

$$\begin{aligned} I_K &= \overline{g}_K n^4 (V - E_K) \\ I_{Na} &= \overline{g}_{Na} m^3 h (V - E_{Na}). \end{aligned}$$

In these equations for the ion currents, $\overline{g}_{ion}$ is the maximal conductance of the ion channel given in $(\mathrm{mm}^2 \ \Omega)^{-1}$, in which Ω is a unit for resistance. These ion channels consist of a number of *activation* and *inactivation gates*, of which the behaviour is modeled by m and n, respectively h. The number of gates, effecting the power of these variables (in this case 3 and 1), varies per ion channel. In short, the gates can either cause ions to flow through the channel, or block the channel. The activation and inactivation variables satisfy certain ODEs themselves, in equation (2.1.5) displayed in its second line. The exact form can be found in for instance [11]. Finally, E_{ion} is the reversal potential of an ion given in mV, also called *Nernst potential*, which is the very potential at which these ions neither flow out of, nor into the neuron: the ion concentrations inside the neuron are at electrostatic equilibrium with those outside it.

In equation (2.1.5) I again resembles the synaptic current and injected current. If we rewrite this, we obtain:

$$\begin{cases} C\dot{V} &= h(V, \mathbf{y}) - \sum_{i=1}^{M} g_{t,i}(t) w_i (V - E_i) \\ \dot{\mathbf{y}} &= f(V, \mathbf{y}), \end{cases} \qquad (2.1.6)$$

in which $h(V, \mathbf{y})$ now contains all currents except for the synaptic currents, which are displayed by the second part of this first equation. The summation is over all M synapses terminating at the neuron corresponding to this

differential equation. This thus models how the postsynaptic membrane potential changes in time, given the various currents including the synaptic currents. In the summation over all these synapses, immediately two other properties of the connections are reflected: the *conductance* $g_{s,i}(t)$ and the *weights* w_i.

2.1.3.1 Conductance of a synapse

The conductance of a synapse describes the reaction of the synapse to an arriving action potential. This reaction may vary in amplitude, as well as in how quick the activation and deactivation of the synaptic current is.

Different models for the synapse are present. The simplest one is a *delta synapse*, in which at the instance of a spike, the synapse opens and immediately closes. Thus only at that timestep the potential of the firing neuron i is added up to the potential of the destination neuron j:

$$V_j(t^+) = V_j(t^-) + V_i(t^-).$$

In the biophysical case though, the neuron needs some time to become active and after reacting to the action potential also needs some time to fully recover and close. Thus this delta synapse is not very realistic. In a more realistic model the conductance has a time constant for the opening of the synapse, and another for deactivating after the action potential. When these time constants are the same, this could be modeled with a conductance following the *alpha funtion* (see figure 2.5a). The synaptic model for this
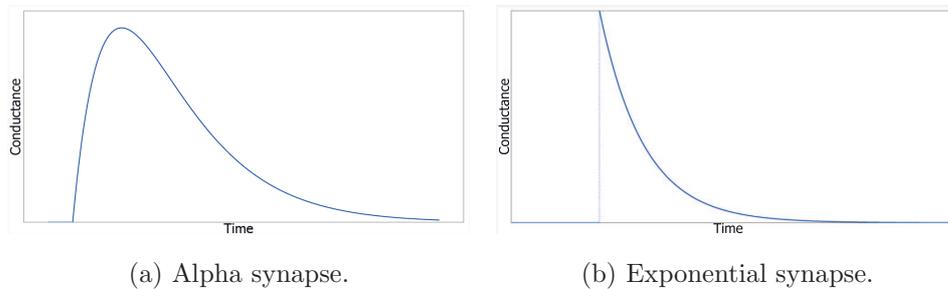

(a) Alpha synapse. (b) Exponential synapse.

Figure 2.5: Sketches of conductances with a) Alpha synapse and b) Exponential synapse.

thesis is more realistic than the delta synapse, yet a little more simplified than the alpha synapse; the *exponential synapse*. At the moment of arrival of the presynaptic action potential, the synapse immediately opens, after which the deactivation is exponential, see figure 2.5b.

To conclude, the amplitude of the conductance can also vary. To separate between the amplitude and the time dependent decay, the conductance is given as:

$$g_{t,i} = g_i \cdot s_i(t),$$

in which, when a spike occurs at $t = t_1$:

$$s_i(t) = \begin{cases} 0 & \text{for} \quad t < t_1 \\ e^{-t/\tau_i} & \text{for} \quad t \geq t_1. \end{cases}$$

Here g_i is the amplitude of the reaction of the synapse to an action potential, and $s_i(t)$ is the time-course of the conductance, as in figure 2.5b, normalized to one. The parameter τ_i is the time constant with which the conductance of the synapse decays. As concluded in the previous section, we will only have one sort of neurons originating from one specific region of the brain, which makes the assumption that all synapses have about the same conductance also plausible. That is why from now on we will change τ_i to τ and g_i to g for every synapse. Both τ and g will become important parameters of our model later on, and when talking about conductance in the remaining of this thesis we mean the amplitude g, since the time course of the conductance is now fixed. With this last formula the model for synapse conductance is fully defined.

2.1.3.2 Weight of a synapse

The probability of a synapse existing between a pair of neurons depends on their distance and so does the amount of appearing synapses. When two neurons are close together, it is quite likely that not one though more connections are established between them. When two neurons are far apart, the existence of one synapse is already unlikely, which makes the appearance of more joint synapses rare.

The above effect of one or more synapses appearing between neurons, is represented by the w_i in equation (2.1.6). There are few publications about how to model these weights with respect to the distance between neurons. One specialised article which does state some results is [6]. Guided by these results we choose for every created synapse i between neuron k and neuron l a weight w_i in the following distance dependent way:

$$w_i = \frac{2}{0.05 + d(k, l)} \cdot a \cdot 10^{-4},$$

in which $d(k, l)$ again is the Euclidean distance between vertices k and l and a is a number picked from a uniform distribution between zero and one.

In the article by Holmgren *et al*, it is shown that synapses with high weights are only found between neurons close to each other, whereas synapses between neurons far away are relatively weak in most cases, corresponding to low conductance. This implies that in general the weight decreases with increasing distance and we choose to let it be inversely proportional to the distance. At the same time, the article shows that a few of the neurons close to each other, share some relatively weak synapses. Thus, it is not per definition true that closeby neurons share strong connections, though chances

are that they do. That is why we introduce a random number a, which is uniformly distributed on the interval $[0, 1]$. In this way, the weight of a synapse between two neurons with distance $d(k, l)$ is picked to be between zero and $\frac{2}{0.05+d(k,l)}$. We multiply this value by 10^{-4} to temper the influence of mutual neurons, thereby reassuring the asynchrony of the network. With these values, the results of Holmgren *et al* can be reproduced.

At this point, the complete computational network has been defined. In the cultured networks, we are not able to measure the electric activity of each separate neuron though, thus we need to clarify how measurements take place.

2.2 Microelectrode Arrays

For the recording of the activity in the cultured network, *Microelectrode Arrays* (MEAs) were used[2]. In our case this MEA is a device that contains 60 electrodes (see figure 2.6b), equally distributed over the region where the neurons are plated (see figure 2.6a). The electrodes have a diameter of 10 μm, whereas the inter-electrode space is 100 μm.
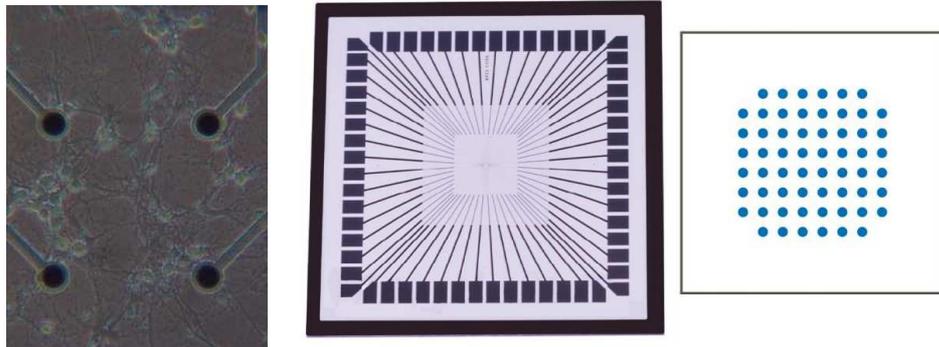
(a) Magnified piece of cultured network with situated electrodes.

(b) Multielectrode Array.

Figure 2.6: a) Electrodes in the region where the neurons are plated. b) Schematic representation of MEA and the distribution of its electrodes.

By using the corresponding software (MC Rack) the membrane potentials of all the separated electrodes can be analysed. In this way for instance spiketimes can be determined in two different ways. The most reliable and simple method seems to be the one that determines a spike based on the standard deviation of the noise. For every electrode it calculates this stan-

---

[2]The exact type can be found at
http://www.multichannelsystems.com/products/60mea10010-ti-wo.

dard deviation, and thereafter defines a spike as a peak in the potential exceeding a threshold of 5 times this standard deviation. These spiketimes can be exported and loaded into MATLAB.

For fitting our computational network to the data of the cultured network, we need to compare the characteristics of both datasets. Therefore it is necessary, that both data are of the same kind. Data created with our computational network concerns single cells, whereas the data originating from the cultured network is based on aggregate information coming from the individual electrodes, that measure in general more than one neuron (see figure 2.6a). Either we need to convert our experimental data to spiking patterns of single neurons, or we should convert the simulated data to signals that are comparable to the data originating from MEA recordings. The first seems to be a not well-defined process, since we have datasets corresponding to all electrodes from which we need to extract spiking patterns corresponding to, by the density of 1000 neurons/mm^2, about 10 times as much neurons. Though methods that take for instance the shape of the spikes in the potentials into account have been developed, it remains a complex task. That is why we choose to convert the results of our computational network, to signals that resemble MEA recordings.

There are two important notes before describing how the conversion takes place. Firstly, with the electrodes of a MEA one measures extracellular behaviour, instead of the intracellular models we have talked about so far. A negative action potential in the intracellular medium of the neuron is measured in the extracellular environment as a positive action potential. Thus if we have a couple of neurons nearby an electrode, instead of simply adding the potentials up, we need to invert all voltages. Secondly, as stated before, the electrodes are not placed in such a way that there is exactly one neuron at the position of one electrode. On the contrary, there can be electrodes recording activity of only a few neurons and only at relatively far distances and there can be electrodes with a cluster of neurons close to the electrode. Intuitively the activity of a neuron situated close to an electrode is measured way better than one far away. Thus instead of just adding up potentials, we need to take some weighted sum, possibly depending on the distance from neuron to electrode.

The above is possible, since after creating the network, all properties are known. For instance the positions are chosen randomly, yet after one has done so, the values can be used. Also the positioning of the electrodes is known from the manufacturer.

A reasonable choice for a weighting factor for this weighted sum is one of the order $R_{i,j}^{-1}$ in which $R_{i,j}$ is the distance from neuron i to electrode j, given in mm. Indeed we choose this as weighting factor, yet transformed in such a way that the contribution of a neuron having exactly the same position as the electrode is one. Moreover, we assume that the electrodes have a certain radius of effectiveness, r, meaning that within a circle of
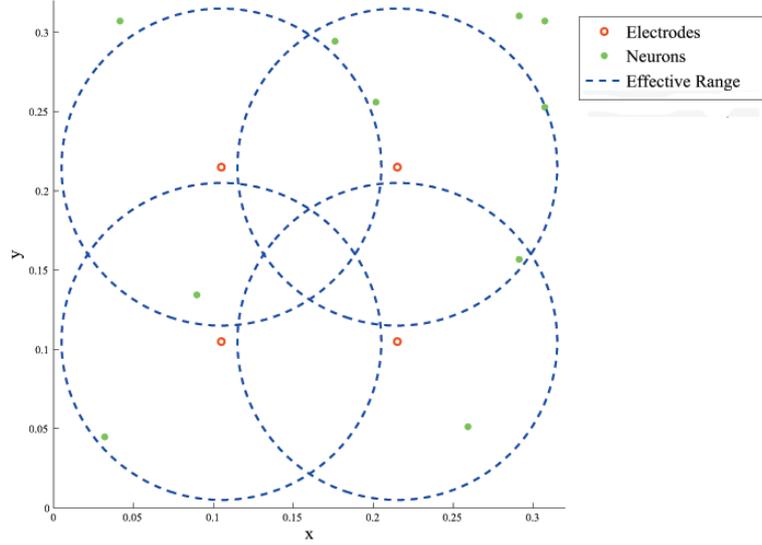
Figure 2.7: Sketch of how we model the MEA chip. In this case 10 neurons, 4 electrodes and their corresponding radius of effectiveness.

this radius activity of neurons is still recorded and outside this circle the particular electrode does not sense the presence of the neuron anymore (see figure 2.7). Therefore we also transformed the weighting factor in such a way that at $R_{i,j} = r$ it is zero. Then, the weighted sum for transforming our cell-based data to data which is comparable to the MEA data, is:

$$M_j(t) = \sum_{i \in D_j} -\frac{r - R_{i,j}}{r(R_{i,j} + 1)^2} \cdot V_i(t),$$

in which D_j is:

$$D_j = \left\{ i \in \{1, 2, \ldots, N\} \,|\, R_{i,j} \leq r \right\},$$

in which N is the total number of neurons. Thus, with every electrode, we create a weighted sum of the neuron-based membrane potentials $V_i(t)$ over the neurons that are within the effective range of an electrode. In this way we obtain a signal of weighted potentials for every electrode, just as we have with the recordings of the cultured networks, although this model remains completely phenomenological. Discussing results, this will therefore be a point which we will return to since it is not assured that this is the correct way of modeling.

Later on in this thesis, we will also discuss spiking patterns corresponding to these electrodes. Spikes in the potentials of an electrode are referred to as a spiking neuron that is detected in the potential trace of these electrodes.

Chapter 3

Gauss-Newton optimisation

In the previous chapter we defined models to create computational data. Moreover we described how we can convert this data in such a way that it can be compared to Microelectrode Array recordings. The goal of this part of the thesis is to try to fit the last few undetermined parameters in such a way that we can recreate, by simulations, the recorded data originating from the cultured network as well as possible, both with and without Aβ. If an effective method for fitting can be developed, this might contribute to systematically looking for significant differences in properties of neural networks, in our case between Aβ-treated and untreated networks.

Of course, one first has to establish how such a fit of computational data to recorded data can be achieved. This is described in the first section. After that we describe in mathematical terms how the behaviour of a network can be approximated, statistically speaking, after which we describe the Gauss-Newton optimisation procedure used as an attempt to solve our problem. We end the chapter by testing the method on a benchmark problem.

3.1   Parameter estimation

In this project we would like to find a computational representative for our cultured network, with and without Aβ. To do this, we will try to fit the data that we create with our simulated network, to the available data that is recorded using MEA chips. The question remains though, what this fitting entails.

Data fitting in this case, means varying certain unknowns in one's model in such a way that one ends up with a model that can reproduce the experimentally recorded behaviour. In the most common ODE or PDE settings, this means that we can change parameters to change the dynamical behaviour. There is no topology at all, since for instance a diffusion equation is an isotropic differential equation.

In our network setting though, we have two categories that might change,

as can be seen in chapter 2. At first there is the dynamical behaviour of every neuron and corresponding synapses. These are all caught in as many parameters as we would like; for instance the Izhikevich parameters, conductances and time constants of synaptic decay. By changing a parameter we automatically change the dynamical behaviour, and if the spanned parameter space has a dimension equal to the number of parameters, we can describe every possible dynamical behaviour in our networks.

The second category is network structure, which we can change by adding or taking away connections and neurons, as well as changing the density of neurons. This network structure might also be changed, although the problem with this is how to parametrise the network structure. There is no clear choice of parameters which defines all possible networks instead of just a subset.

Moreover, keeping such a parameter for a property of the network structure fixed to one value, might lead to different realisations of the network. With another realisation one generically also has other dynamical behaviour. Thus, when one would want to change the network structure as well, to fit the available data, one has to calculate results over different realisations and average these results with every step in the parameter space. This is computationally time consuming.

Following the previous arguments, in this thesis the geometry of our computational network is completely fixed; that is, we pick the positions of all neurons from a 2D uniform distribution, and we know the pairs that share synapses. The fitting is done with respect to the properties of the dynamical behaviour only.

Varying for instance the Izhikevich parameters would change the behaviour of the single neurons and therefore the activity in the whole network. The fact is though, that as explained in section 2.1.2.5, certain facts about these neurons are known. Firstly, the figures corresponding patch-clamp data displayed in the article by Kuperstein *et al* were available and moreover we know that the neurons used for the cultured networks are hippocampal neurons. From this, we derived a reasonable set of parameters for our Izhikevich neurons in the previous chapter, thus we choose not to optimise these.

What is left are the synaptic parameters. Indeed, in the Introduction we mentioned that thoughts are that Aβ dysregulates presynaptic processes, which raises the idea that in these synaptic parameters the first phenomena occur when injecting Aβ. As shown, ion channels are important in the conductance of an action potential through a synapse. These ion channels are proteins that can bind physiologically relevant chemicals such as neurotransmitters. Thereby it will not come as a surprise that certain other molecules can bind to ion channels as well. For example, there is Tetrodotoxin, which can block channels from the outside, suppressing the ability of a neuron to conduct signals. Moreover, there are toxins (for instance α-Scorpion) that

slow the rate of inactivation [14]. This suggests that Aβ can affect those synaptic properties as well, which makes the choice to optimise these parameters biophysically plausible. Therefore we indeed choose to only manipulate the synaptic conductance g and the exponential time constant τ corresponding to the decay of the synaptic activity after a spike. Later on, we will see that we conclude that other possibilities would have been possible as well. Having clarified the above, we have to establish how we actually want to compare the behaviour of two different networks.

3.2 Network characteristics

Until now we have discussed fitting the computational data to the data extracted from a cultured network. Yet how can one actually distinguish between two different networks? In other words, how can we describe the behaviour of a network, with for instance certain network characteristics, in a well-defined way. Here well-defined means that we prefer to have different characteristics for different network behaviour. In this section we will introduce certain target values that are properties of our networks, which we will use as a description for network behaviour. Moreover, with these target values an objective function is defined, representing the error that we make in estimating the MEA data by those generated by our computational network with a specific choice of g and τ.

3.2.1 Target Values

To find a definition for certain characteristics which represent our network behaviour, we should dive deeper into what we want to achieve. Since all mathematical models are an approximate representation of reality, the best we can obtain is an as accurate as possible estimation of the correct values for g and τ. Since we do not have the cell-based data for our cultured neurons as explained in section 2.2, we choose to convert our computational data to data concerning electrode data, in the sense that we wish to map our data on the behaviour of individual neurons onto aggregate data over nearby (in the Euclidean sense) neurons. After this has been done, we have a timeseries of aggregate potential recordings for every electrode at our disposal. Characteristic for networks are spiking patterns corresponding to its neurons. Determining, in turn and in some appropriate way, these spiking moments in our data, we can extract a timeseries of spiking times for each electrode.

    As in other researches into neuronal networks [21], we describe the characteristics of the timeseries by the moments of the underlying probability distribution. Regarding our timeseries, we take the *interspike intervals* (ISIs) per electrode and try to estimate how they are distributed. This can be achieved by estimating the moments of the interspike intervals, starting

by the mean. Suppose we have the spiketimes t_i, $i = 1, 2, ...S$, then the mean ISI is defined to be:

$$\overline{\Delta t} = \frac{1}{S-1} \sum_{i=1}^{S-1} \Delta t_i,$$

in which $\Delta t_i = t_{i+1} - t_i$.

In literature it is more common to talk about frequencies of spiking, thus what we will use as characteristic for the behaviour of our networks, is the *Mean Firing Rate* (MFR) of a network:

$$\text{MFR} = \left(\overline{\Delta t}\right)^{-1}.$$

Here, these values are scaled in such a way, that the frequencies are given in Hertz.

With solely the MFR being fixed, numerous firing patterns might satisfy this characteristic. The same goes for most timeseries; with only an estimation of the mean, one knows almost nothing about the underlying probability distribution. Thus by only using MFRs as our description for the behaviour of a network, we will find a couple of g and τ satisfying the right MFR quite easily, though it will be far from unique. For instance, we can have a regular spiking neuron satisfying the same frequency of firing as another neuron that is bursting.

This may introduce us to a second statistical measure, namely the second moment or variation. The variation reveals the amount of regularity in the interspike intervals, which plainly is a big difference between regular spiking and bursting neurons. Bursting neurons have small interspike intervals inside a burst, alternated with big interspike intervals outside a burst. This gives the variation a relatively larger value than with a regular spiking neuron.

For this reason we define the *Coefficient of Variation*, being a scaled version of the standard deviation $\sigma_{\Delta t}$:

$$\sigma_{\Delta t} = \sqrt{\frac{1}{S-1} \sum_{i=1}^{S-1} \left(\Delta t_i - \overline{\Delta t}\right)^2}.$$

Now the coefficient of variation is this standard deviation scaled by the mean of the interspike intervals:

$$\text{C}_{\text{V}} = \sigma_{\Delta t} \cdot \left(\overline{\Delta t}\right)^{-1}.$$

In statistics, based on the first two moments, the mean and variation, one knows the general shape of the probability distribution. In our case, by using the Mean Firing Rate and Coefficient of Variation as target values to describe neuronal network behaviour, the MFR acts as a measure for

amount of activity in our network and the C_V distincts between the various firing patterns, like bursting or regular firing. In the remaining, these target values are used as characteristics for neural network behaviour and moreover for our objective function for the optimisation later on.

One final note about the way these statistical moments are used should be made. By the uncertainty appearing in the initial conditions of all membrane potentials, a transient stage in which the MFR and CV still need to stabilise might occur. For this reason one might run simulations with several initial conditions after which the results can be averaged, though we choose to use statistics averaged over time and moreover we do not use the first second of our simulations to exclude the transient stage. In the section about the optimisation method we will also analyse the time needed for these target values to stabilise.

Spike detection Based upon spike times the first two moments are used to characterise network behaviour. Thus it remains to introduce a method to estimate the moments at which a spike occurs. This can be done by using the timeseries of potentials of every electrode. In a potential trace of a spiking neuron, a clear definition of a spike is present, namely that point where the potential crosses $V = 30$mV and is reset to its reset value c. This that plainly in figure 3.1a we define three spikes. When converted to data which is comparable to the MEA data though, potentials do not have such fixed ranges anymore, since depending on the amount of surrounding neurons and their distance to the electrode the potentials are added with different weights. Besides that, neurons do not fire all at once which is reflected on a potential trace of MEA data in the sense that it contains a sequence of spikes with highly varying amplitude. The above causes the fact that determining what is a spike is a genuinely hard problem, especially since we do not intend with a MATLAB script as opposed to manual detection.
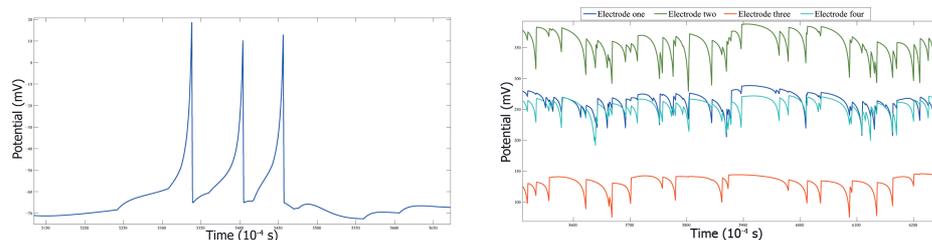
(a) Neuron membrane potential.

(b) Computational MEA data.

Figure 3.1: a) Example of neuronal membrane potential in time with three spikes. b) Example of data of a computational network converted to MEA data.

The most important thing to notice about spikes in MEA data, is that in

the timestep before a spike, the potential decreases, and in the timestep after the spike it increases. This zero-crossing of the derivative of the membrane potential, is the first criterion which we use to detect a spike.

To prevent noisy jumps of small amplitude to be identified with the event spike, a second criterion is that the upward jump after this zero-crossing should be of a certain minimum threshold magnitude. This criterion is derived from the fact that the Izhikevich neurons are defined to make the reset jump after a spike (see equation (2.1.3)). This immediately raises a point of discussion, since there is no way to select this threshold value in a natural way. Regarding figure 3.1b one can actually see that it is not fully clear which jumps should be selected as spikes and which should not be.

3.2.2 Objective function

The objective function should plainly be a measure for how close one is to the desired value. In our case it thus has to give a measure for how well our computational network behaves with respect to the recorded data of the cultured network.

In other words, the most rational choice for an objective function would be the error we make in Mean Firing Rate and Coefficient of Variation. For these purposes, we define the following function:

$$\mathbf{F}(g, \tau) = \left(\begin{array}{c} \mathbf{MFR}(g, \tau) \\ \mathbf{C_V}(g, \tau) \end{array} \right).$$

Here both components of $\mathbf{F}(g, \tau)$ are vectors, containing Mean Firing Rates and Coefficients of Variation of all electrodes, since the best solution to our optimisation problem would be if every electrode in the computational setting resembles the recorded data. Apart from that, using MFRs and $\mathbf{C_V}$s of every electrode instead of just an average over the whole network, allows us to also regard asynchrony in the network.

Regarding the data, let $\mathbf{MFR}^*$ and $\mathbf{C_V}^*$ be the Mean Firing Rate and Coefficient of Variation corresponding to the recorded MEA data and define:

$$\mathbf{F}^* = \left(\begin{array}{c} \mathbf{MFR}^* \\ \mathbf{C_V}^* \end{array} \right).$$

Then we can define the following objective function:

$$E(g, \tau) = \frac{1}{2}\|\mathbf{F}(g, \tau) - \mathbf{F}^*\|_2^2. \tag{3.2.1}$$

in which $\| \cdot \|_2$ represents the 2-norm, thus $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \ldots + x_n^2}$.

With this objective function, we have a measure of how close a simulation with a fixed g and τ is to the available data. There is still one remark to be made: the values of $\mathbf{MFR}$ are usually not of the same order as $\mathbf{C_V}$. Mean Firing Rates are mostly of order 10 to 100, whereas the Coefficient of

Variation is order of 1. Thus to be sure that the influence of the **MFR** to the error is of the same order as the influence of $\mathbf{C_V}$, we need to scale both by dividing **MFR** by the maximum of $\mathbf{MFR}^*$ and similarly for $\mathbf{C_V}$. From now on, if we talk about the objective function, this is with respect to the scaled target values.

Finally, to try to obtain the synaptic parameters for which our estimation is best-fitted to the data, we will now plunge into an optimisation method aimed at minimising this error function.

3.3 Gauss-Newton optimisation method

Inspired by the research by Hahn and McIntyre [4], we will use a Gauss-Newton optimisation method to try to find a couple of g and τ that minimises the error in Mean Firing Rate and Coefficient of Variation between our simulations and the available MEA data. The Gauss-Newton instead of the Newton method is used, since we have two target values per electrode, which comes down to more equations than unknowns. Since a non-square matrix has no inverse, the Gauss-Newton method uses a generalised inverse to determine a next iteration step. The optimisation method starts with an initial value and then iteratively tries to find a zero of the objective function, in our case the error function. This implies that besides our computational network, we will need to introduce the iteration steps of the optimisation method and a way to extract the target values from the data. Both will be discussed in the sections directly below.

3.3.1 Iteration steps

As a result of our iteration steps, we would like to find a minimiser for the error function set up in equation (3.2.1). Generically, we will not find the minimum exactly, especially since we made all kinds of model assumptions. That is why we choose a certain tolerance in error with which we are satisfied. That is, when the optimisation method finds a couple of g and τ that lead to simulations with MFR and $\mathrm{C_V}$, such that the error is smaller than this tolerance, we regard these g and τ as our solution. The value of this minimum score will be discussed later on, since our solution might depend on it.

To start the Gauss-Newton method, we need an initial value. Therefore we pick certain ranges for both g and τ wherein the optimisation method can search for optimal values. Since within these bounds there is no indication what the optimal values should be, we choose the initial values randomly by drawing from a uniform distribution with the boundaries equal to these ranges and then run the method several times.

Starting from these initial values, the iteration is started. To obtain a

next set of parameters, an update rule is needed, which is the following:

$$\left(\begin{array}{c} g_{i+1} \\ \tau_{i+1} \end{array} \right) = \left(\begin{array}{c} g_i \\ \tau_i \end{array} \right) - \lambda (J^T J)^{-1} J^T \mathbf{R}(g_i, \tau_i), \qquad (3.3.1)$$

in which $\lambda = 1$ at the first step and

$$\mathbf{R}(g_i, \tau_i) = \mathbf{F}(g_i, \tau_i) - \mathbf{F}^*,$$

is called the remainder. The Jacobian J equals:

$$J = J(g_i, \tau_i) = \left(\begin{array}{cc} \mathrm{D}_g R_1(g_i, \tau_i) & \mathrm{D}_\tau R_1(g_i, \tau_i) \\ \mathrm{D}_g R_2(g_i, \tau_i) & \mathrm{D}_\tau R_2(g_i, \tau_i) \end{array} \right).$$

In equation (3.3.1) the term $(J^T J)^{-1} J^T$ is a matrix known as the generalized inverse of J.

The Jacobian is computed numerically. In short, for a small perturbation ε in every parameter the change in function value $F(g_i + \varepsilon, \tau_i)$ is calculated and divided by ε. That provides us with the first column of the Jacobian, the other column is calculated in a similar way.

When, by following this iteration a new point in our parameter space is found, the algorithm checks whether the step leads to a decreases of the error with respect to the previous step. If so, λ remains unchanged and with these new parameters we go into the next iteration, unless the error is smaller than the tolerance. If not, we refine the stepsize by halving the value for λ and again we check if this step decreases the error. This is done five times at most, after which the method reapproximated the Jacobian for a bigger values of ε. With the new estimate for the Jacobian, again we perform equation (3.3.1) and we repeat the above. The Jacobian is also reapproximated five times at most, after which, if no point in parameter space that improves the error has been found, the program takes a random perturbation in both g and τ. With these new randomly perturbed values for the parameters we enter the next iteration. One last criterion is that we only allow the procedure to perform 50 iterations, to avoid entering an infinite loop of iterations. The whole procedure can be seen in a overview diagram in the paper by Hahn and McIntyre [4].

As a benchmark for our procedure, we create a computational network with fixed g^* and τ^*. By running simulations for these parameter values we obtain certain target values MFR^* and $\mathrm{C_V}^*$, which then are set as target values for our optimisation procedure. If the procedure works perfectly, it should be able to return those chosen values for gr and τ. Moreover, to gain some insight into the optimisation and the amount of global and/or local minima and maxima, we plot the error function. When for instance several local minima and maxima are present, this might influence the convergence of our Gauss-Newton method. Besides that, by observing the error function

we also found out that the time horizon of our simulation has of influence on the error function. Before we discuss these subject, we derive a theoretical statement about fixed points of our optimisation problem.

Fixed point in theory   In theory, we can show that generically one unique fixed point exists. From equation (3.3.1) we derive that our fixed point should satisfy the following condition:

$$(J^T J)^{-1} J^T \mathbf{R}(g, \tau) = 0,$$

and thus

$$\mathbf{R}(g, \tau) \in \mathcal{K}er \left\{ \left(J^T J \right)^{-1} J^T \right\}, \qquad (3.3.2)$$

in which $\mathcal{K}er\{A\}$ is the kernel of a matrix A, which is also referred to as nullspace.

The statement in equation (3.3.2) is equivalent to:

$$\mathbf{R}(g, \tau) \in \mathcal{K}er \left\{ J^T \right\}, \qquad (3.3.3)$$

since:

$$
\begin{aligned}
x \in \mathcal{K}er\{B^{-1}A\} &\Leftrightarrow B^{-1}Ax = 0 \\
&\Leftrightarrow Ax = 0 \\
&\Leftrightarrow x \in \mathcal{K}er\{A\}.
\end{aligned}
$$

Suppose we have an n by n system which has to be solved. Then, generically, we have one solution to the equation $Ax = 0$ and thus one element in our kernel, namely the zero vector. Removing one equation, gives us an underdetermined $n - 1$ by n system, which has infinitely many solutions. There is one free variable, which can be chosen to be any value, after which other variables are fixed with probability one. In other words, there is a whole curve of solutions; the kernel of A is one-dimensional. More specifically, when fixing this free variable to be the zero element, generically, the others will be zero as well.

In this case, when having m electrodes, J^T is an element of $\mathcal{M}^{2 \times (2m)}$, which are the real matrices having 2 rows and $2m$ columns. Thus an underdetermined system with a solution space of dimension $2m - 2$ has to be solved. The parameters g and τ that we are optimising for, span a 2D-space, which ensures that $\mathbf{R}(g, \tau)$ in equation (3.3.3) is a surface in the full space of $2m$ dimensions. From this, we can conlude that generically the surface corresponding to $\mathbf{R}(g, \tau)$ and the solution space $\mathcal{K}er \left\{ J^T \right\}$ intersect transversally. Outside a set with measure zero in which those $\mathbf{R}(g, \tau)$ and $\mathcal{K}er \left\{ J^T \right\}$ are contained where at least one of the spanning vectors of the tangent space of $\mathbf{R}(g, \tau)$ or $\mathcal{K}er \left\{ J^T \right\}$ is a linear combination of the other's spanning vectors, the tangent spaces span the whole $\mathcal{R}^{2m}$. Thus, generically the tangent spaces of $\mathbf{R}(g, \tau)$ and $\mathcal{K}er \left\{ J^T \right\}$ do not coincide and have dimensions 2 and $2m - 2$ respectively. Thereby, together they span the entire
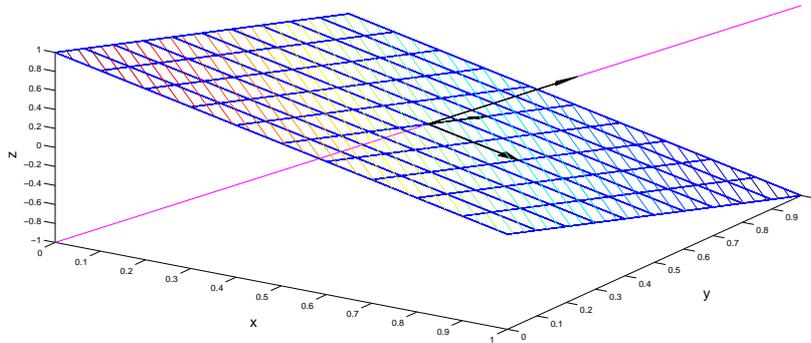
Figure 3.2: Sketch of a transversally intersecting surface and line in $\mathcal{R}^3$.

space and automatically they intersect transversally. In three dimensions, this is equivalent to stating that a surface and a line intersect transversally and thereby that their tangent spaces span the whole $\mathcal{R}^3$. Outside the set with measure zero in which the tangent of the line is tangent to the plane, this is indeed true. An example is shown in figure 3.2, in which the black arrows in the surface span the tangent space of the surface and the third arrow spans the tangent space of the line. All three together indeed span the whole $\mathcal{R}^3$.

Concluding, the fact that $\mathbf{R}(g, \tau)$ and $\mathcal{K}er\left\{J^T\right\}$ intersect transversally implies that generically there is one isolated fixed point.

3.3.2 Time horizon
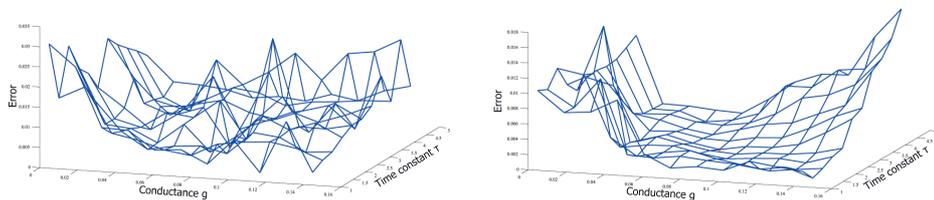
(a) Time horizon $T = 2$ seconds.   (b) Time horizon $T = 40$ seconds.

Figure 3.3: A plot of the error function for a certain grid in g and τ, with a time horizon of a) $T = 2$ seconds and b) $T = 40$ seconds. The red dot represents (g^*, τ^*).

With our benchmark procedure, we choose to have a network of 40 neurons, with a computational MEA of 4 electrodes. With this network we can plot an error function, since after MFR* and C$_V$* are known, this is a funtion of g and τ only. Thereby, interesting notions about the time horizon of our simulations and corresponding stabilisation of our target values MFR and C$_V$ become apparent.

For this analysis, we choose the values $(g^*, \tau^*) = (0.075, 3)$. The error is calculated on a grid in parameter space surrounding this chosen point and then plotted in 3D. An example plot of the error function is given in figure 3.3a. The figure shows such an amount of peaks and thus local maxima and minima, that a clear global minimum cannot be pointed out. Non-smoothness of this kind causes difficulties for our optimisation method with finding the right couple of g and τ.
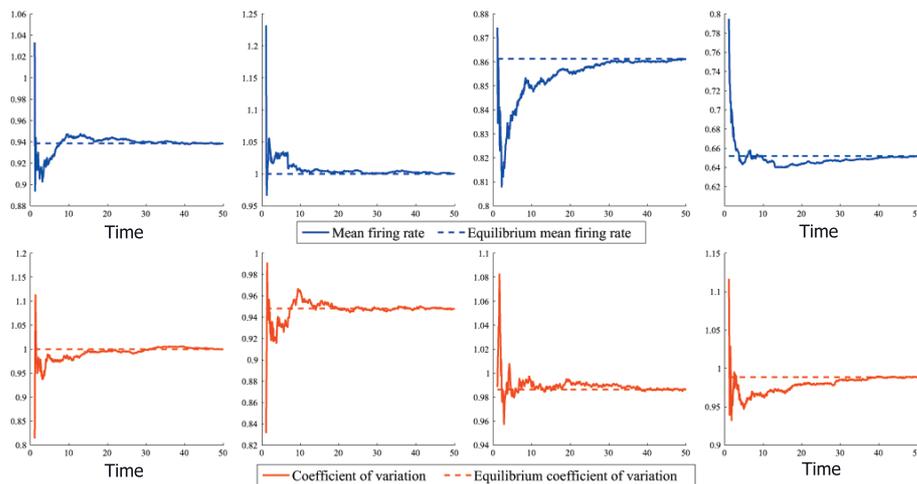

Figure 3.4: The evolution of MFR (blue) and C$_V$ (red) values of every electrode (1 to 4 from left to right) in time. For comparison, the dashed plateaus represent the equilibrated target values. We observe that it takes some time for the values to equilibrate.

Though, given that our error function is non-smooth like this, this might imply that something unforeseen is going on. If not, then this can be problematic for the Gauss-Newton method, since recapitulating, the Gauss-Newton iterations need an estimation of the Jacobian matrix of the vector $\mathbf{R}(g_i, \tau_i)$ (equation (3.3.1)). Computing derivatives of a surface that is non-smooth plainly leads to difficulties. Plainly, there are several ways to smoothen a function, like for instance using a Gaussian kernel to mollify the data. Nevertheless, this smoothening also affects the position of the global minimum such that after mollification it might well be that the outcome of our optimisation is not trustworthy at all [20]. That is why we do not use

any of those methods.

The fact that the time horizon of our simulations is part of the cause for this non-smoothness becomes clear though, when we plot the Mean Firing Rates and Coefficients of Variation in time (see figure 3.4). In these graphs, the values for MFR and C_V are plotted, in which the horizontal axis represents the timestepping. That is, at $T = 30$, the simulations ran for 30 seconds and both Mean Firing Rate and Coefficient of Variation were calculated over the whole time period, excluded the first second to omit the transient time as explained in section 3.2.1. The dashed plateaus represent the target values calculated with regard to the whole 50 seconds and are plotted for comparison purposes.

For the first plot of the error function (see figure 3.3a), the simulation time to calculate new statistical moments in every iteration, equaled to two seconds. As seen from figure 3.4 after two seconds these target values fluctuate as much as the error function does. This might be caused by the fact that they are not yet equilibrated. For instance, we see that some of the Mean Firing Rates are not even equilibrated after 30 seconds.
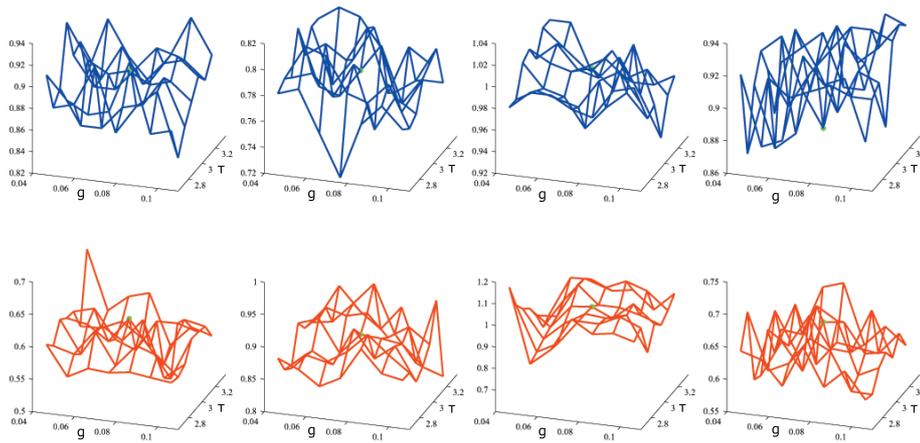

Figure 3.5: The values for MFR (blue) and C_V (red) plotted in the g-τ plane, with $T = 2$ seconds. The green dot represents (g^*, τ^*). From left to right the subplots show target values corresponding to respectively the first to fourth electrode.

This definitely gives a hunch to a cause for the non-smoothness of the error function. The error function is calculated from these target values and since they are not equilibrated after 2 seconds, the error function will not be either. When we indeed plot these target values, see figure 3.5, in the parameter space for $T = 2$ seconds, one sees that these fluctuate as well. Inspired by figure 3.4, we plotted the same statistical moments with a simulation time $T = 60$ seconds in figure 3.6. These are still non-smooth, though less than before. There is a drawback though, and that is

the computational expense that joins the increase of the time horizon. Thus our limited computational power forces a tradeoff between the decrease of non-smoothness and simulation times.
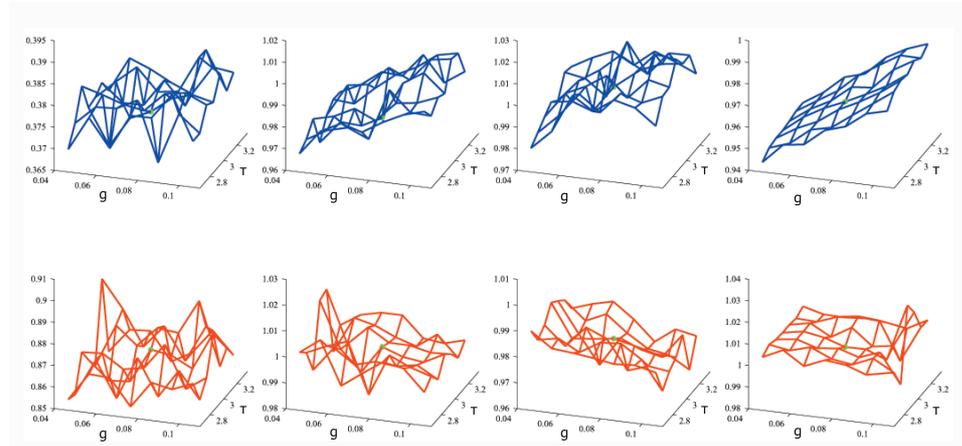

Figure 3.6: The values for MFR (blue) and C_V (red) plotted in the g-τ plane, with $T = 60$ seconds. The green dots represent (g^*, τ^*). From left to right the subplots show target values corresponding to respectively the first to fourth electrode.

Fortunately, looking at an error function corresponding to a simulation time of 40 seconds, see figure 3.3b, we already see improvement in smoothness. This might lead to better convergence of our Gauss-Newton implementation, and although extending from 2 to 40 seconds simulation per calculation of the MFR and C_V is computationally expensive, this is a necessity. If one does not increase the time horizon drastically, one measures transients producing meaningless optimisation results.

Still, if we compute the error function on a denser mesh around (g^*, τ^*), as shown in figure 3.7, fluctuations remain. Nevertheless, from now on we will be using a time horizon of 40 seconds for the reasons explained above.

3.3.3   Conductance versus exponential time constant

In the previous section we showed error functions, also on a relatively large domain around (g^*, τ^*), for instance figure 3.3b. Apart from statements about the time horizon, another interesting fact can be seen in this graph, namely the occurrence of a valley in this error function. In this section, this is explained via the total current that flows through a synapse.

Figure 3.3b shows a clear valley in the g-τ-plane in which error values are all relatively low. There are biophysical arguments supporting this observation and to show this we regard the total current going through a synapse when it fires. By integrating the amount of current that flows through the
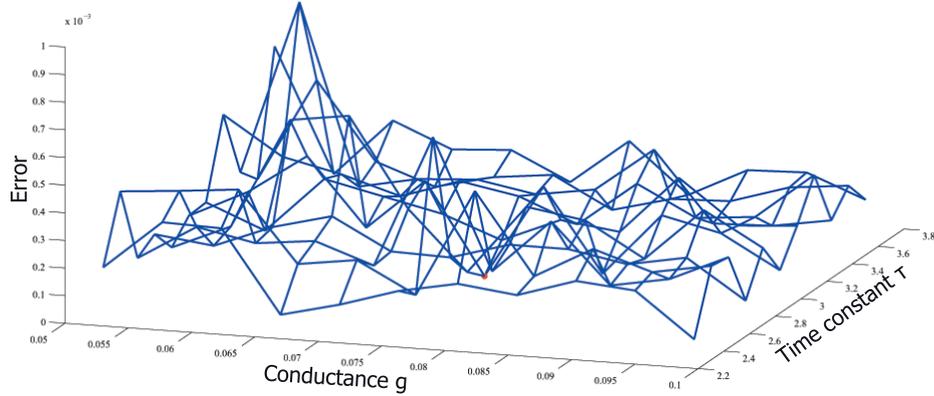
Figure 3.7: An error function with a simulation time of 40 seconds, yet now on a smaller domain than in figure 3.3b. Again, the red dot represents (g^*, τ^*).

synapse at time t, that is the area below figure 2.5b, we obtain the total current that flowed through one such synapse until time t. Since we model the synapses as exponential synapses, this is given by:

$$I(t) = \int_{s=0}^{s=t} g \cdot \mathrm{e}^{-\mathrm{s}/\tau} \mathrm{ds} = g\tau \left(1 - \mathrm{e}^{-\mathrm{t}/\tau}\right).$$

Taking the limit $t \to \infty$, one sees that the current through a synapse corresponding to one spike is proportional to the product of g and τ. This means that along a curve of $g \cdot \tau = $ constant we have about the same current injected into the postsynaptic terminal. The reason we review this total current, is that to a large extent it determines the spiking behaviour of a neuron and thus both Mean Firing Rate and Coefficient of Variation. That is why we would expect a valley to appear in the error function beforehand. Suppose there is a minimum value for our error function, then changing g and τ along the indicated curve, will not change the behaviour of the network too much. Thus this will have a small impact on the error, therefore creating a valley in the error function. On the contrary, when moving away from this curve, the total injected current changes and for instance the Mean Firing Rate changes. Moreover, we could reach different firing pattern regimes, causing the Coefficient of Variation to change, giving higher errors.

The existence of the valley might again lead to complications in the Gauss-Newton method, which is also something we will see in the next section, in which we show results corresponding to our benchmark.

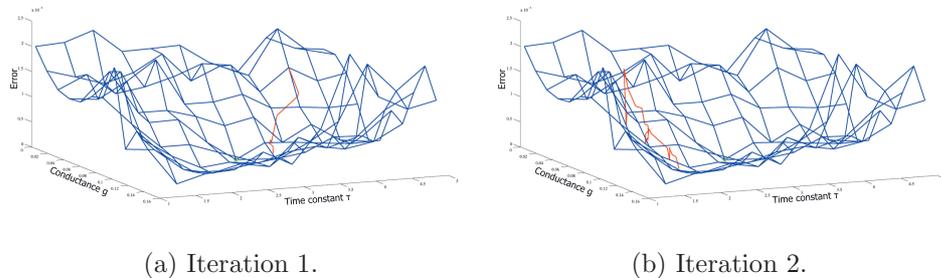## 3.4 Results on a benchmark procedure

(a) Iteration 1. (b) Iteration 2.

Figure 3.8: Error function with path of our iterationsteps (red) for two optimisation. The green dots represent (g^*, τ^*).

As indicated by the difficulties that appear with our Gauss-Newton method, convergence to a fixed point is not self-evident. In this section, as a benchmark, we fix our computational network and even so the values for $(g^*, \tau^*) = (0.075, 3)$ and try to recover those values with our optimisation method. Our fixed network in this case, consists of 90 neurons and 9 electrodes, with which target values are computed. By construction, we have a global minimum where the error is exactly zero.

Results are shown that are created with the Gauss-Newton optimisation procedure explained in this chapter so far. One of the first results with the two-dimensional optimisation, is a clear lack of convergence to the (g^*, τ^*) that we chose. This can be reviewed in figures 3.8a and 3.8b. In these figures the path of the iteration procedure is shown in red with the error function in blue, including the green dot which represents (g^*, τ^*). Both figures display the same error function, with different initial points for the optimisation procedure. It can be seen that neither of them converges to the desired point. Moreover, by the fluctuation in the red path of figure 3.8b in particular, it can be seen that indeed the error function contains even more fluctuations than shown with the current grid.

The optimisation method seems to iterate into the valley, yet being there, with the non-smooth surface and small errors it has difficulties with converging to the global minimum. As already said before, the results of the procedure are dependent on the threshold score we give it. With a relatively large minimum score (0.05), it converges quickly, though to a value which does not have to be near the global minimum and that moreover changes with every computed optimisation. When we decrease the threshold (for instance 0.005), the trouble inside the valley arises and the method keeps refining stepsizes, reapproximating the Jacobian and generating random steps. In both figures of the 2D optimisation shown above, when we iterated longer, the optimisation procedure had difficulties finding points with a lower error.

Moreover we did not per se approach the optimal point either. Thus, by the non-smoothness of our error function the Gauss-Newton method cannot reach values of the error that are needed to find our global minimum.
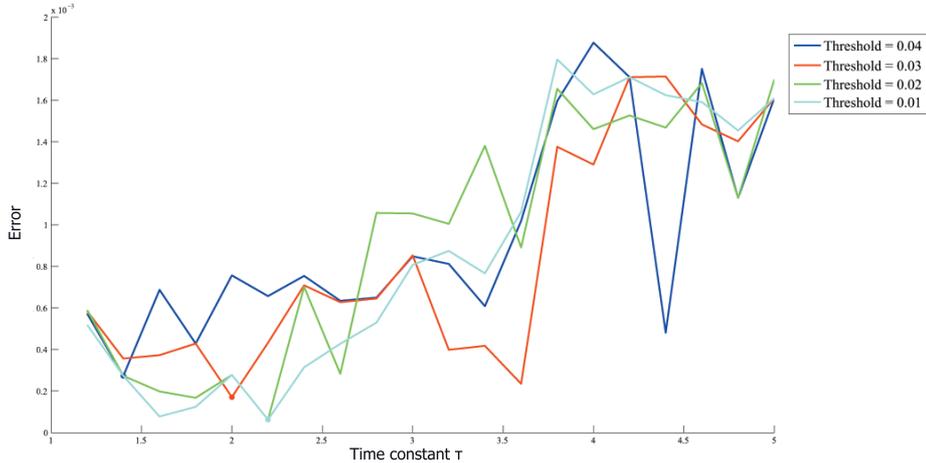

Figure 3.9: 1D optimisation: the minimum error found for every τ and every threshold. The dots coincide with the overall minimum on the curve for the corresponding threshold score, in which the green and cyan minima coincide.

Since apparently the two-dimensional optimisation lacks convergence, we try to reduce the optimisation to one dimension, by only using the conductance as optimisation parameter. As a sort of semi-two-dimensional optimisation, we can perform this one-dimensional method for several, lets say n, values for τ. Then, if the 1D optimisation converges, we should have n couples of g and τ that are positioned on the valley curve. The global minimum can then be tried to find in various ways, though the simplest is calculating the error on a discrete set of points on that curve and determine the minimum of it. Since plainly the results depend on the threshold value, we run the optimisation for decreasing minimum scores, to see if results will improve with every refinement. Every decrease like that intensively increases the computation time, yet if indeed results improve with every step, it might be able to extrapolate to a nearly zero threshold score. Corresponding results are shown in figures 3.9 and 3.10. For every fixed τ and threshold score we used the same initial condition $g = 0.1228$ to enter the optimisation, determined as described in subsection 3.3.1.

Figure 3.9 shows a line connecting all minimum errors that have been achieved for every optimisation with fixed τ-value, one such curve for every different threshold value. Moreover, the dots in the graph represent the chosen τ-values for every threshold corresponding to the smallest error. The figure shows no clear sequence of improvement with every decrease in threshold. In contrary, errors subsequently increase and decrease for fixed
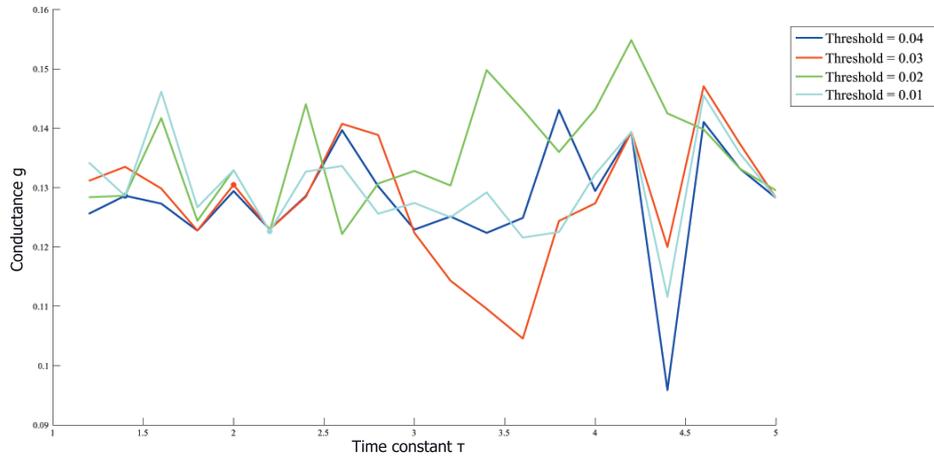
Figure 3.10: 1D optimisation: the values for g found for every τ and every threshold. The dots coincide with the overall minimum on the curve for the corresponding threshold score, in which the green and cyan minima coincide.

τ-values and decreasing thresholds, even for τ fixed to the optimal value. Moreover, none of the found minima is situated within 25% error of this optimal value.

Regarding figure 3.10, the values found for g in every optimisation for fixed τ are displayed, again for all four threshold values. The dots represent the g-values chosen by the semi-2-dimensional optimisation for every threshold. Just as our error function these results again form a non-smooth curve and, for every threshold value, do not seem to resemble any kind of valley curve. Unfortunately, even for τ fixed to the optimal value, it is not per se true that with decreasing the minimum score the procedure recovers values closer to our global minimum. With these observations we can state that the option of extrapolating to zero threshold value seems to be inappropriate. Besides that, the above implies that, since the valley is not found, taking the minimum error over all one-dimensional optimisations does not lead to reliable outcomes either.

The above are arguments for the fact that in this case, with an error function that is still non-smooth, the Gauss-Newton method does not converge to the global minimum. One of the reasons for the above graphs corresponding to the one-dimensional optimisation to be non-smooth, might be because we have a fixed threshold value and the procedure has troubles with reaching relatively low errors. That is, with a low minimum score, one τ-value might lead to no convergence at all, where another one, coincidentally or not, finds an iteration step to a parameter value that does lead to an error low enough. Yet another optimisation might be such, that after few iterations it finds a value for g which satisfies the threshold value, though could have done even

better when performing more iterations. All of them stop when the desired minimum value is reached though, which will not necessarily lead to smooth outcomes.

As mentioned before MEAs cause spike detection to be a genuinely hard problem. Moreover, the way MEAs are modeled can not be supported with biophysical arguments, since it is a way of aggregating potential data based on geographical properties. Therefore, in the following we will discuss computational networks in which MEAs have been removed and we describe network behaviour and error functions based on data of single neurons.

Finally, in this chapter data that we have at our disposal was discussed. Apart from the fact that our computational framework to simulate networks that resemble this data does not work yet, analysing the data also revealed that the activity in these measurements was low. The Mean Firing Rates in the recordings in general did not exceed 2 or 3 Hertz, which did not compare to values seen so far in any of our computational networks. That is why in this thesis do not regard the data. Hopefully in future research, the desired computational framework can again take recorded data into account.

Chapter 4

Network behaviour

In this chapter we will have a look at network behaviour based on single-neuron data. Therefore, instead of using Mean Firing Rate and Coefficient of Variation for every electrode as our target values, we use these for every neuron. At first, we analyse the behaviour by 'eyeballing', that is plotting error functions and other graphs and try to read neuron and network properties from this graph. Thereafter, we try to perform Gauss-Newton on these networks in which MEAs are no longer implemented.

There is one extra change in our networks with regard to the previous chapter. In this chapter, since we expect our statistical results to be better with larger networks, we enlarge our network to 600 neurons, unless otherwise stated. With regard to the time horizon investigation, simulations are all done with $T = 40$ seconds. Moreover, to regard more than one optimal point in this thesis, in this chapter we take $(g^*, \tau^*) = (0.045, 5)$. With other values for (g^*, τ^*) similar results are obtained, though they might not be as descriptive as the ones in this chapter.

4.1 Dependence on synaptic conductance g and time constant τ

4.1.1 Error functions and target values

As a first glance at graphs with single-neuron properties, in figure 4.1a we plotted an error function for a network of 600 neurons, with a simulation horizon of 40 seconds. An advantage of performing all calculations based on single-neuron data instead of MEA data, is that spike analysis does not fail in this case. This implies that overall the calculation of MFR and C_V improves, which, as we will see, leads to error functions that have less radical changes when changing g or τ. This is also reflected in the error functions, since overall these are smoother than for instance figure 3.3b. Moreover, what figure 4.1a also shows, is that the upper bound of the error on a fixed domain

(a) Error function, $T = 40$.

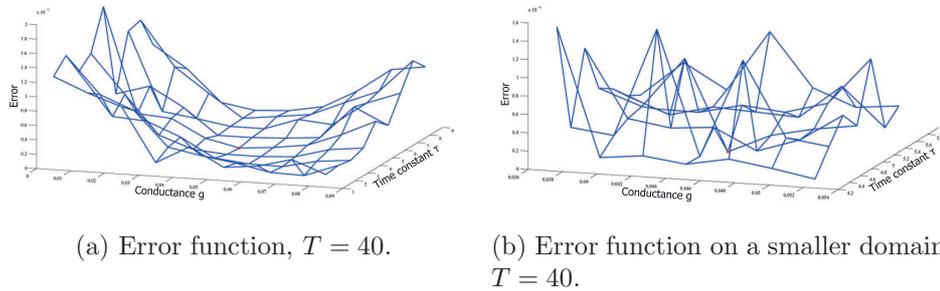(b) Error function on a smaller domain, $T = 40$.

Figure 4.1: Error functions in the single-neuron case on a) a relatively large domain and b) a relatively small domain. As before, the red dot represents (g^*, τ^*).

in parameter space is smaller. This is an interesting statement, because it would imply that we can simulate certain desired behaviour better. In figure 4.1b another error function for the single-neuron case is plotted, yet on a finer mesh. This illustrates that, although the spike detection should not be a problem anymore, the error function on this scale remains non-smooth.

To see how a change in conductance or time constant of the synaptic decay affects the network behaviour, in this section we will consider graphs of the target values of our neurons. Since we work with 600 neurons, we do not plot all target values. We choose to display the Mean Firing Rate and Coefficient of Variation belonging to the spiking data of the neuron with the highest amount of synapses, since this neuron determines most of all neurons the overall network behaviour.

In figure 4.2a the Mean Firing Rate and Coefficient of Variation of this neuron are plotted in the g-τ-plane. Comparing to the plotted values of MFR and C_V in the previous chapter, for instance figure 3.6, these single-neuron target values depend on g and τ in a much smoother way. Besides the statistical moments of this most connected neuron, in figure 4.2b we also plotted the mean MFR and mean C_V for the entire network. Together with this graph and figure 4.2a a good idea about the influence of g and τ on the statistical moments corresponding to the spiking behaviour of our network can be obtained.

What can be deduced from the graphs, is that in most regions of the parameter space, when increasing the conductance of a synapse also the Mean Firing Rate increases. Also the Coefficient of Variation increases when an increment in conductance is realised. An increase of the exponential time constant of the same order of magnitude is not as influential as the conductance, yet when plotted on a bigger τ-domain we see about the same resulting behaviour in statistical moments. Later on, we will see that these observations can partially be explained from a biophysical point of view.

Mathematically, the observation that when increasing both parameters

will lead to comparable results in the target values can be interpreted theoretically. No exact formulas for Mean Firing Rates and Coefficient of Variation are present with respect to g and τ. Yet, in section 3.3.3 we showed that a relation between g and τ exists. Expanding on that, indeed observations by picture are that along similar curves the target values remain approximately on a same level, whereas walking along a line where both g and τ increase gives a steep increase in our target values. This coincides with the observations about the g-τ-curve that causes the valley in the error function. These expectations are probably underpinned by the fact that level curves will in theory be orthogonal to the gradients of the error function, although we have not checked that for this specific case. By the absence of exact formulas expressing MFR and C_V in g and τ, we cannot provide the exact directions of the gradient, yet one might argue that in the direction where MFR and C_V of most neurons have the steepest increase, the gradient of the error function coincides with that direction. Indeed, especially figure 4.2a, indicates level curves in the target values that are similar to the discussed valley.

Now that target values of the most connected neuron and mean target values of the whole network are discussed, in the next section results corresponding to the neurons with highest amount of synapses will be isolated and discussed.

4.1.2 Membrane potentials and spiking patterns

Apart from analysing target values and error functions, since we now use single-neuron properties and membrane potentials, in this subsection we consider the influence of changes in g and τ to the single-neuron potentials and spiking patterns. Again, we only plot the most connected neurons.

Figures 4.3 and 4.4 show several plots of both membrane potentials and raster plots. The first series of plots, figures 4.3a to 4.3e, show membrane potentials of the neuron with the highest amount of synapses. The network structure for creating all five figures remained the same, apart from the fact that we changed our parameters g and τ, as can be seen from the captions of the figures. In figures 4.4a to 4.4e raster plots of the ten neurons with highest connectivities are shown. For every spiketime we plotted a little star, resulting in a series of stars for all spikes of those neurons.

From the plots of membrane potentials it is hard to recover any clear change in target values, yet what can be seen is a subtle change in behaviour. Comparing for instance the three figures in which the conductance g increases (figures 4.3a, 4.3c and 4.3d), we see that the potential trace in time becomes less regular. It seems as though a transition to nearly bursty behaviour takes place. Moreover, counting spikes, we do see what we already noticed in the precious section, namely that the number of spikes on a fixed time interval increases.
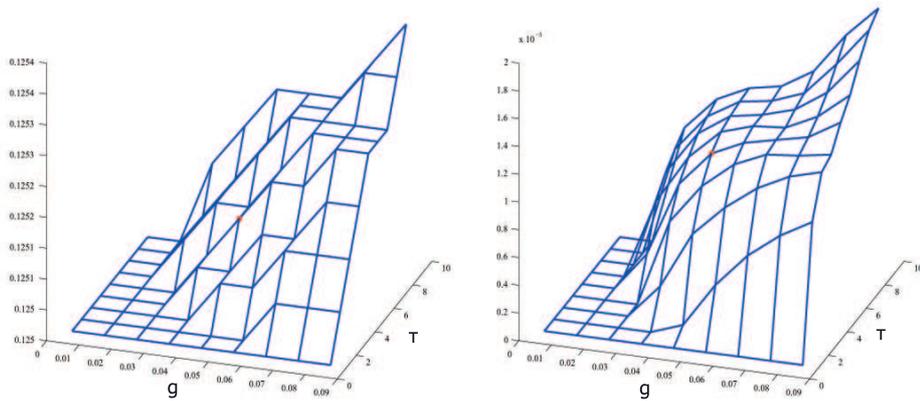
(a) Single neuron, $T = 40$.
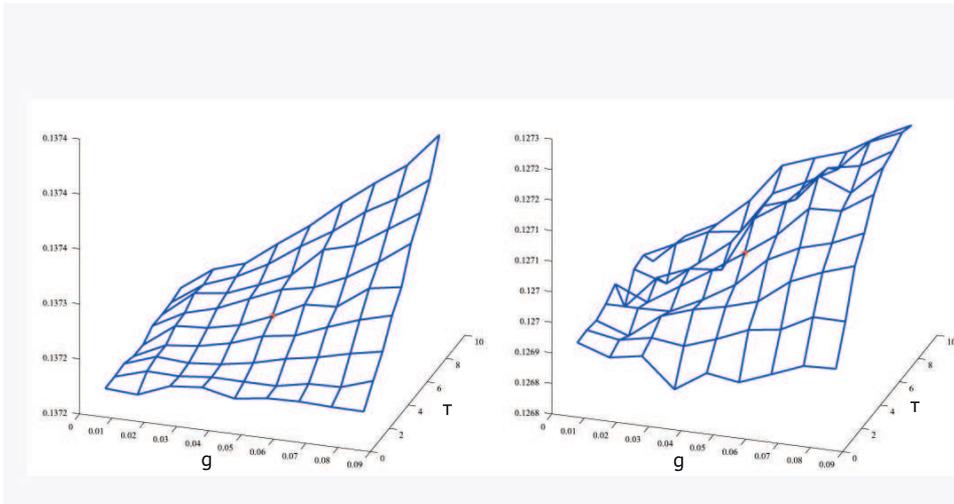
(b) Mean target values, $T = 40$.

Figure 4.2: a) Mean Firing Rate (left subplot) and Coefficient of Variation (right subplot) in g-τ-plane with $T = 40$ seconds for the neuron with highest amount of synapses. b) Mean MFR (left subplot) and mean C_V (right subplot) of all 600 neurons. In both cases the red dots represent the optimal point.

A second result from the graphs of membrane potentials, is that it is influenced less by τ. The spiking patterns do not change as much as with g, while the increase in τ with the figures 4.3b, 4.3c and 4.3e is actually bigger than the increase in g. This was also seen in the previous section with the investigation of the target values.

Regarding the raster plots these indeed show that with an increase in g the spikes start to cluster to patterns that start to resemble bursts with most of the neurons. Moreover it is again seen that with a bigger increase in τ these spikes also seem to cluster, though not as much as with an increase in g.

After observing these influences of g and τ to different properties of network behaviour, next we will discuss the upper bounds of the error functions with single neurons.

4.2 Error estimation

In this section we analyse the fact that with single-neuron data, closeby the optimal point (g^*, τ^*), the error function does not exceed numbers in the order of 10^{-4}. We discuss what this implies for our estimations and simulations of for instance the Mean Firing Rate and Coefficient of Variation with our chosen (g^*, τ^*).

The fact that our error is at most of order 10^{-4}, by equation (3.2.1), implies the following:

$$\|\mathbf{F}(g, \tau) - \mathbf{F}^*\|_2^2 \leq 10^{-4},$$

in which $\| \cdot \|_2$ again represents the 2-norm.

When we write out this expression, using $\mathbf{u}$ for the Mean Firing Rate and $\mathbf{v}$ for the Coefficient of Variation while not writing the g and τ dependence for simplicity, we obtain:

$$\frac{\|\mathbf{u} - \mathbf{u}^*\|_2^2}{\|\mathbf{u}^*\|_\infty^2} + \frac{\|\mathbf{v} - \mathbf{v}^*\|_2^2}{\|\mathbf{v}^*\|_\infty^2} \leq 10^{-4}, \tag{4.2.1}$$

in which, when having n neurons, $\mathbf{u}, \mathbf{v} \in \mathcal{R}^n$. In this equation we wrote out the scaling of Mean Firing Rates and Coefficient of Variation discussed at the end of section 3.2.2, in which

$$\|\mathbf{x}\|_\infty = \max_i x_i.$$

Equation (4.2.1) defines a region, which for one neuron is an elliptical surface in the **MFR**-$\mathbf{C_V}$-space in which our estimations are bounded. With this estimation, the worst case scenario is that one of the two vectors is estimated perfectly, and the other one with the biggest possible error. This implies (a similar expression can plainly be derived for $\mathbf{v}$):
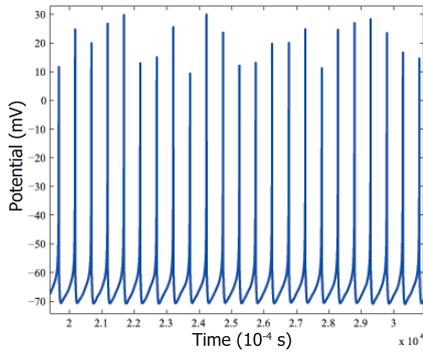
(a) $(g, \tau) = (0.01, 3)$
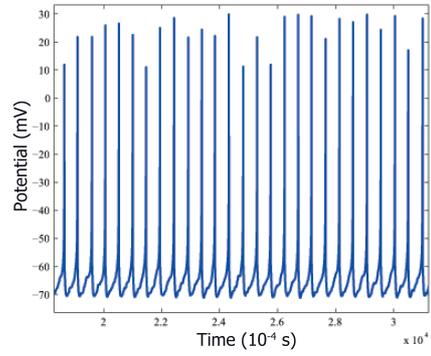
(b) $(g, \tau) = (0.075, 0.5)$
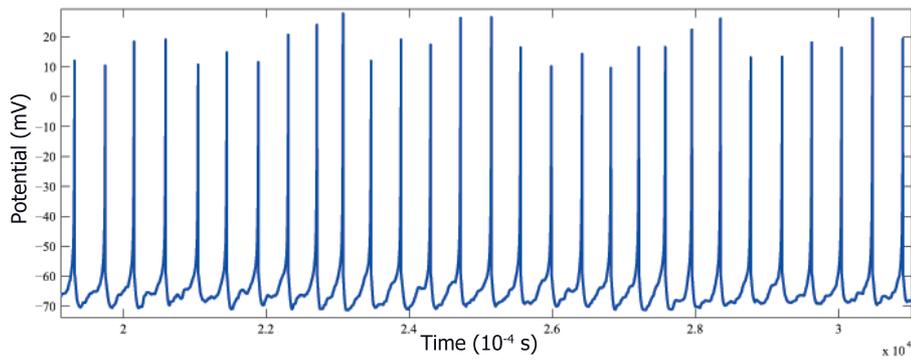
(c) $(g, \tau) = (g^*, \tau^*) = (0.075, 3)$

(d) $(g, \tau) = (0.14, 3)$

(e) $(g, \tau) = (0.075, 5.5)$

Figure 4.3: Membrane potentials of the neuron with highest connectivity for different values of (g, τ).
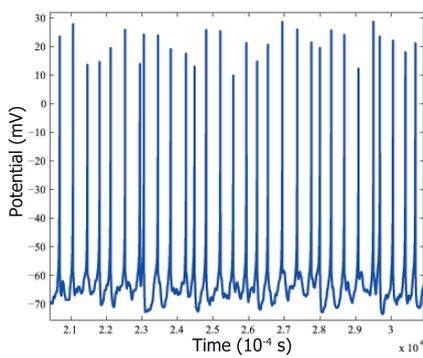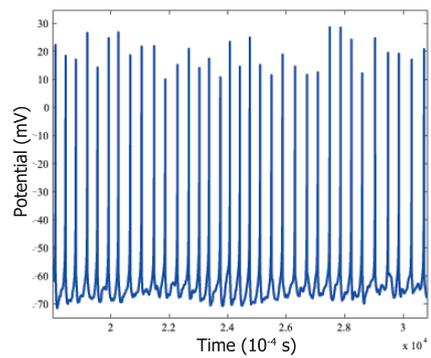
(a) $(g, \tau) = (0.01, 3)$

(b) $(g, \tau) = (0.075, 0.5)$

(c) $(g, \tau) = (g^*, \tau^*) = (0.075, 3)$

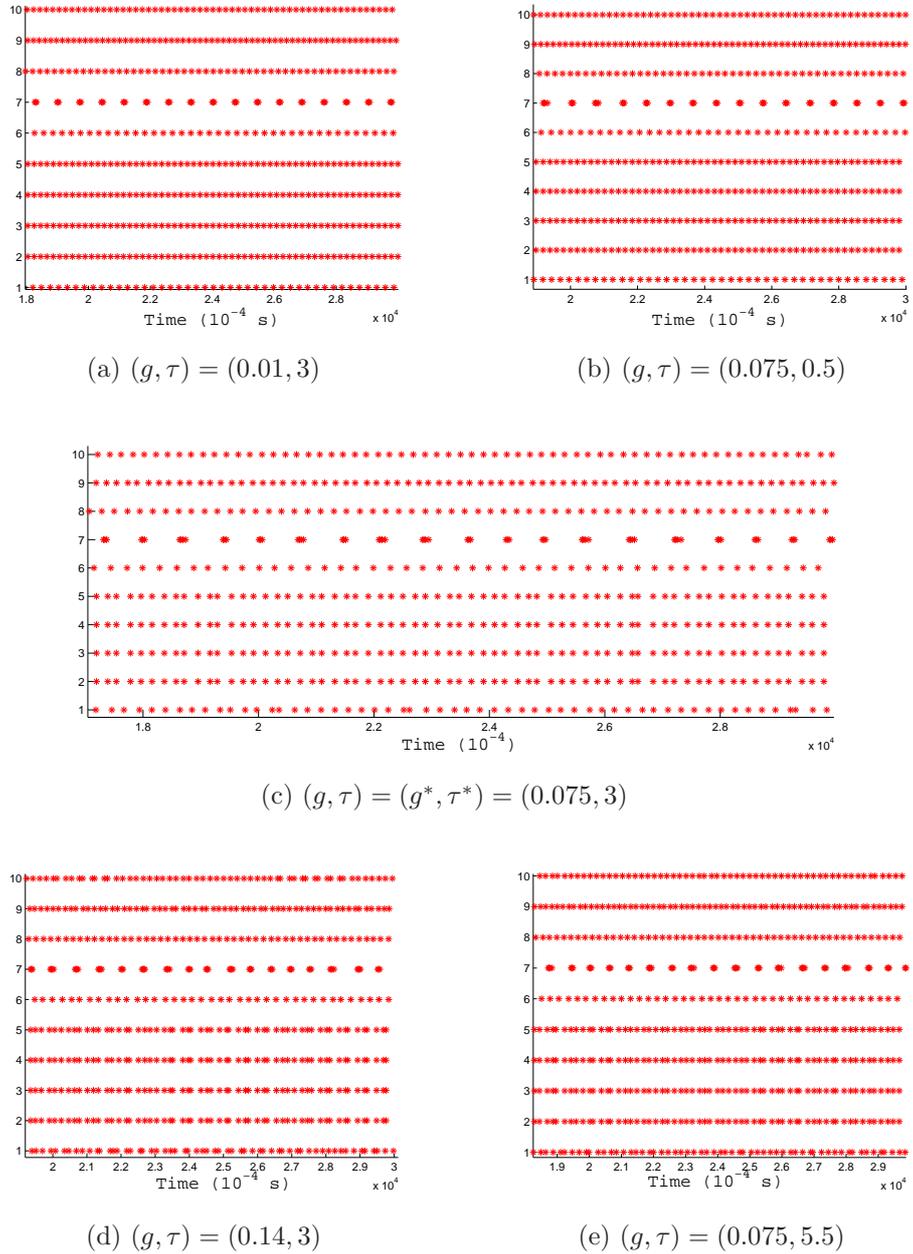(d) $(g, \tau) = (0.14, 3)$

(e) $(g, \tau) = (0.075, 5.5)$

Figure 4.4: Raster plots of the ten neurons with highest connectivity for different values of (g, τ).

$$\sum_{i=1}^{n} |u_i - u_i^*|^2 = 10^{-4} \|\mathbf{u}^*\|_{\infty}^2,$$

in which again the worst case scenario is estimating all components spot on, except for one which is estimated with the highest possible error, leading to:

$$|u_{i_0} - u_{i_0}^*| = 10^{-2} \|\mathbf{u}^*\|_{\infty}.$$

Using the triangular inequality, we can obtain the following bounds on the Mean Firing Rate of one neuron in this worst case scenario:

$$u_{i_0}^* - 10^{-2} \|\mathbf{u}^*\|_{\infty} \le u_{i_0} \le u_{i_0}^* + 10^{-2} \|\mathbf{u}^*\|_{\infty}.$$

Summarising, from the fact that our error function reaches at most values of order 10^{-4} we derived that with our simulations we estimate the Mean Firing Rates with a mistake of at most one percent of the maximum Mean Firing Rate. The same result holds for the Coefficient of Variation and is derived in exactly the same way.

4.3   Single-neuron Gauss-Newton

Since the error function and target values appear to depend on g and τ in a smoother way, we try the Gauss-Newton optimisation method for single neurons as well. Since in the previous chapter we used target values corresponding to every electrode, we change that in this case. We now have target values for every neuron. It should be realised that this is solely done as a benchmark procedure, because we do not have measurements of 600 single neurons and their target values. Though when this does not work, then there is no use to proceed with this method, since having statistical moments of every single neuron is about the best information one can get for converging to our optimal point.

Except excluding the MEAs and changing these target values, the method remains exactly the same as described in section 3.3. Moreover, we also use the $(g^*, \tau^*) = (0.075, 3)$ as used in that section again. A result is shown in figure 4.5, in which the red path shows the optimisation iterations and again the error function is shown as well.

This figure shows that, most likely caused by the error function obtaining small values and being flat on this mesh though again containing fluctuations on a finer mesh, the method does not converge to our global minimum. More specifically, the method cannot find any point in parameter space where the error is lower then with the initial point, not even when we replace the initial point on the slope of the valley. It keeps on refining stepsizes, recalculating Jacobians and performing random steps, causing immense computation times. Since time was too restricted to proceed any
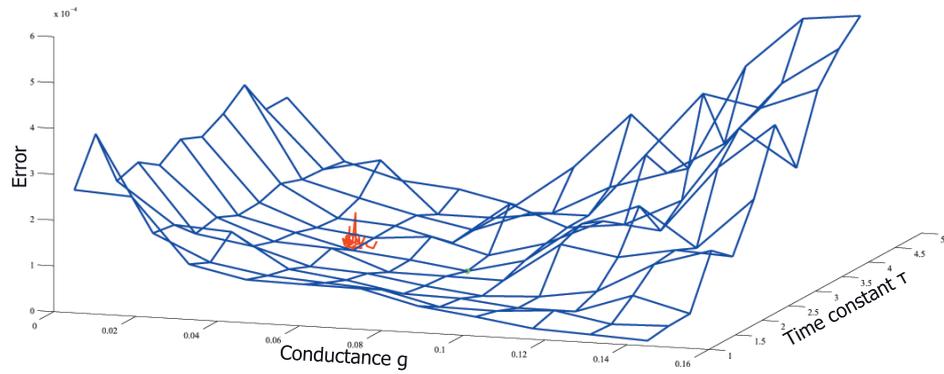
Figure 4.5: Single-neuron error function with a path (red) corresponding to the Gauss-Newton iterations in this single-neuron case. The green dot represents (g^*, τ^*).

further in this method, these are the only results we could show. Still, the smoothness of the error function suggests that in future research one should prefer single-neuron data instead of MEA data.

Chapter 5

Conclusions and recommendations

Based upon observations and results from previous chapters, here we will state final conclusions and moreover discuss some of them from a biophysical point of view.

Gauss-Newton method

By trying to recover a couple of chosen optimal values for g and τ in a benchmark procedure, we observed that the optimisation method we used, a Gauss-Newton mean least square error method, does not convergence to the constructed global minimum. Thus, the need for a computational framework to link experimental data to synaptic parameters is not yet met.

In this benchmark procedure we constructed a relatively small network and chose (g^*, τ^*). By using the corresponding Mean Firing Rates and Coefficients of Variation we attempted to recover those values for g and τ by using the optimisation method. Plainly, when one deals with data from measurements, one would be satisfied with points in parameter space not being the exact global minimum. Parameter values with which our computational network simulates data with similar statistical moments when compared to recordings, could be regarded as solutions. Though when already in a simple benchmark model the original (g^*, τ^*) cannot be recovered, then chances are that the output of the computational framework with data is not representative for the cultured network. Indeed, with the two-dimensional Gauss-Newton, we observed that the method iterates to different values for g and τ every time we run the optimisation. Moreover, it stops iterating when we set the threshold to a relatively large value, whereas most of the times with a small minimum score no parameter values with lower corresponding error were recovered. From these observations, we can conclude that one cannot assume convergence to the desired minimum value.

Since the two-dimensional optimisation did not converge, we developed

a one-dimensional optimisation method with the conductance g as optimisation parameter and in which τ was put in manually. This we did for several τ-values to create a semi-2D method and to investigate the possibility to construct the curve which we characterised as a valley in the graph of an error function. Moreover, we ran this optimisation for several values of the threshold error to see if we can extrapolate this value to zero. Yet, already with recovering a point in the valley with a low error not coinciding with our optimal point, could be satisfying. This would apparently have provided us with parameter values of which simulations resembled the desired Mean Firing Rate and Coefficient of Variation. Unfortunately, also this one-dimensional method does not convergence in a robust manner.

A prominent reason for this, is that error functions created in this benchmark procedure are highly non-smooth. At each iteration step, Gauss-Newton relies not only on the function value, yet also on the value of its derivatives collected in the Jacobian matrix. This makes the use of this method in this case unreliable. Up to now, we found no way to smoothen the function in such a way that the optimisation becomes successful, whereas we suggested not to use smoothing methods since they might affect the position of the global minimum.

One of the conclusions which we can state, is that the time horizon of every simulation in all iterations is influential to whether we obtain reliable statistical moments. When chosen to be $T = 2$ seconds for example, this leads to non-smooth error functions and to improve the smoothness of the error function we need time horizons of 40 seconds or more.

Besides the non-smoothness causing difficulties with determining the position of the global minimum, we also found that there is a relation between the parameters that we optimise for. This in turn causes the appearance of a valley in which values for the error function were relatively small. This makes determining the position of a clear minimum when zooming in on this valley even harder. Closely around (g^*, τ^*) the error function seems to be relatively flat, yet with the same fluctuations seen elsewhere. It is likely that this too is a partial cause for the non-robustness of our method.

Finally, we observed that the way we model MEAs cause extra difficulties, especially since detecting spikes becomes a genuinely hard problem. Moreover, the modeling of MEA concerns aggregating potential data based on geographical properties, which is biophysically indefensible. Therefore we advanced in analysing network behaviour with single-neuron data, which we will discuss in the next section.

Network behaviour

We have analysed both network and single-neuron behaviour in computational networks in which MEAs were not included. As a first observation the error function smoothened again, most likely caused by the absence of

difficulties with spike detection by omitting the MEAs. Moreover, the maxima of the error functions on a fixed domain, turned out to be smaller when compared to the aggregated data. This implies that we are able to simulate the desired behaviour better without MEAs.

Besides these observations, we investigated by means of 'eyeballing graphs' what the influence of g and τ to both statistical moments and individual behaviour is. Therefore we plotted mean target values for the whole network next to target values, membrane potentials and raster plots for the neurons with highest amount of synapses.

From a biophysical point of view, increasing the conductance of the synapses, allows more current to flow through. Since we have excitatory synapses, allowing more current implies that all postsynaptic membrane potentials increase with a bigger amount when a presynaptic neuron fires. This implies that the postsynaptic neuron is more likely to fire itself, thus following this reasoning one would expect the Mean Firing Rate to increase. Indeed, we saw that by increasing the conductance both MFR and C_V increased.

Following biophysical reasoning, increasing τ might not directly imply an increase or decrease in target values. Increasing τ means that synapses are active for a longer time after a spike, keeping the postsynaptic membrane closer to the firing threshold for a longer period. This reasoning would explain an increasing Mean Firing Rate, though at the same time it might also be reasonable that increasing τ leads to longer refractory periods of the synapse. Our results though state that increasing τ also increases both Mean Firing Rate and Coefficient of Variation.

Finally, we developed the Gauss-Newton method for this single-neuron case. Error functions seemed to be flat yet contained fluctuations on a denser mesh, which again lead to the fact that the iterations did not converge to the constructed global minimum.

Recommendations on future work

As shown, the current calibration of our computational framework does not allow us to accurately find a desired point in parameter space. Yet, several statements in this thesis and moreover such an essential research topic as Alzheimer's disease, should only stimulate more research into a computational framework that can assist in a quest for the role that $A\beta$ plays in the appearance of AD. Therefore, in this section we propose topics for future research.

At first, we showed a relation between the parameters that we chose to optimise for. With these related parameters we performed a two-dimensional optimisation. We recommend that in future research one also considers other parameters to be subject to this procedure. Besides this, in data that was at our disposal no measurements were performed regarding g and τ.

Moreover, considering the possible harms that Aβ causes these parameters were biophysically most plausible to change if Aβ is injected. For these two reasons we chose to optimise for g and τ. Yet with current technology the conductance and time constant of decay corresponding to a synapse are measurable on single-neuron basis. Thus we propose that in future measurements on both Aβ-treated and control networks the values of g and τ will become part of the experimental data. As a consequence, one could use the computational framework to optimise for other parameters that influence the network behaviour. For example, a change in the Izhikevich parameters influences the simulation results, which suggests optimising for both mean and variance of these five parameters. The use of a normal distribution is still valid when creating large networks. Besides this, the weights, of which little is known, influence the behaviour of our computational networks as well. Therefore, future research benefits from an optimisation procedure regarding these synaptic weights.

One last notion about weights versus conductance and time constant as choice for optimisation parameters, is another difference from which an extra suggestion might arise. Weights plainly are a network property, since it is assumed that there is a distance-dependent distribution of the weights throughout the network. The conductance and time constant we optimised for though, can also be interpreted as single-synapse properties. Since they are the same for every synapse, defining them for one single synapse coincides with defining them for the whole network. That is why we suggest that, for g and τ, instead of using computational networks as we did, an experiment in which they are measured on single-neuron scale will be performed.

The most prominent problem in our computational method, was the fact that our error function is non-smooth. It might be that in our case this function will never be smooth, since we work with the discrete notion of counting spikes. This implies that when the error changes, since we work with finite time, our total amount of spikes can increase with a minimal amount of one. Divided by the total number of spikes this is generally small, yet it is a jump. Unless one would be able to work with infinite time, which in principle we are not, this means that our function without modifications will not entirely be smooth.

We would recommend though, that when one has more computational power at hand, the whole procedure of constructing the desired computational framework is repeated with inclusion of an averaging over several network realisations. In this way varying initial conditions as well as positions and other network and neuron properties, are averaged out. The same holds for the effect of one extra spike discussed in the previous paragraph. The error function could smoothen out by doing this, since with a change in g and τ the change that we would expect to see, is one of average network behaviour. This could improve the optimisation procedure, though it worsens the computation time.

58

Finally, we did not regard the real data. When one does arrive at that stage in future research though, we suggest that as opposed to what was suggested in this thesis, simulation results of the computational networks are not converted to data resembling MEA data. This means that one has to go through the efforts of converting the experimental recordings that is MEA-based to data based on single neurons. This is worth the effort though, since we concluded that excluding MEAs in computational neuronal networks smoothens both the dependence of the target values on g and τ and the error functions.

Bibliography

[1] A. Alzheimer (1907) Über eine eigenartige Erkrankung der Hirnrinde. *Allgemeine Zeitschrift für Psychiatrie und phychish-Gerichtliche Medizin* **64**, pp. 146-148.

[2] T.V.P. Bliss and G.L. Collingridge (1993) A synaptic model of memory: Long-term potentiation in the hippocampus. *Nature* **361(6407)**, pp. 31-39.

[3] Y.A. Chen and R.H. Scheller (2001) SNARE-mediated membrane fusion. *Nature Reviews Molecular Cell Biology* **2(2)**, pp. 98-106.

[4] P.J. Hahn and C.C. McIntyre (2010) Modeling shifts in the rate and pattern of subthalamopallidal network activity during deep brain stimulation. *Journal of Computational Neuroscience* **28(3)**, pp. 425-441.

[5] J. Hardy and D.J. Selkoe (2002) The amyloid hypothesis of Alzheimer's disease: Progress and problems on the road to therapeutics. *Science* **297(5580)**, pp. 353-356.

[6] C. Holmgren, T. Harkany, B. Svennenfors, Y. Zilberter (2003) Pyramidal cell communication within local networks in layer 2/3 of rat neocortex. *Journal of Physiology* **551(1)**, pp. 139-153.

[7] A.L. Hodgkin and A.F. Huxley (1952) A quantative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* **117(4)**, pp. 500-544.

[8] B.T. Hyman, H.L. West, G.W. Rebeck, S.V. Buldyrev, R.N. Mantegna, M. Ukleja, S. Havlin, H.E. Stanley (1995) Quantitative analysis of senile plaques in Alzheimer disease: Observation of log-normal size distribution and molecular epidemiology of differences associated with apolipoprotein E genotype and trisomy 21 (Down syndrome). *Proceedings of the National Academy of Sciences of the United States of America* **92(8)**, pp. 3586-3590.

[9] E.M. Izhikevich (2003) Simple model of spiking neurons. *IEEE Transactions on Neural Networks* **14(6)**, pp. 1569-1572.

[10] E.M. Izhikevich (2004) Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks* **15(5)**, pp. 1063-1070.

[11] E.M. Izhikevich (2007) Dynamical systems in neuroscience: The geometry of excitability and bursting. *The MIT Press*, Massachusetts.

[12] I. Kuperstein, K. Broersen, I. Benilova, J. Rozenski, W. Jonckheere, M. Debulpaep, A. Vandersteen, I. Segers-Nolten, K. Van Der Werf, V. Sybramaniam, D. Braeken, G. Callewaert, C. Bartic, R. D'Hooge, I. Cristiano Martins, F. Rousseau, J. Schymkowitz, B. De Strooper (2010) Neurotoxicity of Alzheimer's disease Aβ peptides is induced by small changes in the Aβ42 to Aβ40. *EMBO Journal* **29(19)**, pp. 3408-3420.

[13] E. Korb and S. Finkbeiner (2011) Arc in synaptic plasticity: From gene to behavior. *Trends in Neurosciences* **34(11)**, pp. 591-598.

[14] I.B. Levitan and L.K. Kaczmarek (2002) The neuron: Cell and molecular biology. *Oxford University Press*, New York.

[15] R. Llinas, I.Z. Steinberg, K. Walton (1981) Relationship between presynaptic calcium current and postsynaptic potential in squid giant synapse. *Biophysical Journal* **33(3)**, pp. 323-351.

[16] S. Nakanishi (1992) Molecular diversity of glutamate receptors and implications for brain function. *Science* **258(5082)**, pp. 597-603.

[17] M.E.J. Newman (2003) The structure and function of complex networks. *SIAM review* **45(2)**, pp. 167-256.

[18] S.M. Potter and T.B. DeMarse (2010) A new approach to neural cell culture for long-term studies. *Journal of Neuroscience Methods* **110(1-2)**, pp. 17-24.

[19] G. Sberna, J. Sáez-Valero, K. Beyreuther, C.L. Masters, D.H. Small (1997) The Amyloid β-protein of Alzheimer's disease increases acetylcholinesterase expression by increasing intracellular calcium in embryonal carcinoma P19 cells. *Journal of Neurochemistry* **69(3)**, pp. 1177-1184.

[20] S. Shafii, S.E. Dillard, M. Hlawitschka, B. Hamann (2012) The topological effects of smoothing. *IEEE Transactions on Visualization and Computer Graphics* **18(1)**, pp. 160-172.

[21] W.R. Softky and C. Koch (1993) The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *Journal of Neuroscience* **13(1)**, pp. 334-350.

[22] W. Thies and L. Bleiler (2012) 2012 Alzheimer's disease facts and figures. *Alzheimer's and Dementia* **8(2)**, pp. 131-168.

[23] A.M. Thomson, V.E. Walker, D.M. Flynn (1989) Glycine enhances NMDA-receptor mediated synaptic potentials in neocortical slices. *Nature* **338(6214)**, pp. 422-424.

[24] A.J. Vincent, R. Gasperini, L. Foa, D.H. Small (2010) Astrocytes in Alzheimer's disease: Emerging roles in calcium dysregulation and synaptic plasticity. *Journal of Alzheimer's Disease* **22(3)**, pp. 699-714.

[25] M. Yamakage and A. Namiki (2002) Calcium channels - Basic aspects of their structure, function and gene encoding; anesthetic action on the channels - A review. *Canadian Journal of Anesthesia* **49(2)**, pp. 151-164.

[26] J. Yao, R.W. Irwin, L. Zhao, J. Nilsen, R.T. Hamilton, R.D. Brinton (2009) Mitochondrial bioenergetic deficit precedes Alzheimer's pathology in female mouse model of Alzheimer's disease. *Proceedings of the National Academy of Sciences of the United States of America* **106(34)**, pp. 14670-14675.