Modeling the Effects of the Extracellular Matrix on Neuron Dynamics

by

Emilie Fjørner

Thesis

for the degree of

Master of Science

Faculty of Mathematics and Natural Sciences
University of Oslo

August 8, 2015

Abstract

In this thesis I have taken a step towards establishing a mathematical model of the effect of the extracellular matrix on the firing rate of the neuron it surrounds, namely the Kazantsev model, as a model that can be used to further research the effects of the extracellular matrix in general, and the perineuronal nets in particular. Part of this work has been studying the dynamics of the model system, where we see that the model facilitates two separate activity dependent steady states, and how a short period of high activity can lead to a shift between these states. In the activity region between these states we see a tendency for the system to fluctuate relatively strongly in terms of the concentration of the ECM components. It remains to be studied further whether this fluctuation is simply an artifact in the model, or if it reflects legitimate behavior. The model describes a system where two contrasting time scales are at play, and to facilitate this I have an effort to make the model more computationally inexpensive by simplifying the spike modeling within the model. This is done by means of a simple integrate-and-fire model.

"Don't be afraid to scrape the paint off and do it again. This is the way you learn, trial and error, over and over, repetition. It pays you great dividends, great, great dividends."

Bob Ross

Acknowledgements

With this master thesis I am nearing the end of my education here at the university of Oslo. Although I am excited to see what my future will bring, it is with a sadness that I leave this era of my life behind me. I have greatly enjoyed my time here and I have learned a lot, both about physics and myself, and I have met many wonderful people.

I would like to show my appreciation for Fagutvalget as well as Fysikkforeningen and the wonderfully social and stimulating environment I got to be a part of during my time at the Lillefy study area during my bachelor's degree.

The good social environment continued as I began my master's degree, and I would like to thank the people at Computational Physics for being such open, generous, and friendly people. I would particularly like to thank the people I have shared an office with, first Mathilde and Øydis during my first year, and Wilhelm and Marte Julie during my second year. You have all helped make my days brighter.

I would of course also like to thank my supervisors, Morten Hjorth-Jensen, Marianne Fyhn and Gaute Einevoll. Thank you for all your help and encouragement, with a special thanks to Gaute for introducing me to the fascinating field of neuroscience, and for your helpful input during our regular meetings.

My family also deserve thanks, in particular my parents. You are always there to help me when I need it, such as when my moving date is three days before my thesis deadline, and I truly appreciate it.

Finally, my biggest and most loving appreciation goes to Jonas van den Brink. Your help, encouragement and support has been irreplaceable. Thank you for helping me with my figures, for your feedback on my work, and for always being there when I need you.

Contents

Chapter 1

Introduction and Motivation

Neuroscience is a wide, extensive and exciting field. It is a field where people from a wide range of disciplines and backgrounds work together towards a common goal of getting a better understanding of the complex and fascinating part of biology that is our nervous system. Even though we today believe we have a good understanding of the core units making up the nervous system, the nerve cells, and how they function individually, a lot of work remains before we can claim to fully understand the system they make up, and its many functions.

This is partly due to the great complexity of the nervous system and the brain in particular; billions of nerve cells interact in a multitude of ways giving rise to a myriad of varying functionality. Another complicating factor is the level of redundancy seen in this organ which can obscure a researche's view.

One of the bigger questions in this field that still needs answers concerns the creation and storage of long-term memories. There is currently little definitive knowledge regarding this undeniably important question. The process in question would require a mechanism facilitating storage of information for up to the duration of a life time, without too much deterioration occurring.

There are several current competing theories, most of which are based around different macromolecules present in the post-synaptic spines [Tsien, 2013]. However, due to metabolic turnover the lifetime of these molecules will be far shorter than that of our memories. These theories therefore require a process for copying information from old to new molecules using a mechanism robust enough to withstand thousands of duplications without too severe degradation of the original information.

A theory offering a differing approach is based on a molecular structure that can be found surrounding mature neurons, perineuronal nets (PNNs). The idea is that very long-term memories can be stored as holes in this matrix. These nets are believed to deter the formation of new synapses while stabilizing existing connections, and their formation has been found to be dependent on firing activity [Reimers et al., 2007].

In this thesis I have built my work around a model that (to my knowledge) is

the first major mathematical model focusing on the interplay between the extracellular matrix (of which PNNs are a special case) and the neuron it surrounds in terms of levels of firing activity and matrix composition.

This project is thus placed in the exciting junction between biology, physics, and numerical modeling, and relates to the early stages of a compelling theory for the formation of long-term memories and the functionality of the extracellular matrix. In it we will take a step towards establishing a numerical model that can contribute to further research and advances within this field.

1.1 Overview of This Text

This text is made up of eight chapters, plus a bibliography. The first three chapters after this one will give an introduction to the principles and background needed to get an understanding of the rest of the work. The first of these, chapter 2, begins by giving a quick overview of the neurobiology of the nervous system and in particular the brain. It also delves into the functionality of the above mentioned perineuronal nets and their suggested relation to the formation of long-term memories.

In chapter 3 we move on to look at the computational side of this field, focusing on the background needed to model the electrical activity in nerve cells. For this purpose we introduce the Hodgkin-Huxley model, and we also look at a simpler model called the integrate-and-fire model.

Following this, chapter 4 introduces the model that is the basis for this thesis work, the model I have dubbed the Kazantsev model. We look at the different components making up this model, from the spike generation modeled according to the Hodgkin-Huxley model from the previous chapter, to the components of the extracellular matrix.

Chapter 5 deals with my implementation of the model from chapter 4. Here I explain my choice of structure as well as programming language and environment, and I discuss the numerical schemes chosen.

In chapter 6 we find my results from exploring the implemented model. This chapter can be seen as having two parts; first the model is used to reproduce results from the original paper in order to verify my own implementation, before I investigate the model further. This includes studying its behavior under varying conditions, as well as seeing if it can be successfully simplified or expanded upon, depending on the needs at hand.

Chapter 7 brings us to a discussion. What have I found and what does it tell us? Should the model be modified, and if so, in what way? Here I also look towards future work and I contemplate what the way forward entails for the Kazantsev model.

Finally, in chapter 8, we reach a quick summary of the thesis as a whole.

1.2 Code

The code produced during my work on this thesis and used to create the results presented in the following chapters is available from:

https://emiliefj@bitbucket.org/emiliefj/neuron_model_with_ecm.git

Chapter 2

Introduction to Neurobiology

The focus of neuroscience lies in working towards understanding the mechanics of the nervous system, and how the (human) brain works. It is a highly interdisciplinary field with contributers stemming from a wide range of other fields, including biology, medicine, computer science and even linguistics. This is partly due to the broad scope of the field. Neuroscience can be described as the study of the nervous system from its cellular and molecular level all the way up to the behavioral level. Many aspects of the system are studied, including its development, structure, function and anatomy; its relation to learning, human behavior and psychology; and of course its medical aspects. At the same time this is done using a multitude of techniques ranging from behavioral studies to various imaging techniques, through to computational modeling.

In this chapter the goal is to give the reader a brief introduction to neurobiology, focusing on the nervous system and its structure. Although not necessarily essential to understanding content of the later chapters, I think it can be useful as a way of setting the scene for what is to come. Hopefully it will provide an overview of the greater picture, before we dive into the details. The contents of this chapter as well as the next are based mainly on what is given in [Squire et al., 2008], [Sterrat et al., 2011], and [Thompson, 1993].

2.1 The Nervous System

Almost all multicellular animals have some form of a nervous system, but its complexity varies greatly. In vertebrates the nervous system is divided into the central nervous system (CNS) and the peripheral nervous system (PNS). The CNS is made up of the brain and spinal cord, while the PNS includes all the nerves and ganglia outside of this, including motor neurons, the autonomic nervous system as well as the enteric nervous system. The PNS connects the CNS to the rest of the body, including limbs and organs.

The CNS makes up the majority of the nervous system, and it is distinguished

from the PNS not only by location but also by composition. While the PNS mainly consists of axon bundles, or nerves, stretching out to the extremities of the human body, the CNS is made up of so-called gray and white brain matter. Both these substances contain glial cells, with white matter contain more of them, and gray matter consisting mostly of neurons.

2.1.1 The Brain

The center of the nervous system is arguably the brain, and it is also the most complex of the organs, at least in vertebrates. Mammalian brains all have a similar structure, and this holds true also for our human brain. The main distinction of the human brain is its comparatively larger and more developed cerebral cortex.

The cerebral cortex is the thick, outer layer of the brain. It has a strongly folded structure which maximizes its surface area. When thinking of the brain it is typically this characteristic wrinkly and folded structure we imagine. It is commonly divided into sections referred to as "lobes"; the frontal, the parietal, the temporal, and the occipital lobe, see figure 2.1. The division into these lobes is mostly arbitrary, reflecting only the bones of the skull that overlie them. The lobes therefore do not indicate any functional division and all contain brain areas of limited functional relationship. The exception is the occipital lobe which is only involved in tasks relating to vision.

The cerebral cortex with its lobes makes up the greater part of what is known as the cerebrum. The cerebrum is split into a left and a right hemisphere, and it is the largest component of the human brain. Underneath it we find the hippocampi, one in each hemisphere. The hippocampus plays an important role in spatial navigation and the formation of long-term memories [Vianna et al., 2000].

The brain stem, a stalk like structure connecting the spinal cord with the cerebrum, lies underneath the cerebrum and to the back of the brain. In addition to transmitting signals between the brain and the rest of the body, it helps regulate the sleep cycle, as well as cardiac and respiratory function, and it is also crucial in maintaining consciousness. Behind it we find the cerebellum or "little brain" whose main function relates to motor control. It may also be involved in language, mental imagery and learning.

The thalamus as well as the hypothalamus are located in between the cerebrum and the brain stem. The thalamus has been found to act as a type of relay point between different sub-cortical ("below the cortex") areas as well as the cerebral cortex. It is believed to process as well as transfer the incoming signals. In addition it plays a central role in regulating both sleep and alertness. The hypothalamus has been found to a have a variety of functions, most importantly it links together the nervous system and the endocrine system. It also synthesizes

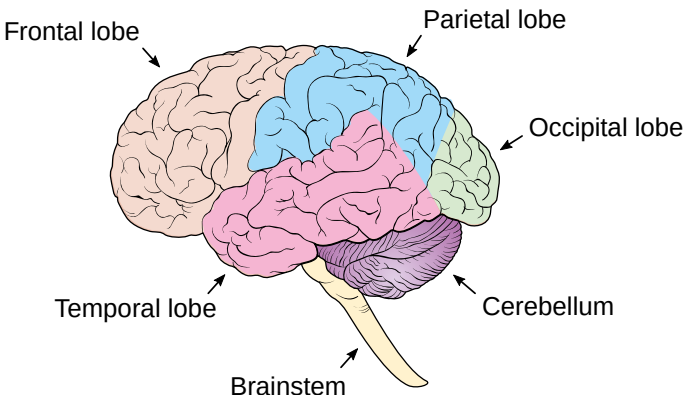

Figure 2.1: A schematic illustration of the human brain, showing the location of its main areas, as well as the lobes of the cerebrum. Figure adapted from Cancer Research UK / Wikimedia Commons.

and secretes neurohormones, such as oxytocin.

2.1.2 Neurons

The main building blocks of the nervous system are nerve cells, often called neurons. A nerve cell is a highly specialized, electrically excitable cell, and the human brain is estimated to contain roughly 85 billion of them [Williams and Herrup, 1988]. These neurons are interconnected through an even larger number of connections called synapses, where nerve signals are transferred between individual nerve cells.

Although different neurons can vary widely with respect to shape, size, and electrochemical properties, they share some common features. Like other cells a neuron is made up of a cell body surrounded by a cell membrane, and within this we find a nucleus as well as cytoplasm, mitochondria and other organelles. The cell body of a neuron is called the soma, and in contrast to other cell types the neurons have structures, or neurites, extending away from this soma, called dendrites and axons, see figure 2.2. Dendrites propagate incoming signals from other neurons to the soma, while outgoing signals are transmitted along the axon. Each neuron may have a large number of dendrites, but only one axon.

The cell membrane separates the inside and the outside of the cell. It is made up of a lipid bilayer making it impermeable to organic molecules and charged ions. However, the membrane does contain several pores called ion channels and ion pumps. These pores in the membrane are made up of proteins and they make it possible for selected ions to pass through the membrane.

There are many different so-called ion channels, and they are typically divided into two categories, active and passive. Active ion channels can be either open or closed to penetrating ions depending on factors such as the electric potential

across the membrane and the presence of neurotransmitters, while passive ion channels are "passive" in that they have a constant permeability. Generally ion channels are only permeable to a specific ion type.

Unlike ion channels, ion pumps do not simply allow ions to pass through, put actively pump specific ions and molecules against the concentration gradient.

With the use of ion channels and ion pumps, a potential difference is maintained between the inside and outside of the cell, by controlling the movement of charged ions across the membrane. This potential difference is known as the membrane potential, and the value it has when the neuron is unperturbed is called the resting membrane potential. It typically has a value of about -65 mV. As we shall see later, neurons make use of this membrane potential and the cell membrane's selective permeability when transferring information between each other.

Neurons are typically divided into three main groups which differ according to specialization:

- Motor neurons are neurons which cause muscle contractions when stimulated. The cell body of a motor neuron is located in the spinal cord, while its axons extend to the various parts of the body and control the muscles there.
- Sensory neurons are neurons whose function is to respond to various sensory input, such as sight, feeling, sound etc., and transmit the information to the brain and spinal cord.
- Finally, the interneurons connect together neurons in the same region, thereby facilitating communication between sensory or motor neurons, and the central nervous system.

Neurons can also be classified according to location, shape, or the type of transmitter they synthesize and release.

2.1.3 Glial Cells

The nervous system also contains another specialized cell type called glial cells. As their main function is to provide the networks of neurons with structural and metabolic support they are often referred to as the support cells of the neurons. The glial cells provide support mainly by holding the neurons in place and isolating them from each other, supplying them with necessary nutrients, and by destroying various pathogens and removing dead neurons. In newer research they have also been found to contribute in some degree to neurotransmission [Auld and Robitaille, 2003].

About half of the total volume of the brain and spinal chord is made up of glial cells, but the ratio of glia to neuron varies strongly in different parts of the

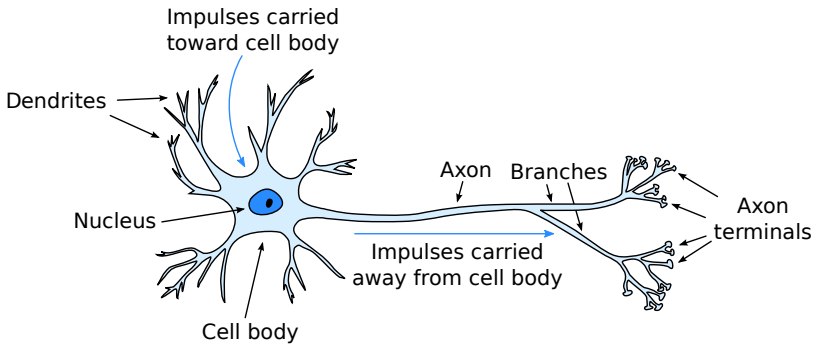

Figure 2.2: A schematic illustration showing the typical structure of a neuron. Figure adapted from [wpclipart, 2015].

brain. For instance about two-thirds of the cells in the cerebral cortex are glial cells, while they in the cerebellum make up less than one-fifth of the total number of cells.

2.1.4 The Action Potential

Most biological cells have an electric potential difference across the cell membrane, as we have seen is the case for neurons. However, in excitable cells such as neurons, this potential fluctuates and these fluctuations can lead to the generation of electrical signals called action potentials. The action potential is the basis of the communication between neurons. It is by means of the transmission of these electrical signals that information is transferred through the nervous system. The action potential itself is a fluctuation of the membrane potential identified by a characteristic, rapid rise followed by an equally rapid fall, see the illustration given in figure 2.3.

When measuring the voltage across the cell membrane of a neuron, one finds that there is an electrical potential difference across this membrane of about -65 mV. This is the resting membrane potential, and in the absence of perturbations the potential difference will stay in this steady state. The generation of an action potential is initiated by synaptic input from a connected neuron perturbing the affected neuron. Depending on the type of input, this can have two different effects on the membrane potential. Either the input is what is known as excitatory, which leads to a depolarization of the membrane (a rise in membrane potential), or it is inhibitory, which leads to a hyperpolarization instead (a fall in membrane potential below the resting potential).

Excitatory input may lead to the creation of an action potential. Often, several excitatory inputs must happen within a small time frame to result in an action potential. The excitatory input causes a local change in the permeability

of the membrane, allowing charged ions to flow through. This leads to an increase in the membrane potential, and if this response is large enough for the potential to exceed a threshold value (typically about 15 mV above the resting membrane potential) an action potential is created.

As mentioned in the previous section, the permeability of some ion channels will depend on the value of the membrane potential. This is for instance true for the sodium channels, and when the membrane potential exceeds the aforementioned threshold these channels respond by opening, allowing positively charged sodium ions to enter the cell. The membrane potential then rises in a rapid fashion. Following this change in membrane potential the potassium channels also open, and potassium ions exit the cell. This leads to a drop in the membrane potential, which generally overshoots the resting membrane potential leading to a hyperpolarization of the cell, before the resting value is finally reached. This phase of the action potential where the cell is hyperpolarized is known as the refractory period, during which the neuron is incapable of firing another action potential. The refractory period typically lasts about 1 ms. Figure 2.3 shows an illustration of a typical action potential.

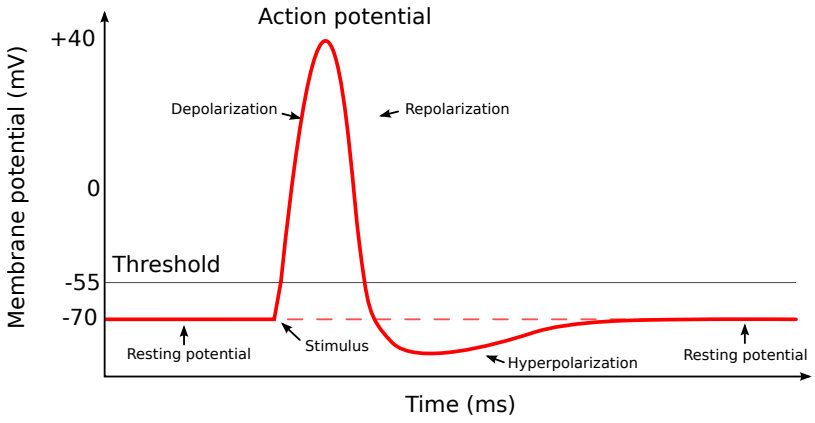

Figure 2.3: Plot of the membrane potential during a typical action potential. The potential is at its resting value, when a stimulus pushes it over the threshold value. This results in a sharp increase (depolarization) followed by a sharp decrease (repolarization) of the membrane potential, before it again returns to its resting value. Notice how the potential first undershoots the resting potential (hyperpolarization).

Very few ions need to cross the membrane of the cell for this large change in membrane potential to occur, and as such the changes in the internal and external concentrations of the involved ion species can in most cases be neglected.

The resulting action potential will propagate along the axon of the neuron spreading the signal to connected neurons. Different neurons display a large

disparity in how frequently they fire an action potential. This frequency is known as the firing rate of the neuron.

2.1.5 Synapses

Communication between neurons is facilitated by the action potentials, but the actual transmission of a signal from one neuron to another occurs through the synapse. A synapse is a connection between the axon of one neuron, and the dendrite or soma (or even axon) of another. There are two main categories of synapses; electrical and chemical.

The main mediators of neuronal communication are the chemical synapses. In a chemical synapse the action potential from the pre-synaptic neuron stimulates the pre-synaptic terminal in such a way that neurotransmitters are released. These neurotransmitters, which are endogenous messenger chemicals, then diffuse across the synaptic cleft, the narrow gap between the pre- and post-synaptic cell. In the membrane of the post-synaptic cell there are specialized neurotransmitter receptors which bind to the diffusing neurotransmitters. This binding can have different effects depending on the type of receptor that is activated. Some receptors are excitatory, and increase the receiving neuron's firing rate, others are inhibitory, decreasing the firing rate, or they can be modulatory, meaning that their effect is not directly linked to firing rate.

An electrical synapse is an electrical conductive link between two bordering neurons. Here the membrane potential can be directly transmitted between neurons through a type of pore that connect the neurons directly across the intracellular space between them. Such a pore is known as a connexon, and it is a type of channel made up of six proteins. Each electrical synapse contains several such pores and these pores actually cross the membrane of both cells, thereby connecting their cytoplasms, and allowing ions and signaling molecules to travel from one neuron to the next.

The electrical synapses allow for much more rapid transfer of signals between neurons, but they admit for less varied response to these signals. Where a signal stemming from a chemical synapse can lead to everything from a larger response in the post-synaptic neuron than the original signal, to a negative response, the resulting signal transferred by an electrical signal will always be equal or smaller than the signal it originates from.

2.2 Perineuronal Nets and Long Term Memory

Our bodies as well as all other forms of living tissue are made up up a large number of different cells, and surrounding these cells is what is known as the extracellular space. This space is filled with an intricate structure called the extracellular matrix (ECM). The relative amount of ECM varies widely between

different types of tissue, with it being most abundant in connective tissue, and comparatively scarce in the central nervous system (CNS) [Frantz et al., 2010]. It is not only the amount of the ECM that varies, its structure also differs strongly between the different types of tissue, and this results in the wide range of existing tissue types. For instance the ECM in bone tissue creates a thick, mineralized structure, and as a result bone tissue is hard and rigid. In other parts of our body, such as for instance the brain, the tissue is found to be much more soft and elastic, as the ECM here has a different composition.

The main components of the ECM are fibrous proteins and proteoglycans, as well as water. Proteoglycans are large molecules consisting of a protein core with long carbohydrate polymer chains called glycosaminoglycans (GAGs) attached. Of the fibrous proteins collagen is the most abundant. It is strong and stretch-resistant, and as such it provides strength to tissue. Other notable examples of fibrous proteins are elastin, which gives tissue a stretchy property, and fibronectin which connects cells with the collagen fibers.

By varying its composition the ECM can give tissue a wide range of different properties, but it has also been found to have a strong impact on the cells it surrounds. In addition to providing structural support for these cells, the ECM has been found to influence cell differentiation, development, and function.

As mentioned there is comparatively little ECM in the CNS compared to other tissue. However, the ECM that is located in the CNS has been found to be highly involved in brain development and it seems to be directly involved in regulating plasticity in this area [Wang and Fawcett, 2012, Tsien, 2013, Kwok et al., 2011]. A specialized form of ECM known as the perineuronal nets (PNN) is found only in the CNS. These nets surround the soma and dendrites of certain sub-populations of neurons, and are mainly found in the cortex, thalamus, hippocampus, brainstem and spinal chord. Although these structures were discovered in the late 19th century, they were not widely studied until the past few decades. An image showing the PNN in the cortex of a rat is shown in figure 2.4.

PNNs surround mature neurons, and although the full scope of their role is not yet fully understood, they have been found to have several functions, including stabilizing the milieu of the neurons they envelope and protecting them from harmful agents [Karetko and Skangiel-Kramska, 2009]. The PNNs have also been found to limit neuronal plasticity and regeneration, and restrict synapse formation [Tsien, 2013]. It has been proposed that very long-term memories may be stored as the holes in the PNNs by strengthening selected synapse connections while deterring the formation of new ones. This theory is based on the Hebbian plasticity hypothesis, popularly summarized with the saying “cells that fire together, wire together”. Due to being linked to neuroplasticity and memory formation, the PNNs have become an increasingly popular research topic in the later years.

Neuroplasticity or brain plasticity is a term describing the possibility of

changes occurring in synapses and neuronal pathways in response to varying conditions. Plasticity refers to the quality of being modifiable or changeable, and it was long thought that the brain was not a particularly plastic organ. However, more recent research has shown otherwise [Greenough, 1988]. Plasticity is tightly linked to learning, memory and neural development.

Brain plasticity is greatest by far during so-called critical periods. These are periods where the brain exhibits strongly heightened plasticity allowing the CNS to undergo large-scale changes, both physical and functional. These critical periods occur in early development, and are the reason why it is much easier for a child to learn a new language, for instance, than it is for an adult. In the adult brain the plasticity is much lower, although some events such as for instance brain damage can lead to a temporary increase in plasticity in and around the damaged area.

As it turns out, the end of the critical period coincides with the formation of PNNs, and removal of these nets through the use of a bacterial enzyme has been shown to increase plasticity in adult rat brains matching that of a critical period [Hockfield et al., 1990]. This discovery is what led to the belief that PNNs and plasticity are tightly linked.

Now that we have gotten a brief introduction to neurobiology, we move on to look at how biological entities like neurons are modeled in the field of computational neuroscience.

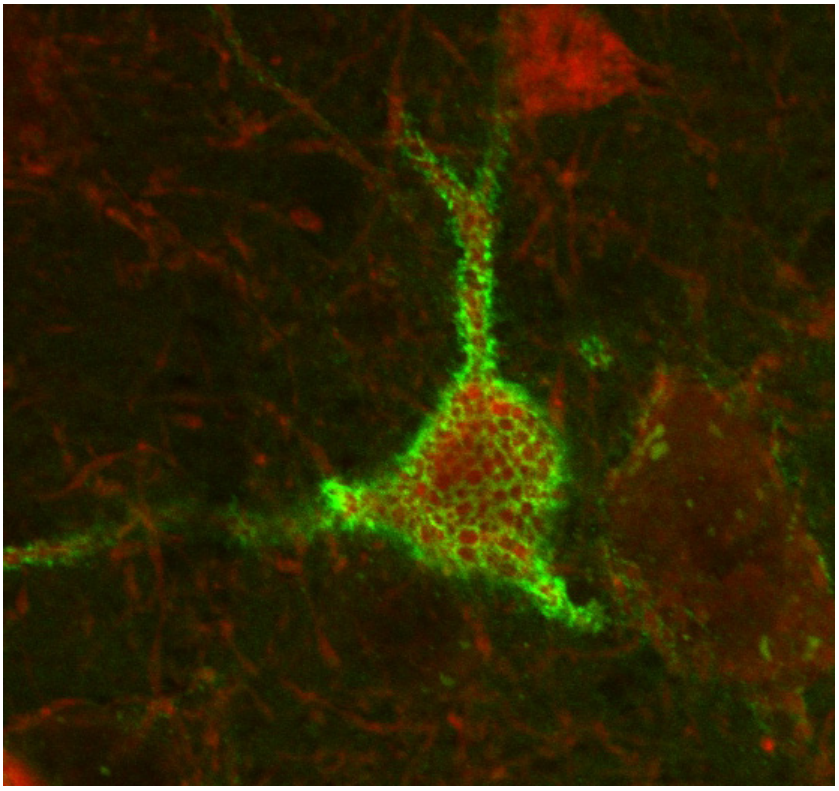

Figure 2.4: Image created with the use of confocal microscopy showing the PNN surrounding a neuron in a rat cortex. The PNN is shown in green due to being dyed with Wisteria Fluribunda Agglutin (WFA). Image taken by Kristian Lensjø.

Chapter 3

Computational Neuroscience

The nervous system is a large and complex system. It can be studied at many levels, and as such a wide range of models have been and continue to be constructed and analyzed to describe it and understand how it functions. This includes models of the individual nerve cell, synapses, the movement of specific ions, and the behavior of networks of neurons, to mention some examples.

One such model, the Hodgkin Huxley model, aims to model the neuron and its electrical properties. Created in the early 1950s, this model is often regarded as a gold standard amongst biological models. There are several reasons for this. The model is conceptually simple, its components have direct connections to the biological constituents, and it is experimentally verifiable to a high degree of accuracy. To this day the model is still widely used, albeit often with some additions or alterations.

In this chapter the goal is to give an overview of the Hodgkin-Huxley model, but to better understand the basis of this model and how it came to be, we begin by looking at the mechanisms behind the electrical activity of neurons. We will look at what drives the movement of charged ions in the neuron, and how the semi-permeable neuronal membrane influences the resting membrane potential. We follow this with an overview of the Hodgkin-Huxley model and its components, before ending the chapter by looking at a model taking a simpler approach, the integrate-and-fire model.

3.1 Modeling Electrical Activity in Neurons

The Hodgkin-Huxley model is a mathematical model of how action potentials are initiated and propagated. The model is the result of the impressive work done by Hodgkin and Huxley studying the squid giant axon, which resulted in five landmark articles published in 1952 as well as the 1963 Nobel Price in Physiology or Medicine. Although later experiments have revealed some shortcomings of this model, it is still widely used both in its original and extended forms.

The Hodgkin-Huxley model is a quantitative model of a neuron's active membrane properties, and it is based on experiments performed on the squid giant axon, popularly used in experiments due to its large size (up to 1 mm in diameter), making it easier to perform measurements on. The base of the model is the electrical circuit shown in figure 3.1. It is very common to describe the neuronal membrane in this way, using an equivalent electrical circuit, due to the electrical properties of its components. The membrane itself is in many ways like a capacitor, while the ion channels can be represented by a battery and a resistor, as they allow for current to flow across the membrane.

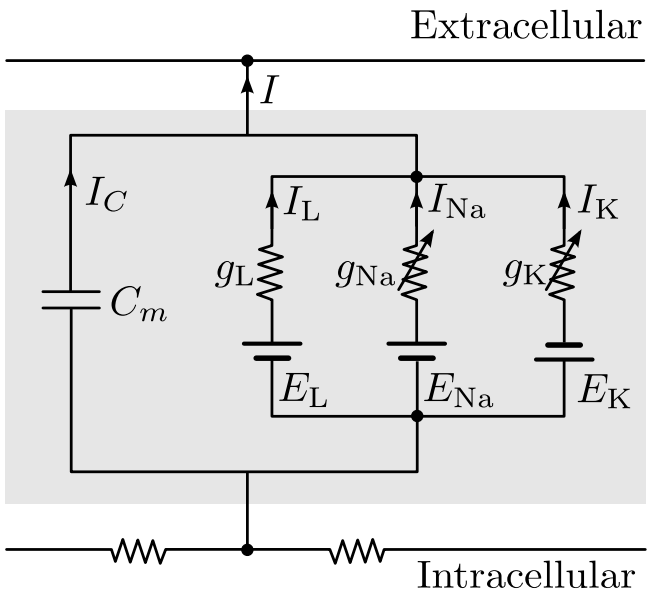

Figure 3.1: The equivalent circuit of a cell membrane compartment as in the Hodgkin-Huxley model. Figure inspired by figure 3.2 on page 50 in [Sterrat et al., 2011].

In later years many different ion channels have been discovered, but they were not known in Hodgkin and Huxley's day, and as so their model includes only three ionic currents: a potassium current, I_K, a sodium current, I_{Na}, as well as a leak current, I_L. This leak current is the current resulting from other ion types flowing through the membrane, mostly chloride.

Before we introduce the specific expressions making up the Hodgkin-Huxley model we will look at the concepts and dynamics that stand behind it, and how these can be described mathematically. This will give us a better understanding of the model and the reasoning behind it.

3.1.1 The Nernst-Planck Equation

Both the extracellular and the intracellular medium contain charged ions, with the mediums being separated by the cell membrane. The potential is more negative on the inside than the outside, creating an electric field across the membrane. In this way the membrane acts as a capacitor. As we saw in the previous chapter the basis of the electrical activity in neurons is the flow of charged ions across this membrane. We will now first look more closely at the dynamics governing this flow.

The electrical field across the cell membrane leads to an electric force on the charged ions. The magnitude of this force will depend on the charge of the ions, as well as the strength of the field, and the force will lead to a drift of charged ions, called electric drift. Across the membrane the particles move in narrow ion channels, and within these channels movement can be seen as occurring in one dimension. The resulting flux for a particle species X moving in one dimension becomes the sum of these contributions:

$$J_{\mathrm{X,drift}} = -\frac{D_{\mathrm{X}}F}{RT} z_{\mathrm{X}}[\mathrm{X}]\frac{\mathrm{d}V}{\mathrm{d}x}. \tag{3.1}$$

In this expression D_{X} is the diffusion coefficient of molecule X, z_{X} is this ion type's valency, meaning the charge of the ion in number of elementary charges, R is the gas constant, F is Faraday's constant, and T is the temperature.

In addition to this electric drift, a difference in the concentration of the various ions on the inside versus the outside of the cell will give rise to diffusion. Diffusion is the net movement of particles down their concentration gradient. This means that the particles are inclined to move from areas of high concentrations to areas of low concentration, thereby evening out the concentration difference.

The molar flux of one particle species X, in one dimension, resulting from diffusion is according to Fick's law given by:

$$J_{\mathrm{X,diff}} = -D_{\mathrm{X}}\frac{\mathrm{d}[\mathrm{X}]}{\mathrm{d}x}, \tag{3.2}$$

The total flux of ion type X then becomes:

$$J_{\mathrm{X}} = -D_{\mathrm{X}}\left(\frac{\mathrm{d}[\mathrm{X}]}{\mathrm{d}x} + \frac{Fz_{\mathrm{X}}}{RT}[\mathrm{X}]\frac{\mathrm{d}V}{\mathrm{d}x}\right). \tag{3.3}$$

This equation is called the Nernst-Planck equation, here given in its one dimensional form. As we have now seen it describes the movement of charged ions in the form of flux. In other, more precise, words it describes the amount of ion species X flowing through a unit area cross-section of the membrane, per time. In our applications we are more interested in the current resulting from this flux, or more specifically the current density. The flux describes the movement

of the ions themselves, while current is instead the movement of charge, and as such the current density can be found as:

$$I_{\mathrm{X}} = F z_{\mathrm{X}} J_{\mathrm{X}}, \tag{3.4}$$

since Faraday's constant is given by Avogadro's constant (the number of particles in one mole) multiplied by the unit electric charge.

3.1.2 A Permeable Membrane

Now we have seen how the concentration gradient as well as the electric field across the neuronal membrane affects the movement of the charged ions on both sides of it. At the time of Hodgkin and Huxley's ground breaking work, not much was yet known of the membrane itself, and it was for instance not yet known that it contained ion channels allowing for specific ions to pass through the membrane. Yet the membrane potential was seen to change rapidly during the creation of an action potential, and this implied that the membrane had to be permeable to ions. Let us now take a look at what the implications of this selective permeability are.

We begin by looking at a membrane that is permeable to only one ion type. Other ions are not allowed to cross the membrane. From the Nernst-Planck equation we see that the flux of this ion type will be a combination of electric drift and diffusion. If we imagine that one side of the membrane has a higher concentration of ions, the result will be a flux across the membrane due to diffusion. This will create a surplus of charge on one side of the membrane, resulting in electric drift in the opposite direction of the diffusion. Eventually these two contributions will balance each other out, and the net flux becomes zero.

At this steady state we have what is called the equilibrium potential, E_{X}, of the ion type; the potential difference across the membrane where there is no net flux.

The Nernst-Planck equation gives us this steady state:

$$[\mathrm{X}]\frac{\mathrm{d}[\mathrm{X}]}{\mathrm{d}x} = -\frac{F z_{\mathrm{X}}}{RT}\frac{\mathrm{d}V}{\mathrm{d}t}. \tag{3.5}$$

Solving this with the concentrations on the inside and outside of the membrane denoted $[\mathrm{X}]_{\mathrm{in}}$ and $[\mathrm{X}]_{\mathrm{out}}$, respectively, we are left with the following expression for the equilibrium potential:

$$E_{\mathrm{X}} = \frac{RT}{F z_{\mathrm{X}}} \ln \frac{[\mathrm{X}]_{\mathrm{out}}}{[\mathrm{X}]_{\mathrm{in}}}. \tag{3.6}$$

E_{X} is then called the Nernst potential for ion type X, often also referred to as the equilibrium or reversal potential.

3.1.3 The Goldman-Hodgkin-Katz equations

The neuronal membrane is, however, not only permeable to one ion type, but to several. To find the equilibrium potential in this case we can use the result known as the Goldman-Hodgkin-Katz (GHK) current equation [Sterrat et al., 2011]. This formula predicts the current across the membrane resulting from a specific ionic species for a given membrane potential V. Making several assumptions;

1. The ions cross the membrane independently,
2. The electric field within the membrane is constant, and
3. The movement of ions is determined by the internal concentration gradient and the electric field across the membrane,

they found the following expression for the current resulting from the flux of ion species X:

$$I_{\mathrm{X}} = P_{\mathrm{X}} V \frac{(z_{\mathrm{X}} F)^2}{RT} \left(\frac{[\mathrm{X}]_{\mathrm{in}} - [\mathrm{X}]_{\mathrm{out}} \exp(-V \frac{z_{\mathrm{X}} F}{RT})}{1 - \exp(-V \frac{z_{\mathrm{X}} F}{RT})} \right). \tag{3.7}$$

In this model of the membrane the permeability, P_{X}, is treated as homogeneous, and proportional to D_{X}.

To find an expression for the equilibrium potential we set the total membrane current I to zero, use one GHK current equation for each of the contributing ions, and solve for voltage. For a membrane permeable to sodium, potassium and chloride ions, the result is:

$$E_{\mathrm{m}} = \frac{RT}{F} \ln \frac{P_{\mathrm{K}}[\mathrm{K}^+]_{\mathrm{out}} + P_{\mathrm{Na}}[\mathrm{Na}^+]_{\mathrm{out}} + P_{\mathrm{Cl}}[\mathrm{Cl}^-]_{\mathrm{in}}}{P_{\mathrm{K}}[\mathrm{K}^+]_{\mathrm{in}} + P_{\mathrm{Na}}[\mathrm{Na}^+]_{\mathrm{in}} + P_{\mathrm{Cl}}[\mathrm{Cl}^-]_{\mathrm{out}}}. \tag{3.8}$$

It is readily seen that this reduces to the earlier result from the Nernst-Planck equation, for a membrane permeable to only one ion type.

The GHK current equation gives us an expression for the current resulting from the flux of a given ion type for membrane potential V. However, it does require that we know the concentration of said ion type both outside and inside the cell. As it turns out, a simpler approximation of this equation will often suffice in the voltage range where cells normally operate, namely the straight line:

$$I_{\mathrm{X}} = g_{\mathrm{X}}(V - E_{\mathrm{X}}). \tag{3.9}$$

Here E_{X} is the equilibrium potential of ion type X, and g_{X} is the conductance per area. Conductance is simply the multiplicative inverse of resistance.

Now that we have gained a better understanding of how we model current flow across the neuronal membrane, and the membrane voltage resulting from

this flow of charged ions, we move on to introduce the Hodgkin-Huxley model and its components.

3.2 The Hodgkin-Huxley Model

As previously mentioned Hodgkin and Huxley included three currents in their model, namely a sodium current, a potassium current and a leak current. In their experimental results they found that the conductances for the potassium and sodium ions were dependent on voltage, and in searching for a way to model this behavior they found that the concept of gating particles gave a good fit. The idea is that the membrane contains a number of gates, and these gates can be either open or closed to passing ions. The state of each gate is decided by a set of independent gating particles. These gating particles can each be either in a permissive or in a non-permissive state. For a gate to be open, all its gating particles must be in the permissive state. One can express the movement of a gating particle between its two states, permissive (P) and non-permissive (NP), as a chemical reaction:

$$\mathrm{NP} \underset{\beta}{\overset{\alpha}{\rightleftharpoons}} \mathrm{P}.$$

Here the variables α and β are rate coefficients describing the rate of the reaction in either of the two directions. The voltage dependence of the conductances is incorporated through these rate coefficients, whose value will depend on the current membrane voltage.

Hodgkin and Huxley found the conductance of the membrane to potassium ions to be best described by a gate made up of four identical and independent gating particles, which they dubbed n:

$$g_{\mathrm{K}} = \overline{g}_{\mathrm{K}} n^4, \tag{3.10}$$

where $\overline{g}_{\mathrm{K}}$ is the maximum potassium conductance.

The gating variable n then represents the probability of a potassium gating particle being in the permissive state, and the rate coefficients for the movement of the potassium gating particles between the permissive and non-permissive state are dubbed α_n and β_n. The first order kinetic equation then gives how n changes with time:

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(1-n) - \beta_n n. \tag{3.11}$$

The cell membrane of a typical neuron will contain many ion channels of each type. There will then be a large number of potassium gating particles, and the proportion of these that are in the permissive state will then be approximately equal to n, the probability of one particle being in the permissive state. The

proportion of gates that are open is then given by the same expression giving the probability of one gate being open, n^4, as seen above. The specific expression with n to the fourth power was chosen because it gave a good fit to the experimental data.

Similarly the sodium ionic conductance was found to be best described by two different gating particles, m and h. Equivalently to n, m can be described as an activation particle giving the rate of gating particles that are in the permissive state, and its time development is governed by the equivalent equation:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(1-m) - \beta_m m. \tag{3.12}$$

The variable h on the other hand is often referred to as an inactivation variable, and it represents a gating particle that can be in one of two states, an inactivated state, or a non-inactivated state. The behavior of this variable with time is given in the same way as for n:

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(1-h) - \beta_h h, \tag{3.13}$$

with α_h and β_h being the rate coefficients of the transition between the two states of the particle.

The resulting model of the sodium conductance uses three independent m gating particles, and one inactivation particle h:

$$g_{\mathrm{Na}} = \overline{g}_{\mathrm{Na}} m^3 h. \tag{3.14}$$

The final current, the leak current, was modeled using a simpler approach, as they assumed this current was a resting background current, and as such its conductance is independent of the voltage:

$$g = \overline{g}_{\mathrm{L}}. \tag{3.15}$$

Putting all these elements together we arrive at the full Hodgkin-Huxley model:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = I_{\mathrm{ext}} - (I_{\mathrm{K}} + I_{\mathrm{Na}} + I_{\mathrm{L}}), \tag{3.16}$$

with the various currents modeled as:

$$I_{\mathrm{K}} = g_{\mathrm{K}}(V - E_{\mathrm{K}}) \tag{3.17}$$

$$I_{\mathrm{Na}} = g_{\mathrm{Na}}(V - E_{\mathrm{Na}}) \tag{3.18}$$

$$I_{\mathrm{L}} = \overline{g}_{\mathrm{L}}(V - E_{\mathrm{L}}). \tag{3.19}$$

Here E_{X} is the equilibrium potential of ion type X, as we saw earlier in this chapter.

The net contribution of axial current from neighboring regions is represented by the variable I_{ext}. In experimental conditions known as space clamp conditions the membrane potential is constant over the membrane, and as a result this term reduces to zero.

To summarize we have now seen that the Hodgkin-Huxley model uses a set of four ordinary differential equations to model how the membrane potential, V, changes with time. These differential equations are highly nonlinear, and as such the system can not be solved analytically. Instead we must use numerical methods to solve them approximately. This is what Hodgkin and Huxley did, solving for each time step by hand. Naturally this was a lot of work, and as a result they did not calculate the entire slope of every action potentials. They saw that the action potentials all had the same shape in the repolarization phase, and as such stopped once they had passed the peak. This is shown clearly in the top half of figure 3.2, taken from the final of their iconic 1952 articles. Thankfully, today we can let computers do the dirty work for us.

3.3 Integrate-and-Fire Neuron

The Hodgkin-Huxley model of a neuron is based on experimental measurements of real neurons, and is constructed with the goal of accurately reproducing their behavior. As such it includes elements such as ion currents, a membrane capacitance and so forth, elements known to play a part in determining a neuron's electrical activity. Building a model that represents the neuron being modeled as accurately as possible is an obvious objective, and as such the Hodgkin-Huxley model has been expanded and modified in a myriad of ways to create far more detailed and complicated models.

The integrate-and-fire model of a neuron, as well as similar models, represent a different approach all together. In this model all but the essentials are stripped away, leaving us with a very simple, if not very biological model. This may seem counter-intuitive, but does offer some benefits. For one, a simpler model is also one where it will be easier to understand and interpret the core mechanisms at work. Simplicity can bring clarity where added details can confuse and obstruct. Simpler models are also useful if one wants to study networks of neurons. Simulating each and every neuron to painstaking detail is then often computationally unfeasible, and describing only the essential function of the neurons is often satisfactory.

In the integrate-and-fire model one has moved away from the standard procedure of describing the activity of the neuron using a set of coupled, non-linear differential equations. Instead of accurately trying to model the underlying mechanisms, this phenomenological model is based on the simple fact that an action potential is fired whenever the threshold for the membrane potential is reached. As action potentials are generally virtually identical, there is no need to model

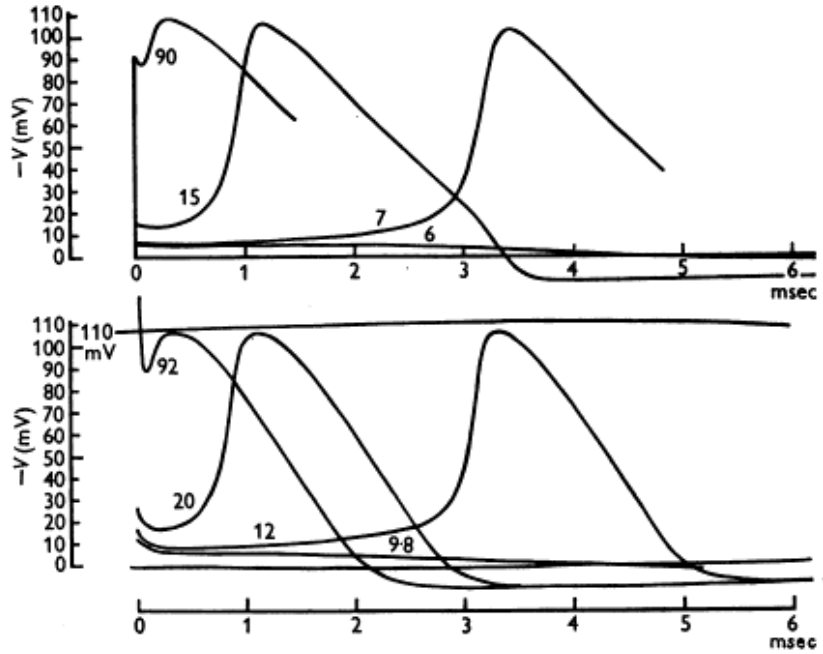

Fig. 12. Upper family: solutions of eqn. (26) for initial depolarizations of 90, 15, 7 and 6 mV (calculated for 6° C). Lower family: tracings of membrane action potentials recorded at 6° C from axon 17. The numbers attached to the curves give the shock strength in mμcoulomb/cm². The vertical and horizontal scales are the same in both families (apart from the slight curvature indicated by the 110 mV calibration line). In this and all subsequent figures depolarizations (or negative displacements of V) are plotted upwards.

Figure 3.2: Figure taken from the last of the four 1952 articles by Hodgkin and Huxley, namely "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve" [Hodgkin and Huxley, 1952].

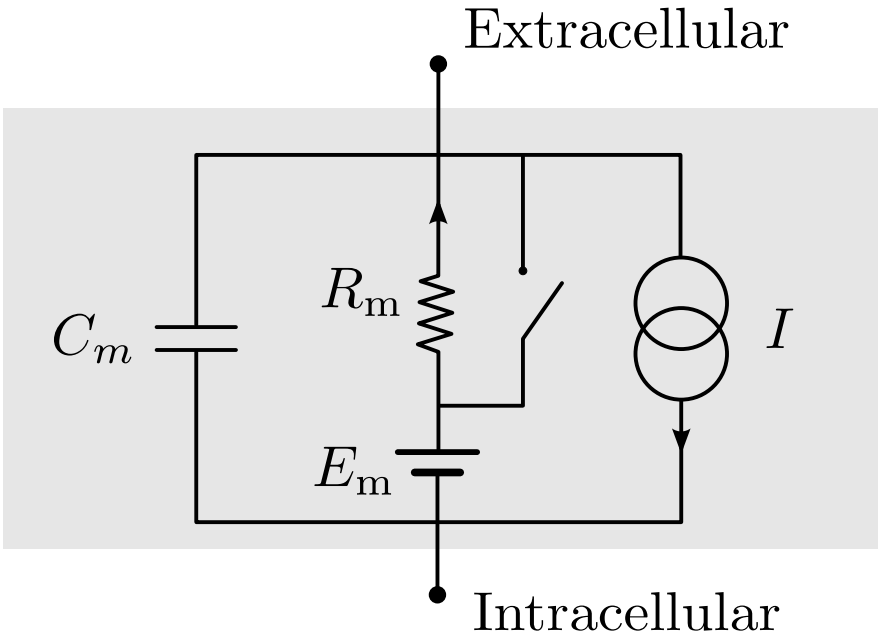

Figure 3.3: The RC-circuit that is the basis of the IF model, with a switch that closes when the threshold needed for an action potential to be generated, is reached. Figure inspired by figure 8.5(a) on page 204 in [Sterrat et al., 2011].

them explicitly.

The basis of the integrate-and-fire model is a simple RC-circuit with a reset mechanism in the form of a switch, as seen in figure 3.3. Incoming current leads to the membrane voltage increasing, and when the threshold value is reached, the switch closes, effectively short-circuiting the membrane resistance and thereby bringing the membrane potential back to its resting value.

Below this threshold value the voltage follows the equation for an RC circuit found from Kirchoff's current law:

$$C_m \frac{\mathrm{d}V}{\mathrm{d}t} = -\frac{V - V_m}{R_m} + I. \tag{3.20}$$

Here C_m represents the membrane capacitance as before, while I is the total current into the cell. The parameter V_m represents the resting value of the membrane potential V, while R_m is the membrane resistance seen in figure 3.3.

The current I could be any incoming current, be it from an electrode or connected synapses, and due to this current the membrane voltage V increases with an initially high derivative that decreases exponentially. If the current is large enough, the voltage will reach the threshold, θ, a spike is fired, and the switch closes resetting the voltage to V_m.

For a (constant) above-threshold current I this model will lead to a constant firing frequency for the modeled neuron, with an increase in I giving an increase in this frequency. If the current is below the threshold, the neuron will never fire.

To make the model more realistic one can also implement a refractory period following the firing of a spike, in which the voltage does not increase (above the threshold).

Chapter 4

The Kazantsev Model

The starting point of this thesis has been the article "A Homeostatic Model of Neuronal Firing Governed by Feedback Signals from the Extracellular Matrix" by Kazantsev et al. [Kazantsev et al., 2012]. In this article a first attempt is made at creating a model that simulates how neuronal spiking is influenced by the surrounding extracellular matrix (ECM). The model is based on existing experimental results indicating how the various components of the ECM influences the firing rate of the neuron it surrounds, as well as how this firing again impacts the production and destruction of these components. This knowledge is used in a qualitative way to create a simplified mathematical model.

The following chapter gives an overview of the model, as presented in [Kazantsev et al., 2012].

4.1 Model Outline

The focus of this model is to gain insight in how the ECM affects the neuron it surrounds in terms of how it influences the neuron's firing rate. The activity of the neuron is modeled using the standard Hodgkin-Huxley formalism receiving input from connected neurons in the form of a Poisson spike train. The ECM is represented through the concentration of its various components, and these concentrations will influence the neuron's tendency to fire an action potential, the strength of the synaptic input, as well as the concentrations themselves. The illustration given of the model in the article is shown in figure 4.1, whereas a more detailed schematic figure giving an illustrative overview of the model components is shown in figure 4.3.

4.1.1 The Neuron and the Average Activity

The generation of spikes by the neuronal cell is modeled using the standard Hodgkin-Huxley equations, with a few minor additions. The membrane potential

is governed by the following differential equation:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -(I_{\mathrm{mem}} - I_{\mathrm{th}} + I_{\mathrm{syn}}). \tag{4.1}$$

Here I_{mem} is the sum of the transmembrane currents responsible for spike generation; a potassium current, a sodium current, as well as a leak current:

$$I_{\mathrm{mem}} = I_{\mathrm{Na}} + I_{\mathrm{K}} + I_{\mathrm{leak}}. \tag{4.2}$$

These transmembrane currents are given by the standard Hodgkin-Huxley formalism, as seen in chapter 3:

$$I_{\mathrm{K}} = g_{\mathrm{K}}(V - E_{\mathrm{K}}) \tag{4.3}$$
$$I_{\mathrm{Na}} = g_{\mathrm{Na}}(V - E_{\mathrm{Na}}) \tag{4.4}$$
$$I_{\mathrm{L}} = \overline{g}_{\mathrm{L}}(V - E_{\mathrm{L}}). \tag{4.5}$$

The line over the conductance for the leak current indicates that this is treated as a constant. The other two conductances are treated in the standard way using the concept of gating particles, with:

$$g_{\mathrm{Na}} = \overline{g}_{\mathrm{Na}} m^3 h, \tag{4.6}$$

and

$$g_{\mathrm{K}} = \overline{g}_{\mathrm{K}} n^4. \tag{4.7}$$

Here, the gating variables $m,\ n$, and h follow the equations:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \alpha_x(1 - x) - \beta_x x, \quad x = m, n, h. \tag{4.8}$$

The term I_{th} is an applied current which regulates the effective spike excitation threshold. Higher values of I_{th} result in hyperpolarization of the neuron, which means that a larger input is needed for the membrane potential to reach the threshold value, and an action potential to be generated. The final current contribution, I_{syn}, describes the total synaptic input to the neuron, here modeled in the form of a Poisson spike train.

The typical duration of an action potential is in the magnitude of about 1 ms, but the changes occurring in the ECM are very much slower than that, taking from hours to days. Because these regulations in ECM concentration are happening on a very long timescale compared to the typical duration of an action potential, a variable is introduced describing the average activity of the neuron. This variable, Q, is given by:

$$\frac{dQ}{dt} = -\alpha_q Q + \beta_q H_q(V), \quad H_q(V) = \frac{1}{1 + \exp(-V/k_q)}. \tag{4.9}$$

Here α_q is a rate constant, and β_q is a scaling coefficient with $0 < \alpha_q < \beta_q$. k_q is the inverse slope of the activation function $H_q(V)$, restricted by $k_q < 1$. For n spikes generated with an average frequency $f = n/T$, $Q(n)$ will then converge to the limit in the following way:

$$Q_\infty = \lim_{\substack{(n,t)\to(\infty,\infty) \\ f=\text{const.}}} Q(n) = \frac{\beta_q \tau \exp(-\alpha_q/f)}{(1-\exp(-\alpha_q/f))} \approx \frac{\beta_q \tau}{\alpha_q} f. \tag{4.10}$$

This variable, Q, can be seen as a sliding window average of the neuron's firing activity, and it quantifies the activity level of the neuron over longer timescales. It is this variable that will influence the creation and destruction of the various components of the ECM, a process happening over quite long timescales.

4.1.2 Neuronal Input

In the Kazantsev model the synaptic input is modeled in the form of a Poisson spike train. This input stems from the fact that a neuron, such as the one modeled here, will be part of a large network of neurons communicating through the sending and receiving of action potentials. In the Kazantsev model the assumption is made that the spiking events experienced by the neuron can be seen as being uncorrelated, in which case modeling them as Poisson processes is a sensible approach.

The synaptic input being in the form of a Poisson spike train means that the timing of each incoming spike event follows the discrete probability distribution known as the Poisson distribution. Let us first look a bit more closely into this distribution, before we continue on to see how we can make sure that I_{syn} follows it.

The Poisson distribution gives the probability that a certain number of events will occur in a finite time interval. If a discrete random variable X is the number of successes resulting from a Poisson experiment, we call it a Poisson random variable and its probability distribution will be this Poisson distribution. The Poisson distribution is given by the Poisson formula:

$$P(k;\lambda) = Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{4.11}$$

Here $k = 0, 1, 2, ...$ represents the number of successes (or firing events in our case) while λ represents the average number of successes within a region, or the expected value of X. For a Poisson random variable it so happens that the expected value coincides with that variable's variance:

$$\lambda = \mathrm{E}(X) = \mathrm{Var}(X). \tag{4.12}$$

We begin by defining what we will refer to as the neural response function for a spike train, $\rho(t)$:

$$\rho(t) = \sum_{i=1}^{k} \delta(t - t_i). \tag{4.13}$$

This function represents a sum of pulses, with each pulse corresponding to an action potential. In this expression k is the total number of spikes or action potentials in the spike train, and t_i are the spike occurrence times. The unit impulse signal $\delta(t)$ is defined as:

$$\delta(X) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

The neural response function can be seen as a list of spike times in the spike train, and from it we can find an expression for the spike count, n, meaning the number of spikes in the spike train in the time interval between t_1 and t_2:

$$n = \int_{t_1}^{t_2} \rho(t)dt. \tag{4.14}$$

The neural response function also gives us an expression for the instantaneous firing rate:

$$r(t) = \langle \rho(t) \rangle. \tag{4.15}$$

Combining equation 4.14 and 4.15 we find the following expression for the average spike count:

$$\langle n \rangle = \int_{t_1}^{t_2} r(t)dt. \tag{4.16}$$

If the time interval $t_2 - t_1 = \delta t$ is small enough the average spike count can be approximated by:

$$\langle n \rangle = r(t)\delta t. \tag{4.17}$$

Extending on this, δt can be reduced until the probability of more than one spike occurring within this interval is small enough to be ignored. Then the average spike count becomes equal to the probability of firing a single spike.

Combining the result from equation 4.17 with the Poisson formula from equation 4.11, we get:

$$P(n \text{ spikes during } \Delta t) = \frac{(r\Delta t)^n e^{-r\Delta t}}{n!}, \tag{4.18}$$

for the probability of n spikes occurring in the time interval Δt. Here we have assumed a homogeneous Poisson process, which implies that the instantaneous firing rate is a constant, $r(t) = r$.

Let us take a quick recap. So far we have defined the neural response function $\rho(t)$ and used this to find an expression for the average spike count, which, for a homogeneous Poisson process became $r\Delta t$. From this we were then able to find the expression in equation 4.18 for the probability of finding n spikes within the interval Δt.

Now, in order to implement the synaptic input in the model what we need is the spike times; when does the neuron being modeled experience synaptic input from the surrounding neurons? To do this we begin by looking again at a time interval. We call this time interval τ. What is now the probability that no spikes occur within this interval? Using equation 4.18 with $n = 0$ and $\Delta t = \tau$ we find:

$$P(\text{next spike occurs after } \tau) = e^{-r\tau}. \tag{4.19}$$

The probability that the next spike occurs before the time interval τ is up is then:

$$P(\text{next spike occurs before } \tau) = 1 - e^{-r\tau}. \tag{4.20}$$

Equation 4.20 is the cumulative distribution function for the probability of a spike occurring within the interval τ. The probability density function for the waiting time until the next spike is the derivative of this cumulative distribution:

$$p(\tau) = \frac{d}{dt}\left(1 - e^{-r\tau}\right) = re^{-r\tau}. \tag{4.21}$$

Now our work has paid off, and we have arrived at an expression for the probability density function for the time in between the spikes in the spike train. Drawing from this distribution gives us the time until the next spike of the synaptic input, for synaptic input in the form of a Poisson spike train.

Each spike is given an amplitude drawn from the following probability distribution:

$$P(x) = \frac{2x}{b^2} \exp\left(-\frac{x^2}{b^2}\right), \tag{4.22}$$

$$\int_0^{+\infty} P(x)\ \mathrm{d}x = \Gamma(1) = 1. \tag{4.23}$$

Here Γ is the gamma function, and b is the scaling factor accounting for the effective strength of the synaptic input.

4.1.3 The Extracellular Matrix

In this model the ECM surrounding the neuron is represented through three variables. While Z describes the concentration of ECM molecules, R describes the

concentration of ECM receptors, and P represents the concentration of extracellular proteases.

As mentioned the basis of this model is various experimental results quantifying the effects of the ECM on the neurons it surrounds, and vice versa. Experimental observations have shown that neuronal firing stimulates ECM production. The average activity, or firing rate, is represented by Q, and an increase in Q will therefore lead to an increased value for Z. In addition to this, increasing concentrations of ECM molecules have been shown to increase the excitation threshold for action potential generation. This is accounted for through the threshold current's dependence on the concentration Z, $I_{th}(Z)$.

The ECM may slowly dissolve spontaneously and degrade due to the activation of extracellular proteases, and experimental results show that higher activity levels lead to an increase in production, secretion and activity of numerous proteases. To account for this behavior P will therefore increase when Q is large, while increasing concentrations of P will lead to a decrease in Z.

ECM receptors such as integrins, also play an important factor, and the assumption is made that their concentration, R, depends on the average activity level. Experiments indicate that R is higher if the average neuronal activity decreases below some critical level. This too is accounted for in the model.

In other words, increased neuronal activity is taken to increase the concentration of ECM molecules as well as proteases, while decreasing the production of receptors. At the same time the available proteases will cause degradation of the ECM molecules.

To account for how the ECM then influences the firing rate of the neuron, the production of ECM molecules and the expression of ECM receptors are assumed to be statistically uncorrelated processes, and the synaptic strength, meaning the strength of the synaptic connection to other neurons, is taken to be proportional to the product ZR. Figure 4.3 shows an illustration of these effects.

To account for the ECM-mediated homeostatic regulation, the article proposes a two-part feedback circuit to summarize these effects. This feedback circuit aims to describe the basic effects the ECM-influence has on neuronal excitability as well as the efficacy of synaptic transmission. This circuit comprises two basic feedback mechanisms:

- The excitation threshold is regulated through I_{th} in an activity dependent way. I_{th} will depend on both the concentration of ECM molecules and proteases, and will lead to fluctuations in the resulting firing rate of the neuron.

- The concentration of ECM-receptors will influence the effective strength of the synaptic input to the neuron, which again will influence the firing rate.

This circuit is in [Kazantsev et al., 2012] visualized as in figure 4.1.

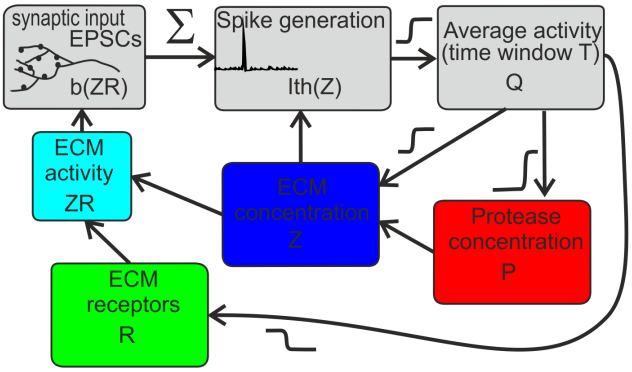

Figure 4.1: Schematic illustration of ECM-induced homeostatic regulation of average activity, as given in [Kazantsev et al., 2012].

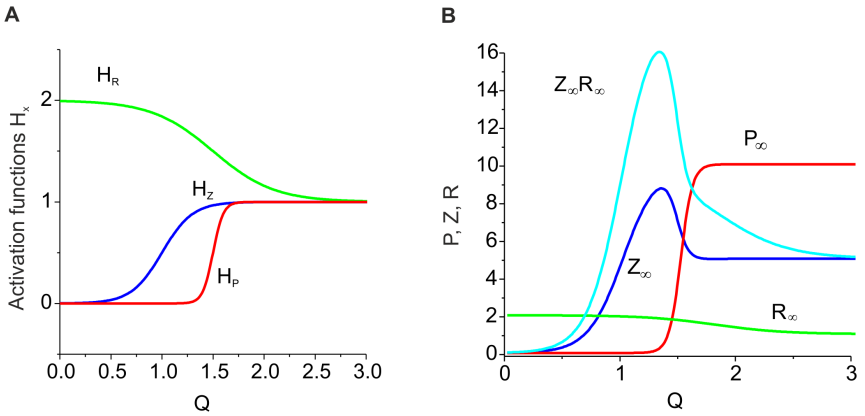

Figure 4.2: Figure (A) shows how the activation functions of Z, R and P depend on Q, while (B) shows steady state distribution profiles for Z_∞, P_∞ and R_∞ as a function of Q.

On the basis of recreating the mechanisms described, the following mathematical model for ECM-induced homeostatic regulation of firing rates is proposed. The following three differential equations govern the time evolution of the concentration of ECM molecules, proteases and receptors:

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = -(\alpha_z + \gamma_p P)Z + \beta_z H_z(Q, z_0, z_1, \theta_z, k_z) \tag{4.24}$$

$$\frac{\mathrm{d}P}{\mathrm{d}t} = -\alpha_P P + \beta_p H_p(Q, p_0, p_1, \theta_p, k_p) \tag{4.25}$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = -\alpha_r R + \beta_r H_r(Q, r_0, r_1, \theta_r, k_r). \tag{4.26}$$

Here the activation functions $H_{z,p,r}$ are approximated with a two-level sigmoid function of the following form:

$$H_x(Q, x_0, x_1, \theta_x, k_x) = x_0 - \frac{x_0 - x_1}{1 + \exp\left(-\frac{(Q-\theta_x)}{k_x}\right)},\ x = z, p, r. \tag{4.27}$$

These activation functions describe the activation kinetics for the concentrations of ECM molecules, proteases and ECM receptors. The parameters x_0 and x_1 are the asymptotic levels as $Q \to \pm\infty$, respectively, θ_x is the activation midpoint, and k_x is the inverse slope of the activation curve. The parameters α_x define the rate of spontaneous degradation of ECM concentration, proteases and ECM receptors, while β_x describes the activation rate of these variables.

The parameters where chosen based on phenomenological observations.

In addition, the neuronal dynamics are modulated following two feedback functions:

$$I_{th} = I_{th}(Z) = I_0(1 + \gamma_Z Z) \tag{4.28}$$

$$b = b(ZR) = b_0(1 + \gamma_{ZR} ZR). \tag{4.29}$$

Through equation (4.28) the activity level of the neuron is modified by changing the effective excitation threshold. This is modeled in the simplest form, by changing the depolarization level necessary to elicit an action potential. The feedback gain is described by the parameter γ_Z.

In a similar way, the synaptic weights are modified following equation (4.29), depending on the product ZR and with gain γ_{ZR}. The result is a potentiation or depression of the synaptic input. The feedback mediated by proteases is implemented in equations (4.24)-(4.26) as a nonlinear relaxation of the ECM concentration controlled by the gain parameter γ_P.

Now that the Kazantsev model has been introduced, we move on to look at my implementation of it.

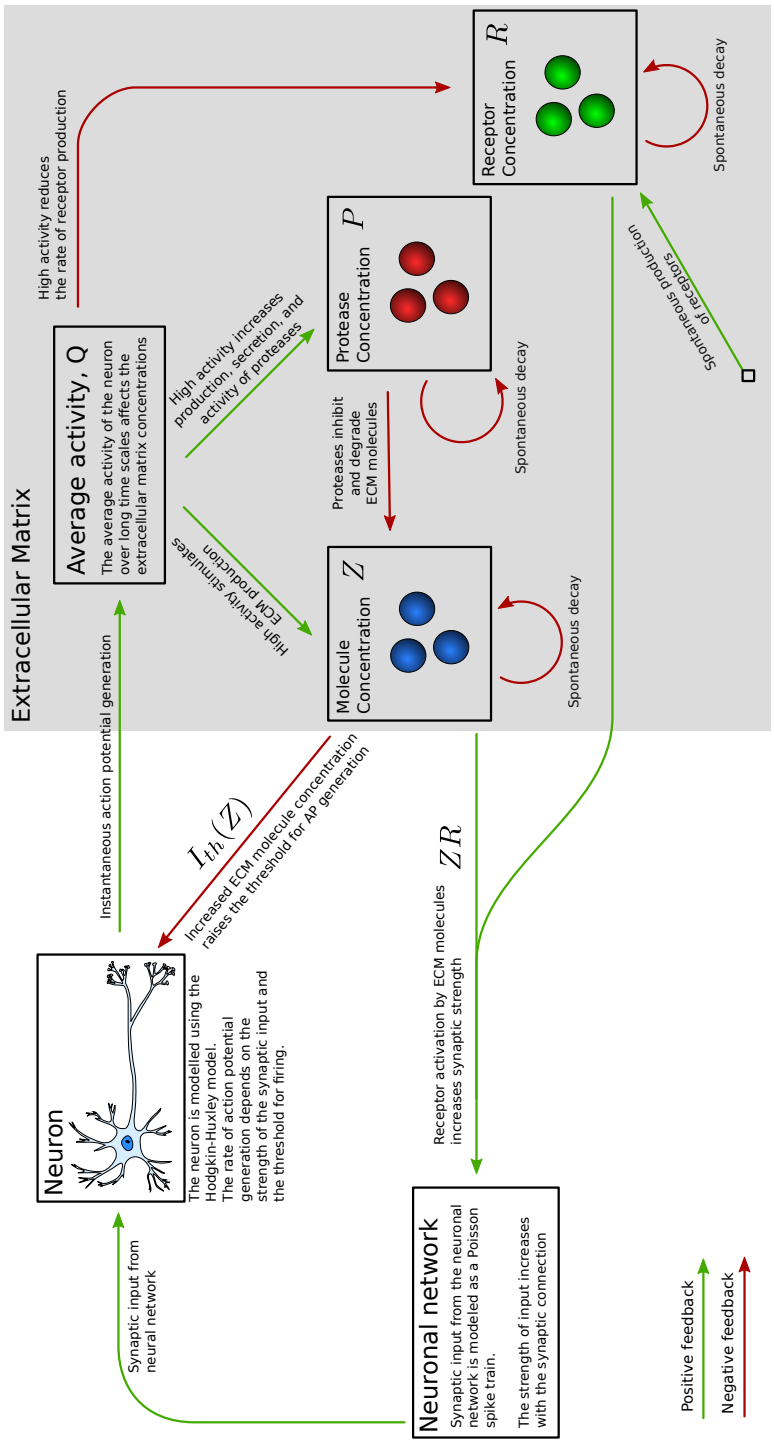

Figure 4.3: A schematic illustration of the model introduced in [Kazantsev et al., 2012].

Chapter 5

Implementation

When implementing the model from [Kazantsev et al., 2012] introduced in the previous chapter, the Kazantsev model, there were many choices that I had to make. My starting point was the set of differential equations making up the model, and from that I had to determine what programming language to use, what environment to program in, whether I was to use an object-oriented approach, as well as how to test my program, amongst other things. In the following I will give an overview of the various aspects of my implementation, and I will attempt to convey my reasons for making the choices that I have made.

5.1 Programming Language and Environment

The system to be implemented is fairly small, consisting of only eight ordinary differential equations. These were, however, coupled and highly non-linear. Due to it being a relatively small system I still figured I could choose programming language quite freely. Nevertheless, the system at hand was one where two contrasting timescales were at play, and as such I wanted to choose one were the performance and efficiency would not be unnecessarily limiting.

When I was only getting started, I knew only what the model I wanted to implement looked like, while it was still unclear where I would go from there, and how I would choose to modify or explore the model further. As such I naturally wanted to use a flexible language, where my options would be wide open.

I also needed to settle on a general structure for the program to be made, and I felt using object-oriented programming seemed most natural for the problem at hand. I will go further into the details of the structure in the following section. With all these preferences in mind a very logical choice was to use the programming language `C++`.

Having made my choice of language a natural next step is the environment in which to work. My personal preference is to always choose as simple of an approach to this as possible, to avoid unnecessary errors and complications that

are due to the framework and not the program itself. My experience with various developing environments have demonstrated how the debugging process can be frustratingly complicated by bugs present in the framework, or the interplay between the framework and the program being made.

Despite my general dislike for such frameworks, however, they have several redeeming qualities. It was easy to see that the model to be implemented would lead to a fairly large program, which could quickly become disorganized. With larger programs consisting of several separate files, frameworks can be a big help in keeping the whole thing neat and organized. A common feature is also a simplified process of compilation and build for the program, which can be a great relief. These features will also be useful if my code is to be used be others after me, who will then have an easier time grasping the structure of the program, and how to use it.

Taking all this into regard I decided to use the development environment `Qt Creator` [Project, 2015], which seemed to fit my needs nicely. I chose this particular framework partly as I was familiar with it already, and have had positive experiences using it. Another reason for choosing `Qt Creator`, specifically was that I also wanted to integrate testing into my program in a simple, straightforward, and clear cut way, and I knew that `Qt Creator` worked well integrated with the unit testing framework `UnitTest++`.

5.2 Unit testing

Unit testing is a very useful way of testing one's implementation. The idea is to individually test each unit, with a unit being a module or component of a program, with the goal of testing independently that each unit behaves as intended. A unit can have various sizes, from an individual method to an entire class.

Imagine for instance that you have made a method that takes two integers, a and b as input, and the desired output is the product of these numbers. We call this method `muliply`, and have it be a method in the class `Calculator`. A typical unit test will then look as follows (in `C++` using the `UnitTest++` testing framework):

```
#include <unittest++/UnitTest++.h>
#include <calculator.h>

TEST(Calculator) {
    Calculator calc;
    CHECK(calc.multiply(3,5) == 15);
}
```

```
int main()
{
    return UnitTest::RunAllTests();
}
```

Running this test should then give you an output with information about the result, whether it succeeded or failed, how long it took, and if it failed the expected as well as the actual value is printed.

A successful run will an output similar to this:

```
Success: 1 test passed.
Test time: 0.01 seconds.
```

whereas a failed test may give:

```
 ../../projectname/tests/main.cpp:95: error:
Failure in Calculator: Expected 15 but was 5
FAILURE: 1 out of 1 tests failed (1 failures).
Test time: 0.01 seconds.
```

You might also want to include a test that checks for multiplying two negative numbers, a negative and a positive number, and so forth. We want to check that the output is as expected for any possible logical input the function may get, and it may also be a good idea to test unreasonable input, to check the error handling within the function.

Generally it may be a good idea to write the tests for a method before the method itself. Then you assure that you have thought through what the desired behavior is, and that the method you create is easy to test. The tests then act as a sort of design document, specifying the desired interface and behavior of the method to be implemented.

Unit testing is beneficial for several reasons. Making the unit tests continually as the units are being created allows errors and bugs to be discovered early, and forces the developer to think more thoroughly at what the desired behavior of the unit is. When making unit tests throughout the development one is also more likely to write "testable" code, meaning code that is more simply tested, containing smaller, more specific functions that are more likely to be reusable later on. It is also a good way to notice errors that turn up unexpectedly later in the development process. The resulting behavior should then remain unchanged, and the unit tests will warn you if this is not the case.

`UnitTest++` [Blume, 2015] is a framework for unit testing in `C++` that is very light weight and easy to use. It integrates nicely with `Qt Creator`, and as such it

seemed a very good choice for my needs. It can be set up to run all tests at every new build, so that changes that cause failure in one or more tests are noticed right away.

Let us now move on to look at the structure of my program.

5.3 Program Structure

As previously mentioned I have implemented the model using object-oriented programming. In this programming paradigm the main constituents are objects. Each object is an instance of a class, and as such contains the methods defined in the class. This allows for a lot of flexibility and tends to make the final program a more organized one, that is easier to get a quick overview of.

In figure 5.1 one can see a schematic overview of the program structure. The top half, labeled *Interface*, shows how the model and solver are separated into two different classes, while experiments can be made by allowing a solver object to work on a model object. Unit tests were of course made for the methods of both these classes. In addition I created several support classes, for tasks such as writing the results to file.

This way of implementing the model, separating the implementation into a problem/model class and a solver class allows for a great deal of flexibility. The *Solver* class has very little dependency on the specifics of the implementation of the particular model to be solved. As such it can be used to solve many similar problems without needing to be changed or adapted to the specific problem at hand. A solver object simply advances the model one step in time using one of the numerical schemes at its disposal (currently three).

In my implementation, shown schematically in the lower half of figure 5.1, I chose to create a virtual parent class dubbed *Neuron Model*. This virtual class acts as an interface, and a specific neuron model can be constructed as a daughter class, implementing the virtual methods in *Neuron Model*. This way it is easy to create a model that can be solved by use of the *Solver* class. The *Solver* class I have designed solves a model problem implementing the *Neuron Model* interface by accessing the right hand side of the voltage equation, as well as the right hand side of ODEs governing any gating or non-gating variables present in the model. To solve the issue of different models having different numbers of gating and non-gating variables, these variables are stored in arrays. The solver can be used both with a constant size time step as well as an adaptive one, as the user can specify the step size for each step taken.

Looking at the specific model problem at hand, the Kazantsev model, we see that it can be can be viewed as an extension of the Hodgkin-Huxley model [Hodgkin and Huxley, 1952]. The Hodgkin-Huxley model is simply one specific model for the electrical activity of a neuron, and as such I created a model class dubbed *HodgkinHuxley*, implementing the virtual *Neuron Model* class. The

Kazantsev model was then implemented as a daughter class of this Hodgkin Huxley class, to minimize redundancy.

This is a useful approach for several reasons. For one, creating a Hodgkin-Huxley class allows for objects to be created that are instances of this class. The Hodgkin-Huxley model is a very popular model, and having it implemented in this way, separate from the Kazantsev model, allows me to test my implementation against the thoroughly verified results of others. It is also handy in that very many existing neuron models are extensions of the Hodgkin-Huxley model, often with additional ion currents added. Having the Hodgkin-Huxley model implemented as a base model lets these more extensive models be implemented more easily. In addition it will be easier to see how the addition of the extra components present in the Kazantsev model affects a neuron, as comparing with a neuron lacking these additions is now straightforward.

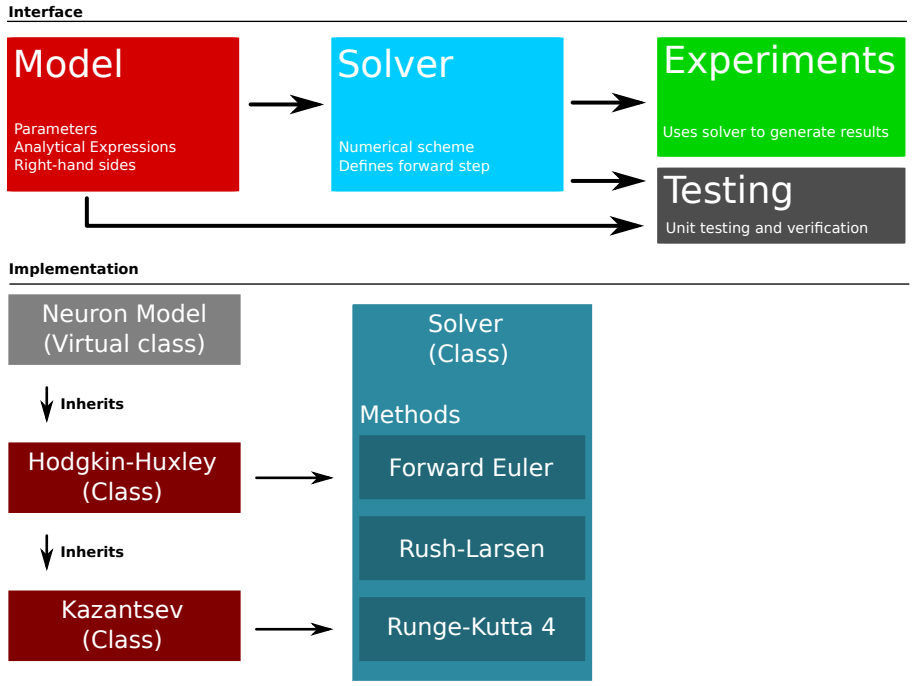

Figure 5.1: A schematic overview of the program structure.

The various components shown in figure 5.1 are made so as to be easily reused and expanded. Letting the *Neuron Model* class be a virtual class allows for fairly few restrictions in regards to which models can be implemented in this set up, and solved by the methods in the *Solver* class. A wide variation of models may be implemented as its descendants. It simply acts as an umbrella class, with the *Solver* class being able to be used on any object that inherits from the neuron model class and as such "is" a neuron model.

In the following section we embark on the next step, which is to look more closely at this *Solver* class, and the numerical methods it employs.

5.4 The Solver

With the model implemented in a way that allows for a fair amount of flexibility when it comes to expanding on the model or changing it, we want a method for solving such a general model. Regardless of modifications the model to be solved will consist of a set of ordinary differential equations, one describing the time evolution of the membrane voltage V, a set for the gating variables, and a set for the remaining time dependent variables, which I have chosen to refer to as non-gating variables.

A natural choice is to make a solver class containing the methods for solving such a model system, and the goals for this class is for it to be intuitive and easy to use, preferably allowing for several solver methods. It should also be general enough to be able to solve all model systems that inherit the neuron system virtual class without needing any modifications.

5.4.1 Choice of Numerical Scheme(s)

The goal when solving a system of differential equations is generally to determine how the system evolves in time (and/or another variable, such as space, depending on the system). Such systems quickly become too complicated to be solved analytically, and this is when we turn to numerical solvers. The `Solver` class is currently implemented with a choice of three different numerical schemes for solving the system of ordinary differential equations that make up a specific neuron model. New solver schemes can also be added in a relatively straightforward way, should it be needed.

There are several reasons why having multiple numerical schemes to choose between is desirable. For one thing, comparing the results when using different schemes is a good way to check your implementation for errors and bugs. If you get equivalent results using several numerical schemes, that points to a correct implementation of the schemes themselves.

Another clear advantage of having more than one numerical scheme to choose between is that different schemes naturally have varying strengths and weaknesses, as well as different stability characteristics. Depending on the problem at hand and your desired level of accuracy in the resulting solution, solvers of different complexities are needed.

It might be tempting to say that a more complex and thereby (hopefully) more accurate solver is always the desired choice, but increased complexity goes hand in hand with an increased need for computational power. When solving large systems and/or for long time periods, any unnecessary increase in complexity is

undesirable as it will lead to a noticeable increase in running time. An example where this is an important consideration is when doing parameter fittings. In such situations the model to be fitted needs to be run for every parameter set included in the fitting, which can naturally be very time consuming. The larger number of parameter sets tested, the more accurate the parameter fitting will generally be.

Weighing your needs for accuracy, stability and efficiency, you can then choose the numerical scheme best suited to your needs.

The three numerical schemes that I have chosen to implement are the explicit forward Euler scheme, the Runge-Kutta scheme (RK4), as well as the Rush-Larsen scheme. In the following I will introduce each of these solver schemes, and discuss their respective strengths and weaknesses.

The Explicit Forward Euler Scheme

We begin by looking at the simplest of the implemented solvers. A general initial value problem to be solved can be formulated as follows:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y(t)), \quad y(t_0) = y_0. \tag{5.1}$$

We wish to find an algorithm allowing us to solve this problem numerically. A simple first approach is the standard Euler method. This method can be derived by considering the Taylor expansion of the function y around the point t_0. Expanding to the second order this becomes:

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{1}{2}h^2 y''(t_0) + \mathcal{O}(h^3). \tag{5.2}$$

Inserting equation 5.1 into this expression, and renaming h as Δt while discarding the second-order and higher order terms, we are left with:

$$y(t_0 + \Delta t) = y(t_0) + \Delta t f(t_0, y_0) \tag{5.3}$$

which is generalized to:

$$y_{n+1} = y_n + \Delta t f(t_n, y_n), \tag{5.4}$$

with $y_n = y(t_n) = y(n * \Delta t)$ for time step n.

This is the explicit forward Euler method, where the next step forward is calculated using the previous step and the right hand side of the differential equation. The local truncation error, E, of this method, meaning the error after one time step, is easily found by comparing the result when taking one step using the method, $y_1 = y_0 + \Delta t f(t_0, y_0)$, with the Taylor expansion from equation 5.2. The difference between these results is:

$$E(\Delta t) = y(t_0 + \Delta t) - y_1 = \frac{1}{2}\Delta t^2 y''(t_0) + \mathcal{O}(\Delta t^3). \tag{5.5}$$

From this we see that the error when taking one step with the forward Euler method is approximately proportional to Δt^2 for small step sizes. This is called the local truncation error. Solving for a total time T, means we take a total of $\frac{T}{\Delta t}$ steps, and from this the global error, GE, of the method becomes:

$$GE(\Delta t) = \mathcal{O}(\Delta t^2) \times \frac{T}{\Delta t} = \mathcal{O}(\Delta t) \tag{5.6}$$

As such we see that the forward Euler scheme is not amongst the most accurate methods, and why it is referred to as a first-order method. In addition to this fairly large global error, the Euler method has a tendency to be numerically unstable. As such it is not the most reliable of solver methods, often requiring the use of very small time steps to give acceptable results. Nonetheless it is a regularly used solver scheme, primarily due to its simplicity and ease of implementation.

The Explicit Runge-Kutta Scheme

A far more numerically accurate scheme is the Runge-Kutta method, also known as Runge-Kutta 4 or the fourth order Runge-Kutta method. It is the most popular of a family of similar Runge-Kutta methods containing both implicit and explicit methods. It is far more complicated than the simple Euler method described above, but with the reward of increased accuracy and stability.

When using numerical integration to solve a system of differential equations the standard procedure is to divide the time interval (or that of the equivalent variable) into very many small intervals, under the assumption that for such a small interval we can assume that the slope of our function is approximately constant. Where the various methods vary is generally in how this slope is determined.

In the Euler method described above we saw that the slope was based simply on the value at the previous interval or time step. It is easy to see that this procedure may lead to an under- or overestimation of the slope. Many methods are variations of this approach, such as the midpoint method where the slope is instead determined from the value obtained halfway between the previous step and the one being calculated, at $t = t_n + \frac{\Delta t}{2}$. Then we arrive at the expression:

$$y_{n+1} = y_n + \Delta t f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} f(t_n, y_n)). \tag{5.7}$$

The Runge-Kutta method takes this general approach a few steps further. In order to calculate a more accurate value for the slope it adopts a weighted average

of four different approaches. As such the next step given by the Runge-Kutta method becomes:

$$y_{n+1} = y_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{5.8}$$

Each of these four k's are different slope approximations, and the hope is that they together will produce a more accurate result than they do on their own. The first one, k_1 is found by simply using Euler's method from the previous section, k_2 takes it a (half-) step further with the midpoint method, k_3 uses the midpoint method again, but now using the result for k_2, while k_4 is based on the value at the end of the interval. The expressions become:

$$k_1 = f(t_n, y_n), \tag{5.9}$$

$$k_2 = f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2}k_1), \tag{5.10}$$

$$k_3 = f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2}k_2), \tag{5.11}$$

$$k_4 = f(t_n + \Delta t, y_n + \Delta t k_3). \tag{5.12}$$

Being a fourth order method, the local truncation error of the Runge-Kutta method is in the order of $\mathcal{O}(\Delta t^5)$, giving a fourth order global error. This can be found in the same way as we did for the forward Euler method, by Taylor expanding y and finding the difference between the resulting expression and that shown in 5.8. This is a big improvement in accuracy compared to the forward Euler scheme, but the with the cost of a higher complexity. The Runge-Kutta scheme is more computationally demanding, and the implementation is more troublesome. Being an explicit method it will also have some stability issues. I will expand on these stability issues later on in the chapter.

The Rush-Larsen Method

The final of the three methods implemented is the Rush-Larsen method. This method was especially designed with the goal of solving systems like the Hodgkin-Huxley model in a labor efficient way, without losing accuracy. It takes advantage of a handy rewrite of the ODEs for the gating variables. The Rush-Larsen method is a popular solver, particularly when studying myocardial cell models, by virtue of being easy to implement while outperforming other simple integrators like for instance the forward Euler method.

As we have previously seen the gating variables in the Hodgkin-Huxley model all satisfy the following first order differential equation:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \alpha_{x_i}(1 - x_i) - \beta_{x_i}x_i. \tag{5.13}$$

Here α_{x_i} and β_{x_i} are the rate coefficients of x_i, gating variable i. In the standard Hodgkin-Huxley model we operate with three gating variables, h, n and m, but other more extensive models may have included many additional ionic currents, and as such many more gating variables.

The rate coefficients will be dependent on the current value of the membrane potential, but let us now assume that they are instead constants. We rewrite equation 5.13 as:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \alpha_{x_i} - x_i(\alpha_{x_i} + \beta_{x_i}). \tag{5.14}$$

It is then clear that under this assumption the solution of the differential equation will approach a constant steady state. Calling this constant solution $x_{i,\infty}$, we get:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = 0 = \alpha_{x_i} - x_{i,\infty}(\alpha_{x_i} + \beta_{x_i}), \tag{5.15}$$

giving:

$$x_{i,\infty} = \frac{\alpha_{x_i}}{\alpha_{x_i} + \beta_{x_i}}. \tag{5.16}$$

With this expression for $x_{i,\infty}$, and the substitution $\tau \equiv \frac{1}{\alpha_{x_i}+\beta_{x_i}}$ we can rewrite the differential equation as:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{x_{i,\infty} - x_i}{\tau}. \tag{5.17}$$

These manipulations have led us from the original expression in equation 5.13, where the ODE was expressed in terms of the rate constants α_{x_i} and β_{x_i}, to the above expression using instead $x_{i,\infty}$ and τ. This may seem arbitrary, but the reward for our work is that we are now able to solve this equation analytically. Using (for instance) the substitution $u = x_{i,\infty} - x_i$, we find that the solution becomes:

$$x_i = x_{i,\infty} - (x_{i,\infty} - x_{i,0}) \exp(-t/\tau). \tag{5.18}$$

Now naturally the rate coefficients will not actually be constants, but are instead dependent on the membrane voltage V, but we can make the assumption that they are constant over the duration of one time step, Δt. The scheme for gating variable x becomes:

$$x_{n+1} = x_{n,\infty} + (x_n - x_{n,\infty}) \exp(-\Delta t/\tau_n), \tag{5.19}$$

with:

$$\tau_n = \frac{1}{\alpha_n + \beta_n}, \tag{5.20}$$

and:

$$x_{n,\infty} = \alpha_n \tau_n. \tag{5.21}$$

In these expressions n indicates the current time step as in the previous numerical schemes, and I have dropped the i indicating that this is the i'th gating variable, for simplicity.

For the remaining (non-gating) variables, including the membrane voltage V, a simple forward Euler scheme is commonly used.

5.4.2 Stability of the Schemes

The methods I have implemented are explicit ones, and this comes with the price of limited stability. Explicit methods are methods where the next step of the solution is calculated directly from the current values of the variables. In contrast implicit methods will also use information from a later time step. Implicit methods are generally more complex to implement, and require greater computational expenses. They are also more difficult to program in a general as opposed to problem specific fashion. However they have the upside of being numerically stable methods, meaning that they produce well behaved solutions even for large time steps.

The typical form of an explicit method is as follows:

$$y_{n+1} = y_n + \Delta t \times f(..). \tag{5.22}$$

The next step in the solution is calculated using the previous step, plus the time step multiplied with some function whose form will vary between the different schemes. It is easy to see from this that a too large Δt can quickly lead to trouble with a rapidly growing solution (in the positive and/or negative direction), thus giving instability for the method.

For the Rush-Larsen method the situation is a bit more complicated. The membrane voltage, as well as any non-gating variable, is in this scheme solved using the simple forward Euler method, and as such the stability here is as for the standard explicit forward Euler scheme. The differential equations for the gating variables are instead solved using a exponential integrator. We saw in the section on the Rush-Larsen method that the analytical solution for these differential equations was as in equation 5.18. Looking at this expression we see that the last term contains a decaying exponential, meaning that that the solution for x_i will approach $x_{i,\infty}$, as the difference $(x_{i,\infty} - x_i)$ decays exponentially.

We know that the values of the gating variables indicate a probability that a gate is open, and as such they can only take values between 0 and 1. Looking at the Rush-Larsen scheme given in equation 5.19, we see that in this scheme x_{n+1} can never take a value that is greater than 1, even for a large Δt. The solution can never "blow up", it will simply approach x_∞.

Taylor expanding the exponential gives:

$$\exp(-\Delta t/\tau_n) = 1 - \left(\frac{\Delta t}{\tau}\right) + \frac{1}{2}\left(\frac{\Delta t}{\tau}\right)^2 - \frac{1}{6}\left(\frac{\Delta t}{\tau}\right)^3 + \mathcal{O}(\Delta t^4). \tag{5.23}$$

For a small enough Δt the higher order terms will approach zero, and we are left with $1 - \frac{\Delta t}{\tau}$. Using this in place of the exponential in equation 5.18 gives:

$$x_i = x_{i,\infty} - (x_{i,\infty} - x_{i,0})(1 - \frac{\Delta t}{\tau}), \tag{5.24}$$

which is easily seen to not have the property of assuring that $x_i < 1$. This expression also reduces to the explicit forward Euler scheme when expanding the parentheses. In other words the Rush-Larsen scheme reduces to a forward Euler scheme for small Δt. This quality of the Rush-Larsen scheme is utilized in a version of the scheme often labeled an adaptive Rush-Larsen scheme. In this variant the forward Euler scheme is used also for the gating variables, so long as $\frac{\Delta t}{\tau}$ is small enough.

This also illustrates what constitutes a "small enough" time step for the explicit forward Euler method to be numerically stable. When τ is big, this means that the state of the gating particle changes more slowly, and a larger step size can be used.

5.4.3 A Note on Stiffness

When solving systems of differential equations numerically, it is often beneficial to examine the stiffness of the equations at hand. As there is no universally accepted theoretic definition of stiffness, this is not always easy to do. Generally, a differential equation is regarded as stiff if an impractically small step size is required for the numerical solution to remain stable. Stiffness is regarded as a property of the differential system itself, and if not addressed it can lead to notable inaccuracies in the final solution.

A large number of cell models, including the Kazantsev model that we examine here, consist of a system of ODEs. These models are often based on the Hodgkin-Huxley model, and contain ODEs that are typically non-linear, and generally include a transmembrane potential, a set of ionic concentrations, as well as a number of gating particles. For these systems stiffness can often become an issue, with some of the equations being stiff, at least in specific intervals. To avoid instability in the solution it is therefore beneficial to address this stiffness and find ways to deal with it.

How can stiffness be dealt with? Often it is a matter of using an appropriate solver method. As mentioned in the previous sections different solvers have varying strengths and weaknesses, and as such some solvers are better suited to solve

stiff equations than others. It is the level of stiffness in an equation that will determine if a specific numerical method will be able to solve a model efficiently (without requiring en unrealistically small step size). The difficulty then lies in determining the level of stiffness in the system, and as stiffness has not been precisely defined this is not necessarily trivial.

One common approach when determining stiffness is to regard the ODE as stiff on a time interval (with respect to the numerical method used and the error tolerance chosen) if stability requirements force the numerical method to advance the solution using smaller step sizes than what is dictated by accuracy requirements [Ascher and Petzold, 1998].

Another approach that can give information on the stiffness of the system uses the eigenvalues of the Jacobian matrix of the equation set. Given an initial value problem as in equation 5.1, the Jacobian matrix can be found from $\mathbf{J}= \frac{\partial f}{\partial y}(t, y)$. The stiffness of such an initial value problem is associated with the eigenvalues of this Jacobian matrix, evaluated over time. If the equation at hand is stiff, these eigenvalues typically have large negative real parts. We see this from the fact that a large negative eigenvalue λ means that the time step Δt has to be small, else $\lambda \Delta t$ will lie outside the stability region of the numerical method [Marsh et al., 2012].

There are several options when it comes to combating stiffness. Using implicit methods as explained above will assure a stable solution, but it is not always feasible to do. Other solutions include using self-correcting methods, but this also tends to increase the complexity of the implementation, and the computational demand at each time step. The extra required computational demand may be offset by the possibility for using larger time steps, but sometimes simply assuring the time step chosen is small enough is the easiest solution. Using multiple solver schemes and comparing the results will also give confidence in the found solution.

Chapter 6

Exploring the Model

Having looked at the more technical aspects of my implementation we move on to look more closely at the model itself, and how it behaves. A first step in exploring the model further is to assure first and foremost that my implementation of it is correct, and that the model behaves as expected. Once it has been assured that the model implementation is satisfactory, we continue ahead and explore the model and its behavior. The goal is to get an understanding of the workings of the model and how it relates to experiments, while examining how it can be simplified or expanded upon, depending on need.

6.1 Reproduction of the Kazantsev Model

As the model described in [Kazantsev et al., 2012] has been the basis of this thesis an important first step has been to evaluate whether my implementation of the model behaves as expected, by reproducing some of the results included in the article. This is a good strategy to check for errors in my own implementation of the model, before the model is explored further.

As I have been using unit testing to test all methods used in the model, I could be fairly confident that my methods did what I expected them to. Still, it is wise to investigate whether or not they together create the expected behavior. We start by looking at the core of the model, the modeling of the spike generation in the neuron.

6.1.1 Testing the Core of the Model

The Kazantsev model is based on the Hodgkin-Huxley formalism and uses it to model the activity of the neuron, and as such this can be seen as the core of the model. It is essential for the rest of my implementation that this part of my implementation works as intended, and we test this by comparing my model with available results.

The Hodgkin-Huxley model of a neuron is a thoroughly tested and much used model, and as such it should be more than possible to verify my implementation against existing results. To do this I decided to recreate figure 3.11(a) in [Sterrat et al., 2011], which is seen in figure 6.1. In this plot there is no input current, not even a pulse. Instead the simulation is started with the initial membrane potential being at various distances above the resting membrane potential. With all parameters chosen as in [Sterrat et al., 2011], I attempt to recreate this result and the plot I get using my implementation is shown in figure 6.2.

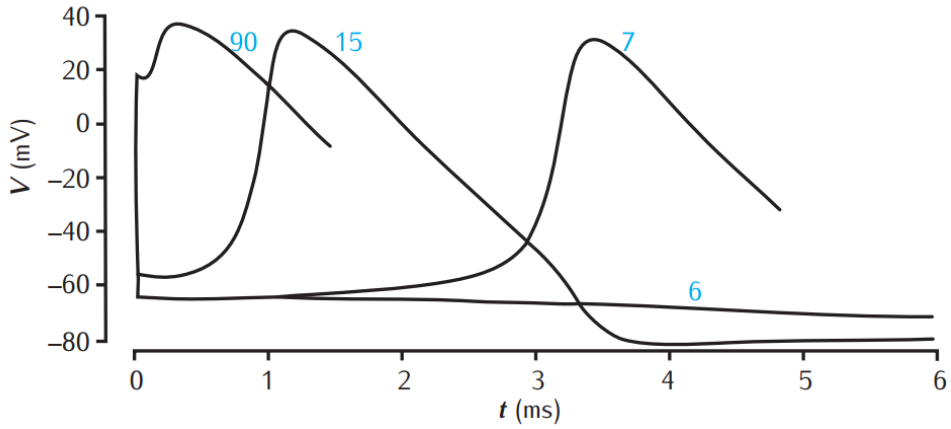

Figure 6.1: Solutions of the Hodgkin-Huxley equations for various initial polarizations above resting potential, as indicated by the numbers on the lines. Figure taken from [Sterrat et al., 2011]

Comparing these figures we can conclude there seems to be a good agreement between my implementation and the expected behavior for a Hodgkin-Huxley system. The resulting action potentials produced have very similar shapes, last for the same time period, and reach a similar amplitude. The firing thresholds also seem to agree, with $V_0 + 7$ mV resulting in a spike in both cases, while $V_0 + 6$ mV does not lead to a spike for either.

With the model foundation working as it should, the next step is to look at the synaptic input current in to the system.

6.1.2 Synaptic Input

The neuron modeled in the Kazantsev model receives synaptic input in the form of excitatory post synaptic current (EPSC) from surrounding neurons. This input is included in the form of a Poisson spike train. The amplitudes of these incoming spikes obey a probability distribution given by the expression:

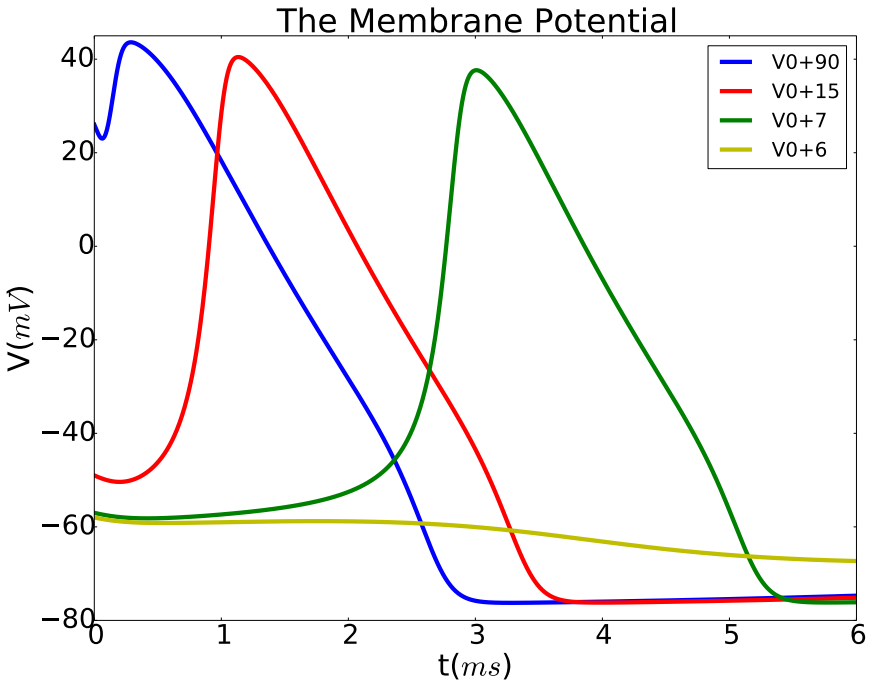

Figure 6.2: Experimental solutions of the Hodgkin-Huxley equations for various initial polarizations above resting potential, using my implementation of the Kazantsev model, removing the threshold current I_{th} and the synaptic input current I_{syn} so we are left with a simple Hodgkin-Huxley system.

$$P(x) = \frac{2x}{b^2} \exp\left(-\frac{x^2}{b^2} \right). \tag{6.1}$$

Comparing this theoretical distribution with the amplitudes generated in my implementation, as seen in figure 6.3, we see that they seem to be in very good agreement. The figure shows the result for two different scaling factors of the input, $b = 6$ and $b = 10$, in the form of a histogram of the current amplitudes produced, with the red dotted line indicating the matching theoretical distribution. The histograms follow the theoretical distribution nicely for both scaling factors, and as such it seems the amplitudes of the incoming spikes are as they should be.

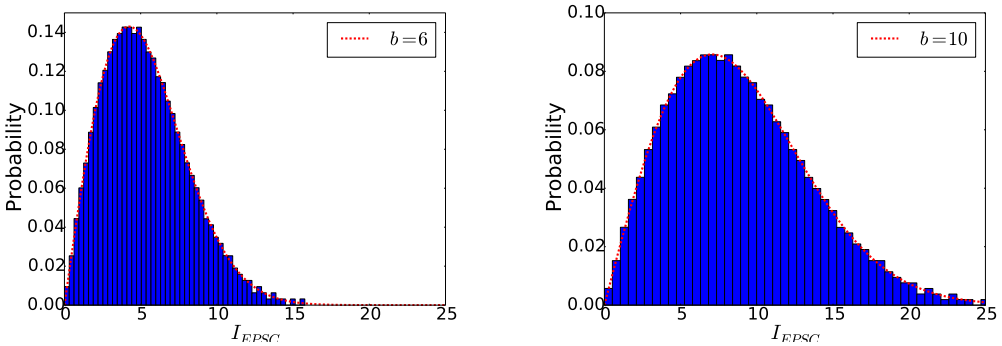

Figure 6.3: Histogram showing the probability densities of the amplitude of the synaptic input, together with the theoretical distribution for $b = 6$ (a), and $b = 10$ (b).

Knowing that the amplitudes of the incoming excitatory post synaptic current are as expected, there is still the possibility that they do have the correct amplitude, but are produced at the wrong frequency. Figure 6.4 shows a histogram of the probability distribution for interspike times, meaning the time in between spikes, produced by my implementation. The histogram is plotted together with a red line showing the expected distribution, given by:

$$P(t) = f_{\text{in}} \exp(-f_{\text{in}} x). \tag{6.2}$$

As seen in figure 6.4 the interspike times produced look to be in very good agreement with the theoretical distribution.

Having verified that the incoming EPSC given as input to the neuron has the expected frequency as well as amplitude, I feel confident that my implementation of the synaptic input works as intended. With this part of my implementation verified, together with the Hodgkin-Huxley model of the neuron, we move on

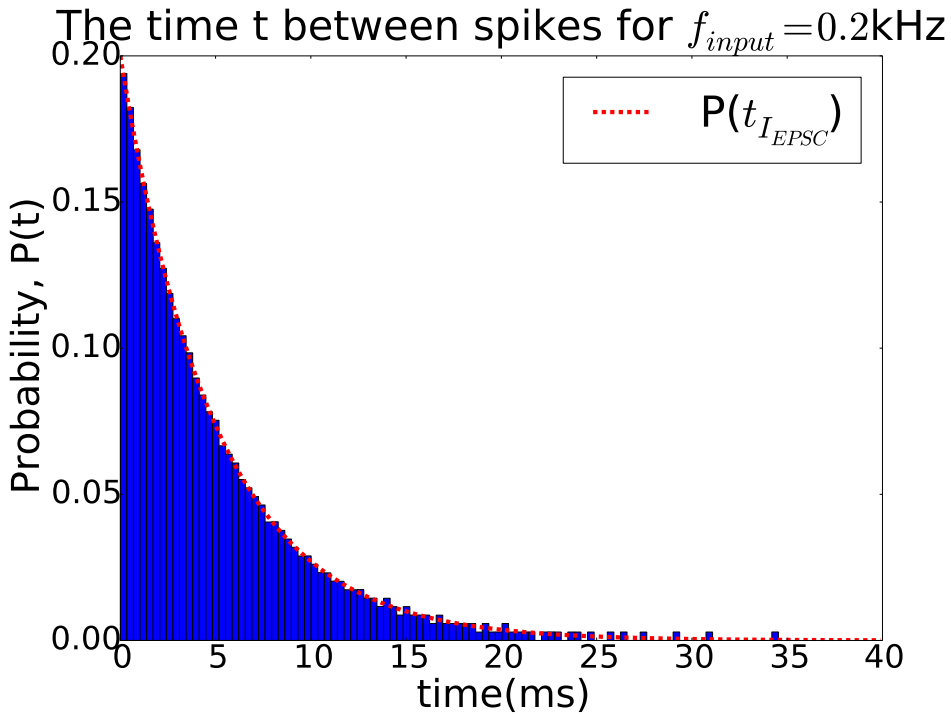

Figure 6.4: A histogram showing the probability density of the time between the spikes of the synaptic input. The red line shows the theoretical distribution. The input frequency $f_{in} = 0.2$ kHz.

to look at the resulting firing activity of the model, and the average activity resulting from it, Q.

6.1.3 Firing Activity

A key component in the model is the variable Q, representing the average firing activity of the neuron being modeled. Figure 6.5 illustrates how this variable in [Kazantsev et al., 2012] is showed to relate to the input frequency f_{input}, which is the characteristic frequency of the synaptic input. The solid line shows the logistic curve fitted to the data, given by the expression:

$$Q = A_2 + \frac{A_1 - A_2}{1 + (f_{input}/f_0)^2}, \tag{6.3}$$

with $A_1 = 0.02861$, $A_2 = 2.80585$, and $f_0 = 0.83012$ Hz [Kazantsev et al., 2012].

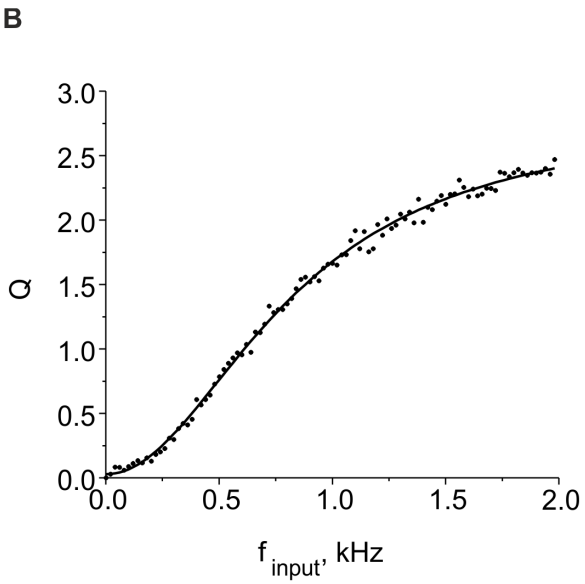

Figure 6.5: The average activity Q as a function of the input frequency of the Poisson synaptic input. Figure taken from [Kazantsev et al., 2012].

An attempt at recreating this plot is shown in figure 6.6. In creating this plot the model is run for two minutes for each value of f_{input}, and the value for Q is found by averaging over the last minute. In figure 6.7 we see that after about a minute of run time, the value for Q has stabilized around its steady state value, and as such this is a sensible procedure.

The red dots in figure 6.6 show the average activity Q measured for various values of the input frequency f_{input}. The blue line shows the line given by equation

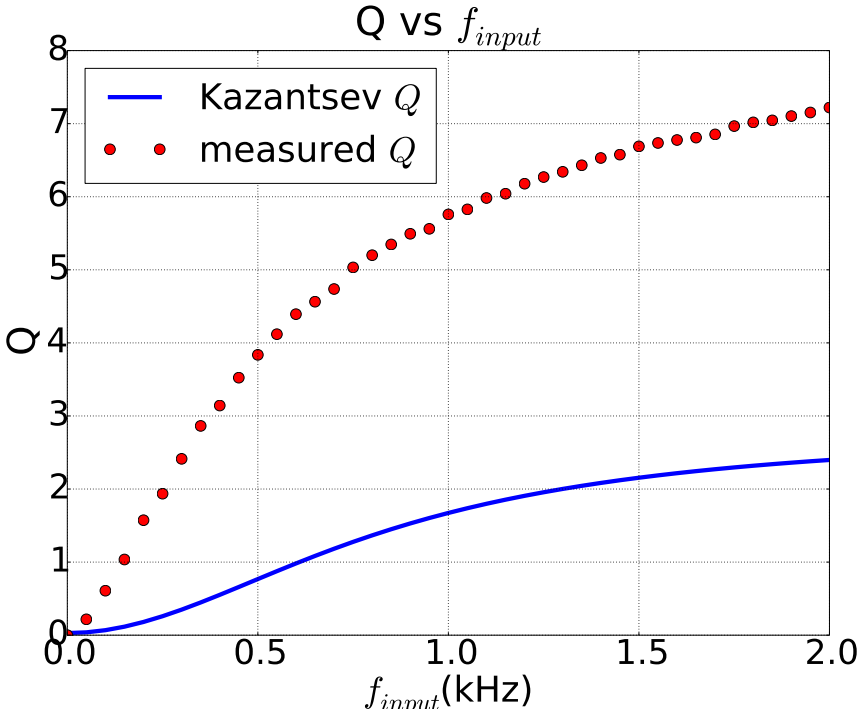

Figure 6.6: The average activity Q as a function of the input frequency of the Poisson synaptic input in my reproduction of the Kazantsev model. The figure is created by running the model for two minutes so that the average activity stabilizes, for each of the input frequencies tested. Each data point is the average activity measured in the last minute of the run. Parameter values as given in figure 2 in [Kazantsev et al., 2012]. Solver is explicit forward Euler with a time step $\mathrm{d}t = 0.0005$ ms.

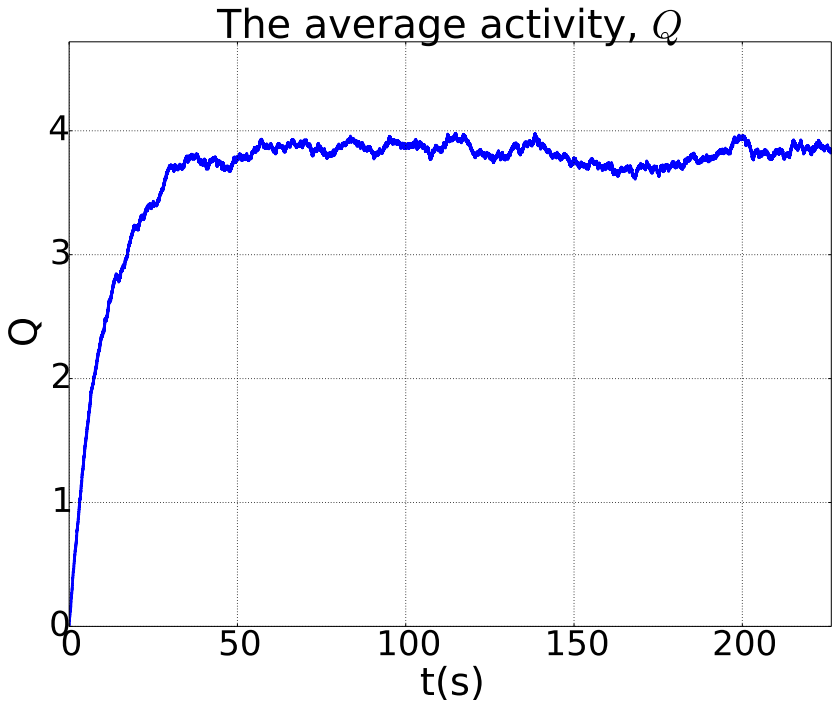

Figure 6.7: The average activity Q seen as a function of time to show that it stabilizes around its steady state value after about a minute. Parameter values as given in figure 2 in [Kazantsev et al., 2012]. Solver is explicit forward Euler with a time step $\mathrm{d}t = 0.0005$ ms.

6.3, which is seen to be a good fit with the data points in figure 6.5. In figure 6.6 however, the measured data point lie high above this line for all input frequencies. In other words, my implementation results in a markedly higher average activity, given the same input frequency. This is clearly a serious discrepancy, and there are multiple possible reasons that could explain this dissimilarity.

In the article, a number of the parameters used when creating the plot are not mentioned explicitly, so these had to be determined in some other way. To do this I have in general opted for using parameters corresponding to those used in other parts of the paper. The parameters for the Hodgkin-Huxley part of the model however, meaning the modeling of the ionic transmembrane currents, were not given anywhere, and as such had to be found elsewhere. I used the parameters given on page 61 in [Sterrat et al., 2011]. This is of course not a perfect solution, but one that may possibly have led to at least some of the discrepancy seen.

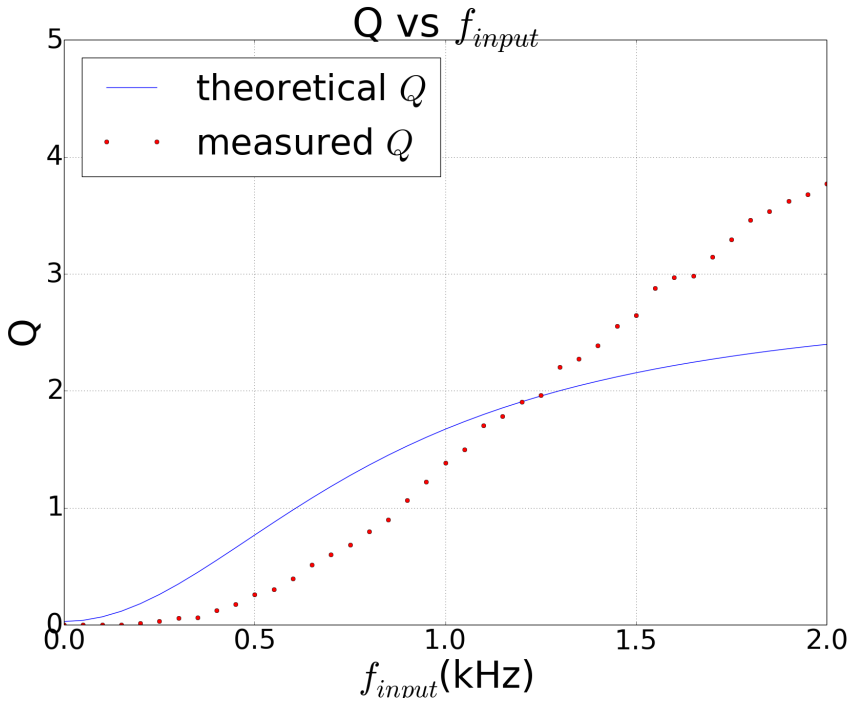

Figure 6.8: The average activity Q as a function of the input frequency of the Poisson synaptic input in my reproduction of the Kazantsev model. Figure created as 6.6, but with the scaling of the synaptic input reduced to $b = 2.5$.

Let us explore this discrepancy further. Looking at figure 6.6 we see that the measured values for Q give a plot that does seem to have a similar shape to the curve fitted to the results in the article. It looks like the disparity between the results lie in the amplitude/scaling as the behavior seems to be quite similar oth-

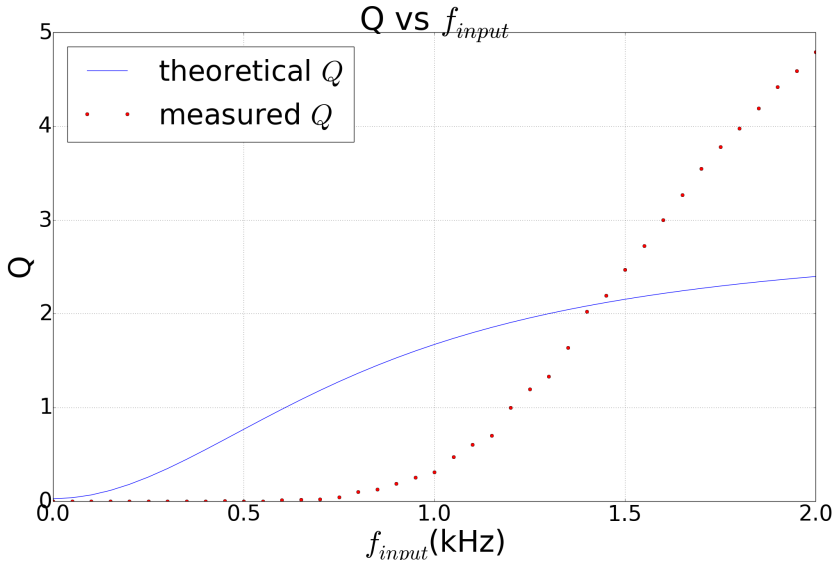

Figure 6.9: The average activity Q as a function of the input frequency of the Poisson synaptic input in my reproduction of the Kazantsev model. Figure created as 6.6, but with the threshold current increased to $I_{\text{th}} = 15\ \mu\text{A/cm}^2$.

erwise. We move on to explore this further by modifying parameters controlling this scaling.

A first exploration was done by simply lowering the scaling factor, b, of the synaptic input, as lowered synaptic input is expected to give a lowered firing frequency. The result is shown in figure 6.8. I also attempted to increase the spike excitation threshold, and the result is seen in figure 6.9. In both these attempts of lowering the activity we see a tendency to undershoot the activity for low input frequencies, and overshoot for higher input frequencies. Lowering the synaptic input does seem to give a noticeably better fit to the given curve, but there are still clear discrepancies.

When looking at the dependence of the average activity, Q, on the current I_{th}, we see again that although the general trend is as in the article, the values for the activity are noticeably higher. A plot is shown in figure 6.11, and comparing with the original plot from [Kazantsev et al., 2012], shown in figure 6.10, we see this clearly.

It is worth mentioning that I have also checked that my error is not simply in the implementation of Q, it is the actual activity of the neuron which is higher than expected. I confirmed this by reproducing the plot in figure 6.6 from the membrane voltage V alone. I did this by using the membrane voltage produced by my implementation to find the spike frequency and spike duration. A value for Q can then by found by using equation 4.10 in chapter 4. The results from

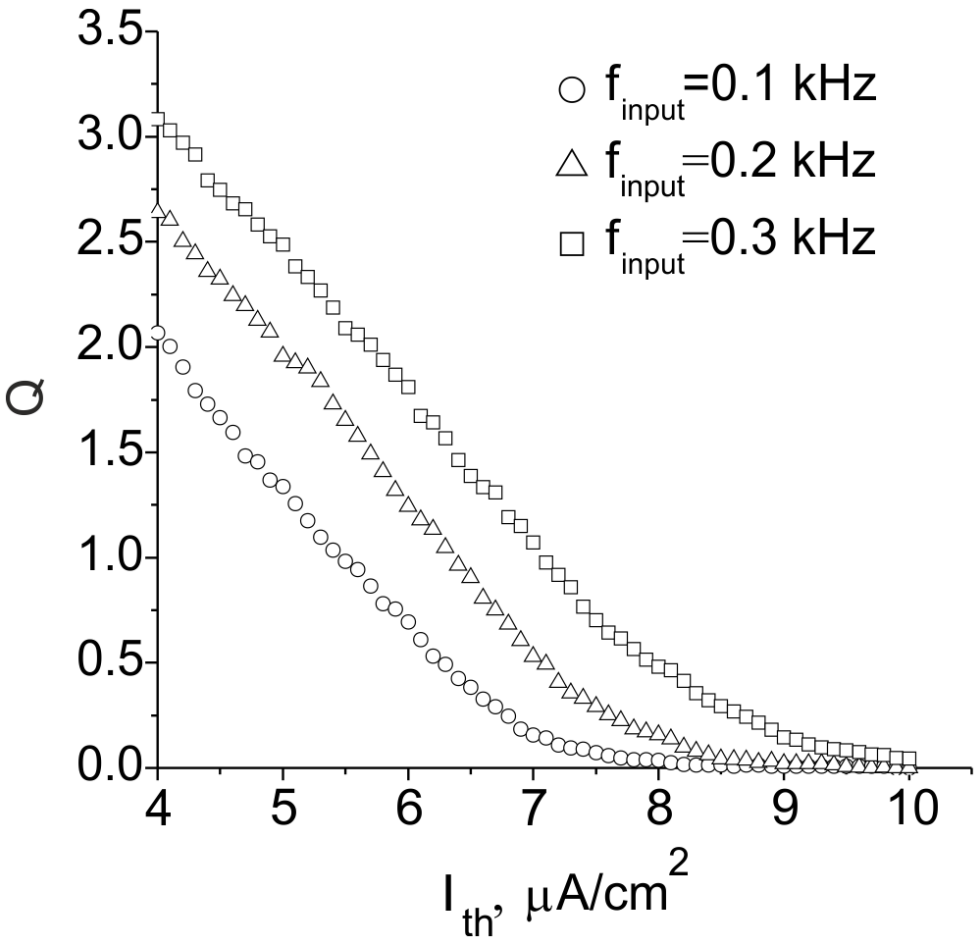

Figure 6.10: The average activity Q as a function of threshold current I_{th}. Figure taken from [Kazantsev et al., 2012].

this procedure were near identical to that seen in figure 6.6. It is then clear that the discrepancy does not lie with Q, but instead with the membrane voltage produced by the neuron model.

With the base system working as it should, as well as the synaptic input, I can find no obvious error in my implementation of the Kazantsev model. As an extra measure of quality control I did also re-implement the model using the programming language `Python`, and my results were identical to my original implementation. As such I draw the conclusion that the discrepancy is not due to an error in my implementation, and in my further work I will simply expect a higher level of activity than what has been found in [Kazantsev et al., 2012].

I now move on to look at the molecules, receptors and proteases of the extracellular matrix (ECM), responsible for the homeostatic regulation of firing activity in the model.

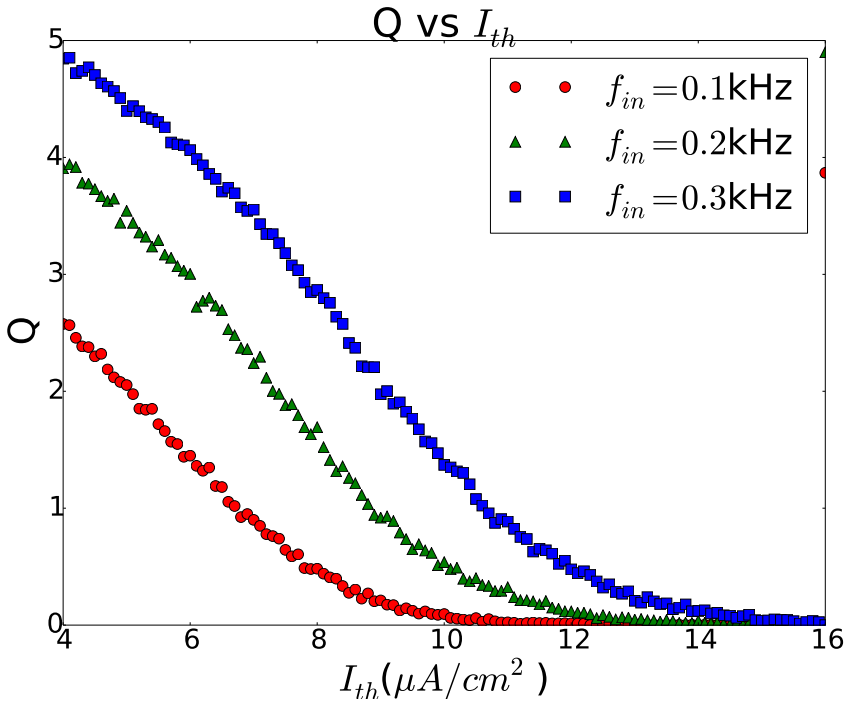

Figure 6.11: The average activity Q as a function of the threshold current I_{th} in my reproduction of the Kazantsev model. Figure created using the same parameters as in figure 6.10.

6.2 Steady State Behavior of the Extracellular Matrix

The key part of the Kazantsev model is its inclusion of the extracellular matrix and this structure's effect on the neuron it surrounds. Being an early attempt at modeling this interplay, it is modeled in a quite straightforward way, where the various molecules associated with the ECM are included in the form of three coupled differential equations. These molecules affect and modulate the activity of the neuron through two additional equations for the synaptic strength and the threshold current. A first step now that the implementation of the core of the model has been tested and verified, is to look more closely at these ECM molecules and their effect on the model dynamics.

The differential equations controlling the dynamics of the ECM molecules, Z, receptors, R, and proteases, P, are all made up of a simple decay term, as well as a growth term controlled by an activation function, with Z having an additional decay term. The shape of these activation functions are shown in figure 6.12.

From this figure there is clearly seen to exist two different steady states, one

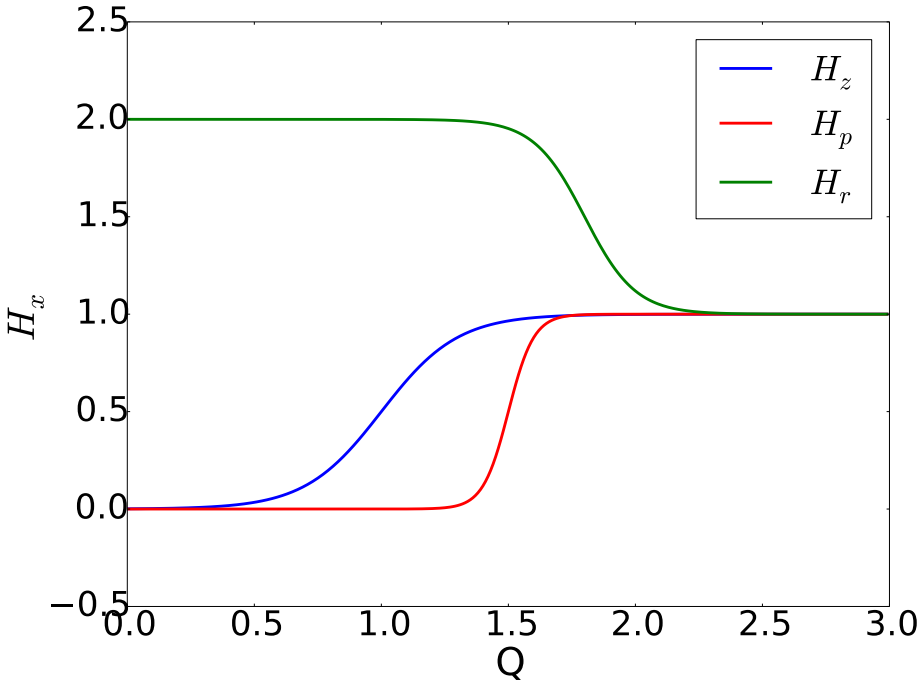

Figure 6.12: The activation functions for the various molecules of the ECM, as a function of the average activity Q. Reproduction of figure 4 in [Kazantsev et al., 2012].

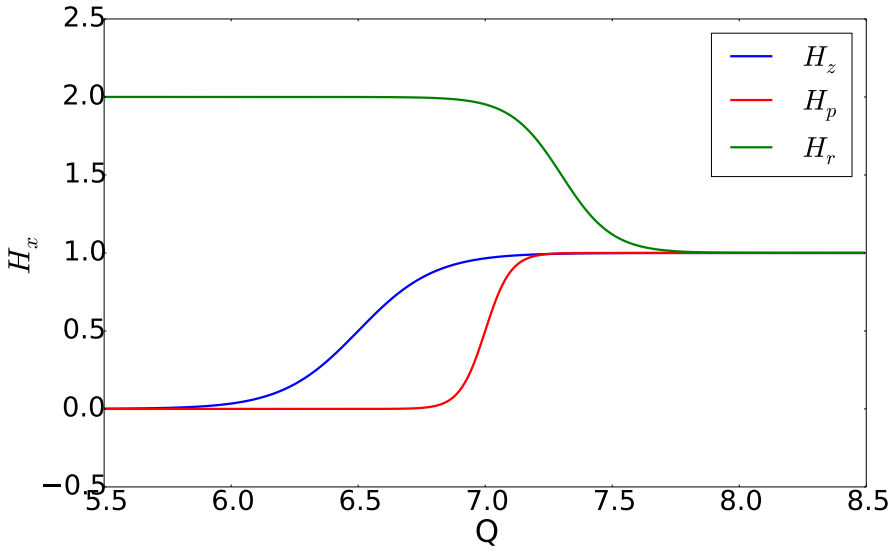

Figure 6.13: The activation functions for the various molecules of the ECM, as a function of the average activity Q, when shifted to the right by 5.5 points.

for low activity, with $Q < 0.5$, and another in the high activity regime, where $Q > 2.5$. Now, as we have seen, the activity in my implementation is noticeably higher than what is seen in [Kazantsev et al., 2012], and as so it is natural to shift the activation functions shown in figure 6.12 towards higher activities. Recall from the chapter on the Kazantsev model, that the activation functions are given by:

$$H_x(Q, x_0, x_1, \theta_x, k_x) = x_0 - \frac{x_0 - x_1}{1 + \exp\left(-\frac{(Q-\theta_x)}{k_x}\right)}, \quad x = z, p, r. \tag{6.4}$$

Here, θ_x, gives the shift of the function in terms of $Q = 0$, with bigger values for θ_x giving a shift towards higher activity. In figure 6.12 the values used are $\theta_z = 1$, $\theta_p = 1.5$, and $\theta_r = 1.8$. I shift all these upwards by 5.5, giving $\theta_z = 6.5$, $\theta_p = 7.0$, and $\theta_r = 7.3$. The result is shown in figure 6.13.

We move on to look at the behavior of the various molecules associated with the ECM. We begin by exploring the two steady states, looking first at the low activity steady state, which is seen from figure 6.13 to be found for an average activity $Q < 6$. Comparing this with figure 6.6, we see that with input frequency f_{in} lower than about 1.1 kHz should give the desired activity level.

Running the model then with $f_{\text{in}} = 0.5$ kHz, we expect to see the various molecules approaching a steady state. Looking at the result from such a run in figure 6.14, we see that a steady state is indeed reached, and it is reached

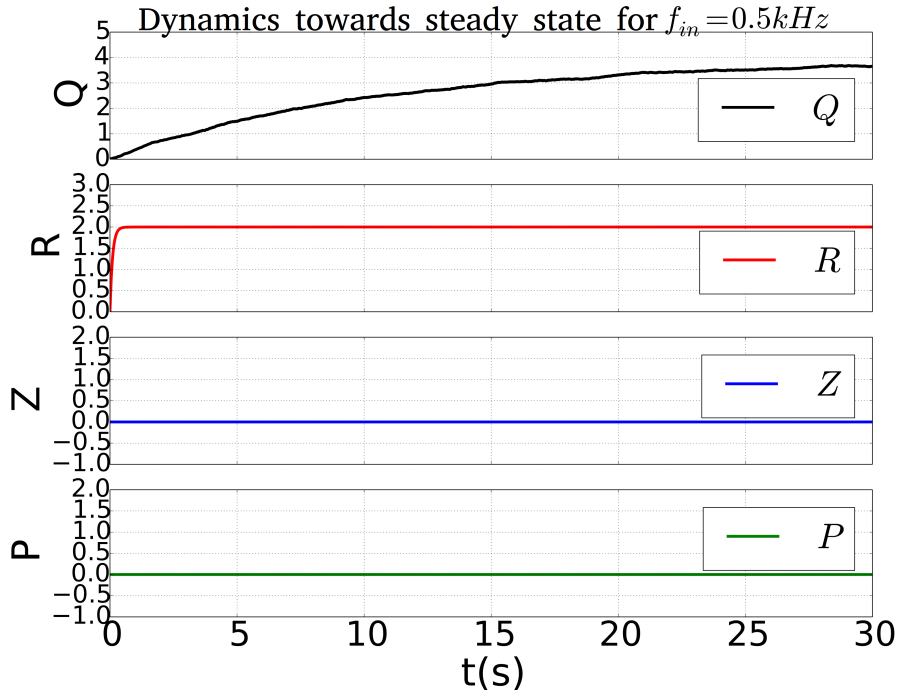

Figure 6.14: Plot of Q, R, Z and P for an input frequency giving a low firing activity. Parameter values for the simulation: $\tau = 1.0$ msec, $b_0 = 6.0$, $\gamma_{ZR} = 0.065$, $I_0 = 4.5$ μA/cm^2, $\gamma_Z = 0.0345$, $\alpha_z = 0.001$ msec^{-1}, $\gamma_p = 0.1$, $\beta_z = 0.01$ msec^{-1}, $z_0 = 0$, $z_1 = 1$, $\theta_z = 6.5$, $k_z = 0.15$, $Z_0 = 0.0$, $\alpha_p = 0.001$, $\beta_p = 0.01$ msec^{-1}, $p_0 = 0$, $p_1 = 1$, $\theta_p = 7.0$, $k_p = 0.05$, $P0 = 0.0$, $\alpha_r = 0.01$ msec^{-1}, $\beta_r = 0.01$ msec^{-1}, $r_0 = 2.0$, $r_1 = 1.0$, $\theta_r = 7.3$, $k_r = 0.1$, $R0 = 0.0$, $\alpha_q = 0.0001$ msec^{-1}, $\beta_q = 0.01$ msec^{-1}, $q_0 = 0$, $q_1 = 1$, $\theta_q = 0$, $k_q = 0.01$, $Q0 = 0.0$, $C_m = 1.0$, $g_Na = 120$ mS/cm^2, $V_{Na} = 50$ mV, $g_K = 36$ mS/cm^2, $V_K = -77$ mV, $g_l = 0.3$ mS/cm^2, $V_l = -54.4$ mV. These are the parameter values used in all the following figures unless other values are given.

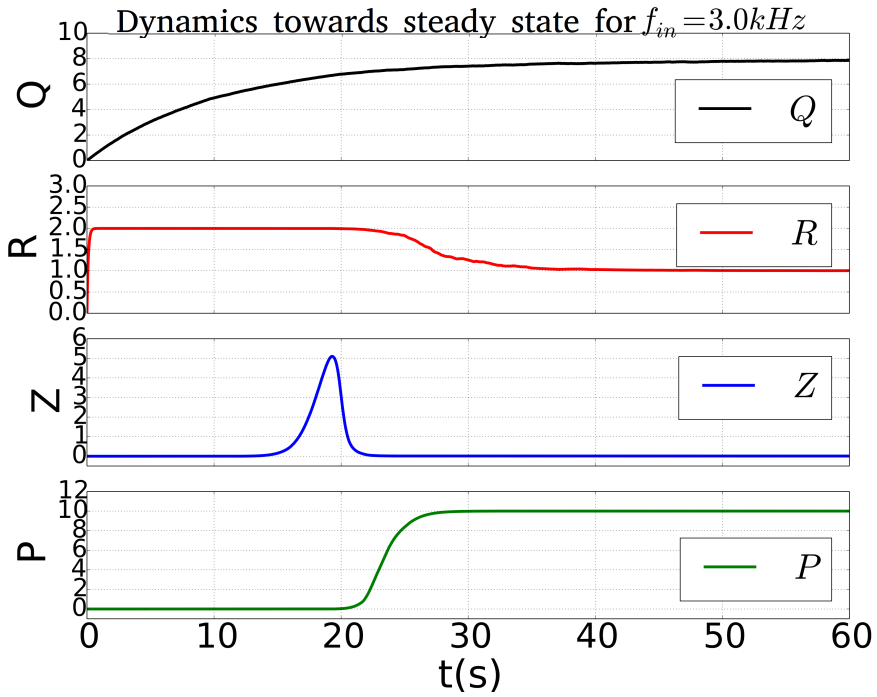

Figure 6.15: Plot of Q, R, Z and P for an input frequency giving a high firing activity.

quite quickly. Not surprisingly the ECM molecules Z and the proteases P stay put at their initial value of zero. Both have activation functions that are in the inactivated region and equal zero for this low activity level, meaning that they experience no activation. The ECM receptors, R, rise quickly to their steady state level of $R = 2$, the increase from the initial value of 0 to the steady state value of 2 happening in less than a second.

In figure 6.15 we see a similar plot but for the high activity steady state. In this run of the model the input frequency is set to $f_{\text{in}} = 3.0$ kHz. Here we see the same tendency, though it takes a bit longer reaching the steady state. This can be easily explained by the seen trend in the plots for the average activity Q. It takes activity some time to grow, and as such the activity noticed by the various ECM molecules matches the low steady state in the beginning, before it increases into the high activity regime. We see the shift between the two steady states when Q is between 6 and 8, with Z responding by first increasing rapidly in response to its activation function rising from 0 to 1. It then drops again quite rapidly, as the decay of Z rises with Z as well as P, and it finally reaches its steady state slightly above zero. The proteases follow suit with a large increase following its own activation function being "activated", while the ECM receptors decrease as their activation function gradually lowers from 2 to 1.

Looking at figure 6.15 we see that the concentration of ECM molecules, Z, is only slightly above zero in the high activity steady state. If we look to figure 4B in the article where the Kazantsev model is described [Kazantsev et al., 2012], given in chapter 4 as figure 4.12, we see an expected value for Z of around five in this steady state. The article gives the relation between the steady state value of Z, and the activity Q , as:

$$Z_\infty = \frac{\beta_z H_z(Q, z_0, z_1, \theta_z, k_z)}{\alpha_z + \gamma_p \beta_p H_p(Q, p_0, p_1, \theta_p, k_p)/\alpha_p}. \tag{6.5}$$

But when solving this equation in the high activity limit using the parameter values given in the figure (which coincide with the parameter values used in my plots), the result is only slightly above zero, with for instance $Z_\infty(Q = 10) \approx 0.01$. From this I gather that there is a mistake in the article, and that the desired behavior of Z is closer to what is seen in figure 4B in [Kazantsev et al., 2012], that the results seen when using the given parameter values. A higher value of Z for high firing activity seems more in place with the indication that increased firing activity leads to increased production of ECM molecules [Dityatev et al., 2007], and it is also more interesting then to examine how changes in Z affect Q.

In order to obtain a behavior similar to that shown in figure 4B in the article, I change the parameters for spontaneous degradation and activation of ECM molecules, α_z and β_z. In the figure these parameters are given as $\alpha_z = 0.001$ msec^{-1} and $\beta_z = 0.01$ msec^{-1}, but in order to achieve a Z_∞ that rises to a maximum almost reaching nine, and then stabilizing around five, I set

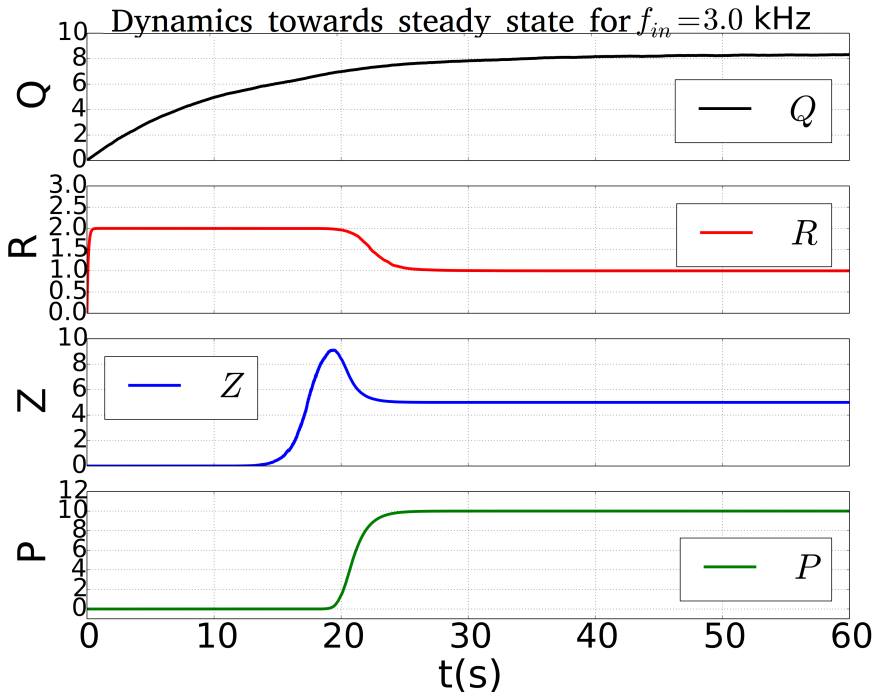

Figure 6.16: Figure as in 6.15, but with updated parameters for the activation function H_z, $\alpha_z = 1$ msec^{-1} and $\beta_z = 5$ msec^{-1}.

$\alpha_z = 1$ msec^{-1} and $\beta_z = 5$ msec^{-1}. With this change implemented, the plot we saw in figure 6.15 becomes instead as in figure 6.16. The behavior of Z is now as expected in this high activity regime.

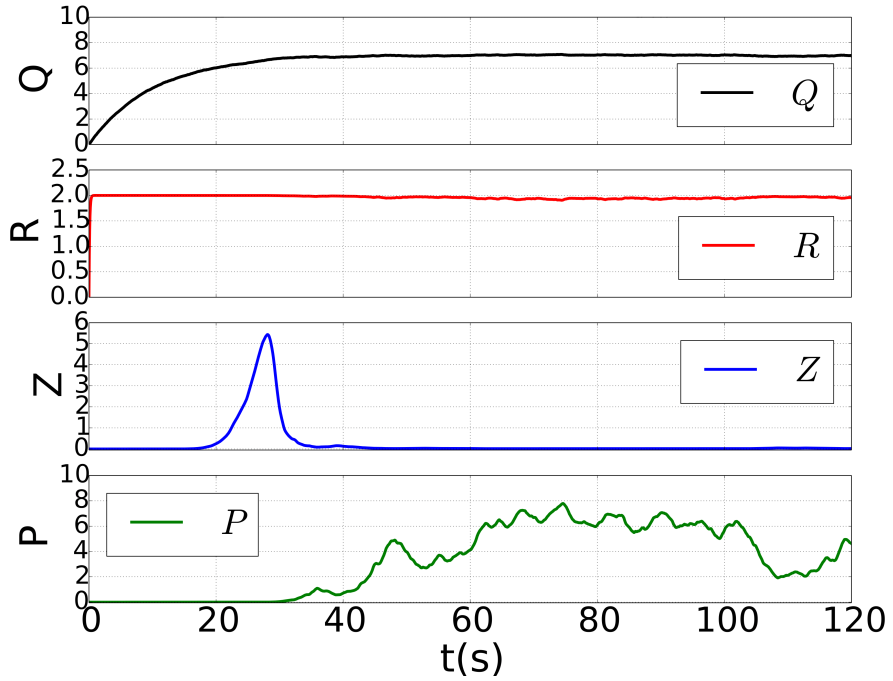

Figure 6.17: A plot showing the result when running the Kazantsev model using $f_{\text{in}} = 1.75$ kHz. All other parameter values as in figure 6.14.

Now let us look more closely into the region between the two steady states, to see how the system behaves in this region. I expect the plots to be more varied in this region, with the random behavior of the synaptic input having more of an impact on the model dynamics. Note that in the following figures, figure 6.17, 6.18 and 6.19, I have used the values for α_z and β_z given in the article $\alpha_z = 0.001$ msec^{-1} and $\beta_z = 0.01$ msec^{-1}, as I noticed the error in Z at a late point in my work, and did not have time to change it. However, the results are expected to be fairly equivalent. Figure 6.17 shows a plot of Q, R, Z and P using an input frequency of $f_{in} = 1.75$ kHz, giving an average activity Q of about 7. This places us approximately in the middle of the activation function for the proteases, as seen in figure 6.13, and as seen in figure 6.17 the result is a strongly varying concentration of proteases.

In figures 6.18 and 6.19 we see the same tendency in the concentrations for Z and R, for activity levels correspondent with the slope of their activation functions.

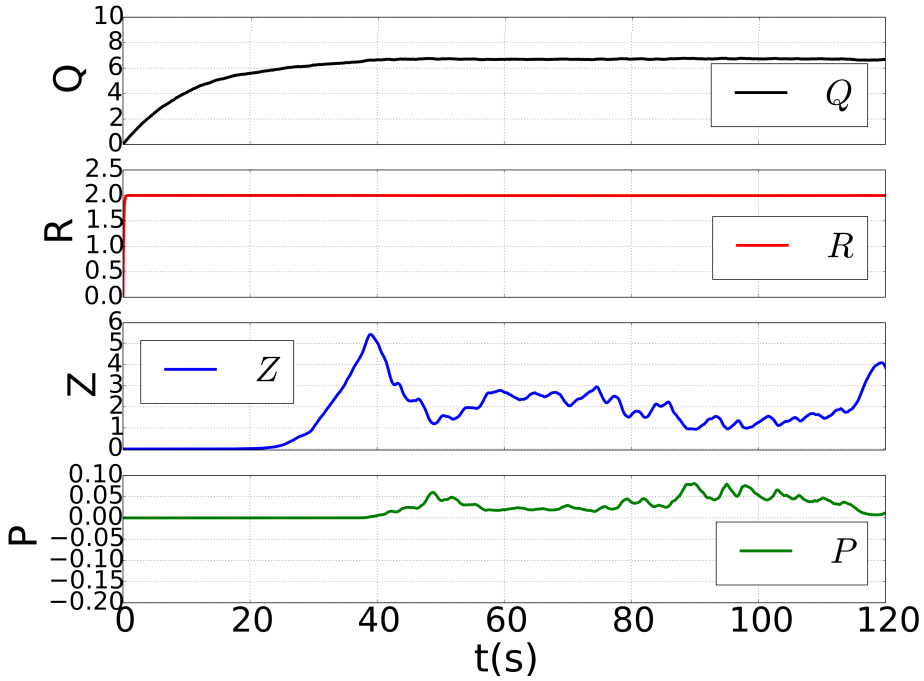

Figure 6.18: A plot showing the result when running the Kazantsev model using $f_{\text{in}} = 1.40$ kHz. All other parameter values as in figure 6.14.

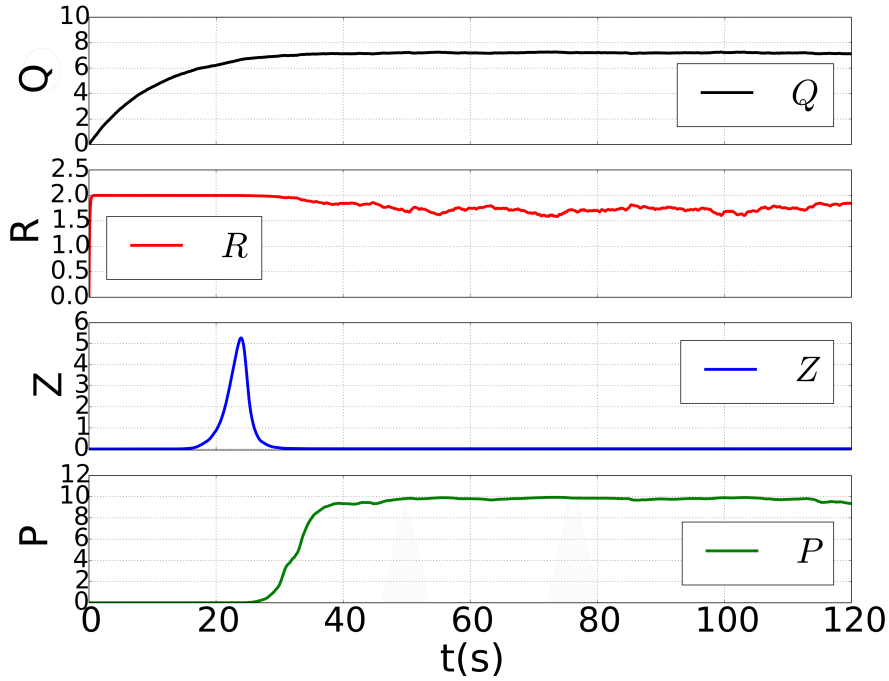

Figure 6.19: A plot showing the result when running the Kazantsev model using $f_{\text{in}} = 2.00$ kHz. All other parameter values as in figure 6.14.

From this we see that there are indeed two steady states for the system, one for low activity and one for high activity, as expected. We also see that in the activity region between these steady states the concentrations Z, P and R are seen to fluctuate with fluctuations that can become quite large.

It is also interesting to look at the consequence of a short period of strongly increased synaptic input. In figure 6.20 I ran the Kazantsev model with synaptic input $f_{\text{in}} = 0.5$ kHz, placing the system in the region for the low activity steady state. After 5 ms, a strong synaptic pulse is added, lasting for 2 ms, and we clearly see how this pulse perturbs the system, pushing it over in the high activity state. The system eventually returns to the lower activity steady state, but this happens long after the original pulse. We can compare this with the plots shown in figure 6.21. Here the Kazantsev model is run without synaptic input, and as such it is solidly in the low activity steady state. After 5 ms the same pulse is added as in figure 6.20, and we see this has a very similar effect. However, now that that there is no synaptic input, Z and R cannot influence the strength of the synaptic input, and as a result the system stays in the high activity region for a noticeably shorter duration, about half as long (note the different run times). Another reason for the prolonged lingering in the high activity region is likely be that the total incoming current is larger in figure 6.20, and further investigations are needed to distinguish these effects.

6.3 Modifying the Kazantsev Model

Now that we have tested and validated the model implementation, it is time to look more closely at the model dynamics at play, to see if we can learn more about the model, and how the model can be used to learn more about the brain, and the interplay between neurons and the ECM.

6.3.1 A Simplified Model with an IF Neuron

An interesting aspect of the Kazantsev model is that the mechanisms it models operate on widely different timescales. The modeling of the electrical activity of the neuron including the generation of action potentials has a timescale of about 1 ms, whereas the changes in the structure and composition of the ECM is a process with a timescale of hours or even days. In [Kazantsev et al., 2012] this complication is solved by assuming that the time scale can be adjusted by a tuning of the necessary parameters. Still it is clear that if we want to model this system to get insight into the variations in ECM composition that happen over the duration of hours or days, and how this again affects the neuronal activity, we are interested in running long simulations. However, this is quite computationally expensive, largely due to the portion of the model involved in spike generation.

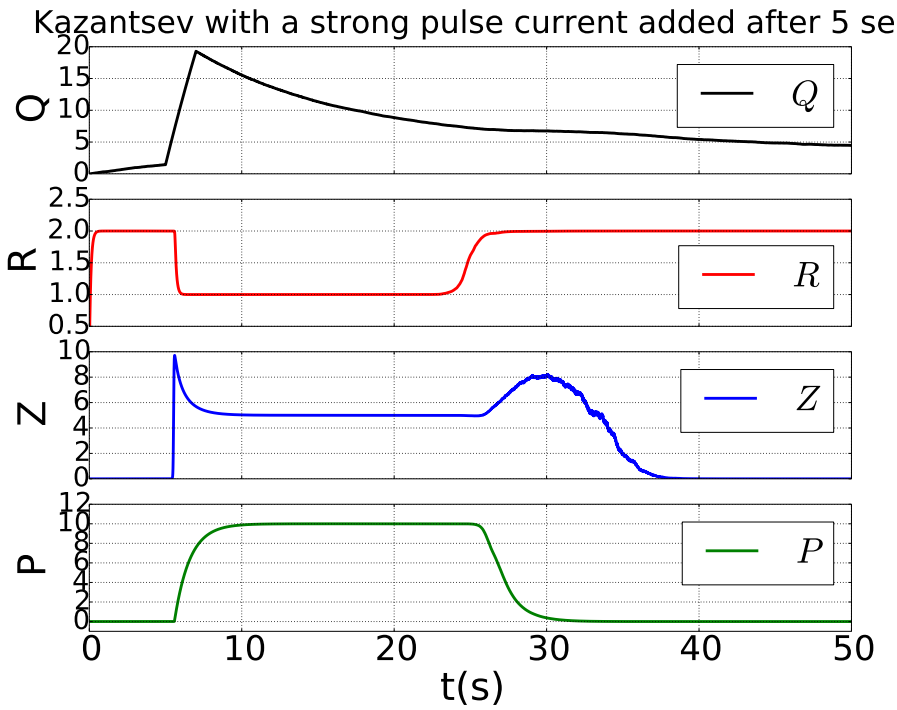

Figure 6.20: The Kazantsev modeled using an input frequency $f_{\text{in}} = 0.5$ kHz. After 5 ms a strong input current is added, lasting 2 ms. A plot showing the result when running the Kazantsev model using $f_{\text{in}} = 1.75$ kHz. The parameter values are as in figure 6.14, but with $\alpha_z = 1$ msec^{-1} and $\beta_z = 5$ msec^{-1}.

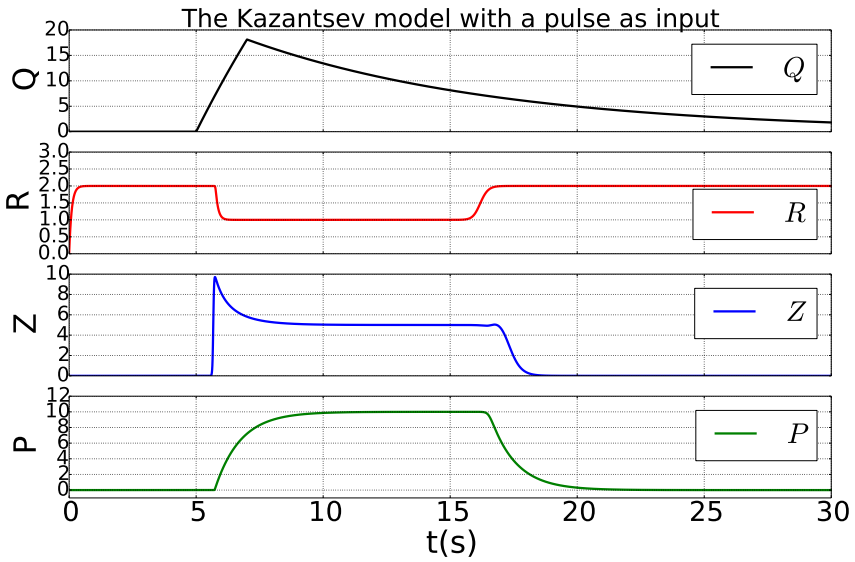

Figure 6.21: The Kazantsev modeled without any synaptic input at all. After 5 ms a strong input current is added, lasting 2 ms. The parameter values are as in figure 6.14, but with $\alpha_z = 1$ msec^{-1} and $\beta_z = 5$ msec^{-1}

Now, what we are interested in in this model is how the firing activity of the neuron affects the ECM and vice versa. As the firing activity is included through the average activity Q, which is simply increased by $\Delta Q = \beta_q \tau$, where τ is the spike duration, for each generated action potential, decaying exponentially between spikes, it seems the computational labor used to model the spike generation is unnecessarily excessive. The Kazantsev model as it stands now consists primarily of eight differential equations, of which three are due to the explicit modeling of the membrane currents.

It is then tempting to look for ways to simplify the model. And one option is to forgo modeling the membrane currents entirely, and instead model the electrical activity of the neuron as a simple integrate-and-fire (IF) neuron. This effectively removes three eights, almost half, of the computational cost of the model.

We want to check if such an integrate-and-fire model of the neuron can replace the Hodgkin-Huxley model used until now. A first step is then to compare the behavior of these two models. We begin by running the model for two minutes, using an integrate-and-fire neuron with simple constant input current $I = 1.544\mu$ A/cm^2 as the only input current. The resulting Q is seen in figure 6.22. Plotted together with it is the result when using the standard Kazantsev model with Hodgkin-Huxley modeling the cross membrane currents, with $f_{\text{in}} = 3.0$ kHz. Apart from the integrate-and-fire version of the model not displaying the irregularity of the standard model the two seem to be in very good

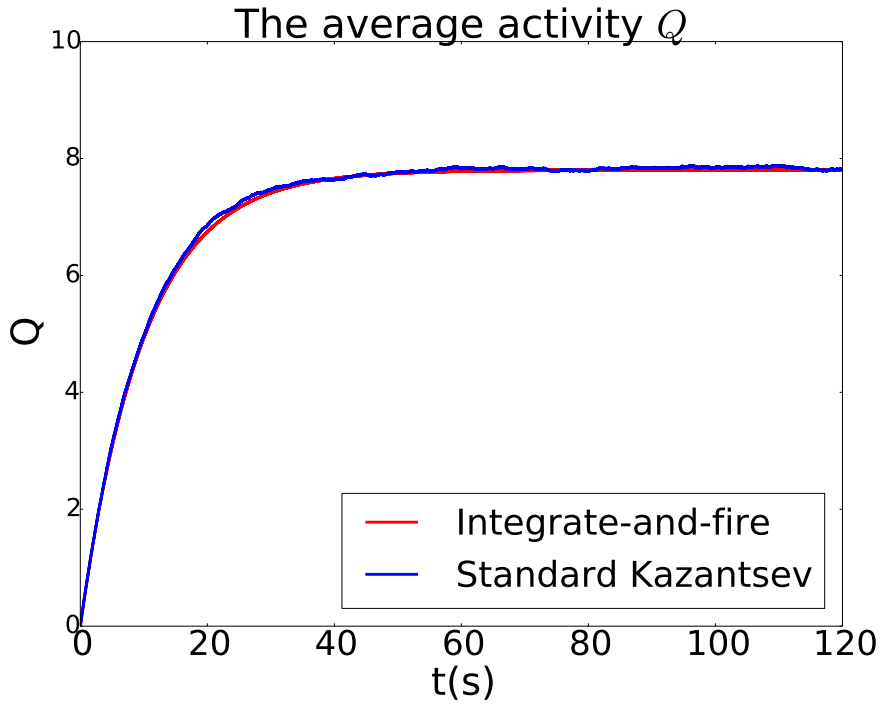

Figure 6.22: A plot comparing the average activity, Q, in the Kazantsev model when the neuron is modeled using the Hodgkin-Huxley model, versus the more simple integrate-and-fire neuron. The Hodgkin-Huxley model gets synaptic input in the form of a Poisson spike train, with $f_{\text{in}} = 3.0$ kHz, while the current into the cell is a constant $I = 1.544\mu\text{A/cm}^2$ for the integrate-and-fire neuron. Parameters used: $R_{\text{m}} = 22\ k\Omega$, $E_{\text{m}} = -64$ mV, $\theta = -49$ mV, $C_{\text{m}} = 1\ \mu$F.

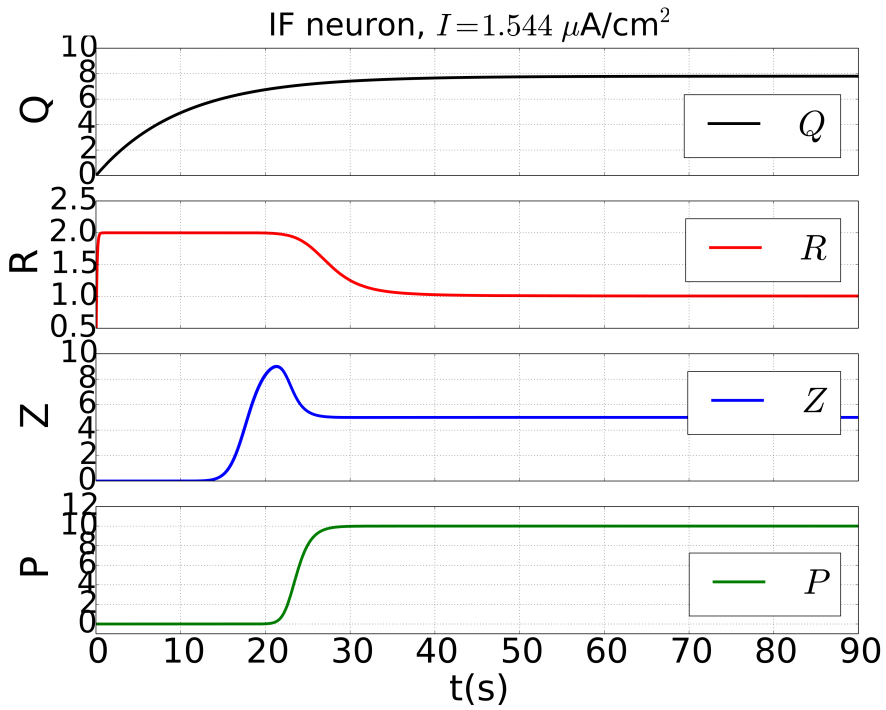

Figure 6.23: A plot of the average activity, Q, and the concentration of Z, R and P in the Kazantsev model when the neuron is modeled using the integrate-and-fire model. The current into the cell is set at a constant $I = 1.544\mu$ A/cm^2.

agreement, reaching the steady state in a very similar manner. This irregularity is due to the standard model experiencing a random synaptic input, and as such it is not reproduced using an IF neuron with constant input current.

Let us now look at the variables representing the concentrations of the various ECM components, and how using and IF model affects these. A plot of the concentrations over time is shown in figure 6.23, where the neuronal activity is modeled as an integrate-and-fire neuron in the same way as in figure 6.22. Comparing this plot with the same plot for a Hodgkin-Huxley style neuron, as seen in figure 6.15, we see that the plots exhibit the same trends and seem to be in very good agreement.

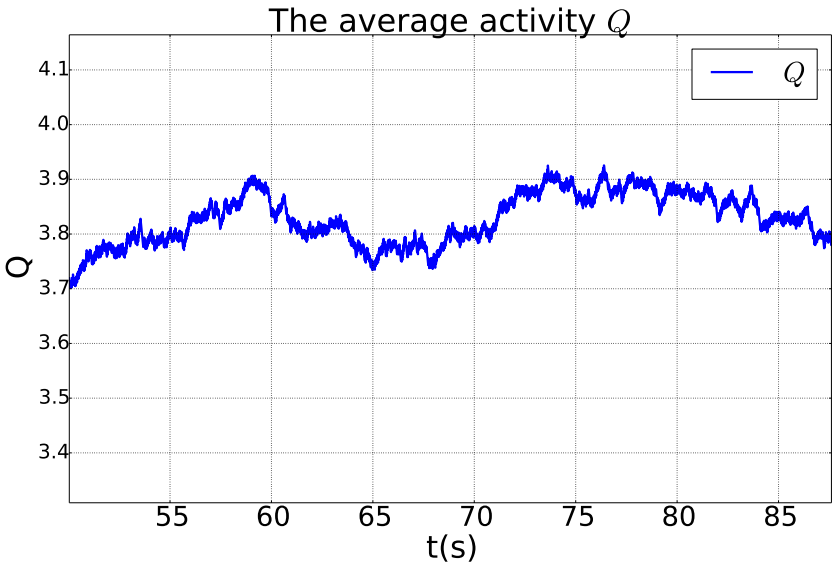

Figure 6.24: A plot illustrating the random dynamics of the average activity Q that arrive due to the random nature of the synaptic input I_{syn} of the standard Kazantsev model.

We have now seen that the much simpler integrate-and-fire (IF) neuron seems to give results that are in good agreement with the original Hodgkin-Huxley modeled neuron. However, as of now the IF model has been given a constant input current. This we saw gave the same general behavior as when using Hodgkin-Huxley formalism, except for giving smoother plots, in particular for Q. This is not easily seen in for instance figure 6.23 due to the long run time, but when taking a closer look it is evident that the randomness of the synaptic input I_{syn} transfers to the average activity Q, as seen more clearly in figure 6.24.

To recreate this behavior as well, while still using an IF neuron for computational efficiency, we simply let the incoming current I in the IF neuron be in the form of a Poisson train. This is done in the exact same way as for I_{syn} in

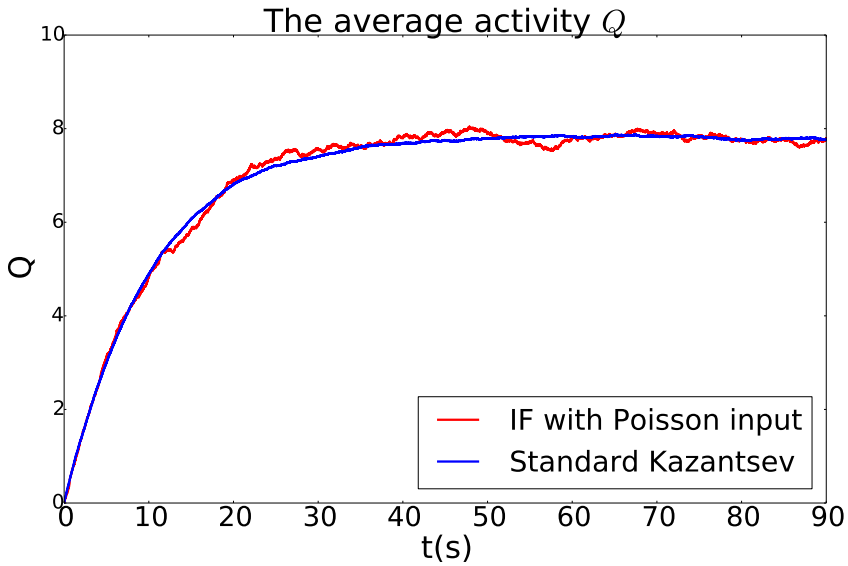

Figure 6.25: A plot comparing the average activity, Q, when using the standard Kazantsev model, versus one where the neuron is modeled using an integrate-and-fire neuron receiving synaptic input in the form om a Poisson train. The input frequency $f_{\text{in}} = 3.0$ kHz for the standard Kazantsev model, and 1.28 kHz for the IF model.

Table 6.1: Run times when simulating for various model times for the Kazantsev model with the neuron modeled first as an integrate-and-fire neuron, and then using the Hodgkin-Huxley model. The ratio is simply the runtime of the first divided by the runtime of the last. Both models were run using a forward Euler numerical scheme with $\mathrm{d}t = 0.001$ ms.

	IF	**HH**	**Ratio**
1 min	15.669	25.182	0.622
2 min	29.310	48.910	0.599
3 min	45.075	73.748	0.611
5 min	74.340	122.531	0.607
10 min	147.357	243.036	0.606

the standard Kazantsev model, although it seems the IF neuron requires a lower f_{in} to give the same activity level. A plot comparing the resulting average activity for these two models is shown in figure 6.25. Here f_{in} is set to 3.0 kHz for the standard Kazantsev model, and 1.28 kHz for the Kazantsev model using an integrate-and-fire neuron. We see from this plot that they display a very similar trend, but the random variations in Q are noticeably larger for the latter.

With the results we have seen so far it seems that using an IF model to model the activity of the neuron in the Kazantsev model gives result that are comparable with the full model using Hodgkin-Huxley to model the neuron's activity. The results seen are very similar for both variations of the model, and using random input for the IF version is seen to give similar random variations in the average activity. Our main motivation for doing this substitution, aside from ease of implementation, was an expected lowering of the computational cost of running the model. To see if such a result was achieved we move on to compare the models in terms of computational cost.

I ran both models for a few total simulation times; 1, 2, 3, 5 and 10 minutes, and compared the resulting run times. A table showing a summary of the results is shown in table 6.1. Also shown in this table is the ratio of the run times for the two model variations. What we see is that the ratio stays fairly constant at approximately 0.6, meaning that both models have a linear complexity $\mathcal{O}(N)$, where N is the number of time steps, but the IF-model has a better coefficient. It is also interesting to note that $\frac{5}{8} \approx 0.6$, while the IF-Kazantsev model consists of five differential equations, and HH-Kazantsev of eight.

All in all we have managed to almost cut the run time of the simulation in half by using a simple integrate-and-fire model for the neuron, while still obtaining near identical results.

6.4 Relating the Model to Experiments

The Kazantsev model is a good and necessary first step towards modeling the influence of the ECM on the neurons it surrounds. However, it is only a first step, and as such it is lacking in some aspects. One way forward is by creating a closer link between the model and experiments. As the model is now, there is only a loose connection between the various parameters and experimental measurements. The model is founded on experimental results, but in a qualitative way more than a quantitative one, with the goal of reproducing the general behavior and dynamics reported in various experimental studies. This includes for instance an increase in the production of ECM molecules when the firing activity is increased [Dityatev et al., 2007] [Dityatev and Schachner, 2003].

To change this, and get a firmer relation between experiments and model, a next step is to find a link between the two. How can we measure the quantities represented in the model, and their response times?

In the Kazantsev model we model the average activity of a neuron enveloped by an ECM, receiving synaptic input from the network it is part of, and the concentration of the ECM molecules, ECM receptors and proteases surrounding it. The concentrations of these molecules are regulated by parameters for spontaneous decay and activation, as well as an activation function depended on the neuron's activity level. By manipulating the decay and activation parameters we can influence the system's behavior, and study how the various molecule concentrations influence each other, as well as the neuronal firing activity.

If one in an experiment can measure the spiking from a neuron, giving the average activity, one can then manipulate the composition of the ECM, and see how the firing activity is affected. If one for instance injected an enzyme or other substance that breaks down the ECM molecules, one can compare the resulting effect on the firing activity with the results when simulating this procedure using the Kazantsev model. This can be done by increasing the degradation rate of the ECM molecules in the model, α_{Z}, and looking at the effects on the firing activity, Q.

Now I want to test how the model reacts to a substantial increase in the degradation parameter of Z, α_z, simulating the injection of a substance breaking down the ECM molecules. Figure 6.26 shows a standard simulation where the degradation parameters are left unchanged through its entire duration. The plot of Q is slightly zoomed in, to better show variations when comparing this figure with figure 6.27. In figure 6.27 α_z is increased after simulating for 60 s, and we clearly see the effect of this in the plot for Z. The concentration Z drops rapidly at this point, eventually reaching a value of about 1.5. The degradation of Z can be made to happen over a longer timespan by gradually increasing α_z over several time steps, but right now we are only interested in a first approach to study the effects on the average firing activity Q. Comparing the plot of Q in figures 6.26

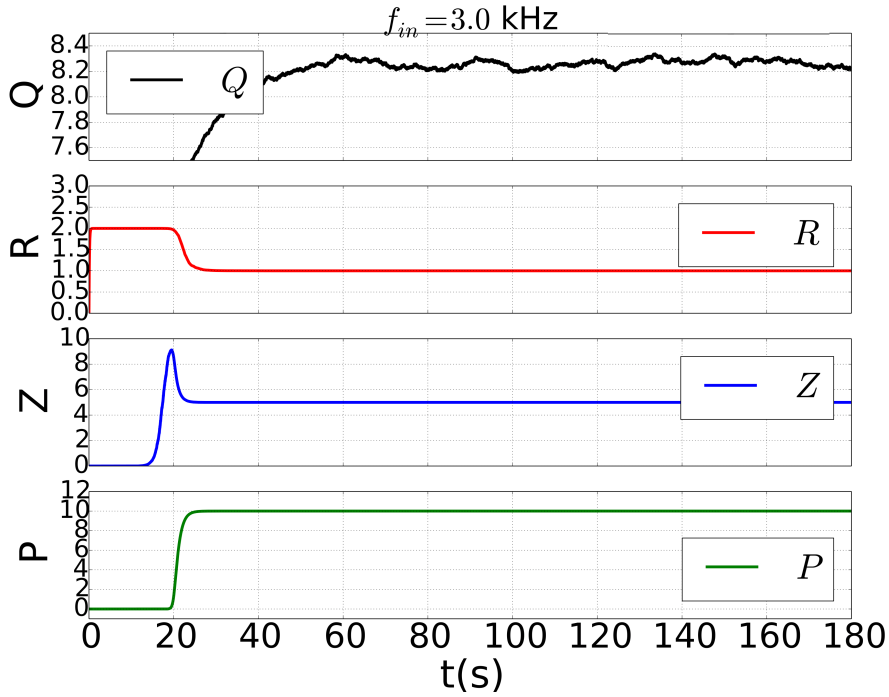

Figure 6.26: A plot showing Q, Z, R and P under normal conditions, with $f_{in} = 3.0$ kHz, and no change in the degration rates during the simulation.

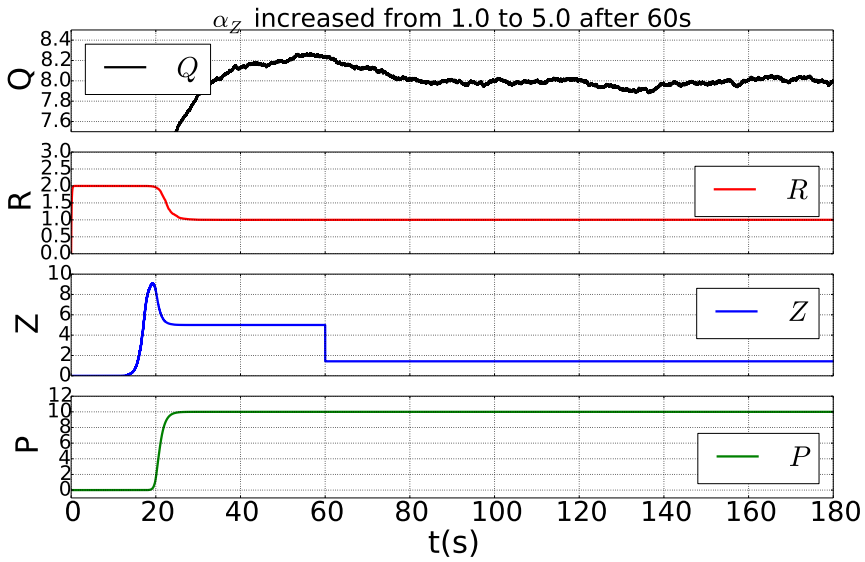

Figure 6.27: This plot shows a simulation under the exact same conditions as in figure 6.26, except the degration rate of the ECM molecules, α_z is increased after 60 s, from $\alpha_z = 1.0\text{msec}^{-1}$ to $\alpha_z = 5.0\text{msec}^{-1}$.

and 6.27, we clearly see that the degradation of Z has led to a decrease in the firing activity of the neuron. This is as we would expect from the model, due to the dependence of the scaling of the synaptic input, b, as well as the threshold current, I_{th}, on Z. The effect on Q is not large enough to affect the levels of P and R, who stay in their high activity steady state.

Chapter 7

Discussion and Suggestions for Further Work

This thesis has revolved around modeling the extracellular matrix and the effect it has on the neuron it surrounds, focusing on the first model developed specifically to model this behavior, the Kazantsev model. I begun by focusing on reproducing the results available for this model, but this process was complicated by a lack of accurate and comprehensive documentation of the procedure and parameters used. The outtake from this exemplifies the importance of facilitating reproducibility. I have studied the dynamics of the model, proposed a simplified version using an integrate-and-fire neuron, and also looked at ways to link the model more closely to experiments.

7.1 Reproducible Science

Reproduction and verification are important components of the groundwork of science, and to make this process possible accurate and detailed information is crucial. It is only when the same results have been obtained in several independent instances that they gain the weight needed to be taken as a good representation of reality. It is by equivalent results being reported independently in several instances, that theories gain credibility, but when information needed to do this is missing, it becomes a difficult task. It is understandable that small errors and bugs are particularly difficult to pick up one when there are (as of yet) no substantial results to compare with, as the case often is when creating a new model. But thorough documentation reduces the risk of this, and allows other to verify your work.

As the dynamics in this case are modeled qualitatively, and the general behavior of my implementation matches that of the original paper in a qualitative way, the discrepancies between the implementations are not so crucial. It is the general behavior we are interested in studying. This is the case for the disparity

between the firing activity in the two implementations of the model. The cause of this inconsistency can stem from a mismatch in parameters, a fault in my implementation or theirs, or a combination of these. As the resulting behavior was found to be very similar qualitatively, this ended up being of lesser importance.

The error in Z_∞ had as such a greater significance, but also seems to be more obviously an error in the original paper, as the behavior reported there did not match the expression given for this quantity.

7.2 Findings in Kazantsev Model

In exploring the Kazantsev model I have looked at the two fairly clear steady states present in the system, one for low and one for high activity. These steady states looked to be quite stable states with very little variation or perturbation. We saw however that in the activity interval between these steady state levels there was a tendency for some rather large fluctuations in the concentrations of Z, P and R. In this area in particular it could be interesting to investigate more thoroughly the effect of temporary perturbations of the system such as a sudden increase of excitatory synaptic input or oppositely a blocking or reduction of input. We saw that such a perturbation in the positive direction could bump the system into the higher activity region, and that the effect of the ECM on the strength of the synaptic input possibly prolonged its stay in this region. This is particularly interesting as it indicates a mechanism that extends periods of higher activity, and memory formation is linked to increased activity. However, the mechanisms here need to be studied more comprehensively.

7.3 Simplifying the Kazantsev model

The model in question is one where the fast dynamics of the spike generation coexist with the slow, modulatory dynamics of the ECM, resulting in two opposing modeling needs. To model the fast spiking accurately we want a small time step and an accurate solver, while the modeling of the slow dynamics requires longer run times, which again is more easily achieved by larger time steps and a solver requiring less computational complexity.

It is the modeling of the spike generation in the nerve cell which requires the most computational power, particularly due to the highly non-linear differential equations for the gating particles. But in the modeling of the effects of the ECM the neurons membrane voltage is not even included directly, but through the activity variable Q. As such the computationally demanding modeling of this voltage seemed unnecessarily excessive. With this in mind I exchanged this part of the model with a far more simple integrate-and-fire neuron model. What we then saw was that this new simplified version of the Kazantsev model was

capable of giving the same or similar results as the original, but at almost half the computational cost. This seems like a clear improvement over the original model. The resulting behavior of the variables were equivalent, except for a lack of statistic fluctuations in the variable Q. We saw then that these fluctuations could be obtained by letting the input to the IF neuron be in the form of a Poisson train of input spikes. This gave larger fluctuations than in the original model, but this seems possible to adjust by modifying the various parameters in the model. It is also worth noting that in doing this the ODEs we are rid of are those for the gating variables, which due to their strong non-linearity can often lead to stiffness in the system. So in addition to creating a smaller system by this substitution, it may now be more robust to an increased dt.

This approach can be taken even further by replacing the explicit modeling of the spike generation with a rate-based model. In the model it is the variable Q that is used to modulate the concentrations of the various components of the ECM. The timing of the individual spikes of the neuron itself are not given much weight as it is the average firing rate that is critical to the model development. This makes sense as we are looking at changes over longer time scales. The individual spike is not expected to have much impact. In a rate-based model the crucial parameter is the firing rate of the neuron. Instead of modeling the generation of each spike in detail through the integration of a differential equation, such a model contains a characteristic function $f(I)$ that converts an input current to a firing frequency directly. An example of such a function is the following sigmoid function [Sterrat et al., 2011]:

$$f(I) = \frac{\overline{f}}{1 + \exp(-k(I - \theta))}, \tag{7.1}$$

where $\overline{f}$ gives the maximum firing rate, while θ represents the firing threshold, as before. The desired random fluctuations of Q can still be obtained, either by giving I as some random input, or by adding noise to the resulting Q. The differential equation $\frac{\mathrm{d}Q}{\mathrm{d}t}$ would however have to be altered slightly, as V is no longer explicitly modeled.

Such rate-based models can be good models despite their simplicity when used in appropriate settings. The Kazantsev model is a good candidate as the timing of individual spikes are not given much weight. The average activity Q can already be seen as a variable for the firing rate in the original model.

7.3.1 Towards Comparison With Experiments

Computational models such as the Kazantsev models and many others are essential to neuroscience as a whole. The nervous system in general and the brain in particular are vastly complex structures and we are dependent on the help of numerical models if we want to get closer to an understanding of its workings.

Models by definition give a simplified view of reality and this allows us to get an overview of complex systems, and a clearer understanding of specific elements of the system.

Experimental results are what tie theories and models to reality and the key to new understanding is often found in the interplay between a model and experiments. In order to create such a link between the Kazantsev model and possible experiments it would be beneficial if the constituents and parameters of the model were given a tighter connection measurable quantities. In particular a more clear conformity between the parameters and variables for the various constituents of the ECM, and measurable properties.

With this in mind a sensible step forward might be to expand the model to include a finer division of the variables relating to the concentrations of the ECM components. This can for instance be done by dividing the ECM receptors into specific receptor types. It is likely that various receptor types will have differing responses to a varied stimuli such as an increase in firing activity. In order to be able to give predictions for experimental results it is beneficial if this and similar behaviors are accounted for. This also encompasses responses to enzymes and the like that can be added to the system to break down (parts of) the ECM, which may for instance only target specific ECM molecules.

Chapter 8

Conclusion

This thesis has been based around the modeling of the effect of the extracellular matrix on the firing rate of the neuron it surrounds. We have looked at an early model for this interplay, the Kazantsev model, and examined the dynamics of the system it describes. In this work we discovered that there were quantitative discrepancies between the results given by my implementation, and the original paper. However, the qualitative behavior was seen to be analogous, and as the model aimed to describe the dynamics in a qualitative way, the disparities were taken to be of lesser concern.

Being a first approach to modeling the aforementioned interplay between the neuron and the surrounding ECM, the model takes a fairly simplistic approach. To move forward one may want to expand upon the model, making it more detailed and with a clearer link to experimentally measurable quantities, with the goal of being able to use the model to make experimentally verifiable predictions.

Bibliography

[Ascher and Petzold, 1998] Ascher, U. M. and Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam.

[Auld and Robitaille, 2003] Auld, D. S. and Robitaille, R. (2003). Glial cells and neurotransmission: an inclusive view of synaptic function. *Neuron*, 40(2):389–400.

[B et al., 2002] B, A., A, J., J, L., and et al. (2002). *Molecular Biology of the Cell: The Extracellular Matrix of Animals*. Garland Science, fourth edition. Available from: http://www.ncbi.nlm.nih.gov/books/NBK26810/.

[Blume, 2015] Blume, C. (2015). libunittest c++ library - a portable c++ library for unit testing. http://sourceforge.net/projects/libunittest/?source=directory. Accessed: 14.06.2015.

[Bower and Beeman, 2003] Bower, J. and Beeman, D. (2003). *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System, Ch 4*. Springer-Verlag, second edition. Available from: http://www.genesis-sim.org/GENESIS/bog/bog.html.

[Celio et al., 1998] Celio, M. R., Spreafico, R., De Biasi, S., and Vitellaro-Zuccarello, L. (1998). Perineuronal nets: past and present. *Trends in neurosciences*, 21(12):510–515.

[Dityatev et al., 2007] Dityatev, A., Brückner, G., Dityateva, G., Grosche, J., Kleene, R., and Schachner, M. (2007). Activity-dependent formation and functions of chondroitin sulfate-rich extracellular matrix of perineuronal nets. *Developmental neurobiology*, 67(5):570–588.

[Dityatev and Schachner, 2003] Dityatev, A. and Schachner, M. (2003). Extracellular matrix molecules and synaptic plasticity. *Nature Reviews Neuroscience*, 4(6):456–468.

[Dityatev et al., 2010] Dityatev, A., Schachner, M., and Sonderegger, P. (2010). The dual role of the extracellular matrix in synaptic plasticity and homeostasis. *Nature Reviews Neuroscience*, 11(11):735–746.

[Frantz et al., 2010] Frantz, C., Stewart, K. M., and Weaver, V. M. (2010). The extracellular matrix at a glance. *Journal of cell science*, 123(24):4195–4200.

[Greenough, 1988] Greenough, W. (1988). Plasticity of synapse structure and pattern in the cerebral cortex. *Cerebral cortex*, 391.

[Hockfield et al., 1990] Hockfield, S., Kalb, R., Zaremba, S., and Fryer, H. (1990). Expression of neural proteoglycans correlates with the acquisition of mature neuronal properties in the mammalian brain. In *Cold Spring Harbor symposia on quantitative biology*, volume 55, pages 505–514. Cold Spring Harbor Laboratory Press.

[Hodgkin and Huxley, 1952] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/pdf/jphysiol01442-0106.pdf.

[Karetko and Skangiel-Kramska, 2009] Karetko, M. and Skangiel-Kramska, J. (2009). Diverse functions of perineuronal nets. *Acta Neurobiol Exp (Wars)*, 69(4):564–577.

[Kazantsev et al., 2012] Kazantsev, V., Gordleeva, S., Stasenko, S., and Dityatev, A. (2012). A homeostatic model of neuronal firing governed by feedback signals from the extracellular matrix. *PloS one*, 7(7):e41646.

[Kwok et al., 2011] Kwok, J. C., Dick, G., Wang, D., and Fawcett, J. W. (2011). Extracellular matrix and perineuronal nets in cns repair. *Developmental neurobiology*, 71(11):1073–1089.

[Marsh et al., 2012] Marsh, M. E., Ziaratgahi, S. T., and Spiteri, R. J. (2012). The secrets to the success of the rush–larsen method and its generalizations. *Biomedical Engineering, IEEE Transactions on*, 59(9):2506–2515.

[Project, 2015] Project, T. Q. (2015). Qt creator manual. http://doc.qt.io/qtcreator/. Accessed: 14.07.2015.

[Purves et al., 2001] Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., LaMantia, A.-S., McNamara, J. O., , and Williams, S. M. (2001). *Neuroscience*. Sinauer Associates, Inc, 23 Plumtree Road,P.O. Box 407,Sunderland, MA 01375-0407, USA, third edition.

[Reimers et al., 2007] Reimers, S., Hartlage-Rübsamen, M., Brückner, G., and Roßner, S. (2007). Formation of perineuronal nets in organotypic mouse brain slice cultures is independent of neuronal glutamatergic activity. *European Journal of Neuroscience*, 25(9):2640–2648.

[Rush and Larsen, 1978] Rush, S. and Larsen, H. (1978). A practical algorithm for solving dynamic membrane equations. *Biomedical Engineering, IEEE Transactions on*, (4):389–392.

[Squire et al., 2008] Squire, L., Bloom, F., Spitzer, N., du Lac, S., Ghosh, A., and Berg, D. (2008). *Fundamental Neuroscience*. Elsevier, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, third edition.

[Sterrat et al., 2011] Sterrat, D., Graham, B., Gillies, A., and Willshaw, D. (2011). *Principles of Computational Modeling in Neuroscience*. University Press, Cambridge, The Edinburgh Builind, Cambridge CB2 8RU, UK.

[Thompson, 1993] Thompson, R. F. (1993). *The Brain: A Neuroscience Primer*. W. F. Freeman and Co, second edition.

[Tsien, 2013] Tsien, R. Y. (2013). Very long-term memories may be stored in the pattern of holes in the perineuronal net. *Proceedings of the National Academy of Sciences*, 110(30):12456–12461.

[tutorialspoint, 2014] tutorialspoint (2014). Polymorphism in C++. http://www.tutorialspoint.com/cplusplus/cpp_polymorphism.htm. Accessed: 08.10.2014.

[Vianna et al., 2000] Vianna, M. R., Alonso, M., Viola, H., Quevedo, J., de Paris, F., Furman, M., de Stein, M. L., Medina, J. H., and Izquierdo, I. (2000). Role of hippocampal signaling pathways in long-term memory formation of a nonassociative learning task in the rat. *Learning & Memory*, 7(5):333–340.

[Wang and Fawcett, 2012] Wang, D. and Fawcett, J. (2012). The perineuronal net and the control of cns plasticity. *Cell and tissue research*, 349(1):147–160.

[wikibooks, 2015] wikibooks (2015). Object oriented programming. https://en.wikibooks.org/wiki/Object_Oriented_Programming. Accessed: 14.07.2015.

[Williams and Herrup, 1988] Williams, R. W. and Herrup, K. (1988). The control of neuron number. *Annual review of neuroscience*, 11(1):423–453.

[wpclipart, 2015] wpclipart (2015). Neuron. http://www.wpclipart.com/medical/anatomy/cells/neuron/neuron.png.html. Accessed: 08.05.2015.