Assessment, Integration and Implementation of Computationally Efficient Models to Simulate Biological Neuronal Networks on Parallel Hardware

Giacomo Bassetto

Department of Information Engineering

University of Padova

A thesis submitted for the degree of

Laurea Magistrale in Bioingegneria (Master of Engineering in Bioengineering)

March 2013

Acknowledgements

I would like to acknowledge the following people for their contribution to the completion of this thesis. Alessandra Bertoldo, my thesis advisor. Stefano Vassanelli, my co-supervisor, who offered invaluable guidance and perspectives. Mufti Mamhum, for the suggestions he gave me while I was writing this thesis and for his friendly support. Special thanks also to Giorgio Urso, who provided software assistance and valuable knowledge in graphic hardware programming.

I would like to extend my thanks to the lab team, Marta, Claudia, Stefano, Mostafiz and Mufti for the friendly environnment they create in the lab.

Abstract

In neuroscience, the existing literature suggests a high level knowledge on the properties of the basic components of the central nervous system (CNS). The next step is to understand how the building blocks of the CNS cooperate and interact to accomplish different tasks (e.g., cognition, response to external stimuli, etc..). To do that, a simulation tool capable of handling large-scale network simulations is needed. The high level of complexity about the simulation in terms of parameter tuning and large-scale of data handling has always required *ad hoc* hardware (super computers or large cluster). However, these resources are expensive and not always available. Modern graphics processors (GPUs), provide a high-performance, programmable, low-cost platform on which large-scale neuronal network simulations can run. The aim of the present work is to develop a simulation tool that is capable of (1) closely mimicking the physiological phenomena of the neurons and their networks, and (2) exploiting the computational capabilities of the new available hardware. To these goals, different models have been extensively analyzed looking for those that provide the best compromise between quality of the predictions and computational efficiency, with the possibility of implementation on parallel hardware. The proposed software will be available to the scientific community in order to provide researchers with a versatile, powerful and easy to access tool to simulate large-scale neuronal networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem overview

The properties of the basic components of the central nervous system (CNS) have been investigated for more than 50 years, and now the mechanisms underlying the computational properties of single neurons have been understood. Also the basic circuitry of cortical networks has been extensively studied (for a review please refer to Thomson & Bannister [2003]). The attention in now focused on the interactions and coding strategies of neurons, to understand how they cooperate in big networks to accomplish specific tasks, like vision, pattern recognition, learning.

Simulation is a very powerful tool to study neuronal networks, because *in silico* models do not need particular preparations to be investigated and, moreover, they allow to explore different experimental conditions with small effort[1].

Up to now the simulation of large-scale[2] spiking neuronal networks (SNN) has required super-computers or large cluster (e.g. Izhikevich & Edelman [2008]), expensive resources that are not always available. Even the simulation of "smaller" networks (few thousands of neurons or less) using a normal personal computer is not comfortable, because the amount of data to manage leads to prohibitive

[1] e.g. a simulator could allow to simulate a network with or without a particular set of connections to test its behavior for the two different configurations, simply changing an ounce of parameters. Performing the same experiment on a real neuronal network might be very difficult, if not impossible.

[2] Large-scale means "on the order of magnitude of tens of thousands of neurons".

computational times.

1.1.1 Proposed solution

Neural networks are "parallel" in nature, as they are made of a lot of similar interacting entities. Modern graphics processors (GPUs), being able to process simultaneously hundreds or even thousands of entities, provide a high-performance, programmable, low-cost platform on which large-scale SNN simulations can run. Exploiting their computational power can boost incredibly the performances of a simulator.

Some work has been done in this direction (e.g. Nageswaran *et al.* [2009], Yudanov *et al.* [2010]), but so far, to our knowledge, no really realistic SNN simulator especially designed for graphic hardware yet exists.

1.2 Motivations

Understanding how biological neuronal networks process information to accomplish high level tasks can be the starting point to develop new biologically-inspired computational strategies. Simulation studies of large population of neurons are limited by the cost of the resources that have been traditionally needed. A simulation software exploiting modern GPUs is an accessible alternative resource.

1.3 Aim of the work

Our intention is to provide the scientific community with an easily accessible tool for large-scale realistic SNN simulations.

We are developing a simulation tool that exploits the computational capabilities of the new available hardware. The most important requirement is that it closely mimics the physiological phenomena of the neurons and their networks. Our attention is particularly focused on the thalamo-cortical system (Figure 1.1), but we cannot exclude that we may extend it to model other types of networks in the future (e.g. the cerebellar cortex). The work is divided in three phases:

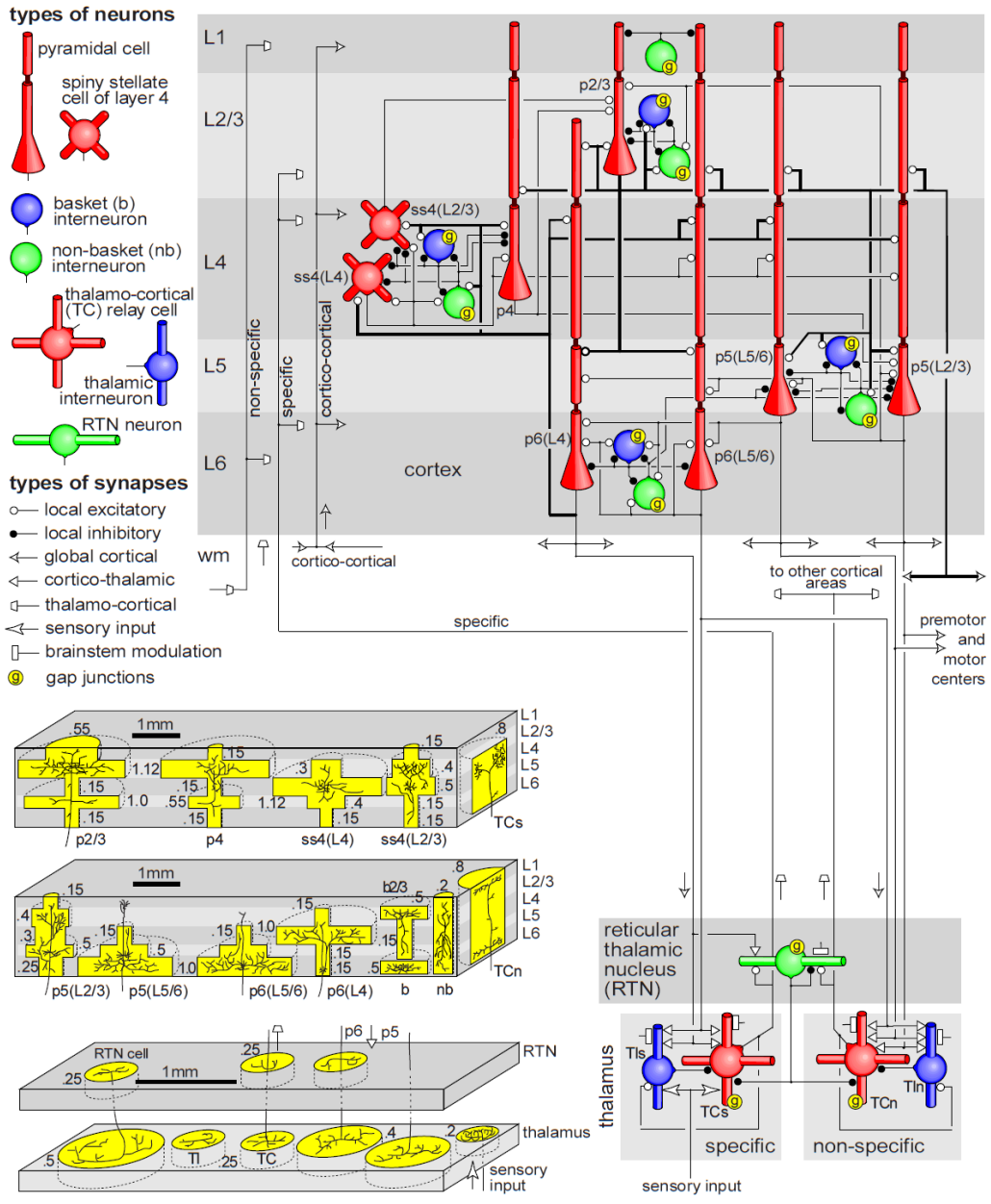


Figure 1.1: Simplified diagram of the microcircuitry of the cortical laminar structure (*Upper*) and thalamic nuclei (*Lower*). Neuronal and synaptic types are as indicated. Only major pathways are shows in the figure; bold lines denote pathways carrying more than 30% of synapses to a particular target. Arrows indicate types and directions of internal signals. Self-synapses denote synaptic connections within the populations. *L1-L6* are cortical layers; *wm* refers to white-matter. From Izhikevich & Edelman [2008].

1. Assess various models of the physiological process that will be simulated, finding the ones that provide good predictions and better adapt to be implemented on parallel hardware.

2. Design and implement the simulation software, using the models selected in section 1.

3. Create a tool for the automatic creation of complex networks (e.g. Figure 1.1) according to the design defined in (2), possibly based on existing shared protocols (e.g. *NeuroML* – http://www.neuroml.org) for compatibility with other existing simulation tools (*Neuron* – http://www.cell.com/neuron, *Brian* – http://briansimulator.org, *GENESIS* – http://genesis-sim.org).

The present thesis focuses on phases (1) and (2).

1.4 Organization

The rest of the thesis is organized as follows. Chapter 2 provides background information on neural physiology, while Chapter 3 gives an overview on CUDA-enabled GPUs' architecture. In Chapter 4 and Chapter 5 neuronal and synaptic mathematical models are introduced and evaluated. Chapter 6 presents the design and the implementation of the software. Finally, some appendices provide additional information about physiology, mathematical models and some code listings.

Chapter 2

Physiology

*I*n this chapter we will describe the physiological basis underlying electric phenomena and communication in the nervous system. For further information about the subjects treated in this chapter, please refer to D'angelo & Peres [2012].

2.1 Morphology

Neurons in the mammalian central nervous system come in many different forms. Given the diversity of functions performed by neurons in different parts of the nervous system, in fact, there is a wide variety in shape, size, and electrochemical properties of neurons. From a functional point of view, however, neurons generally have four important zones:

1. an input zone consisting of a variable number of *dendritic* membrane processes extending for hundreds of micrometers, where signal from other cells are integrated. Due to its particular structure, this zone is usually called "dentritic tree";

2. an elaboration site, where integrated signals are processed to elicit a response to be propagated to other cells. This zone is called *axon hillock*;

3. a cable-like membrane projection, generating from the axon hillock, called *axon*, that carries the information along the nerve. Most of the neurons have

only one axon, but it usually undergoes extensive branching, enabling communication with many target cells. To minimize metabolic expense while maintaining rapid conduction, the axons of many nerves are myelinated, that is, they acquire a sheath of myelin, a protein-lipid complex that is wrapped around the axon. The sheaths are formed by glial cells: oligodendrocytes in the central nervous system and Schwann cells in the peripheral nervous system. This sheath enables action potentials to travel faster than in unmyelinated axons of the same diameter, whilst using less energy;

4. the nerve/axon endings, where communication with other cells takes place through particular structures called *synapses*.

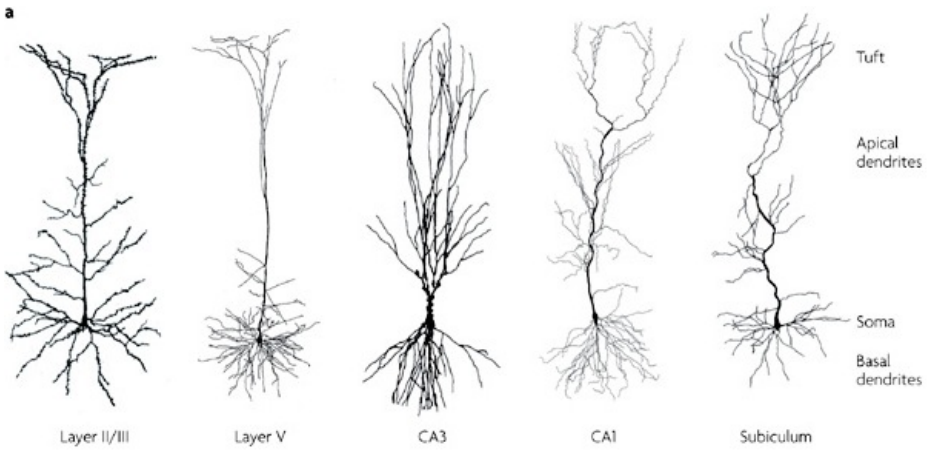


Figure 2.1: Morphology of various type of pyramidal cells in the cerebral cortex

The cell body (called *soma*), containing the nucleus and other important cellular structures, usually corresponds to the root of the dendritic tree and contains the axon hillock, but it can be located within the axon (like in auditory neurons) or attached to the side of the axon (for example, cutaneous neurons). The diameter of the soma can vary from 5 to 50 µm.

2.2 Membrane Potential

The cell membrane is a biologic structure that separates the interior of a cell from the surrounding environment and controls the movements of substances in

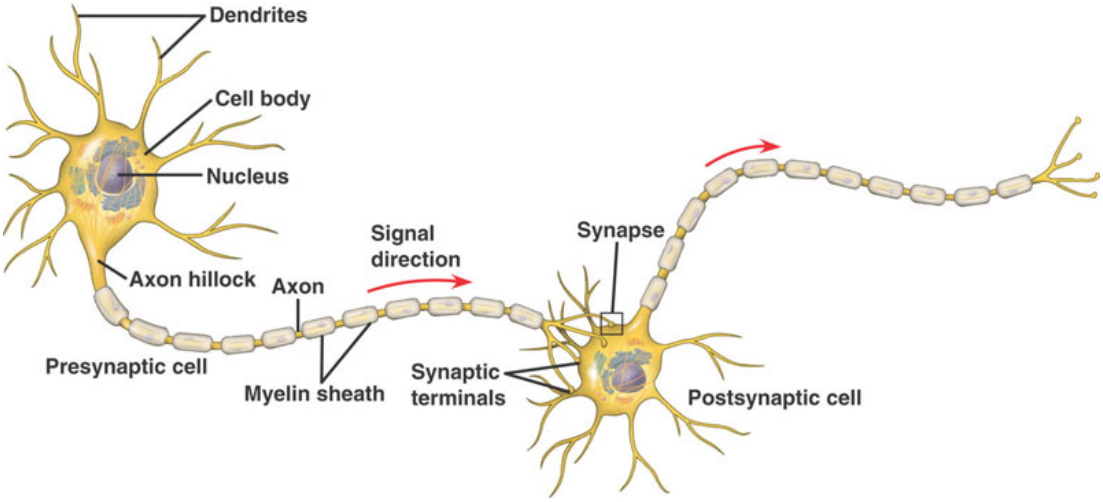


Figure 2.2: Representation of two neurons.

and out of cells. It consists of a phospholipides bilayer where several proteins are embedded. Phospholipides' polar head are hydrophilic and then they spontaneously organize to face the surrounding polar fluid, while their apolar tails are oriented towards the interior of the membrane. Proteins are needed to allow the cell to exchange small polar or charged molecules, as is the case for ions, with the external environment, which otherwise could not cross the membrane because of the interior apolar region. These proteins can be divided in two categories: ion channels, which allow passive diffusion of ions down concentration gradients, and ion transporters/pumps which actively push ions across the membrane to establish those concentration gradients.

When a thin electrode is inserted inside a cell, a constant potential difference can be observed with a reference electrode kept in the extracellular fluid. This potential difference is negative and of few tens of millivolts in amplitude. Since ions are present at different concentrations inside and outside the cell ($[K^+]_{out} < [K^+]_{in}$ while the opposite is true for Na^+, Cl^- and Ca^{2+} ions), their movements are influenced both by the chemical gradients and the electric field responsible of the membrane potential. The observed potential results as a dynamic equilibrium where the net ions flux across the membrane is zero. This is not true for each ion indeed. The value of the membrane potential at which the net flux of a single ion

X is zero is called the *reversal potential* of ion X (E_X) and can be obtained by

$$E_X = \frac{RT}{z_X F} \ln \frac{[X]_{\text{out}}}{[X]_{\text{in}}} \tag{2.1}$$

where $R = 8.314\,472\,\text{J}\,\text{mol}^{-1}\,\text{K}^{-1}$ is the universal gas constant, $F = 96\,485\,\text{C}\,\text{mol}^{-1}$ is the Faraday constant, T is the absolute temperature in K, z_X is the valence of the ion, $[X]_{\text{out}}$ and $[X]_{\text{in}}$ are the concentrations of the ion X respectively outside and inside the cell in $\text{mol}\,\text{m}^{-3}$. (2.1) is called *Nernst equation*. It is worth noting that the reversal potential of a given ion can be very different from the steady-state value of the membrane potential, that is given by the *Goldman-Hodgkin-Katz (GHK) equation*:

$$E_m = \frac{RT}{F} \ln \frac{P_{Na^+}[Na^+]_{\text{out}} + P_{K^+}[K^+]_{\text{out}} + P_{Cl^-}[Cl^-]_{\text{in}}}{P_{Na^+}[Na^+]_{\text{in}} + P_{K^+}[K^+]_{\text{in}} + P_{Cl^-}[Cl^-]_{\text{out}}} \tag{2.2}$$

where

- $[ion]_{\text{out}}$ and $[ion]_{\text{out}}$ are respectively the extracellular and intracellular concentrations of that ion in $\text{mol}\,\text{m}^{-3}$
- P_{ion} is the membrane permeability for that ion expressed in $\text{m}\,\text{s}^{-1}$
- R, T and F are defined as in (2.1)

Ion	Int (mM)	Ext (mM)	Rev (mV)
Sodium (Na^+)	49	440	59
Potassium (K^+)	410	22	0
Chloride (Cl^-)	40-100	560	-65
Calcium (Ca^{2+})	0.0002	10	145

Table 2.1: Intracellular and extracellular concentrations of the most important ions and their respective reversal potentials for the squid axon. Int: intracellular concentration; Ext: extracellular concentration; Rev: reversal potential.

2.2.1 Electric Model of the Membrane

Cytoplasm and extracellular fluid are electrolytic solutions, and then they are electric conductors, while the membrane is an insulator. Moreover, the existence of a potential difference across the membrane means that there is an unbalanced number of electric charges both inside and outside the cell. In this particular case, as the resting membrane potential is negative, there is a surplus of negative charges inside the cell and a surplus of positive charges outside. These charges are attracted by the electric field and lay on the surface of the membrane. Hence the phospholipidic bilayer can be represented with a capacitor. As membranes contains also structures that can be passed through by charged particles (e.g. ion channels and transporters), the electric representation of the membrane must include resistors connected in parallel with the capacitor. The conductance of the membrane is very low and can vary depending on the number of proteins and their conformation (open/closed/inactivated state[1]).

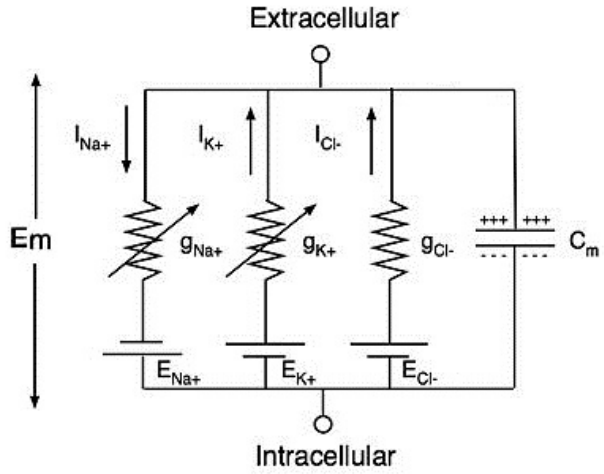


Figure 2.3: Equivalent electric circuit of the cell membrane.

Figure 2.3 shows the electrical model of the membrane: the three conductances represent the permeability of the membrane to each specific ion, while the three batteries represent the voltage difference driving them, as given by (2.1). This

[1] see section 2.3

circuit is described by the first order differential equation (2.3)[2]

$$C_m \dot{V} = -\sum_{\text{ion}} I_{\text{ion}} \tag{2.3}$$

where ionic currents[3] are given by

$$I_{\text{ion}} = G_{\text{ion}}(V - E_{\text{ion}}) \tag{2.4}$$

It is worth noting that for steady-state conditions (2.3) gives $\sum_{\text{ion}} I_{\text{ion}} = 0$, i.e. the sum of all the ionic currents is zero, but that's not true for each of them (see (2.4)).

2.3 Ion Channels

Ion channels are integral membrane proteins with a central pore that allow ions to move passively across the membrane down their electrochemical gradient. These proteins can be in two main conformation, one in which the pore is open (*open state*) and an other in which the pore is closed (*closed state*). A lot of different ion channels exists, but they all share two common properties: selectivity and gating.

Gating is the mechanism governing morphological transition between open and closed state. Depending on the stimulus that opens or closes the pore, ion channels can be divided in *voltage-gated* (they respond to changes of the membrane potential) or *ligand-gated* (conformation state changes when the protein binds a specific ligand molecule) ion channels. These two gating mechanisms are not mutually exclusive, but ion channels generally adopt one or the other. When both of them are present in the same channel, one prevails on the other.

Selectivity refers to the ability of the channel to choose which particular ion specie can pass through it. Thus for example a channel for sodium (Na^+) is permeable to Na^+ ions but not to K^+ or Ca^{2+}.

[2] In this thesis the membrane potential will be labeled either as E_m or V, preferring the former in chemical equations and the latter when describing equivalent electric circuits.

[3] Ionic currents sign convention: positive from inside to outside.

2.3.1 Channels opening and closing mechanisms

Figure 2.4 show a typical current recording through a single channel for a fixed voltage across the membrane. Even if in steady-state conditions, the current

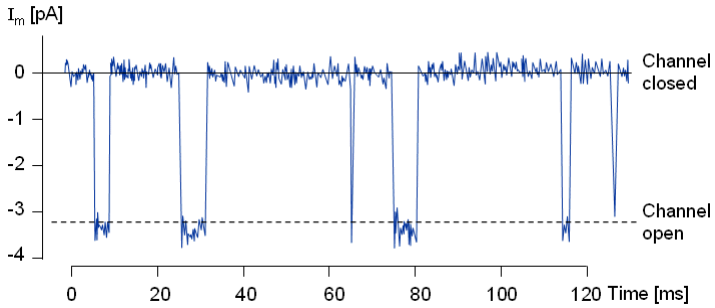


Figure 2.4: Single channel recording of a voltage-gated ion channel.

through the channel switches frequently between two levels (see Figure 2.4): level zero (closed channel) and another one of few picoamperes (open channel). It must be noticed that the duration of the opening and closing events shows a certain degree of variability, suggesting a stochastic mechanism controlling morphological changes in the channel protein. In the simplest case, transitions between the two states can be described as:

$$C \underset{\beta(V)}{\overset{\alpha(V)}{\rightleftharpoons}} O \tag{2.5}$$

where C and O represent respectively closed and open states, while α and β represent the number of transition in each directions occurring in a unit time. Continuous macroscopic currents observed in whole-cell patch clamp recordings arise from the summation of a lot of single "on/off" stochastic currents at the microscopic level.

An other important parameter used to characterize the activity of a ion channel is its opening probability p_O. Depending on the type of gating mechanism, p_O can be described as:

- voltage-gated channels

$$p_O = \frac{1}{1 + \exp\left(-\dfrac{z_g\, e(E - E_{1/2})}{kT}\right)} \tag{2.6}$$

  where e is the elementary charge, k is the Boltzmann's constant, T the absolute temperature. $E_{1/2}$ and z_g are parameters varying from channel to channel and set respectively the potential at which $p_O = 0.5$ and the slope of the curve in that point.

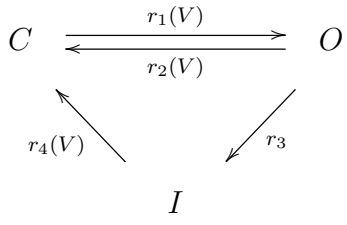
- ligand-gated channels

$$p_O = \frac{1}{1 + \left(\dfrac{K_D}{[L]}\right)^n} \tag{2.7}$$

  known as *Hill's equation*, where $[L]$ is the concentration of ligand. K_D is known as dissociation constant and it is the concentration of ligand needed to make $p_O = 0.5$, while n describe the steepness of the relation between p_O and $[L]$.

Inactivation

Some types of voltage-gated ion channels, when stimulated by a long-lasting depolarizing pulse, are permeable only for a brief period of time, and then they enter a non-permeable state, although the depolarization lasts. This phenomenon is called *inactivation* and can be found, for example, in the Na^+ voltage-gated channels responsible of the action potential. To take in account this additional behavior, one more additional state must be added to the kinetic scheme (2.5), resulting then:



(2.8)

States C and I are indistinguishable analyzing a patch clamp recording, being both of them non conductive states, but note that channels can recover from inactive state only once the membrane become repolarized[4].

2.3.2 Sodium channels

Voltage-gated sodium channels

Voltage-gated Na^+ channels (Na_V) play an essential role in the generation and propagation of action potentials, in the spike after depolarization and in the control of the firing frequency. Closed at the resting potential, Na_V channels rapidly open when the membrane is depolarized. The voltage-dependency of the activation well fits with Boltzmann's equation (2.6). $E_{1/2}$ is very variable and the generally relatively high z_g makes these channels very sensitive to weak membrane depolarization too. Additionally, Na_V channels have a very variable inactivation mechanism.

They can be further divided depending on the properties of the current they generate.

Transient currents: they activate very rapidly, with submillisecond time constants, causing positive feedback inward Na^+ currents that further activate other Na_V channels. This is the underlying mechanism of the fast depolarizing phase of the action potential. Moreover, these channels show a very fast inactivation mechanism. Recovering from inactivation instead requires several milliseconds, forcing the cell in a refractory period.

Persistent currents: those currents differ from transient ones because they are active at lower depolarization levels at which they don't inactivate. They are responsible of the modulation of the firing frequency and of the subthreshold behavior.

[4]Inactivation is not a directly reversible process, since channels must first be closed before they can be reopened again.

2.3.3 Calcium channels

Voltage-gated calcium channels

From a biophysical point of view these channels (Ca_V) can be divided in two classes: *low-voltage activated* (LVA) and *high-voltage activated* (HVA), which exhibit an higher $E_{1/2}$ compared to the first ones.

Low-threshold currents: channels responsible for these currents exhibit fast inactivation kinetics. Most of them is inactivated at resting potential, this means that the membrane must first be hyperpolarized in order to open the channels. They are responsible of the post-inhibitory rebound spikes.

High-threshold currents: these currents are activated at higher threshold and shows a slower inactivation kinetic compared to Ca_{LVA} currents. These channels are important because they activate Ca^{2+}-activated K^+ channels, and are responsible of the bursting firing pattern. Moreover Ca_{LVA}–induced Ca^{2+} fluxes trigger neurotransmitter release in the synapses.

2.3.4 Potassium channels

Voltage-gated potassium channels

These channels (K_V) activate when the membrane depolarizes, and their currents are responsible of the repolarizing phase of the action potential and of the modulation of the firing frequency.

Delayed-rectifier current: this current (DRK) activates at about -50 mV, it has a time constant of few milliseconds around 0 mV and doesn't inactivate. It is responsible of the depolarizing phase of the action potential.

Transient A-type currents: it differs from DRK current as it activates faster, and shows inactivation with time constant comparable to sodium fast current. Its role is the modulation of the firing frequency.

M-type currents: it shows slow activation kinetic after membrane depolarization and no inactivation. As for A-type currents, they play an important role for the regulation of neuronal excitability.

Calcium-activated potassium channels

K^+Ca^{2+}-activated channels (K_{Ca}) are regulated by the intracellular concentration of Ca^{2+}. These channels opens during action potentials after Ca^{2+} ions enter inside the cell through Ca_V channels. They play a role during repolarizing and after spike hyperpolarized phases of the action potential, as well as in the modulation of the firing frequency.

2.4 Action Potential

If a current pulse of proper amplitude and duration[5] is injected into an excitable cell in order to depolarize its membrane, we notice that when the membrane potential reaches a certain level (about $-40\,\text{mV}$) of depolarization (that is often defined *threshold*), it rapidly increases to a positive value of about $30\,\text{mV}$ and then falls down to a value a little bit more negative than the resting potential and then it goes back to the resting value. It can be observed that the amplitude (we could say even more generally the shape) of the recorded voltage trace is independent from the amplitude of the current stimulation, and it is always the same once the potential reaches the threshold. This stereotyped behavior is called *action potential*[6].

2.4.1 Shape of the action potential

The time course of an action potential can be divided in three phases (see Figure 2.5):

[5] The minimum value of the amplitude of the current step that provokes an action potential after an infinite period of time is called *rheobase*.

[6] Action potentials in neurons are also referred as *spikes*. The temporal sequence of action potentials generated by a neuron is called *spike train*. When a neuron produces an action potential it is said to be *firing*.

1. rising phase: in this first phase the membrane becomes more and more depolarized. Initially the depolarization is induced only by the injected current, but, once the potential reaches the threshold value, a great inward flux of Na^+ ions arises. This sudden and great current causes the rapid increase of the membrane potential observed in the voltage traces. The threshold[7] is defined as that value at which the velocity of the depolarization changes.

2. peak and falling phase: with a little delay with respect to the sodium current also a potassium currents appears. This balances the sodium one and thus the potential gets to its peak value. In the meanwhile, the sodium current is shut down, and then the potassium currents repolarizes the membrane back to the resting potential.

3. hyperpolarization phase: at the end of the second phase the potassium current is still persistent, causing the hyperpolarization of the membrane. This phase is also known as spike after hyperpolarization. When the potassium current is finally shut down the cell restores the resting potential through active mechanisms.

When the cell is in phases 1 and 2 it is said to be in its *absolute refractory period*, because the application of a second stimulus would not change the time course of the potential and could not elicit a further action potential until the end of phase 2.

During phase 3 the cell is said to be in its *relative refractory period*, because a second stimulus would unlikely elicit an other spike. Indeed the neuron is still hyperpolarized, and therefore far threshold.

2.4.2 Propagation of the action potential

Passive signal propagation

The axon can be approximated by a cylindrical conductor enveloped by an insulator layer (the myelin sheath) surrounded by an electrolytic solution. If the time

[7] It must be said that the value of the threshold can be variable, because it can depend on the past history of the membrane potential.

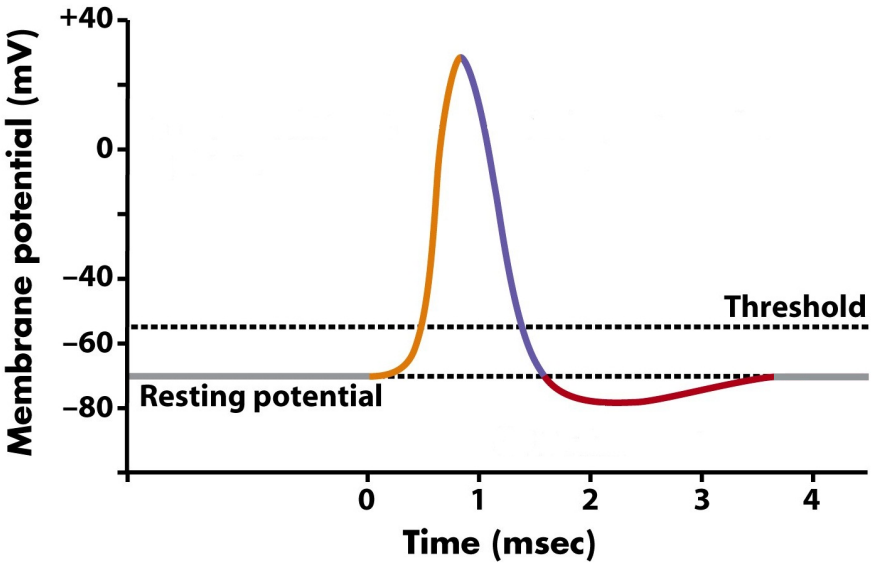


Figure 2.5: Different phases of the action potential: yellow) rising phase; blue) falling phase; purple) hyperpolarization phase.

dependence of the conductance[8] is neglected the axon can be compared to a linear cable. With this assumption, the propagation of the signal is said *electrotonic*.

Figure 2.6 represent the equivalent electric circuit of such a cable: here an horizontal slice of length dx is represented by one internal axial resistance $r_{\text{in}}\,dx$ coupled with the external environment by the parallel of a resistance of value r_{m}/dx and a capacitor of value $c_{\text{m}}\,dx$. It is assumed that these parameters are constant both in time and in space. Applying the formalism of electric circuits we obtain the following relation, which describes, in steady-state conditions and for a constant current injected in $x = 0$, the trend of the membrane potential according on the distance from the origin:

$$E_m(x) = E_0 \exp\left(-\frac{x}{\lambda}\right) \tag{2.9}$$

where E_0 is the value of the membrane potential at $x = 0$. The space constant λ control how rapidly the signal decays moving far from the origin. The space

[8] i.e. the active properties that provide the axon with excitability.

constant can be calculated as:

$$\lambda = \sqrt{\frac{r_{\mathrm{m}}}{r_{\mathrm{in}}}} \tag{2.10}$$

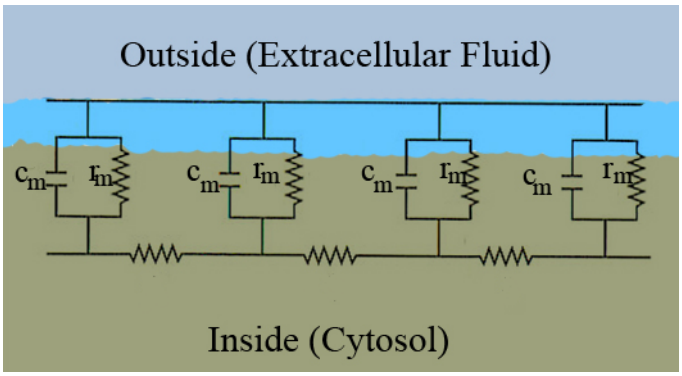


Figure 2.6: Electric representation of an insulated cable surrounded by an electrolytic solution, as it is the case for a passive unbranched dendritic segment.

For physiological values of c_{m}, r_{m} and r_{in}, λ is few millimeters in the best cases, thus passive propagation of cellular electrical signal is meaningful only for short distances. Propagation of the action potential for longer distances requires a continuous regeneration of the signal along the fiber to permit it to reach distal targets.

Local circuits

The propagation of the action potential occurs through active processes, based on voltage-dependent sodium and potassium conductances, together with the passive electrotonic propagation process. Hodgkin and Huxley merged the two descriptions, obtaining a solution consisting in a excitation wave traveling at constant speed along the cable. The point where the action potential is generated is considered as a source of current flowing across the membrane and along the axon.

Figure 2.7 describes this concept. The action potential is traveling from left to right and is in the orange zone. In this region an inward sodium current is present and it propagates on both sides along the axon according to electrotonic

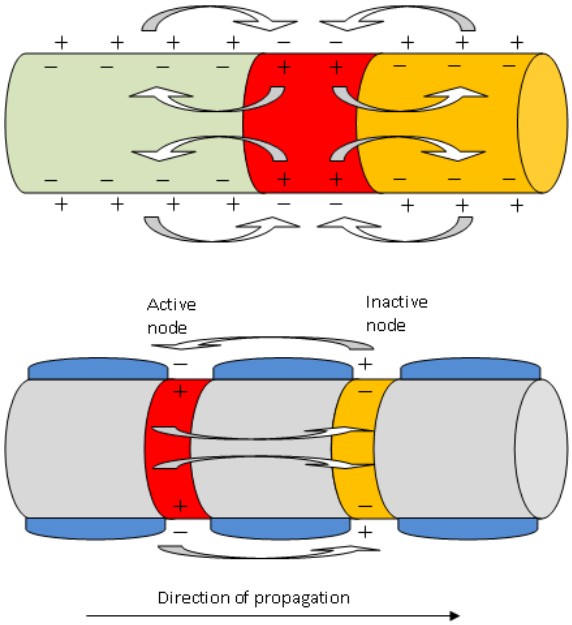


Figure 2.7: Propagation of the action potential. Top: propagation in unmyelinated fiber. Bottom: propagation in myelinated axon.

propagation, causing the depolarization of adjacent patches of membrane. If the stimulation is strong enough, the patch on the right side will generate an action potential. This is not true for the patch on the left side: being in its refractory period, even if the stimulus depolarizes the membrane above the threshold, the generation of the action potential will fail. This mechanism guarantees that the propagation is unidirectional.

The speed of the excitation wave is strongly related to the diameter of the axon (and hence its space constant): the bigger the diameter, the faster the propagation (Rushton [1951]).

Saltatory conduction

Many axons present periodically regions enveloped inside an insulating myelin sheath and free regions. The former are called *internodes* and the latter *nodes of Ranvier*. Myelinated regions are highly insulated from the outside environment, and poor of ion channels, which are concentrated at Ranvier's nodes. The high insulation of myelinated regions causes a strong increment of the space constant

there, which can reach values around 3 mm. This means that the electric pulse can "jump" for a distance greater than the internodal one and it is still strong enough to regenerate the action potential. Conduction velocity in myelinated fibers is then much higher[9] than unmyelinated ones' having the same diameter.

2.5 Synapses

Synapses are functional connection between neurons. Most of them connect axons with dendrites, but also axo-axonic, axo-somatic and dendo-dendritic synapses are known. They allow the transmission of information from presynaptic to postsynaptic neuron. They can be distinguished in two different types: electrical synapses, also called gap junction, and chemical synapses. In this section we will talk about the latter.

2.5.1 Structure

Chemical synapses are made of three distinct elements: the synaptic terminal, also called synaptic bouton, the synaptic space (synaptic cleft), a thin space of about 20 nm that separates the membranes of the pre- and postsynaptic neurons, and the postsynaptic membrane.

The release of neurotransmitter by the presynaptic neuron induce a current on the postsynaptic side.

Presynaptic side

The synaptic bouton is a specialized area within the axon of the presynaptic neuron (the source of the signal) where neurotransmitter molecules are enclosed in small membrane vesicles. They are usually situated in the terminal part of the axon, after the last node of Ranvier. In the terminal part of the axonal membrane, sodium channels are replaced by high-threshold voltage-gated calcium channels so that the depolarization induced by an action potential produces an influx of Ca^{2+} in the presynaptic terminal. Calcium ions interact with calcium-binding proteins that, once activated, induce the fusion of the vesicles with the bouton's membrane

[9] it can exceed $100\,\mathrm{m\,s^{-1}}$

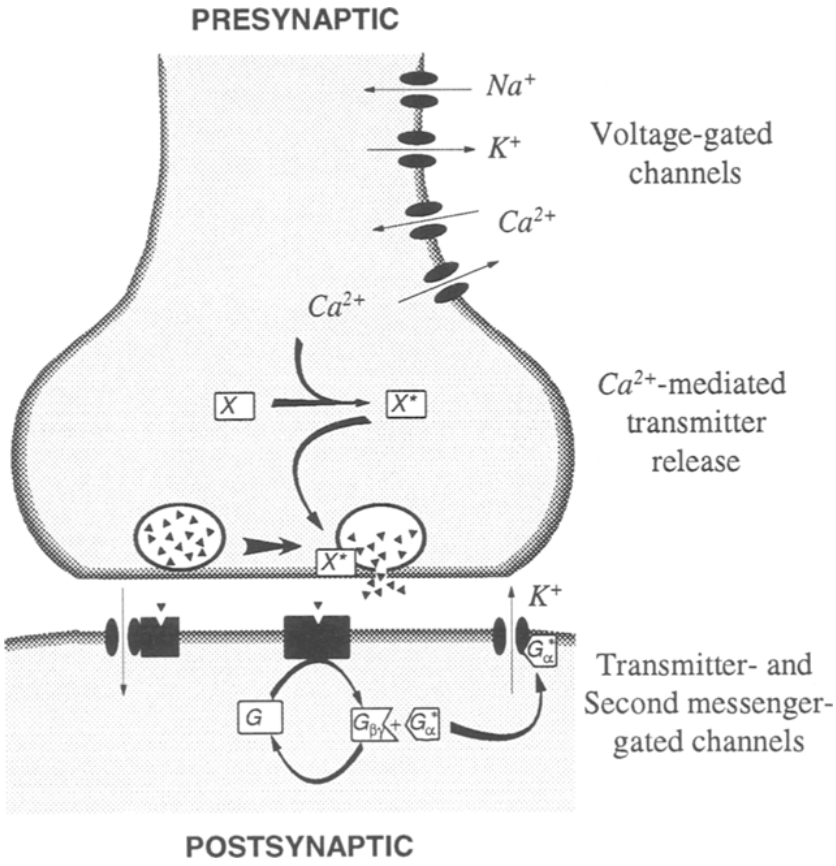


Figure 2.8: Representation of neurotransmitter release process. When voltage-gated channels depolarize the membrane Ca^{2+} enter the cytosol via HVA calcium channel. Ca^{2+} ions bind to a calcium-binding protein (X) and produces an active form (X^*); X^* promotes the exocytosis of synaptic vesicles an release of transmitter molecules (triangles) into the synaptic cleft. The transmitter can induce a postsynaptic current by binding to ligand-gated ion channels (left) or to a secondmessenger-linked postsynaptic receptor (right) which induce a cascade of reactions that end with the opening of K^+ ligand-gated ion channels. From Destexhe *et al.* [1994].

and the following release of neurotransmitter in the synaptic cleft. There, transmitter molecules can bind to specific receptors on the postsynaptic membrane or be recovered by the presinaptic terminal, a process called transmitter reuptake.

Postsynaptic side

Neurotransmitters can bind to two distinct categories of receptors in the postsynaptic membrane:

- *ionotropic receptors*: they are ligand-gated ion channels (e.g. AMPA-kainate or $GABA_A$receptors). They usually produce fast currents, with rising times of some hundreds of microseconds and decaying times of few milliseconds.

- *metabotropic receptors*: for these receptors, the receptor complex and the ion channel are not part of the same protein complex. When the receptor binds to transmitter molecules, it produce a cascade of intracellular reactions that ends up with the opening of the ionic channels. Currents produced after the activation of metabotropic receptors are characterized by slower dynamics (with time constants up to one second), if compared with ionotropic-receptors induced ones, because they are the result of a metabolic reactions. The most important receptor of this category in the central nervous system is the $GABA_B$receptor, found in inhibitory synapses.

As for every ion channel, also for synaptic channels it is possible to find a value of the membrane potential, the reversal potential, at which, even if the channel is open, the net current is zero. If at the resting potential the current across postsynaptic channels tends to depolarize the membrane, it is said to be excitatory (*EPSC: Excitatory Post Synaptic Current*), since it moves membrane potential towards the threshold, otherwise they are said to be inhibitory (*IPSC: Inhibitory Post Synaptic Current*). It is worth noting that a current that usually is inhibitory may become excitatory if certain conditions are met. For example $GABA_A$receptor selectively permeable to Cl^-, thus its reversal potential is the same of the ion, that is around $-65\,\mathrm{mV}$. It is clear that if the membrane potential is at values more negative than the resting one, $GABA_A$current becomes excitatory.

2.5.2 Short term synaptic plasticity

During repetitive stimulation, EPSCs can show modulations of their amplitude, that are together defined short term synaptic plasticity.

Synaptic facilitation consists in a gradual increase of the amplitude of the synaptic response caused by Ca^{2+} accumulation in the presynaptic terminal due to the repetitive stimulation. This happens because the complete extrusion of Ca^{2+} after the first stimulus is complete only after some tens of milliseconds. The release mechanism, being Ca^{2+}-dependent is therefore facilitated.

Synaptic depression represents a progressive decrease of the amplitude of the synaptic response that eventually will tend to a reduced steady state value. Depression is primarily caused by the limited velocity of transmitter reuptake. When a second pulse arrives (typically within a period of some tens of milliseconds) some of the previously used vesicles could be not yet available for the release process, resulting in a lower number of releasing sites (and hence of the total number of released transmitter molecules).

Together with these two mechanisms there is a third one taking place in the postsynaptic side. Many synaptic receptor, in addition to a closed and open state, have a third conformation for which the associated ionic channel is open but not conducting. When a receptor enters this state it is said to be *desensitized*. The desensitization process for synaptic receptors is similar to inactivation for voltage-gated ion channels.

Usually facilitation and depression coexist. At the end of a spike train, however, the functional state of a synapse returns to basal conditions. This fact makes short-term synaptic plasticity a completely reversible phenomenon.

Chapter 3

GPU Architecture

Figure 3.1 shows a simplified view of the CUDA GPU architecture from NVIDIA. It contains an array of Streaming Multiprocessors (SMs). Each SM consists of eight floating-point Scalar Processors (SPs), a multi-threaded instruction unit (Register), a 16 up to 48KB user-managed shared memory and 16KB of cache memory (8KB constant cache and 8 KB texture cache)[1]. Each SM has a hardware thread scheduler that selects a group of threads (called *warp*) for execution. If any one of the threads in the group issues a costly external memory operation, then the thread scheduler automatically switches to a new thread group. By swapping thread groups, the thread scheduler can effectively hide costly memory latency. At any instant of time, the hardware allows a very high number of threads to be active simultaneously.

The following are general best practices to keep in mind while designing a CUDA program:

1. Parallelism: To effectively use the GPU resources, the application needs to be mapped in a data-parallel fashion; each thread should operate on different data. Also, a large number of threads (in the thousands) need to be launched by the application to effectively hide the stalling effects caused while accessing GPU memory.

2. Memory bandwidth: To achieve peak memory bandwidth, each processor should have uniform memory access (e.g., thread_0 accesses address_0, thread_1

[1]From *CUDA Programming Guide*, v4.0

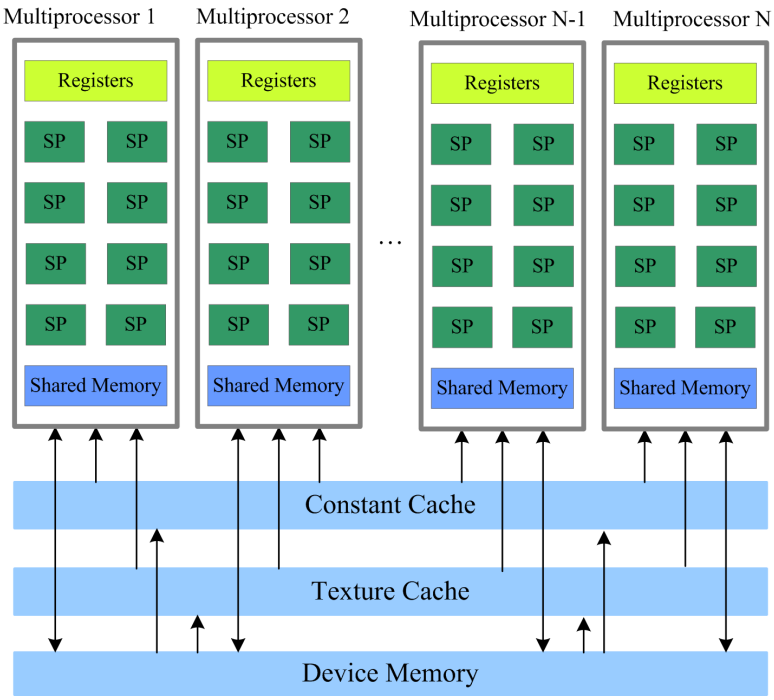


Figure 3.1: Simplified architectural view of a CUDA enabled graphic processor.

accesses address$_0$+4, thread$_2$ accesses address$_0$+8 etc.). If memory accesses are uniform, it is possible to group many memory accesses into a single large memory access (termed coalescing operation) achieving high memory bandwidth. In CUDA 1.2 compatible GPUs (and future families) memory coalescing is performed if all SPs within an SM accesses the same memory segment in any ordering.

3. Minimize thread divergence: By design, the current CUDA GPUs selects a warp of 32 threads, and executes them using a single instruction register. Maximum performance can be achieved if all the threads within the warp execute the same instruction. If different threads within the warp follow different branches, which are termed divergent warps, then this will lead to sub-optimal performance.

It is important to note that the above factors are interrelated, and all four factors need to be optimized for effective execution on GPUs.

Chapter 4

Neurons

4.1 Introduction

It is widely accepted that neurons encode information in the temporal patterns of their action potentials[1], hence a careful reproduction of their "computational properties" is an important issue. In this chapter we are going to analyze several models for this purpose. Finally, we will discuss briefly multi compartmental models.

4.1.1 Selection criteria

We are going to evaluate various models according to the following criteria:

1. Quality of predictions.
2. Computational cost: obviously, if two models are available to describe the same phenomenon, we will prefer the one that is computationally lighter.
3. Number of reproducible neuro-computational properties: models describing different phenomena simply assigning different parameters to the same equations are preferred to models requiring different equations to account for the same phenomena.

[1] Refer to Appendix C: Firing patterns for information about firing patterns.

4. Possibility of implementation (on parallel code[2]): a model needing a fixed amount data to reproduce different behaviors is preferred to models requiring variable amounts of data to reproduce the same phenomena. This is because handling objects of different sizes in parallel code is a difficult, at all trivial, task.

The assessment of the various models must consider all the previous criteria.

4.2 Phenomenological models

As their name says, these models describe the behavior of a neuron with few or no assumptions about underlying physiological mechanism. They are all based on the assumption that spikes are stereotyped events, hence what is important is the correct prediction of their occurring times, rather than their actual shape. Moving from this hypothesis, many models have been proposed to achieve this task. They can be classified into two main families, i.e. *integrate-and-fire* (IF) and *spike-response models* (SRM), according to the approach used to describe the time course of the membrane potential: a set of differential equations for the first, whereas the latter are formulated using filters.

4.2.1 Leak integrate-and-fire neurons

The differential equation governing the simplest integrate-and-fire model is derived from a basic circuit for the cell membrane, consisting of a capacitor C in parallel with a resistor R driven by a driving force E_r. The whole circuit is driven by a time-dependent current $I(t)$[3]. Once the circuit is solved, the following dynamic equation is obtained for the voltage:

$$C\dot{v} = -\frac{v - E_r}{R} + I(t) \tag{4.1}$$

In leak IF neurons the shape of the action potential is not described explicitly, but rather a spike is triggered at time $\bar{t}$ whenever the potential v reaches a certain

[2]See Chapter 3 for an overview of CUDA programming best practices

[3]The nature of $I(t)$ is not specified: it can be an artificially injected current as well as the input current from the synapses.

threshold value $\vartheta > E_r$. Immediately after, the potential is reset to the value E_r. Formally, this is written as:

$$\text{if } \bar{t} \ : \ v(\bar{t}) = \vartheta, \quad \lim_{t \to \bar{t}^+} v(t) = E_r \tag{4.2}$$

and for $t > \bar{t}$ the dynamics are again given by (4.1). A more general version of the leak IF neuron includes an absolute refractory period after the generation of a spike, during which the dynamics is interrupted. The integration then restarts at time $\bar{t} + \Delta_{abs}$ with the new initial condition $v(\bar{t} + \Delta_{abs}) = E_r$. In this way the maximum firing frequency is limited so the neuron cannot fire at an arbitrary high frequency. The current-frequency relation can be found solving the differential equation (4.1) for a constant input current I_0. Hypothesizing that a spike occurred at time $\bar{t}$, we obtain:

$$v(t) = E_r + RI_0 \left(1 - e^{-(t-\bar{t}-\Delta_{abs})/\tau}\right), \quad t > \bar{t} + \Delta_{abs} \tag{4.3}$$

where $\tau = RC$ is the membrane time constant. For $t \to +\infty$, v approaches the steady-state value $E_r + RI_0$. Hence, if $E_r + RI_0 < \vartheta$ no more spikes are triggered, otherwise a new spike occurs and, solving (4.3) for the elapsed time $\Delta T = t - \bar{t}$ required to reach the threshold and calculating its reciprocal, one obtains the following current-frequency relation:

$$f = \left(\Delta_{abs} + \tau \ln \frac{RI_0}{RI_0 - \vartheta - E_r}\right)^{-1} \tag{4.4}$$

The IF model clearly represent a class 1 excitable neuron[4].

Although it is the simplest and computationally cheapest model of a spiking neuron, its application is very limited because it lacks important computational properties, like adaptation or bursting.

Adaptation

Adaptation can be included adding one more differential equation to the model. The new variable mimics a sort of slow high-threshold potassium conductance

[4]Refer to Appendix C for details.

that force the potential to hyperpolarized values:

$$\begin{cases} C\dot{v} = -g_L(v - E_L) - w(v - E_K) + I(t) \\ \dot{w} = -w/\tau_w \end{cases} \tag{4.5}$$

where g_L represents a leak conductance and E_L is the reversal potential of the current mediated by g_L. This current is always present and accounting for the passive properties of the membrane. After each spike, the potential v is reset to a new value v_r as in (4.2), whereas w is updated according to

$$\lim_{t \to \bar{t}^+} w(t) = w(\bar{t}) + d \tag{4.6}$$

This immediate increase of w can be interpreted as the opening of calcium-dependent potassium channels caused by the influx of calcium during an action potential due to HVA calcium channels. A negative feedback variable like w makes the explicit simulation of an absolutely refractory period unnecessary and introduces a relative refractory period "for free".

4.2.2 Nonlinear models

The integrate-and-fire-with-adaptation model represents the vast majority of neurons in the central nervous system, i.e. excitatory pyramidal cells. Indeed these neurons are about 80% of the total number of cortical neuron, and largely exhibit class 1 excitability with adaptation. However, a 20% of neurons are not represented by this model, and even within pyramidal cells other firing patterns are found. Moreover, spikes are not explicitly simulated, but rather added "manually" to voltage traces when a threshold crossing is detected. Hence the need of finding mathematical models which can describe better neuronal dynamics.

The models we are going to study in this section[5] can be written in the generic

[5] Many other models have been proposed, among all we mention the one described by Fitzhugh (Fitzhugh [1961]) and the one proposed by Hindmarsh and Rose (Hindmarsh & Rose [1984]). The main drawback of these solutions is that each of them can reproduce only a limited subset of the firing and computational properties found in neurons (Izhikevich [2004]), and this fact implies that the simulation of different types of neurons requires different models.

abstract form:

$$\dot{V} = f(V) - w + I \tag{4.7}$$
$$\dot{w} = a(bV - w) \tag{4.8}$$

where V is a potential-like variable, while w is a recovery variable needed to take into account firing frequency adaptation and other properties. w somehow represents all the ionic currents that are not directly involved in the generation of the action potential.

For this types of models spiking does not occur depending on the polarization threshold, but rather if the potential tends to escape towards infinity in a finite time. In practice, this is true if V is greater than a given value V_p, that is the theoretical peak value of the voltage during an action potential. If a spike is detected, the model is then reset in a fashion similar to equations (4.2) and (4.6):

$$\begin{cases} V \leftarrow c \\ w \leftarrow w + d \end{cases} \tag{4.9}$$

Adaptative quadratic integrate-and-fire

Izhikevich proposed a model where the $\dot{v} - v$ relations is a second order polynomial (Izhikevich [2003]):

$$\dot{v} = 0.04v^2 + 5v + 140 - u + I \tag{4.10}$$
$$\dot{u} = a(bv - u) \tag{4.11}$$

This model is purely phenomenological, as a matter of facts v and u are dimensionless variables, as well as a, b, c and d are dimensionless parameters, and the same is true for the input current I. If $v = 30$, the model is reset according to equations (4.9)[6].

Although purely phenomenological, the model can account for a large variety of neuronal firing patterns (Izhikevich [2004]). It has been fully characterized

[6]The choice of the cutoff value V_p is critical, as the recovery variable tends to diverge when the potential escapes to infinity.

by its author, who provides the values of the parameters for different classes of cortical neurons/firing patterns (see Figure 4.1).

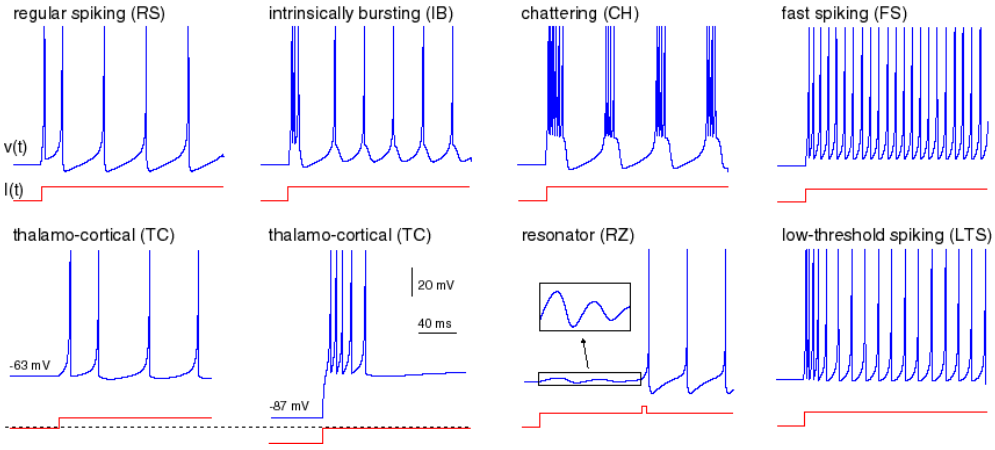


Figure 4.1: Examples of firing patterns and neuron types that can be reproduced by the Izhikevich's simple model. Each inset shows a voltage response of the model neuron to a step of dc-current $I = 10$ (bottom). Time resolution is 0.1 ms.

Adaptative exponential integrate-and-fire (AdEx)

If $\dot{v}$ depends exponentially on v, we obtain the so called *exponential integrate-and-fire* neuron, proposed Brette and Gerstner (Brette & Gerstner [2005]):

$$C\dot{V} = -g_L(V - E_L) + g_L\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + I \tag{4.12}$$

$$\tau_w\dot{w} = a(V - E_L) - w \tag{4.13}$$

where C is the membrane capacitance, g_L the leak conductance, E_L the leak reversal potential, V_T the threshold, Δ_T the slope factor, a the adaptation coupling parameter and τ_w is the adaptation time constant. A full characterization of the model and its parameters can be found in Naud *et al.* [2008]. The model is reset when $v = 20\,\text{mV}$ (Brette & Gerstner [2005]) or $v = 0\,\text{mV}$ (Naud *et al.* [2008]), and the potential is reset respectively to E_L or to an additional parameter V_r.

Unlike the previous model, all parameters of the Adaptive Exponential Integrate-and-Fire model have a biological interpretation:

- In the absence of adaptation, V_T is the maximum voltage that can be reached under constant current injection without generating a spike (rheobase current). In the presence of adaptation the voltage corresponding to the rheobase current is shifted.

- The slope factor Δ_T quantifies the sharpness of spikes. It can be related to the sharpness of the sodium activation curve, when one neglects the activation time constant. In the limit of zero slope factor, the model becomes an integrate-and-fire model with a fixed threshold V_T.

- Spike triggered adaptation (the parameter b) summarizes the effect of calcium dependent potassium channels under the assumption that calcium influx occurs mainly during an action potential. Note that the coupling of voltage and adaptation via the parameter a also contributes to spike-triggered adaptation because of the sharp rise of the voltage during the upswing of an action potential.

- The subthreshold parameters in equation (4.12) can be extracted from experiments by standard linear identification methods, whereas parameters V_T and Δ_T can be extracted from experiments using the technique of dynamic $I - V$ curves (Badel *et al.* [2008]).

4.3 Conductance-based models

Conductance-based models are based on an equivalent electric circuit of the membrane. The double lipid bilayer is represented by mean of a capacitor in parallel with a leak conductance, which is an approximation of the passive properties of the cell. Several current sources are connected in parallel with the base RC circuit, in order to take into account ion-channel[7] and synaptic-mediated as well as artificially injected currents. Applying Kirchhoff's current law to the circuit and considering the relation between voltage and current through a capacitor, we get

[7]For a summary of the types of ion channels that can be expressed by a neuron refer to sections 2.3.2, 2.3.3, 2.3.4. Each type of ion channel mediates a current that takes the same name.

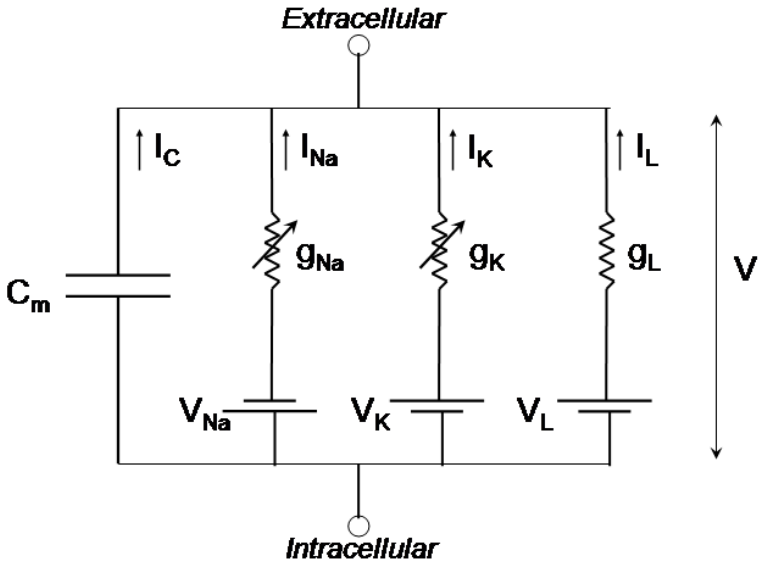


Figure 4.2: Equivalent electrical circuit of a membrane, in which only sodium and potassium ions are considered. The capacitor C represents the capacity of the membrane, while g_L is its passive conductance and V_L the corresponding driving force. They cause a current I_L that tends to keep the membrane at its resting potential. g_{Na} and g_K are instead "active" conductances, and represent the voltage-dependent ion channels embedded in the membrane. V_{Na} and V_K are the reversal potential for sodium and potassium respectively as given by the Nernst equation (2.1).

the following first order differential equation for the time evolution of the circuit:

$$C\dot{V} = -\bar{g}_{leak}(V - E_{leak}) - I_{ion} - I_{syn} - I_{ext} \tag{4.14}$$

where V is the membrane potential, C is the total membrane capacity, $\bar{g}_{leak}$ and E_{leak} are the conductance and the reversal potential for the leak current, and I_{ext}, I_{syn} and I_{ion} are the external-injected, synaptic and ion-channels-mediated currents respectively. The convention followed for their signs is that an inside-out flowing current is considered to be positive.

It is worth noting that even synaptic currents are carried by ions flowing through ion channels,therefore, in principle, we could have identified all the currents that are not artificially injected with the same label. Nevertheless, in order to separate inputs and free evolution of the cell system, we distinguish from ionic currents that depend only on the internal state of the cell (I_{ion})[8] and currents that are driven by external factors (I_{syn}). In this chapter we are focusing of the former, but, although the underlying mechanisms are different, a similar description is possible also for the latter.

I-V relation for ionic currents

I_{ion} results from the sum of the currents carried by the various ions. Each ionic current in turn arises from the contribution of different ion channels permeable to a specific ion. The generic ionic current through a generic population of channels can modeled by two different equations:

- *Electrochemical diffusion*: Ion are present in different concentration inside and outside the cell so, in addition to the electric field, they are subject to chemical gradients. For this reason, ions tend to flow better in one direction rather than in the other, across the membrane. This phenomenon is called rectification, and it is all the more important the more the concentrations are different. If this is the case, the ionic flux across the membrane is better can be modeled by the phenomenological law called *Goldman-Hodgkin-Katz*

[8] i.e. voltage and ionic concentrations. For the most important ionic currents found in neuron, please refer to Appendix B: Ionic currents.

(GHK) flux equation:

$$\Phi_S = p_S(t,V) z_S^2 \frac{F^2}{RT} V \frac{[S]_i - [S]_o \exp(-z_S VF/RT)}{1 - \exp(-z_S VF/RT)} \tag{4.15}$$

where Φ is the flux of ion S across a unit area of membrane (therefore to obtain the value of the current, Φ must be multiply by the total membrane area), p_S is the permeability of the membrane for ion S given by the ionic channel of interest, z_S^2 is the valence of S, F and R are the usual physical constants and T the absolute temperature, $[S]_i$ and $[S]_o$ the concentration of ion S in the intra and extracellular media and V the membrane potential. The GHK flux equation is mostly used by electrophysiologists not only when the ratio between $[S]_i$ and $[S]_o$ is large, but also when one or both change considerably during an action potential.

- *Linear ohmic relation*: Most of the times, however ion concentration inside and outside the cell are comparable, and ionic current can be described by a simpler a linear ohmic relation:

$$I_x = g_i(t,V)(V - E_i) \tag{4.16}$$

where E_i is the value of the reversal potential of the channel[9], G_i, which generally is time and voltage dependent, is the sum of the conductances of all the channels responsible of the current i. This linear relation is suitable for most of the currents found in neurons.

Both models have advantages and drawbacks: GHK equation is more complete, but it requires additional physiologic information (ion concentrations). Moreover, it is hard to find values for channels permeability in literature, and extra work must be done to extract them from experimental data; linear ohmic relation is a much more simple model (also from the computational point of view), but it based on strong approximations. If it can be cleanly(used for sodium or

[9]That is the reversal potential of a specific ion as given by the Nernst equation (2.1), if the channel is selectively permeable to a single ion species; otherwise E_i is given by a weighed sum of the various Nernst potential accordingly to the reletive permeability the channel shows for the various ions.

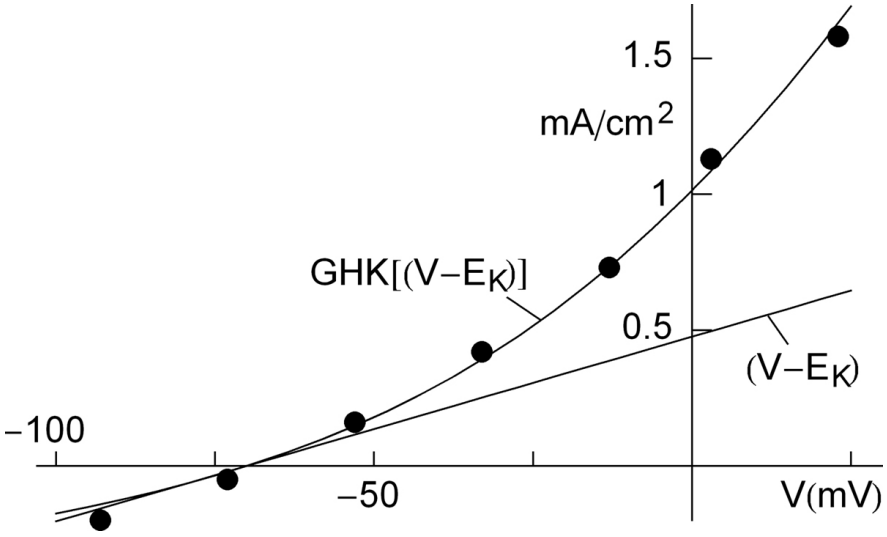


Figure 4.3: Example of current–voltage relation for K^+ ions predicted by GHK (marked line) and using a linear ohmic relation (unmarked line) . It is worth noting how GHK relation approaches two different asymptotes when V diverges from zero.

potassium currents because these ions have comparable intra and extracellular concentration), the same cannot be said in case of calcium. However, it is worth noting that neurons spend most of their time near their resting potential, whose value (between -60 to $-70\,\mathrm{mV}$) is far from calcium reversal potential (about $120\,\mathrm{mV}$), therefore the voltage-current relation can be approximated as linear also in the case of calcium ions[10]. For these reasons, when, in the future, we implement conductance-based models, we will use the ohmic linear relation for each ionic current.

In both (4.16) and (4.15) the value of the conductance and of the membrane permeability are functions of time and membrane potential, but they can also depend on other factors like the intracellular concentration of calcium or other molecules. In the following subsection we will introduce two different theoretical framework to describe their dynamics.

[10]However, in principle, we should still care about intracellular calcium concentration and continuously update its reversal potential using the Nersnt equation (2.1).

Further extension of conductance-based models

Also calcium dynamics, a subject we are not treating in this thesis, can be simulated with conductance-based models. For this reason, it is possible to include also non voltage-dependent currents[11].

4.3.1 Hodgkin-Huxley formalism

The first attempt to describe voltage dependency of ionic conductances was realized by Hodgkin and Huxley on their work about the electrical properties of the squid giant axon (Hodgkin & Huxley [1952]). They focused on fast sodium and potassium currents responsible of the generation of the action potential, but the formalism they introduced can be used to characterize other types to ionic currents found in neurons.

They hypothesized that several independent activation particles must simultaneously occupy a certain position in the membrane, depending on its potential, to allow the flow of ions. This fact can be illustrated by a two-states state diagram:

$$C \quad \underset{\beta(V)}{\overset{\alpha(V)}{\rightleftharpoons}} \quad O \tag{4.17}$$

where C and O represent the closed and open positions respectively, and $\alpha(V)$ and $\beta(V)$ are the voltage-dependent transition probabilities. In order to interpret inactivation phenomenon observed in experimental data, they introduced one or more independent blocking particles, ruled by a similar kinetic scheme, may prevent the flux of current if they occupy a certain position.

Usually, the fraction of activating particles in the open position are represented by the gating variable m, while the fraction of non-blocking inactivation particles by h. According to scheme (4.17), the temporal evolution of the generic gating variable x is governed by a first order differential equation:

$$\dot{x} = \alpha_x(V)(1 - x) - \beta_x(V)x \tag{4.18}$$

[11] e.g. I_C and I_{AHP} currents, which play an important role in the adaptation process of the firing frequency at different time scales. See Yamada *et al.* [1989].

here α and β assume the meaning of transition rates. This equation can also be rearranged into another form more suitable to fit voltage-clamp experimental data:

$$\dot{x} = \frac{1}{\tau_x(V)} \left(\bar{x}(V) - x\right) \tag{4.19}$$

where the steady-state value $\bar{x}$ and the time constant τ_x are given by

$$\bar{x}(V) = \alpha_x(V) / \left[\alpha_x(V) + \beta_x(V)\right] \tag{4.20}$$

$$\tau_x(V) = 1 / \left[\alpha_x(V) + \beta_x(V)\right] \tag{4.21}$$

These functions are usually sigmoid- and bell-shaped respectively.

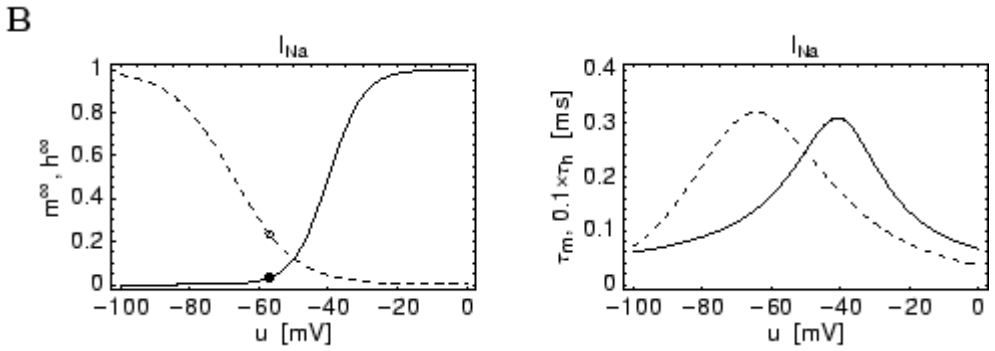


Figure 4.4: Steady-state values and time constant of m and h, the gating variables introduced by Hodgkin and Huxley to describe the voltage dependency on membrane potential of the sodium conductance.

The total conductance for a any given current j is assumed to be proportional to the number of activated gating particles that are not blocked. Mathematically:

$$G_j = \bar{g}_j m^{p_j} h^{q_j} \tag{4.22}$$

where $\bar{g}$ is the maximal conductance, i.e. the product of the conductance of a single j-type channel times the number of channels of that type found in the membrane, p_j is the number activation particles that must be open simultaneously to allow the flux of ion, while q_j is the number of inactivation particles that must simultaneously be removed to remove the block. If a current does not inactivate over the time, $q_j = 0$.

4.3.2 Markov models

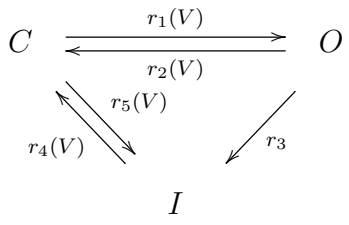
According to Hodgkin-Huxley formalism a channel is considered to be composed of several independent gating particles, all assumed to be equal to one another. Voltage-clamp experiments on single channel recordings instead evidenced that the situation is different: inactivation of sodium channels have been proven to be voltage independent and strongly coupled with the active state (Aldrich *et al.* [1983] and Benazilla [1985]). In principle, this could be true also for other ion channel. These experimental evidences suggest to consider channel proteins on the whole. Markov models[12] are suitable to describe channel kinetics. The base assumption is that conformational changes of the protein, leading to different conductive states, are influenced only by its present configuration (and not from the past hystory).

The simplest possible kinetic scheme to describe a channel that does not inactivate is given by:

$$C \underset{\beta(V)}{\overset{\alpha(V)}{\rightleftharpoons}} O \tag{4.23}$$

This scheme is equal to the one used for gating particles, with the difference that here C and O are the closed and open state of the whole channel and not of single subunits.

To include inactivation one more state is required (I), thus obtaining the following minimal model:



(4.24)

Opening from the inactive state is not possible, and inactivation from the open state is ruled according to a constant rate r_3. Other rates are voltage dependent. Although other schemes are possible adding or deleting some of the connections

[12] See Kinetic schemes for an introduction on Markov models.

between the states, this one is the closest to physiological facts.

4.3.3 Choice of the function for the voltage dependency of the rate constants

In Hodgkin and Huxley's work, the voltage dependence of the rate constants were fit to voltage-clamp measurements using empirical functions of the membrane potential. This approach was used in many other works to characterized also other type of currents (Yamada *et al.* [1989], Reuveni *et al.* [1993], Traub & Miles [1991]). We report the most commonly used for the sake of clarity:

$$f(V) = \frac{A(V - V_H)}{1 + \exp[-(V - V_H)/k]} \quad ,linoid \tag{4.25}$$

$$f(V) = \frac{A}{1 + \exp[-(V - V_H)/k]} \quad ,sigmoid \tag{4.26}$$

$$f(V) = A \exp[-(V - V_H)/k] \quad ,exponential \tag{4.27}$$

This approach is a good solution because it leads to good fits of experimental data, but it is not compatible with the requirement for a model to be mathematically cheap (criteria 3 and 4).

Alternatively, the exact functional form of the voltage-dependence of the rate constants could be deduced from thermodynamics (Destexhe & Huguenard [2000]). This approach has two advantages: first of all, it is physically meaningul, and, second, but maybe even more important, it allows to describe all the rate constants using the same functional form, as we will see next.

Thermodynamic models

Generally, it is assumed that the transition between two states of the channel corresponds to a conformational change of the ion channel protein. Given a transition between two states S_1 and S_2, with a voltage-dependent rate constant $r(V)$:

$$S_1 \xrightarrow{r(V)} S_2 \tag{4.28}$$

the rate of the transition depends exponentially on the free energy barrier between the two states (Eyring [1935]):

$$r(V) = r_0 \exp\left[-\Delta G(V)/RT\right] \tag{4.29}$$

where r_0 is a constant, R and T are the universal gas constant and the absolute temperature, while $\Delta G(V)$ is the free energy barrier, which can be written

$$\Delta G(V) = G^*(V) - G_0(V) \tag{4.30}$$

where $G^*(V)$ is the energy of an intermediate state and $G_0(V)$ is the free energy of the initial one.

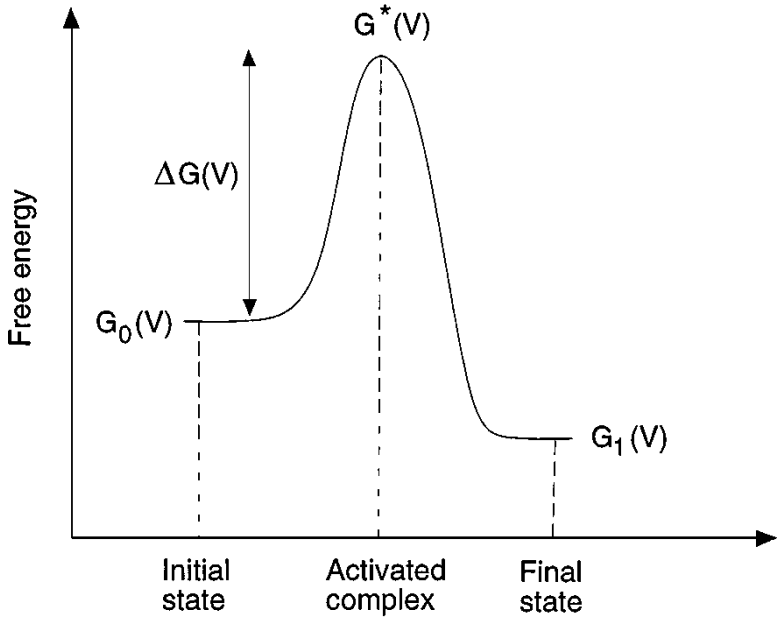


Figure 4.5: Free energy function. From Destexhe & Huguenard [2000].

The relative values of the free energy of the initial and final states (G_1 and G_2) determine the equilibrium distribution between these states, while the kinetic of the transition depends on the size of the barriers. How the transition rates between these conformational states depend on membrane potential is given by the voltage-dependence of the free energy barrier, which is in general very difficult to evaluate. In the following some assumptions are made in order to manage the

problem.

Linear thermodynamic model:

the simplest relation linking the free energy of a state i with the membrane potential is linear:

$$G_i(V) = A_i + B_i V \tag{4.31}$$

where A_i corresponds to the free energy that is independent of the electrical field, and the linear term $B_i V$ to the effect of the electrical field on isolated charges and rigid dipoles. Thus, we can express the energy barrier with respect to any given state as:

$$\Delta G(V) = (A^* - A) + (B^* - B)V \tag{4.32}$$

and then, letting $a = A^* - A$ and $b = B^* - B$, we obtain the following expression for the rate constant α and β of the generic transition between states S_1 and S_2:

$$\alpha(V) = \alpha_0 e^{-(a_1 + b_1 V)/RT} \tag{4.33}$$

$$\beta(V) = \beta_0 e^{-(a_2 + b_2 V)/RT} \tag{4.34}$$

With the assumption that the conformational change consists in the movement of a freely moving gating particle of charge q (Hodgkin & Huxley [1952]), the forward and backward time constant can be rewritten (Borg-Graham [1991]):

$$\alpha(V) = A e^{-\gamma q F(V - V_H)/RT} \tag{4.35}$$

$$\beta(V) = A e^{(1-\gamma) q F(V - V_H)/RT} \tag{4.36}$$

where $\gamma \in (0 \ldots 1)$ is the relative position of the energy barrier in the membrane, V_H is the half-activation voltage and A is a constant; R, F and T are the usual physical quantities.

The drawback of models in which the rate functions are simple exponentials of voltage is that these functions can reach unrealistically high values, which leads to very small time constants and possibly aberrant behavior. Different solutions are possible to avoid this problem:

1. the number of states of the kinetic diagram is increased, and some tran-

sitions are made voltage independent. E.g. scheme (4.17) should become:

$$C_0 \underset{\beta(V)}{\overset{\alpha(V)}{\rightleftarrows}} C_2 \underset{k_2}{\overset{k_1}{\rightleftarrows}} O \tag{4.37}$$

so, when the time constant of the first transition assumes very high values, the evolution of the system towards the open state is rate limited by k_1 and k_2. This solution has several drawbacks: different transition are modeled in different ways, and the model becomes more expensive in terms of memory, which is in contrast with our purpose of minimizing memory occupancy.

2. impose a minimum value to the time constants: smart solution, but we found that it complicates the implementation of the model.

3. force an artificial saturation of the rate functions (Hartshorne *et al.* [1986]):

$$\alpha(V) = \frac{a_1}{1 - \exp[-(V - V_H)/b]} \tag{4.38}$$

$$\beta(V) = \frac{a_2}{1 - \exp[(V - V_H)/b]} \tag{4.39}$$

where a_1 and a_2 are the limit values, b is the voltage sensitivity and V_H the half-activation potential. These parameters are the same for both the rate constants. This solution is as cheap as the linear model, in terms of required parameters, needing only four parameters for each state transition.

Non-linear thermodinamic model

An exhaustive characterization of the free energy function of a conformational state of a channel is impossible, since many different factors contribute to its final value (e.g mechanical constraints, the position of charged amino-acids, etc.). Nevertheless, without any assumption about the actual molecular structure of the channel, the free energy function of the generic conformation i can be expressed

as a Taylor series expansion:

$$G_i(V) = \sum_{j=0}^{+\infty} A_j^{(i)} V^j \tag{4.40}$$

The first order approximation of this expression leads to the linear model of equation (4.31). $A_0^{(i)}$ and $A_1^{(i)}$ assume respectively the same meaning of A_i and B_i in the linear approximation of the free energy funciton. The higher-order terms describe effects such as electronic polarization and pressure induced by V, as well as mechanical constraints (see Figure 4.6). The energy barrier can then be written as:

$$\begin{aligned} \Delta G(V) &= \sum_{j=0}^{+\infty} A_j^{(*)} V^j - \sum_{j=0}^{+\infty} A_j^{(0)} V^j \\ &= \sum_{j=0}^{+\infty} (A_j^{(*)} - A_j^{(0)}) V^j = \sum_{j=0}^{+\infty} a_j V^j \end{aligned} \tag{4.41}$$

where $a_j = A_j^{(*)} - A_j^{(0)}$.

The application of this expression to the open-close scheme (4.17) leads to the following expressions for the forward and backwward rate constants:

$$\alpha(V) = \alpha_0 \exp\left(-\textstyle\sum_{i=0}^{+\infty} a_i V^i / RT\right) \tag{4.42}$$

$$\beta(V) = \beta_0 \exp\left(-\textstyle\sum_{i=0}^{+\infty} b_i V^i / RT\right) \tag{4.43}$$

It is worth noting that, in general, the parameters a_i and b_i are not necessarily interrelated because they represent the energy barrier of two different states (here the "closed" and the "open" ones), which, in principle, may have very different distributions of charges, resulting in different coefficients in eq. (4.40) and therefore resulting in different values for a_i and b_i.

We can truncate the Taylor expansion (4.41) to have a polynomial approximation of the energy barrier. For example, the truncation at the second term gives the quadratic expansion, requiring six independent parameters, that can be

written as:

$$\alpha(V) = A^{-[b_1(V-V_H)+c_1(V-V_H)^2]/RT} \tag{4.44}$$

$$\beta(V) = A^{[b_2(V-V_H)+c_2(V-V_H)^2]/RT} \tag{4.45}$$

Similarly, we can write a cubic expansion, which requires eight independent parameters:

$$\alpha(V) = A^{-[b_1(V-V_H)+c_1(V-V_H)^2+d_1(V-V_H)^3]/RT} \tag{4.46}$$

$$\beta(V) = A^{[b_2(V-V_H)+c_2(V-V_H)^2+d_2(V-V_H)^3]/RT} \tag{4.47}$$

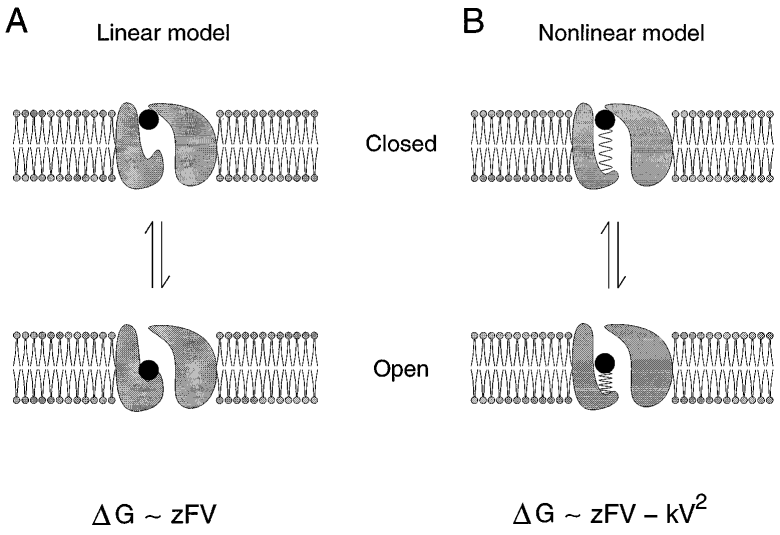


Figure 4.6: Models of ion-channel gating based on the movement of an electric charge inside the channel. A: A freely moving gating charge will result in a free energy that depends linearly on voltage. B: Imposing constraints on the movement of the gating charge will add nonlinear terms in the free energy. The example shown here illustrates the case of a gating charge attached to a spring of constant k, which will result in a quadratic voltage-dependence of the free energy. From Destexhe & Huguenard [2000].

Conclusions about rate constants

Thermodynamic models provide a solid theoretical framework to describe conformational changes of the voltage-activated channel proteins. The use of nonlinear models reduces the number of states needed, and provides better explanation of experimental data Destexhe & Huguenard [2000].

Rate constants are limited for mechanical reasons, indeed a protein cannot change its conformation arbitrarily fast. This effect can be obtained using non linear thermodinamics models (4.44, 4.45) or (4.46, 4.47) or functions that saturates at extreme voltages (4.38). To state which is the best solution, we need to compare both the models according to their ability to explain experimental data. We will make this analysis as soon as they will be available.

4.4 Multi-compartmental models

The value of the membrane potential along a passive, infinitely long and uniform cylindrical dendrite of diameter d, with axial resistivity ρ, membrane specific resistance and capacity R_m and C_m respectively, is conveniently described by the following equation:

$$\tau \frac{\partial V}{\partial t} = \lambda^2 \frac{\partial^2 V}{\partial x^2} + (E_L - V) \tag{4.48}$$

where $\tau = \sqrt{R_m C_m}$ is the membrane time constant, $\lambda = \sqrt{r_m / r_a}$ the length constant and E_L is the leakage potential. r_m and r_a are respectively the specific membrane and axial resistance per unit length of the cable, and are defined as

$$r_m = \frac{R_m}{\pi d}, \quad r_a = 4\frac{\rho}{\pi d^2} \tag{4.49}$$

Similarly we define the capacity per unit length as

$$c_m = \pi d\, C_m labeleq : per_u nit_l ength_c \tag{4.50}$$

Equation (4.48) gives also an analytical solution for the voltage along a finite cable if suitable boundary conditions are imposed.

Although it has been proved that a branched dendrite can be assimilated to a

finite cable if some constraints on the diameters of the various segments are met (Rall [1962]), obtaining an analytical solution from (4.48) for a more generic and realistic dendritic structure is far too complex, and therefore the problem must be solved numerically. In order to solve for the membrane potential, the dendritic tree must be divided into small cylindric compartments with an approximately uniform membrane potential and uniform electrical properties. Adjacent compartments are coupled by the longitudinal resistance that is determined by their geometrical properties (see Figure 4.7).

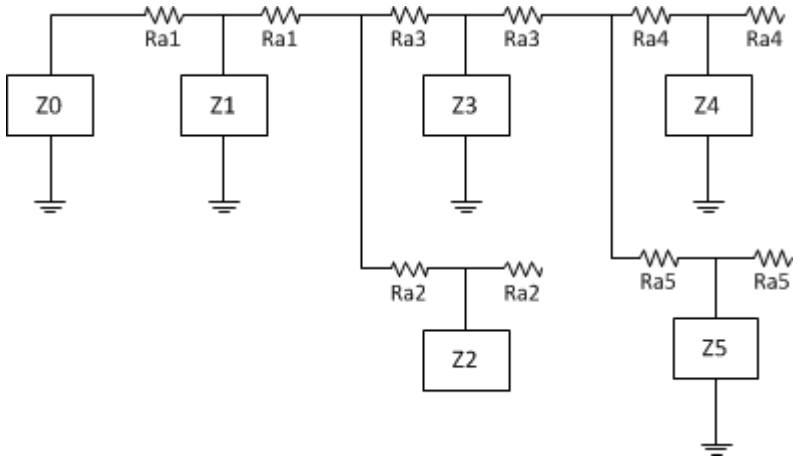


Figure 4.7: Electric equivalent model of a compartmentalized dendritic tree. In this high-level abstraction we do not represent explicitly other electrical properties of the compartments, which are left as responsibility of the generic element Z_i. The axial resistance of each compartment is $2R_{a_i} = r_a \Delta x_i$, where Δx_i is the length of the i^{th} compartment.

The simplest possible compartment is a passive cylinder of length Δx, in which the generic element consist of a parallel between a capacitor $C = c_m \Delta x$ and a conductance $G_m = \Delta x / r_m$ driven by the leakage potential E_L. Synaptic currents, when needed, are provided by a voltage-dependent current source. This passive circuit can be easily adapted to represent also active compartments described either with the conductance-based formalism or with the AdEx model: indeed it is only required to add additional voltage-dependent current sources (consisting on voltage-dependent conductances and a driving force in case of conductance based models) in parallel with the capacitor and the passive leakage conductance. On the other hand the Izhikevich model requires an *ad hoc* representation, the

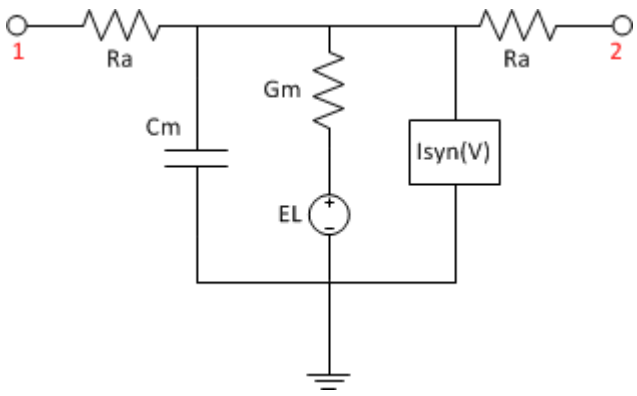


Figure 4.8: Representation of a passive cylindrical compartment.

mathematical model not including a linear voltage-dependent current, as the leakage one is.

4.4.1 Mathematical model of a dendritic tree

The most natural way to represent a dendritic three is a graph, where the edges are the compartments and the nodes the points where the compartments join. Each compartment can be represented, without any loss of generality, as in Figure 4.9.

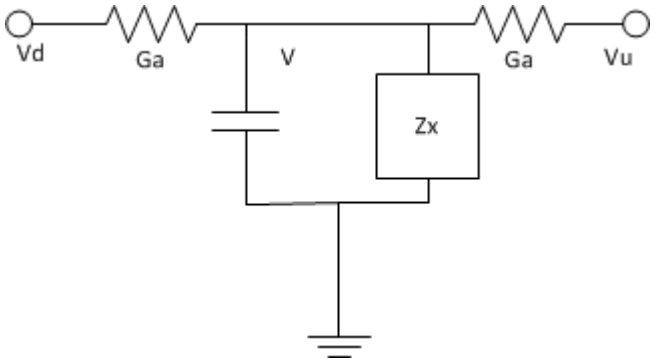


Figure 4.9: Representation of a generic compartment. V_d and V_u are the potentials at the extremites of the compartment (d is for *down*, meaning "toward the root", and u is for *up*, in the sense of "toward the leaves" of the "dendritic tree"), G_a is the axial conductance, reciprocal of the axial resistance R_a

For the Kirchhoff law of currents, we can say that the current through the capacitor of the k-th compartment is equal to the sum of the currents leaving the

central node, and hence, for the current-voltage relation of a capacitor, we can write:

$$C_k \dot{V}_k = G_{k,a}(V_{n_i} - V_k) + G_{k,a}(V_{n_j} - V_k) - f(V_k) \tag{4.51}$$

where $f(V_k)$ is the total membrane current caused by the generic element Z_x, whereas the other two components on the right hand side of the equation represent the current leaving towards other compartments.

The potentials V_{n_i} and V_{n_j} of nodes i and j respectively are obtained as the weighted average of the potentials of all the compartments incident on them. That is, for a generic node ℓ, the following:

$$V_{n_\ell} = \sum_{i \in \Im(n_\ell)} \left(G_k^i V_i\right) \Big/ \sum_{i \in \Im(n_\ell)} G_k^a \tag{4.52}$$

where $\Im(n_\ell)$ is the set of edges incident on node ℓ.

Including eq. (4.52) in eq. (4.51), we can describe a whole tree of n compartments using a single matrix equation:

$$\mathbf{C}\dot{\mathbf{V}} = \mathbf{G_{den}}\mathbf{V} - f(\mathbf{V}) \tag{4.53}$$

where $\mathbf{V} \in \mathbb{R}^n$ is a column vector containing all the membrane potential $\mathbf{C} \in \mathbb{R}^{n \times n}$ is a diagonal matrix where C_{ii} is the capacity of the membrane of the i-th compartment, $\mathbf{G_{den}} \in \mathbb{R}^{n \times n}$ is the square matrix containing the information about the edge linking and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a vector function implementing the dynamics within each compartment such that $f_i(\mathbf{V}) = f_i(V_i)$. In particular this general representation can describe:

- the single-compartment case, when $n = 1$ and obviously $\mathbf{G_{den}} = \mathbf{0}$;
- a passive tree, if $f_i(\mathbf{V}) = G_m(V_i - E_L)$;
- any other model shown in this chapter, when the function $f(\cdot)$ is properly implemented.

4.5 Conclusions

We showed that both conductance-based and phenomenological models can be used to simulate the activity of the cell membrane. In the following paragraph, we will draw some conclusions.

Conductance-based Models

In conductance-based models each known ionic current is described explicitly. According to Hodgkin-Huxley formalism, each ion channel is controlled by two independent voltage-dependent "gating particles", each one characterized by a set of first order differential equations. If, on the other hand, simplified Markov models are used, the whole channel is described by a small system of first order differential equations.

Markov models, in combination with thermodynamic-based rate constants, provide a valid alternative to Hodgkin-Huxley formalism, because all the channels can be described using the same scheme (scheme (4.23) is obtained from (4.24) simply forcing r_3, r_4 and r_5 to zero), changing only the parameters of the rate functions.

Nevertheless, the fact that each firing pattern requires different set of ionic currents, limits the use of this type of models in parallel simulations, unless particular strategies are used[13]. The situation is even more complicated if calcium dynamics is included in the model, because handling electrochemical diffusion in multi-compartmental models (an issue not treated here) is not a trivial task.

For all these reasons, we decided to use a phenomenological model to simulate active properties of the cellular membranes, in this early version of our software. We leave the implementation of conductance-based with a future work.

Phenomenological Models

Despite their simple formulation, some phenomenological models are capable to reproduce (almost) all the electrical responses found in cortical neurons. In par-

[13]We do not analyze these strategies in the present work because we are still studying them and we have not found a definitive solution yet.

ticular, we have analyzed two variant of the IF neuron: the Izhikevich model and the *adaptative exponential integrate-and-fire* model.

We chose to implement the AdEx model for the following reasons:

- its formulation is the same of a passive compartment in which an exponential function of the voltage is added. This means, essentially, that the same algorithm can be used, with slight differences, to simulate both active and passive compartments.
- the Izikevich model shows unrealistic nonlinearities in the subthreshold regime and exhibits a too slow upswing of the action potential compared to real neurons (Izhikevich [2008])
- the dimensionless formulation of the Izhikevich model makes it difficult to be coupled with the parameters of a passive dendritic tree. It would be required to rescale all the parameters describing the tree and the synaptic input, or to rescale the model parameters itself to match the parameters of the dendritic tree's physiology. In both cases, extra work is needed to be done, and the advantage of having a precharacterized space of parameters is lost.

Dendrites (and compartments)

Equation (4.53) allows us to describe the dynamics of a whole neuron, considering the cell body and the dendrites, with a set of first-order differential equations. The structure of (4.53) allows to decouple the inter-compartment interactions from the intra-compartment specific behavior.

Chapter 5

Synapses

5.1 Introduction

As synapses are the most numerous entities within a neural network simulation, our aim was to obtain a model as simple as possible, but capable to take into account all the main features of synaptic transmission. We approached the problem using two different formalisms: the first one uses the framework of linear system theory and handles the problem from a higher level of abstraction, while the second one is closer to the detailed models mimicking the physiological phenomena more accurately.

In this work, which is an early report on a complete simulation software system, we neglected the plasticity caused by synaptic transmitter release for the sake of simplicity of the network at the moment. In the future versions of the model they will be included.

First of all we will analyze detailed models of synaptic conductances. Then we will make some considerations about the transmitter release process, and finally we will treat the above mentioned models.

5.2 Detailed models of synaptic currents

We simulated most important synaptic currents in the central nervous system (i.e. AMPA/kainate, NMDA, $GABA_A$and $GABA_B$mediated currents) with highly

detailed models, indistinguishable from real recordings, in order to get data to be fitted by our simplified models (because large data sets of experimental recordings were not available in this case). Synaptic currents were generated according to simulated voltage clamp experiments. In the following subsections simulation techniques are illustrated.

5.2.1 Presynaptic mechanisms of transmitter release

Transmitter release involves different processes in the presynaptic side (Yamada & Zucker [1992]). An exhaustive model requires several differential equations to describe the phenomenon, but Destexe and coworkers (Destexhe *et al.* [1994]) realized that mechanisms underlying transmitter release are so fast that transmitter's concentration inside the synaptic cleft ($[T]$) can be considered in equilibrium with the presynaptic membrane potential (V_{pre}). They found that this stationary relationship is well approximated by a sigmoid function when an ohmic relation is used to simulate Ca^{2+} currents[1]:

$$[T](V_{pre}) = \frac{[T]_{max}}{1 + \exp[-(V_{pre} - V_p)/K_p]} \tag{5.1}$$

where T_{max} is the maximal concentration of transmitter in the synaptic cleft, $K_p = 5\,\mathrm{mV}$ gives the steepness and $V_p = 2\,\mathrm{mV}$ sets the value at which the function is half-activated.

Some hypothesis were made to obtain this relation: (a) upon invasion by an action potential, Ca^{2+} enters the presynaptic terminal due to the presence of HVA Ca^{2+} current; (b) Ca^{2+} activates a calcium-binding which promotes release by binding to the synaptic vesicles; (c) an inexhaustible supply of vesicles are available in the synaptic bouton, ready to release. While hypothesis (a) and (b) make the model close to physiological reality, (c) sets a strong constraint on what can be modeled: assuming that there are always vesicles ready to be used implies that short-term presynaptic-phenomena-induced plasticity is completely

[1] This means that calcium reversal potential is considered to the Nernst potential from equation (2.1), and that current-voltage relation is linear. The complementary approach is to use the Goldman-Hodgkin-Katz equation (2.2), which leads to a different function for the transmitter-voltage relation.

neglected. However, this does not represent an immediate problem, as modeling these phenomena is not in the purpouse of this work.

5.2.2 Markov models of postsynaptic currents

We required that our detailed model was able to capture three important aspects of receptor gating kinetics:

- *Activation/binding*: at low concentrations of transmitter, the time course of the rising phase of the synaptic current is rate limited by the amount of transmitter present in the synaptic cleft. At higher concentrations, activation is controlled by the opening rate once the transmitter is bound to the receptor. The more molecules of transmitter must bind to the receptor to activate it, the more the activation will be delayed.

- *Deactivation/unbinding*: both desensitization and transmitter removal contributes to the time course of the decaying phase. Deactivation rate is limited either by the closing rate of the receptoror by the rate of unbinding of transmitter from the receptor.

- *Desensitization*: we already mentioned that ligand-gated channels can enter a desensitized state, similar to the inactivated state for voltage-gated ion channels. Desensitization decreases the fraction of channel that open after a repeated stimulation, inducing a form of short term synaptic depression.

AMPA/kainate receptors

AMPA/kainate receptors mediate the prototypical fast excitatory synaptic currents in the brain. Their raise time can be in the submillisecond range, while the decay time constant is about 5 ms. It is believed that the decay time constant is due mainly to transmitter remove from the synaptic cleft rather than to desensitization of the receptor. It is worth noting that different types of neurons express different types of AMPA receptors. For example inhibitory interneurons express AMPA receptors that are about twice as fast in rise and decay times as those on pyramidal neurons (Hestrin. [1993]).

For the simulation of AMPA currents we used the Markov kinetic model as described by Standey and collegues(Standley *et al.* [1993]):

$$\begin{array}{ccccc} C_0 & \underset{R_{r_1}}{\overset{R_{d_1}}{\rightleftharpoons}} & D_1 & \underset{R_{r_2}}{\overset{R_{d_2}}{\rightleftharpoons}} & D_1 \\ R_{u_1} \Big\uparrow \Big\downarrow R_{b_1} & & & & \\ C_1 & \underset{R_{u_2}}{\overset{R_{b_2}}{\rightleftharpoons}} & C_2 & \underset{R_c}{\overset{R_o}{\rightleftharpoons}} & O \end{array} \tag{5.2}$$

where the unblound closed form of the receptor C_0 binds to one molecule of transmitter T, leading to the singly-bound closed form C_1 or to the singly-bound desensitized state D_1. By binding another molecule of T it can then move to states C_2 or D_2, respectively doubly-bound closed and desensitized forms. The receptor can then open from the state C_2, leading to the open form O. For the rate constants are used the values as found in Destexhe *et al.* [1994]: $R_{b_1} = 20\,\mathrm{ms}^{-1}\,\mathrm{mM}^{-1}$, $R_{u_1} = 1.3\,\mathrm{ms}^{-1}$, $R_{b_2} = 10\,\mathrm{ms}^{-1}\,\mathrm{mM}^{-1}$, $R_{u_2} = 2.6\,\mathrm{ms}^{-1}$, $R_o = 0.9\,\mathrm{ms}^{-1}$, $R_c = 0.5\,\mathrm{ms}^{-1}$, $R_{d_1} = 10\,\mathrm{ms}^{-1}\,\mathrm{mM}^{-1}$, $R_{r_1} = 0.0002\,\mathrm{ms}^{-1}$, $R_{d_2} = 0.002\,\mathrm{ms}^{-1}\,\mathrm{mM}^{-1}$, $R_{r_2} = 0.0001\,\mathrm{ms}^{-1}\,\mathrm{mM}^{-1}$.

The AMPA current is then given by:

$$I_{AMPA} = \bar{g}_{AMPA}[O](V - E_{AMPA}) \tag{5.3}$$

where $\bar{g}_{AMPA} = 1\,\mathrm{nS}$ (Destexhe *et al.* [1994]) is the maximal conductance, $[O]$ is the fraction of channels in the open state, V is the postsynaptic membrane voltage and $E_{AMPA} = 0\,\mathrm{mV}$ is the reversal potential.

NMDA receptors

NMDA receptors mediate slow excitatory currents, with a rise time of about 20 ms and decay time constant between 25 ms to 125 ms. The slow activation kinetic is due to the requirement that two molecules of transmitter must bind to open the channel, together with intrinsic slow opening rate of the bound form. The slow decay is belived to be caused by slow unbinding of glutamate from the receptor.

NMDA receptors are blocked by physiological concentrations of Mg^{2+}. This

block is voltage dependent and allows NMDA receptors to conduct only at depolarized membrane potential. For NMDA receptor we used the following model (Destexhe *et al.* [1998]):

$$C_0 \underset{R_u[T]}{\overset{R_b[T]}{\rightleftharpoons}} C_1 \underset{R_u[T]}{\overset{R_b[T]}{\rightleftharpoons}} C_2 \underset{R_c}{\overset{R_o}{\rightleftharpoons}} O \qquad (5.4)$$

$$C_2 \underset{R_d}{\overset{R_r}{\rightleftharpoons}} D$$

where the unbound form C_0 binds to a molecule of T leading to the singly-bound form C_1. With the same rate constant R_b, the receptor move from C_1 to C_2 when it bound one more molecule of transmitter. From C_2 the receptor may enter a desensitized state D or open, entering the state O. In (Destexhe *et al.* [1998]) the following values for the rate constants are provided: $R_b = 5000\,\mathrm{mM^{-1}\,s^{-1}}$, $R_u = 12.9\,\mathrm{s^{-1}}$, $R_d = 8.4\,\mathrm{ms^{-1}}$, $R_r = 6.8\,\mathrm{s^{-1}}$, $R_o = 46.5\,\mathrm{s^{-1}}$ and $R_c = 73.8\,\mathrm{s^{-1}}$.

The NMDA current is described by the following equation:

$$I_{NMDA} = \bar{g}_{NMDA}\, B(V)[O](V - E_{NMDA}) \qquad (5.5)$$

where all the parameters have the same meaning as for AMPA current. $\bar{g}_{NMDA} = 1\,\mathrm{nS}$, $E_{NMDA} = 0\,\mathrm{mV}$. $B(V)$ is the voltage-dependent magnesium block. It has been demonstrated that Mg^{2+} block is an extremely fast process compared to other kinetics of the NMDA receptor, therefore it can be described by an instantaneous function of the postsynaptic membrane potential (Jahr & Stevens [1990]):

$$B(V) = \frac{1}{1 + [Mg^{2+}]_{out} \exp(-0.062V)/3.57} \qquad (5.6)$$

where $[Mg^{2+}]_{\mathrm{out}}$ is the external magnesium concentration (1 to 2 mM in physiological conditions).

GABA$_A$receptors

GABA$_A$receptors mediate most of the fast inhibitory synaptic currents in the central nervous system. They have at least two binding sites for GABA and show a weak desensitization. However, as with AMPA/kainate receptors, deactivation following transmitter removal is the main determinant of the decay time.

The model we used was found in (Destexhe *et al.* [1998]) and is based on the following state diagram:

$$\begin{array}{ccccc} C_0 & \underset{R_{u_1}[T]}{\overset{R_{b_1}[T]}{\rightleftharpoons}} & C_1 & \underset{R_{u_2}[T]}{\overset{R_{b_2}[T]}{\rightleftharpoons}} & C_2 \\ & & \underset{R_{o_1}}{\overset{R_{c_1}}{\rightleftharpoons}} & & \underset{R_{o_2}}{\overset{R_{c_2}}{\rightleftharpoons}} \\ & & O_1 & & O_2 \end{array} \tag{5.7}$$

The transmitter (GABA) can bind to the unbounded closed form C_0 leading to the singly- and doubly-bounded forms, respectively C_1 and C_2. Both forms can open and lead to the open forms O_1 and O_2 respectively. The autors of the paper reported the following values for the rate constant: $R_{b_1} = 20\,000\,\mathrm{mM^{-1}\,s^{-1}}$, $R_{u_1} = 4600\,\mathrm{s^{-1}}$, $R_{b_2} = 10\,000\,\mathrm{mM^{-1}\,s^{-1}}$, $R_{u_2} = 9200\,\mathrm{s^{-1}}$, $R_{o_1} = 3300\,\mathrm{s^{-1}}$, $R_{c_1} = 9800\,\mathrm{s^{-1}}$, $R_{o_2} = 10\,600\,\mathrm{s^{-1}}$ and $R_{c_2} = 410\,\mathrm{s^{-1}}$.

The current is given by:

$$I_{GABA_A} = \bar{g}_{GABA_A}([O_1] + [O_2])(V - E_{Cl}) \tag{5.8}$$

where $\bar{g}_{GABA_A} = 0.5\,\mathrm{nS}$ is the maximal synaptic conductance, $[O_1]$ and $[O_2]$ is the fraction of receptors in the open states and $E_{Cl} = -70\,\mathrm{mV}$ is the reversal potential of chloride, the ion GABA$_A$receptors are permeable to.

GABA$_B$receptors

GABA$_B$receptors are metabotropic receptors, whose response is mediated by K^+ channels that are activated by intracellular concentration of G-proteins activated when the transmitter binds to the receptor. They require high level of presynaptic

activity to elicit a response. The process leading to the opening of potassium channels can be describe by the following kinetic equations (Destexhe & Sejnowski [1995]):

$$\frac{d[R]}{dt} = K_1[T](1 - [R] - [D]) - K_2[R] + K_3[D] \tag{5.9a}$$

$$\frac{d[D]}{dt} = K_4[R] - K_3[D] \tag{5.9b}$$

$$\frac{d[G]}{dt} = K_5[R] - K_3[G] \tag{5.9c}$$

$$I_{GABA_B} = \bar{g}_{GABA_B} \frac{[G]^n}{[G]^n + K_d}(V - E_K) \tag{5.9d}$$

where $[R]$ and $[D]$ are respectively the fraction of activated and desensitized receptor, $[G]$ is the concentration (in µM) of activated G-protein, $\bar{g}_{GABA_B} = 1\,\text{nS}$ is the maximal conductance of K^+ channels, $E_K = -90\,\text{mV}$ is the reversal potential of calcium and K_d is the dissociation constant of the binding of G on K^+ channels. Parameters have the following values: $K_d = 100\,\mu\text{M}^4$, $K_1 = 660\,\text{mM}^{-1}\,\text{s}^{-1}$, $K_2 = 20\,\text{s}^{-1}$, $K_3 = 5.3\,\text{s}^{-1}$, $K_4 = 17\,\text{s}^{-1}$, $K_5 = 0.83\,\text{mM}\,\text{s}^{-1}$, $K_6 = 7.9\,\text{s}^{-1}$ and $n = 4$ is the number of binding sites for G-proteins on the potassium channel.

5.2.3 Generation of the dataset

We generated current traces for each type of receptor, simulating voltage-clamp experiments where postsynaptic potential was hold at $V_{clamp} = -60\,\text{mV}$ using the formalism introduced in this section. Presynaptic voltage was simulated by an Hodgkin-Huxley model with Na^+ and K^+ conductances for the generation of action potentials. We provide two different types of stimulation: in the first experimental protocol a single spike was elicited by the injection of a brief current pulse lasting 5 ms.

5.3 Transmitter release process

Synaptic conductances depend on the concentration of transmitter inside the synaptic cleft, as the gating process of the postsynaptic receptors is ligand-

dependent. Then it is crucial to find a model for this process. In the previous section we showed how it can be considered as an instantaneous function of the presynaptic potential (eq. (5.1)). This relation can be further simplified assuming that transmitter concentration immediately saturates at its maximum value at the onset of an action potential, and that it is immediately removed from the synaptic cleft when the spike ends. The transmitter-potential relation therefore assumes a pulse-like fashion:

$$[T](t) = \begin{cases} [T]_{max}, & \text{if } V_{pre}(t) \geq V_{th} \\ 0, & \text{otherwise} \end{cases} \tag{5.10}$$

where $V_t h$ is the threshold for transmitter release.

The implicit assumption underlying the previous equation is that the presynaptic voltage is generated by mean of a physiologically plausible model. As a matter of fact things are different if a phenomenological model is used to simulate presynaptic membrane potential: because of the hard reset on spike detection, the duration of an action potential might be considerably different depending on the time step of the simulation. Let's assume that the rising phase of the action potential is so rapid that the elapsed time since the crossing of V_{th} and the reaching of the peak value can be neglected. Let the integration time step be Δt: between the peak value of the presynaptic potential and its reset, we have that the potential is above the threshold for transmitter release for an equivalent simulated time of $t_{rel} = \Delta t\,(V_{peak} - V_{th})/(V_{peak} - V_{reset})$. If we consider a simulation run with $\Delta t = 1\,\text{ms}$ and another with $\Delta t = 0.01\,\text{ms}$, we have that in the first case the transmitter is released for a period hundred times longer than in the second one. We underline that the same problem, even if with minor impact, is found also in detailed model for the simulation of the membrane potential, as different parameters for the same ion channel may lead to slightly different time courses of the action potential waveform, that means maybe slightly different durations. This problem can be avoided if we impose that transmitter release must occur

for a given predefined time after a spike is detected, that is:

$$[T](t) = \begin{cases} [T]_{max}, & \text{if } t - t_S < t_r \\ 0, & otherwise \end{cases} \tag{5.11}$$

where t_S is the time of the onset of the last spike and t_r is the duration of the releasing process, dependent on the particular type of presynaptic neuron.

In next sections, when required, we will simulate transmitter concentration using (5.11), in order to prove if a synaptic model is good independently from the method used to simulate the presynaptic potential. Of course, since (5.11) is a strong approximation of reality, we are aware that we might not obtain perfect fits. However we are not interested that the prediction of a model will perfectly overlap with the experimental data, but rather that it can reproduce the essential features of the particular synaptic current of interest.

5.4 Synapses as Linear Time Invariant Systems

Analyzing the problem from a very abstract point of view we can say that what happens in synapses is that a transient current, due to a change of the synaptic conductance, appears at the postsynaptic side when an action potential invades the synaptic bouton. We can treat synapses as a "black box" that convert presynaptic activity into postsynaptic conductances. We decided to use a linear time-invariant (LTI) system to model these phenomena because they are quite flexible, easy to analyze and computationally efficient.

5.4.1 Definition of the input of the system

We followed a top-down approach to model our system: we made very strong abstractions and then went into detail.

Our first hypothesis was to consider the input of the system to be presynaptic spikes, completely neglecting transmitter release dynamics. We justify this decision because spikes are intrinsically impulsive phenomena. Spikes are modeled as Dirac's delta function, hence the change of the synaptic conductance after a

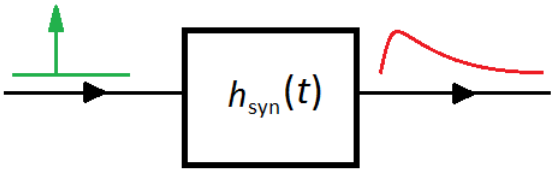


Figure 5.1: Schematic representation of the synaptic I/O system. Action potentials of the presynaptic neuron are modeled as Dirac's delta function (green arrow). The output of the system is the time course of the synaptic conductance (red curve).

spike is the impulse response of the system itself. The input of the system $u(t)$ is provided by a spike detection algorithm, a simple function that returns a impulse whenever the presynaptic potential crosses a given threshold with positive slope:

$$u(t) = \sum_i \delta(t - t_{S_i}) \tag{5.12}$$

where t_{S_i} is the time of i-th spike that satisfies:

$$f(t_{S_i}) = \begin{cases} V_{pre}(t_{S_i}) & = V_{th} \\ V'_{pre}(t_{S_i}) & > 0 \end{cases}$$

In the actual implementation this function compares the value of the presynaptic potential at a given time step and the previous one with the threshold, detecting an action potential at the step t[2] if and only if $V_{pre}(t) \geq V_{th}$ and $V_{pre}(t-1) < V_{th}$.

In our second approach the input of our system was a signal mimicking the concentration of transmitter in the synaptic cleft after an action potential. We slightly modified equation (5.11) to represent an dimensionless normalized signal:

$$u(t) = \begin{cases} 1, & \text{if } t - t_S < t_r \\ 0, & otherwise \end{cases} \tag{5.13}$$

[2]Here t represent a simulation time step, hence it is not a continuous variable, but it assumes discrete integer values.

5.4.2 Choice of the impulsive response function

The input/output relation of a LTI system can always be represented by a linear ordinary differential equation with constant coefficients:

$$\sum_{i=0}^{n} a_i y^{(i)}(t) = \sum_{j=0}^{m} b_j x^{(j)}(t) \tag{5.14}$$

where $y^{(i)}(t)$ and $x^{(j)}(t)$ are respectively the i-th and j-th derivatives of the output $y(t)$ and of the input $x(t)$ ($y^{(0)}(t) = y(t)$ and $x^{(0)}(t) = x(t)$).

Due to memory occupancy reason, we restricted our analysis to second order systems ($n = 2$) with dependency only on the present state of the input ($m = 0$), which require two state variables to be simulated plus six parameters to store the coefficients of the system. Equation (5.14) becomes:

$$a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_0 x(t) \tag{5.15}$$

The impulse response ($h(t)$) of such a system depends on the roots λ_1 and λ_2 of the polynomial $a_2\lambda^2 + a_1\lambda + a_0 = 0$. In particular, they can be:

real and distinct, leading to	$h(t) = A_1\, e^{\lambda_1 t} + A_2\, e^{\lambda_2 t}$	(5.16a)
real and coincident, leading to	$h(t) = A\, t\, e^{\lambda t}$	(5.16b)
complex conjugates, leading to	$h(t) = A_1\, e^{(\sigma+i\omega)t} + A_2\, e^{(\sigma+i\omega)t}$	(5.16c)

Due to the causality of the system, all $h(t) = 0$ when $t < 0$. In our particular case oscillating solutions are of no interest, thus we made use only of the functions (5.16a) and (5.16b). They were modified in order to be more physiologically meaningful and, with reference to eq. (5.16a), to respect some constraints imposed by the particular problem, that is:

1. $h(0) = 0$, that implies $A_1 = -A_2$, thus $h(t) = A\left(e^{\lambda_1 t} - e^{\lambda_2 t}\right)$.
2. $h(t)$ must be positive for $t > 0$, because negative conductances are physically meaningless; this fact requires $\lambda_1 < \lambda_2$.

Finally, we obtained:

$$h_1(t) = g_{max}\,\frac{t}{\tau}\,e^{1-t/\tau} \tag{5.17a}$$

$$h_2(t) = g_{max}\,\left(e^{-t/\tau_1} - e^{-t/\tau_2}\right), \quad \tau_1 > \tau_2 \tag{5.17b}$$

5.4.3 Identification of the model

We fitted the artificially generated data with the linear model. We tested both the delta impulsive input (5.12) and the square-pulse release (5.13).

Delta-pulse input

We first tested our simplest hypotesis, i.e. we condidered synaptic input as described in equation (5.12). In this case the output of the system is the impulse response itself, thus we fitted the currents obtained as describe in section 5.2.3 with the following function:

$$I_{syn}(t) = g(t)\,(V_{clamp} - E_{syn})$$

where $g(t)$ is one of (5.17b) and (5.17a), $V_clamp = -60\,\mathrm{mV}$ and E_{syn} is the reversal potential of each receptor. The fitting procedure was performed using a non linear least squares algorithm. The estimates of the parameters are shown in tables 5.1 and 5.2. It is noticeable how the bi-exponential model is the best for all the receptors according to the weighted residual sum of squares of the estimates.

A comparison between the currents predicted by simple model and the ones obtained with the detailed model is shown in Figure 5.2. Our opinion is that bi-exponential models, even with some discrepancies, well reproduce the time course of synaptic currents for ionotropic receptors (i.e. AMPA, NMDA and $GABA_A$), while they cannot mimic the slow activation phase of the $GABA_B$mediated current.

Square-pulse release

We then tested the model with the second type of input (see eq. (5.13)) assuming that $t_r = 1.5\,\mathrm{ms}$. This is still a simple function, and the output of the system can

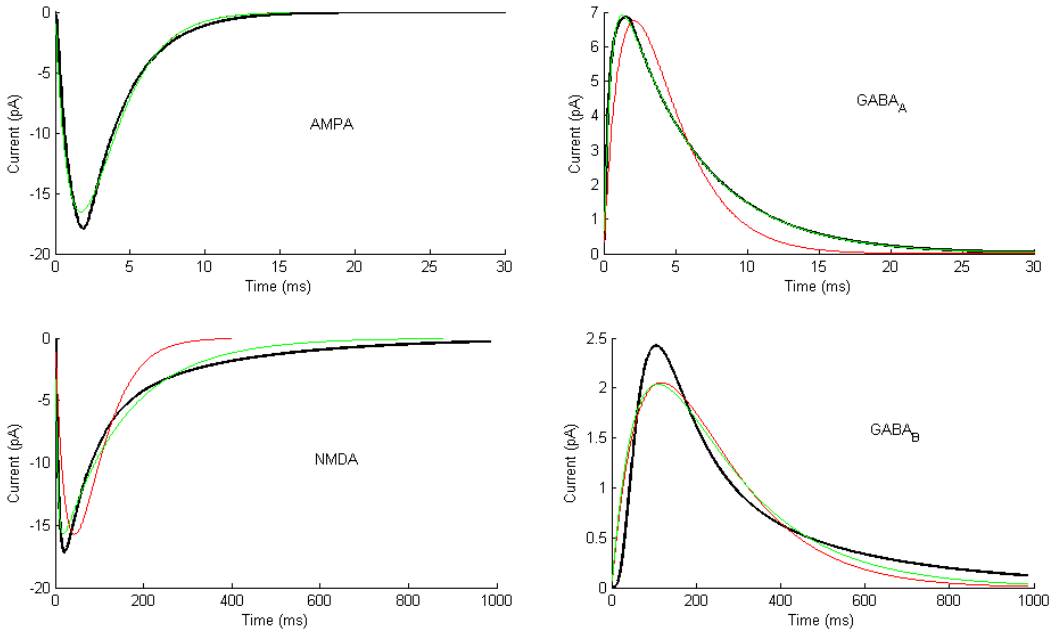


Figure 5.2: Best fits of the mono (red traces) and bi-exponential(blue traces) models to the current traces generated with the detailed models (black traces), when the input of the system consists on a Dirac's delta pulse.

be analytically evaluated once the duration of the pulse (t_r) is known[3]. Solving the integral, we obtain:

$$g_1(t) = \begin{cases} 0, & t < 0 \\ \bar{g}\,[T]_{max}\, e\, \left[\tau\left(1 - e^{-\frac{t}{\tau}}\right) - t\, e^{-\frac{t}{\tau}}\right], & 0 \leq t < t_r \\ \bar{g}\,[T]_{max}\, e\, \left[(t - t_r + \tau)\left(1 - e^{-\frac{t-t_r}{\tau}}\right) - (t+\tau)e^{-\frac{t}{\tau}}\right], & t \geq t_r \end{cases} \tag{5.18}$$

for the convolution with $h_1(t)$, and

$$g_2(t) = \begin{cases} 0, & t < 0 \\ \bar{g}\,[T]_{max}\, \left[\tau_1\left(1 - e^{-\frac{t}{\tau_1}}\right) - \tau_2\left(1 - e^{-\frac{t}{\tau_2}}\right)\right], & 0 \leq t < t_r \\ \bar{g}\,[T]_{max}\, \left[\tau_1\left(e^{\frac{t_r}{\tau_1}} - 1\right) e^{-\frac{t}{\tau_1}} - \tau_1\left(e^{\frac{t_r}{\tau_1}} - 1\right) e^{-\frac{t}{\tau_1}}\right], & t \geq t_r \end{cases} \tag{5.19}$$

[3]From our data we could estimate that $t_r \approx 1.75\,\text{ms}$.

the output when the impulse response is $h_2(t)$.

The estimates are reported in Table 5.3 and Table 5.4, and their fits in Figure 5.3. Once again bi-exponential functions fit better to the data, but the model is still not capable to mimic the slow onset of the $GABA_B$-mediated current.

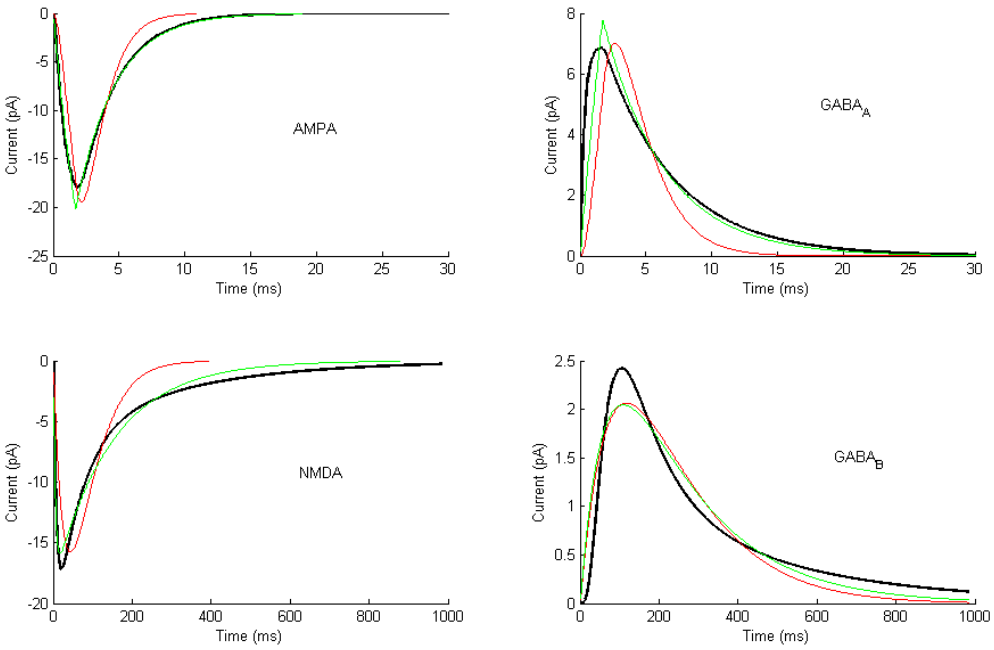


Figure 5.3: Best fits of the mono (red traces) and bi-exponential(green traces) models to the current traces generated with the detailed models (black traces) when the input of the system is a short-lasting pulse of duration t_r.

Final considerations on LTI systems

Comparing the weighted residual sum of squares of the best-fitting impulse response (i.e. the bi-exponential function) returned by the fitting procedure, when input (5.12) and (5.13) are applied, we noticed that the output of the system fits better to the data when it is fed with input (5.13), except the case of the $GABA_A$receptor. The predicted current does not mimic the fast rising phase of the $GABA_A$-mediated current, even if it fits better during the decaying phase. This is surely a limit of our model, but we cannot say that it completely failed for the simulation of $GABA_A$-mediated currents. Many subtypes of the same

Receptor	$\bar{g}_{\max}$ (nS)	τ_1 (ms)	τ_2 (ms)	WRSS
AMPA	3.530	1.905	1.540	30296
$GABA_A$	0.983	5.271	0.510	19847
NMDA	0.309	146.214	5.431	724875
$GABA_B$	0.187	194.188	69.261	85089

Table 5.1: Estimates of the parameters for the bi-exponential model: $\bar{g}_{\max}$, maximal conductance; $\tau_{1,2}$ time constants; WRSS, weighted residual sum of squares.

Receptor	$\bar{g}_{\max}$ (nS)	τ (ms)	WRSS
AMPA	0.276	1.725	30306
$GABA_A$	0.675	2.160	35092
NMDA	0.262	43.999	5426623
$GABA_B$	0.068	119.192	91439

Table 5.2: Estimates of the parameters for the mono-exponential model: $\bar{g}_{\max}$, maximal conductance; τ time constant; WRSS, weighted residual sum of squares.

Receptor	$\bar{g}_{\max}$ (nS)	τ_1 (ms)	τ_2 (ms)	WRSS
AMPA	0.262	2.943	0.005	25141
$GABA_A$	0.551	4.670	0.017	33438
NMDA	0.175	146.674	4.640	699288
$GABA_B$	0.107	196.376	67.248	82018

Table 5.3: Estimates of the parameters for the bi-exponential model: $\bar{g}_{\max}$, maximal conductance; $\tau_{1,2}$ time constants; WRSS, weighted residual sum of squares.

Receptor	$\bar{g}_{\max}$ (nS)	τ (ms)	WRSS
AMPA	0.206	1.074	138533
$GABA_A$	0.419	1.667	87818
NMDA	0.150	43.413	5708173
$GABA_B$	0.039	118.401	89612

Table 5.4: Estimates of the parameters for the mono-exponential model: $\bar{g}_{\max}$, maximal conductance; τ time constant; WRSS, weighted residual sum of squares.

receptors exist, and we used only one detailed model that reproduce the current generated by a specific subtype among many others. This means that currents generated by other subtypes of $GABA_A$receptors may be better fitted by our model.

An other important limitation of this model is that it does not reproduce saturation of the synaptic conductance, nor desensitization of the receptors. As a matter of fact a train of action potentials elicit an unrealistic unbounded increasing response. This is an intrinsic behavior. Let's assume to be in the simplest condition, where a spike train is described by (5.12); the output of the system is the convolution of its impulse response with the input signal. In this simpler situation what happens is:

$$y(t) = h * u(t) = \int_{-\infty}^{+\infty} h(t-u) \sum_i \delta(u - i\Delta t)\, \mathrm{d}t = \sum_i h(t - i\Delta t)$$

It is obvious that if the interspike interval Δt is small enough the i-th signal starts before the transient of the $(i-1)$-th has exausted, and the peak of the $i-th$ repetition would be greater than the peak value of the impulse response.

5.5 Kinetic models for synaptic conductances

In the previous section we realized a phenomenological mathematical model, without considering the physiology underlying synaptic transmission. In this section we will adopt a different approach: we are going to take into consideration physical mechanisms of the gating process, transmitter release and second messaging, developing a model that is based on physiology. The process mainly consists in the simplification of the models introduced in section 5.2.

In all this chapter we consider synaptic currents mediated by ion channel x as generated by a variable conductance multiplied by a potential difference, as in the following formalism:

$$I_x = \bar{g}_x\, s(V - E_x) \tag{5.20}$$

where the variable conductance is obtained weighting the maximal value $\bar{g}_x$ by

the variable s, V is the postsynaptic membrane potential and E_x is the reversal potential for the particular ion channel.

5.5.1 Model of transmitter release

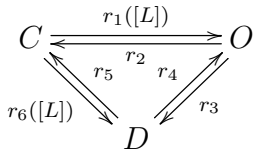
Most of the models developed in this section are based on kinetic schemes[4]. In the particular case of synaptic transmission, the gating process is ligand regulated: this means that some of the rate constants (see equation (A.4)) may depend on the transmitter concentration inside the synaptic cleft. For the simulation of transmitter release we will use equation (5.11).

5.5.2 State diagrams for ligand gated channels

State diagrams represent conformational changes of a protein. The simplest diagram that can be written for a ligand gated channel involves only two states, closed (C) and open (O), where the opening rate is ligand-concentration dependent:

$$C \underset{r_2}{\overset{r_1([L])}{\rightleftharpoons}} O \tag{5.21}$$

Sometimes the introduction of an additional desensitized state (D) is required to take into account time dependency properties of the channel. Scheme (5.21) in this case becomes:



(5.22)

Further simplified schemes can be derived by this one if one or more rate constant are set to zero; in particular scheme (5.21) represent the case in which $r_3 = r_4 = r_5 = r_6 = 0$. In the next paragraphs we will analyze only the second state diagrams, because any consideration valid for it can obviously be applied also to the simpler one.

As we implicitly mentioned talking about rate constants, we are in the context of a large number of proteins, since we want to model macroscopic currents

[4] see Kinetic schemes for an introduction.

produced by a synapse as a whole. In this condition C, O and D represent the fraction of channels that, at any given time, are respectively in the closed, open or desensitized state; hence it follows that

$$C + D + O = 1 \tag{5.23}$$

Thus the number of differential equations in that represent the system[5]can be reduced from three to two setting $C = 1 - O - D$, leading to:

$$\dot{x}_1 = -(r_2 + r_3)x_1 + r_4 x_2 + r_1[L](1 - x_1 - x_2) \tag{5.24a}$$
$$\dot{x}_1 = r_3 x_1 - (r_4 + r_5)x_2 + r_6[L](1 - x_1 - x_2) \tag{5.24b}$$

that we rearrange to obtain the matrix form:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_6 \end{pmatrix} \tag{5.25}$$

where x_1 represents the receptors in the open state and x_2 the ones that are desensitized and $a_{11} = -(r_1[L] + r_2 + r_3)$, $a_{12} = -r_1[L] + r_4$, $a_{21} = r_3 - r_6[L]$, $a_{22} = -(r_4 + r_5 + r_6[L])$, $b_1 = r_1[L]$ and $b_6 = r_6[L]$.

This representation requires to store two floating point variables for each simulated synapse, plus six variables shared among all the synapses that express the same receptor to store the rate constants from which to calculate the coefficients of the equation. Thus memory occupancy requirements are respected. Moreover, this model is also computationally efficient requiring 4 scalar multiplications and 2 scalar additions for the matrix-vector product, plus 8 scalar additions/subtractions and 2 multiplication to compute the coefficients, making a total of 10 scalar multiplications and 10 scalar addition/subtraction for the computation of the derivatives of the variables x_1 and x_2.

[5]see the master equation (A.4) in Kinetic schemes.

Application of the model to AMPA, NMDA and $GABA_A$-mediated currents

Destexhe and colleagues (Destexhe *et al.* [1994]) found that the simple scheme (5.22), or some of its subtypes in which some of the r_i are set to zero, can reasonably reproduce the time course of ionotropic-receptor mediated currents when a 1 mM rectangular pulse of transmitter lasting 1 ms is applied.

For this types of currents the weighting variable s of equation (5.20) is the fraction of open channels at a given time, that is x_1 of equation (5.25).

$GABA_B$receptor

Neuromodulators-mediated transmission, that is the case of $GABA_B$, is based on a complete different mechanism. On one hand there is the activation of the receptors due to the binding with the neurotransmitter, that in turn activates G-proteins. On the other, activated G-proteins interact with potassium ion channels leading them to an open state. Different simplified models can be obtained depending on different working hypothesis:

- the concentration of activated G-proteins ($[G]$) is assumed to be a rectangular pulse of given duration, and the ionic channel is modeled according to a kinetic scheme similarly to ionotropic receptors, but where rate constants depend on $[G]$ rather than on transmitter concentration (Destexhe *et al.* [1994]).

- kinetic equations (5.9) are simplified eliminating one variable (Destexhe *et al.* [1998], Destexhe [1998]). In particular, the desensitized state is drop, and the equations become:

$$[\dot{R}] = K_1[T](1 - [R]) - K_2[R] \tag{5.26a}$$

$$[\dot{G}] = K_3[R] - K_4[G] \tag{5.26b}$$

The fraction of open ion channels represents the weighting variable s in (5.20), that is calculated as

$$s = \frac{[G]^n}{[G]^n + K_d} \tag{5.27}$$

Also equations eqs. (5.26a) and (5.26b) can be put in matrix form like (5.25). Since the number of state variables is always two, the same considerations about memory occupancy and performance we did for ionotropic receptors are valid for this model too.

5.6 Conclusions

Being more flexible than LTI systems and without requiring much more computational time, kinetic models are a good solution for the simulation of the postsynaptic receptor dynamics.

Regarding the presynaptic side, we implemented transmitter release as a square pulse of fixed duration and amplitude, although other more realistic solution exists. We took this decision in order to reduce, for the moment, the number of free parameters of the model while keeping our software able to simulate a functional network. In next versions of the software, the model will be expanded and also presynaptic side-dependent plasticity phenomena will be included, as well as long term plasticity.

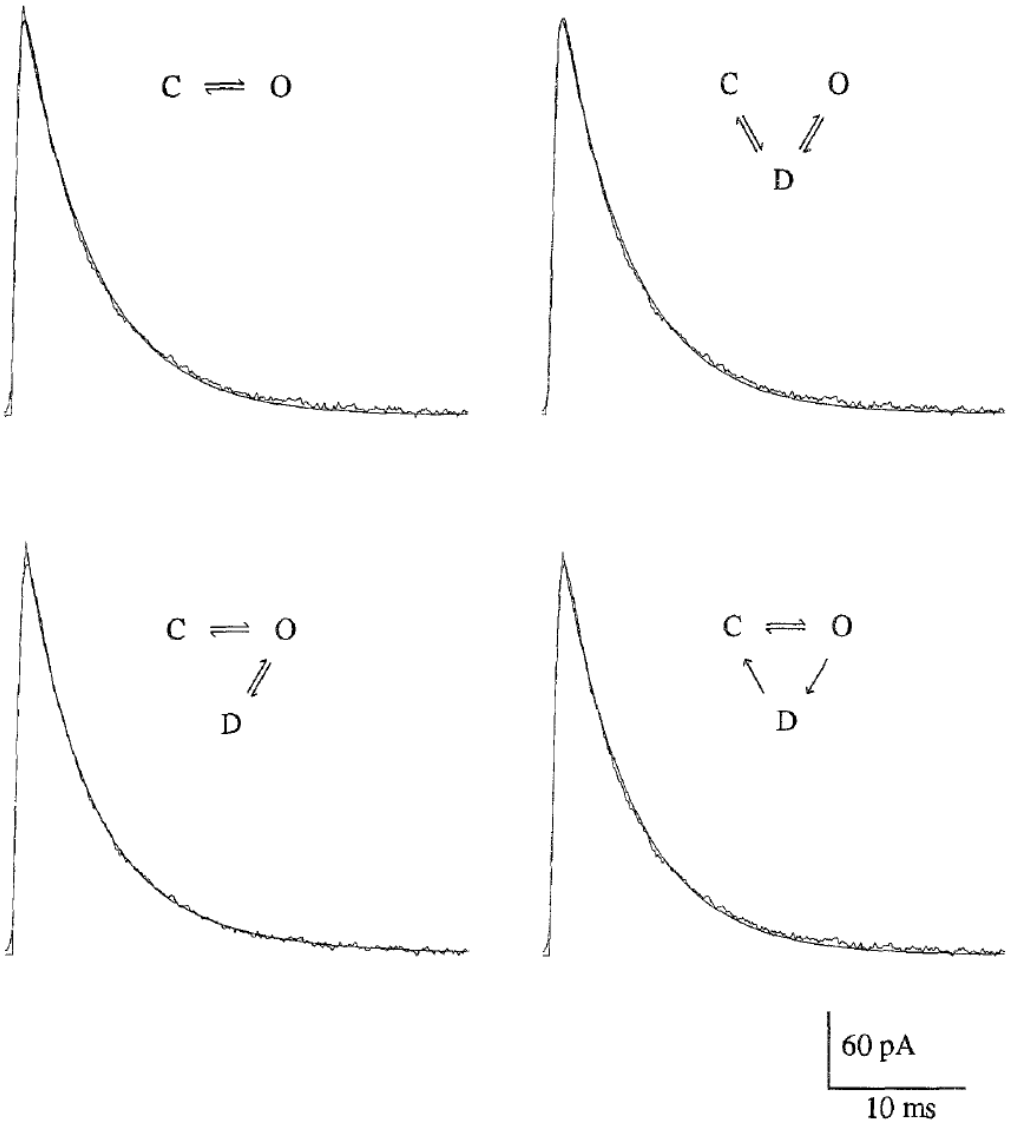


Figure 5.4: Best fits for different simple kinetic schemes (continuous traces) to averaged postsynaptic current mediated by AMPA/kainate receptors (noisy traces) based on a rectangular transmitter pulse as reported in text. Reported from Destexhe *et al.* [1994].

Chapter 6

Design and implementation

6.1 Introduction

In this chapter we will describe how the models previously introduced transformed into software[1]. First of all we will produce a *domain model* of the software, which is an abstract representation of the various entities and their static relationships[2]. We will make an extensive use of class diagrams for this purpose. Then, we will explain implementation strategies.

6.2 Design phase

A neuronal network consists of several neurons interacting each other by mean of synapses. In the general case, a neuron is made of more than one electrical compartment. A synapse takes information from one neuron and propagate it to another one (i.e. it interacts with a specific compartment of the postsynaptic neuron).These concepts are illustrated in Figure 6.1. *Synapses* and *Neuron/-Compartments*, can be analyzed separately, as they play two different roles in a

[1]For introduction on best practices on software design please refer to Gamma *et al.* [1995].

[2]We underline that we are presenting a conceptual view of the software, and that the actual code is not a blind implementation of the abstractions explained here, since further optimizations could be language-dependent and therefore cannot be represented with the *uml* formalism. In particular, not all the structures and concepts of object oriented languages are allowed in CUDA code (for example virtual functions or abstract classes). The actual implementation requires a lower level approach.

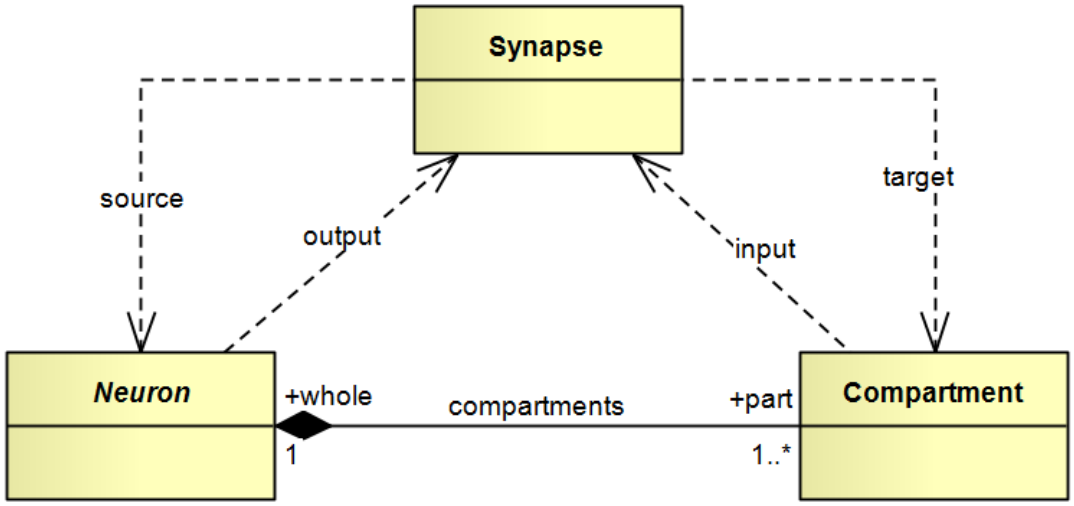


Figure 6.1: Abstraction of a neuronal network.

network, that is *comunication* and *information processing* respectively.

6.2.1 Neurons and compartments

Dendritic tree

Recalling what we said in Chapter 4, section *Multi-compartmental models*, a compartment can be described essentially by mean of three passive electrical parameters and its membrane potential, because we assume that all the ion channels (i.e. what confers active properties to the membrane) are limited to the soma. We do not care about active properties of the membrane for the time being.

The dendritic tree is described by mean of a graph structure, where edges represent compartments and nodes the points where the compartments join. Nodes have the responsibility of averaging the potentials of all the incident edges (see equation (4.52)) and store this information in order to let edges/compartments to calculate the current flowing from/to other compartments. This is done by sharing information through a common buffer where nodes write and edges read. A special case is represented by the cell body, which are both nodes and compartments at the same time (we will consider this aspect in the implementation section). A positive feature of this approach is that no dedicated objects are

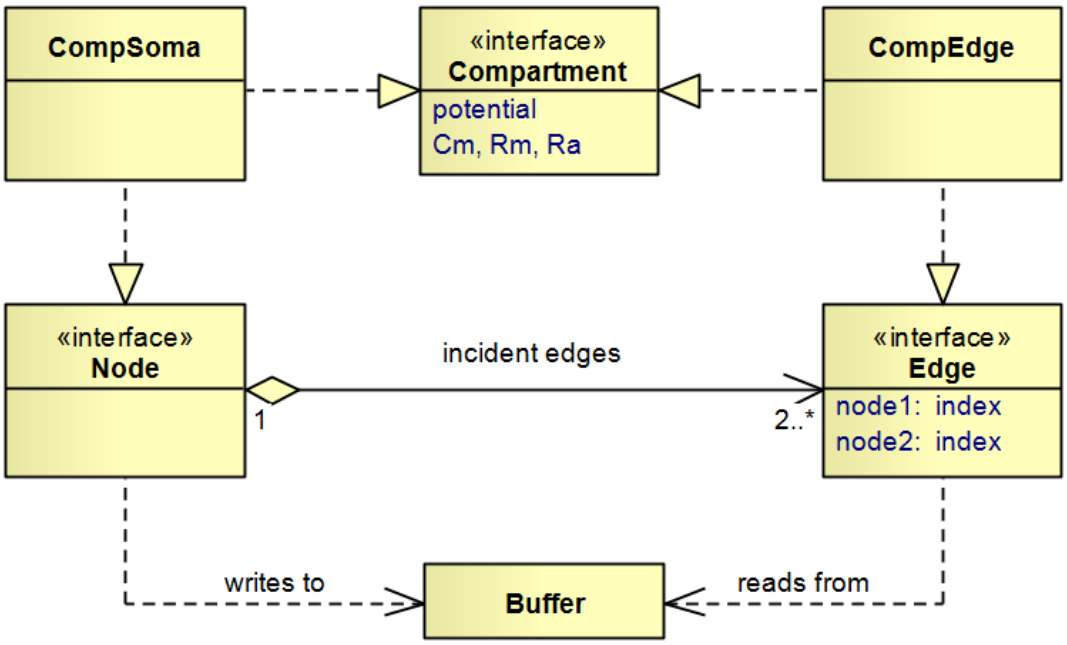


Figure 6.2: Illustration of the logical relationships between node and edges and their corresponding implementations in somata (*CompSoma*) and dendritic compartments (*CompEdge*). Other realizations of the *Node* interface, i.e. all the nodes that are not somata, are not shown. The fields of the *Edge* interface are indexes inside the shared buffer.

required to manage whole neurons, because the particular dendritic structure of any cell is implicitly defined by the runtime relationships between *Edges* and *Nodes*. Moreover, this solution makes network creation easier, because object relations and cross references are minimize. Nevertheless, there are also some drawbacks: being the compartments not aware of their neighbors, there is no concept of "locality", and thus all the information exchange must occur through the slow global memory.

Active compartments

For this first implementation, we hypothesized that dendrites are passive structures. All the active conductances are concentrated in the somatic compartment. Each neuron, then, has one and only one active compartment, simulated using the *adaptative exponential integrate-and-fire* model (equations (4.12, 4.13). This is a temporary solution, as we are planning to extend active behavior also to the dendritic tree.

6.2.2 Synapses

Synapses are junctions where communication between neurons takes place. We distinguish between pre and postsynaptic sites, as they correspond to distinct functional units (see Figure 6.3). The presence of transmitter inside the synaptic cleft is represented by an information flow from *PresynapticBouton* and *PostsynapticDensity*. A concrete implementation may consists of a shared buffer where presynaptic boutons write transmitter concentrations, which are read, in turn, by postsynaptic densities (see Figure 6.5).

Presynaptic side

The process of transmitter releasing is synchronized with presynaptic spikes, or shifted by a fixed amount of time if conduction delay is considered. We made the assumption that all the boutons "separated" from the soma by the same amount of time share the same time courses of the transmitter release, i.e. all the synapses are equal from the presynaptic point of view, and the distinction is made only by mean of a shift in time.

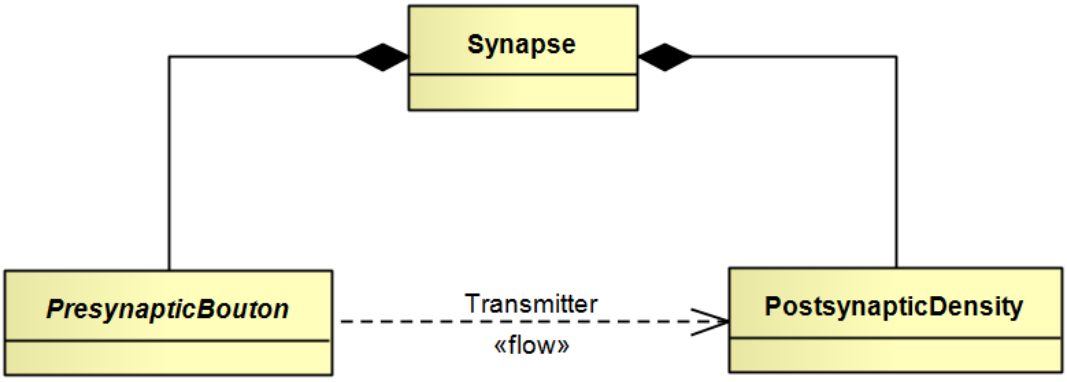


Figure 6.3: Illustration of the two distinct functional units assembling a synapse.

This imply that single synapse dependent phenomena, like stochastic spontaneous release, are neglected, but nevertheless the number of distinct entities is reduced from N (total efferent synapses) to $D_{max}\Delta T^{-1}$ (number of time frames) where D_{max} is the maximum delay and ΔT is the time step of the simulation (at least until $D_{max}\Delta T^{-1} < N$). Considering $D_{max} = 20\,\text{ms}$[3] and $\Delta T = 0.01\,\text{ms}$ we obtain $D_{max}\Delta T^{-1} = 2000$, that is more or less than the estimetad average number of synapses per neuron. This approach is also computationally efficient and, for greater values of Δt, memory cheaper than the representation of single presynaptic boutons. As a matter of facts the updating cost of synaptic boutons can be reduced from $O(N)$ to $O(1)$, if a circular buffer is used to store past history of synaptic activity (see Figure 6.4).

In a circular buffer of size M, the index of the first element – corresponding to the current time step – is any i_0 such that $0 \leq i_0 < M$. The index of a generic frame k in the future, only i_0 must be updated, according to the following rule: $i_0 \leftarrow (i_0 + 1) \text{mod} M$. In this way, no data have to be copied. A two dimensional matrix where each row (or column) is a circular buffer is a *cylindrical array*. In this case, the non circular row index represent, for example, neurons, while the circular column index represents time.

From the presynaptic point of view, we do not represent a physical synapse,

[3]Considering a lower bound of $1\,\text{m}\,\text{s}^{-1}$ for the conduction velocity of the signal, $D_{max} = 20\,\text{ms}$ is equivalent to a covered distance of 2 cm, i.e. we can simulate a cortical area of 4 cm in diameter. If we consider myelinated axons, where conduction velocity is even greater, the size of the simulated network can be further increased.

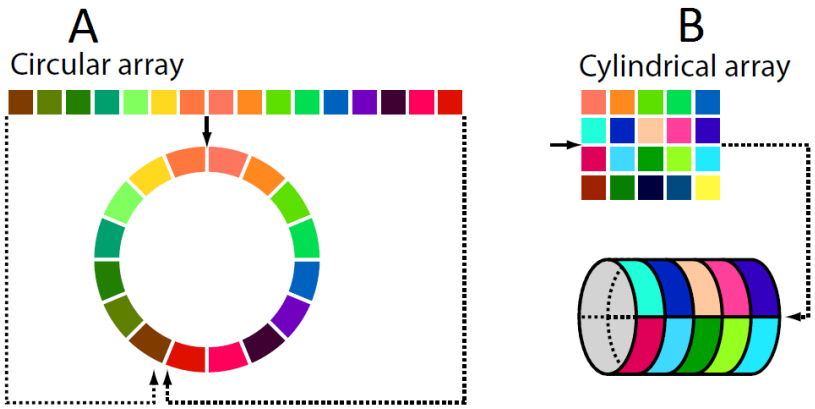


Figure 6.4: Basic structures to store past data. Left: a circular array. Right: a cylindrical array, the generalization of the of a circular array to a two dimensional matrix. Modified from Brette & Goodman [2011].

but rather the concentration of transmitter an hypothetical bouton would release if it was at a certain distance from the soma. How the concentration is actually calculated is of partial importance[4], the only constraint this process has to satisfy is to store the calculated value in the present frame of the buffer.

Postsynaptic side

We assume that each postsynaptic density contains two types of receptors, one mediating a fast synaptic current, whereas the other a slower one[5]. Each receptor complex is characterized by a conductance, a reversal potential for the current and specific kinetics, approximated by a system of two coupled differential equations both in the cases of ionotropic and metabotropic receptor (Chapter 5):

$$\begin{cases} \dot{x} &= F([T])x + G[T] \\ s &= h(x) \end{cases} \tag{6.1}$$

[4]In the present version of the software transmitter release is implemented as a brief rectangular pulse, and all short-term plasticity phenomena are neglected (see Chapter 5, Transmitter release process). However, what we explained in this section holds true even if other solutions are considered.

[5]e.g. AMPA & NMDA for the excitatory synapses, $GABA_A$& $GABA_B$for the inhibitory ones. These are all the receptors we included up to now in our simulation

where $x \in \mathbb{R}^{2\times 1}$ is the state vector, $F([T]) \in \mathbb{R}^{2\times 2}$ and $G \in \mathbb{R}^{2\times 1}$ are matrices that relate the evolution of the state with its present value and the concentration of transmitter $[T]$, and $h : \mathbb{R}^2 \to \mathbb{R}$ is a function that returns the activation of the channel given its present state.

Only two state variables and six additional parameters are needed to describe the evolution of the whole system. However, we assume that any neuron expresses only one receptor of the many available subtypes for each class, and therefore kinetic parameters can be stored only once per neuron. For the same reason, specific conductances and reversal potentials are stored per neuron as well. In this way, only the state must be stored for each synapse, leading to a considerable saving of memory[6].

The output of a receptor is not a current, but, rather, its conductance. Consider, for example, the total synaptic current mediated by $\mathrm{GABA_A}$receptor in a given compartment:

$$I_{GABA_A} = \sum_{j \in GABA_A} i_j = \sum_{j \in GABA_A} [g_j(V - E_{GABA_A})] \tag{6.2}$$

Regardless how each conductance g_j is computed, both V and E_{GABA_A} are constant inside the compartment, then the above expression becomes:

$$I_{GABA_A} = (V - E_{GABA_A}) \sum_{j \in GABA_A} g_j \tag{6.3}$$

i.e. we can perform a cumulative sum of the conductance from the various synapses, and then calculate the current by multiplying the previous result with the appropriate electric potential. Letting N_S be the total number of synapses afferent to a generic compartment and N_R the number of different types of receptors, we reduce the number of scalar multiplication form N_S (that can be large) to N_R (that is 4 in our case).

[6] In this first version of the software that means only four (see note on page 78) sets of kinetic and electrical parameters stored per each neuron, resulting in great memory saving.

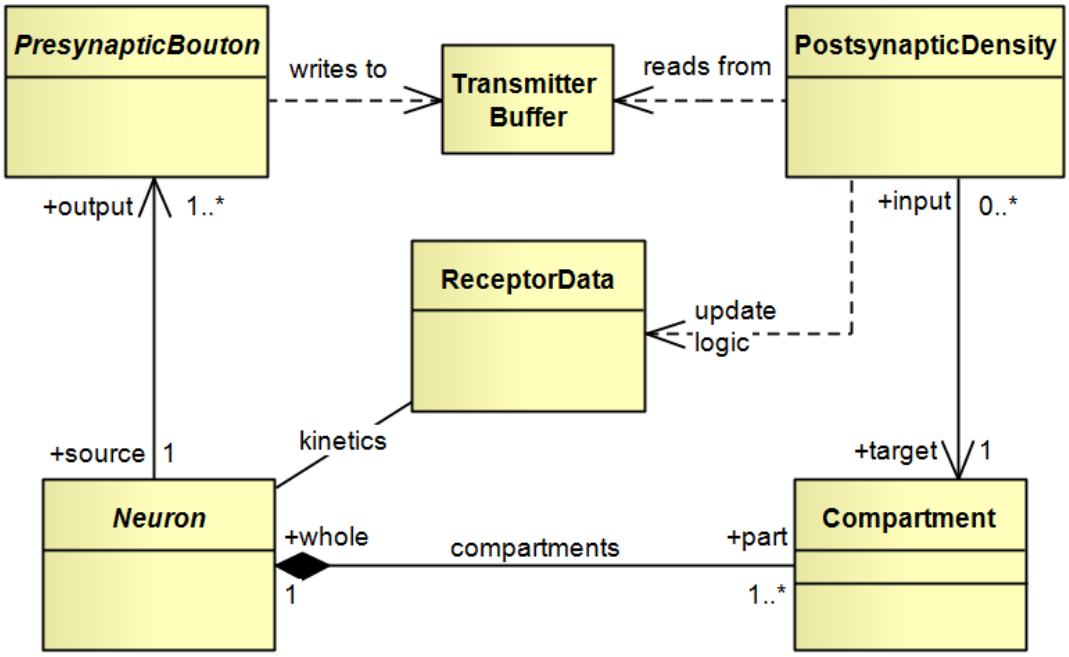


Figure 6.5: Summary diagram for synapses modeling. Each relationship is shown with its own cardinality, for details, see text.

6.2.3 Summary

A summary of what we said about synapses is represented in the diagram shown in Figure 6.5. Boutons are linked to neurons by a many-to-one association, that means their responsibilities can be transferred to neurons, taking care that enough space is reserved for each neuron in order to store all the past time values of the concentration of released transmitter. We could apply a similar concept to compartments and synaptic densities as well, since few sets of kinetic and electrical parameters are shared among the synapses targeting a specific compartment. How the interaction between densities and compartments is implemented depends on the data structure used to represent compartments: with the present design, each compartment store a list of all its afferent synapses.

In this case no pecific object exists to manage neurons as a whole, hence the input for presynaptic boutons is represented by the somatic compartment of the various neurons. Kinetic parameters, shared by the all compartments of the same neuron, can be stored in a look-up table. Each compartment stores an

index referencing the correct entry it needs for its synapses.

6.3 Implementation

Our software, like any CUDA program, is essentially divided in two parts:

- *host code*: this code runs on CPU and is responsible to load data, free memory after usage and coordinate the simulation flow and the main program logic.
- *device code*: this code is executed by the GPU and it is the core of the program, because it implements the models used for the simulation.

The software has been developed using C++ language for host code and C/C++ with CUDA extensions for device code.

6.3.1 Data model

So far we have used high level concepts typical of object oriented programming to explain the design and the program logic, but in CUDA only a restricted subset of C++ features is available. Complex structures like classes must be decomposed in their basic components (i.e. primitive types), all the variables must be stored in big arrays and references must be handled explicitly. The main difficulties concerned the efficient implementation of multicompartmental models, which we are explaining next.

Representation of the dendritic tree

The implementation of the diagram shown in Figure 6.2 is not straightforward, because there are some abstraction relationships. We solved the problem in the following way (illustration in Figure 6.6): edges keep track of their adjacent nodes by mean of an adjacency list (implemented as an array of pair of references), nodes do the same for their incident edges; all the information about electric potentials is stored in one big array; no matter if it is from somas, nodes or dendritic compartments, each type of entity has a reserved space in this array.

Every entity is identified by an index; to access the right memory location in the array of the voltages, this index is offset by an appropriate value. In this way

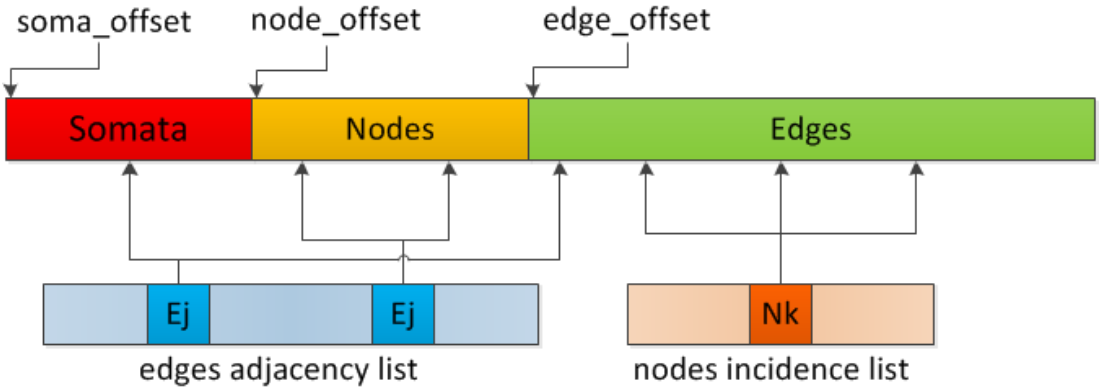


Figure 6.6: Data structures used to represent dedritic trees in memory. The upper rectangle represents the vector of the electric potentials. Each section corresponds to the potentials of a different type of entity. The light blue rectangle is the edges' adjacency list. The light orange one is the nodes' incidence list.

we solved the problem due to the polymorphic behavior of the entities of type *CompSoma* and *CompEdge*.

6.3.2 Memory Analysis

Let N be the number of simulated neurons, M the number of synapses per neuron, k the number of edges per neurons, D the number of delays. The total number of edges is $N_E = kN$, the total number of nodes, in the worst case,is $N_N = (k-1)N$ (the worst case is represented by a cable, where the number of nodes is exactly $k-1$). We used 32 bit floating point representation for real numbers and large integers (1 word). We will perform an upper bound analysis, assuming that all nodes have the same degree equal to a. Table 6.1 reports the memory requirements for every type of entity. Based on the expression of Table 6.1, the total amout of memory required by differeny configurations is shown if Figure 6.7.

Component	Description	Memory requirements (in words)
Soma	dynamics (V, w), incident edges, current, type, transmitter	$(5+a)N + DN$
Edge	potential, adjacent nodes, type	$4kN$
Node	potential, incident edges	$(1+a)(k-1)N$
Synapse	fast and slow dynamics, source, target	$6MN$

Table 6.1: Memory requirements

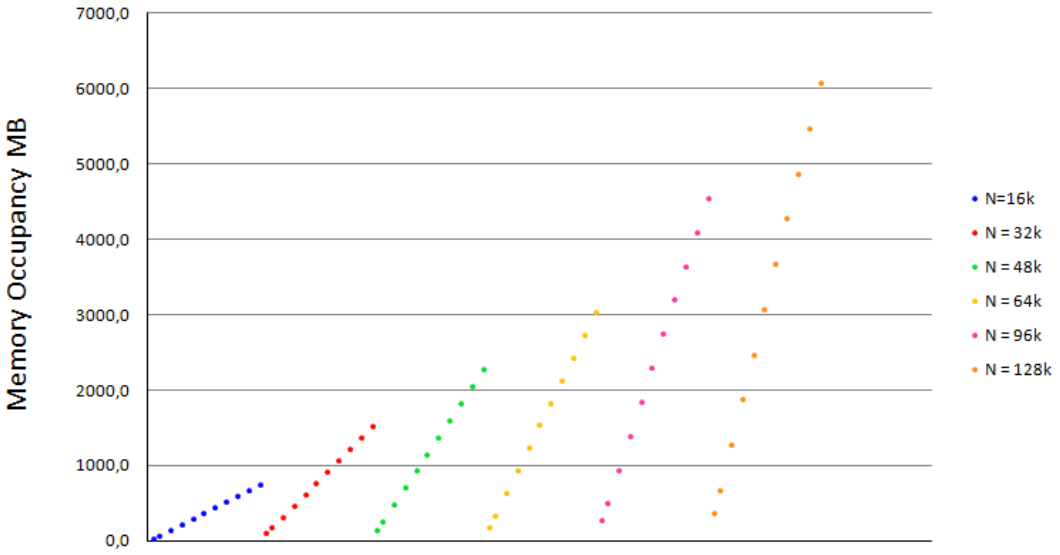


Figure 6.7: Memory requirements for different configurations of the network. Parameters have the following values: $a = 4$, $k = 16$, $D = 1$ (no delays, spike propagation is immediate). Each series correspond to a different configuration: the number of neurons is indicated in the name of the series, while the number of synapses is $M = 100, 200, 400, \ldots, 2000$.

6.3.3 Execution flow

The simulation flow on GPU is shown in Figure 6.8. Each activity represents, more or less, a *kernel* (a function run on GPU):

- *Initialize Network*: performs the initialization of the whole network, setting all dynamic variables to their initial values. After the initialization, the simulation enters the main loop.

- *Update Nodes*: nodes are grouped accordingly to the number of incident edges. All the nodes having the same degree are processed simultaneously, by the kernel *Compute Node Potential*. The potential of each node is calculated according with equation (4.52). For somas, which are nodes with an electric capacity, the total outgoing current is instead computed and stored for later use.

- *Update Edges*: passive compartments are processed in this kernel. It consists of three sub procedures:

  1. the receptors of all the afferent synapses are updated (*Updaqte Synaptic Conductances*); the conductances of the synapses sharing the same dynamics are summed together.
  2. once the have been updated and the total conductance for each receptor is available, synaptic current are calculated (*Compute Synaptic Currents*);
  3. the total outgoing current is calculated (summing the contributes from synapses and from other compartments with the leakage current) and the potential is updated according with the current-voltage relation of a capacitor (see eq. (4.51))

- *Update Somas*: the update phase of the somata is subdivided in the same way as the one of the edges.

- *Update Transmitter Release*: this function updates the concentrations of the transmitter released by the various neurons.

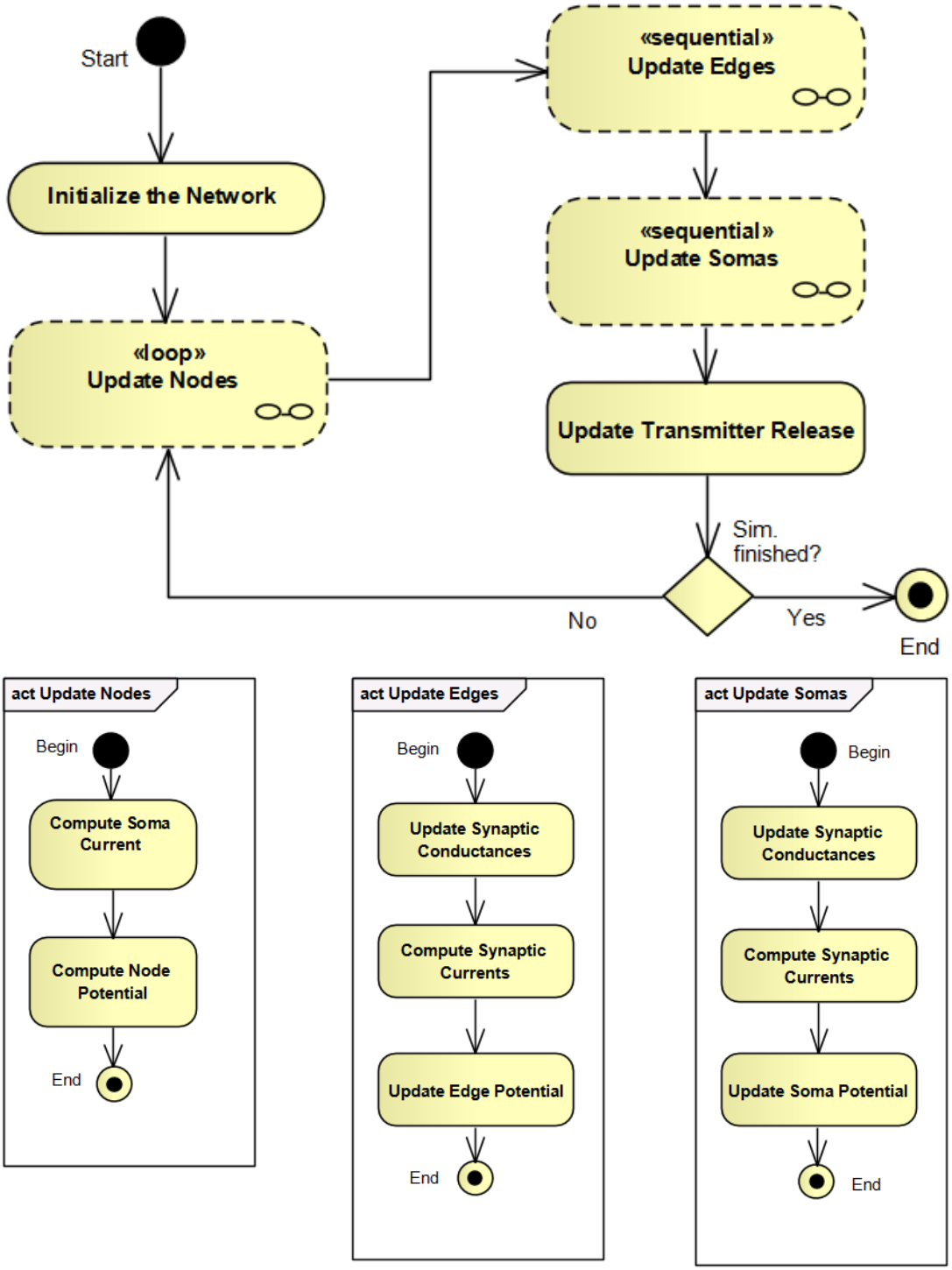


Figure 6.8: Top: flowchart of the simulation using GPU showing various kernels. Each activity roughly correspond to a GPU kernel. Bottom: detailed representation of the activities show in the main flow chart.

6.3.4 Results

We simulated a small-world network (Figure 6.9) of compartmentalized neurons. We chose this topology because the creation of a network with this connection pattern is particularly easy. Each neuron projectes and recives 288 synapses. The number of synapses is a parameter used to create the network. Each neuron had extensively branched and widespread dendrites, for a total of 224 compartments. This chose a very high value because we wanted to test our program in the case of a very high number of compartments.

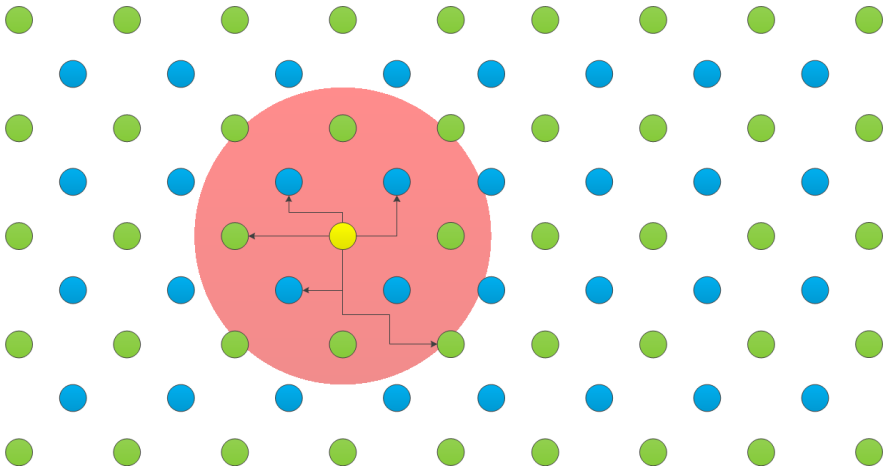
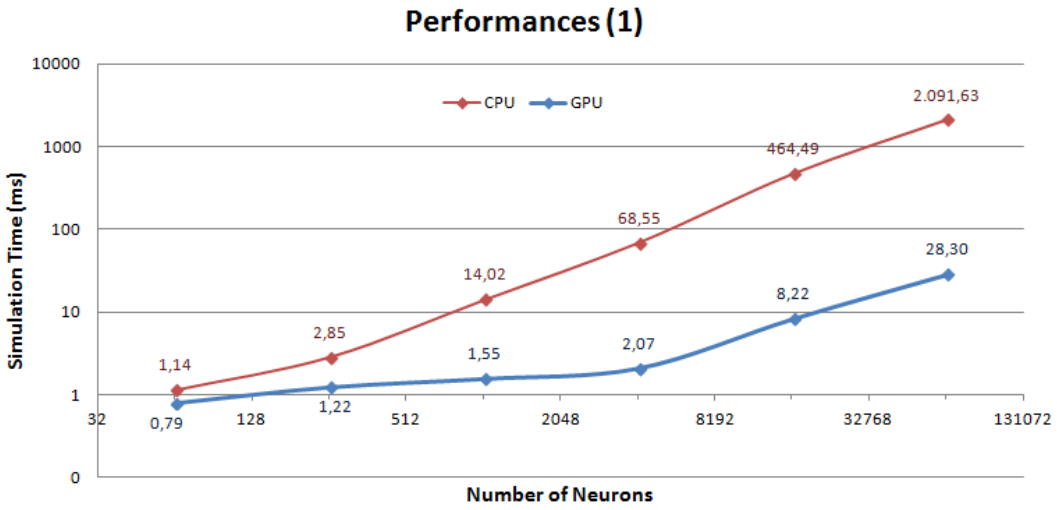
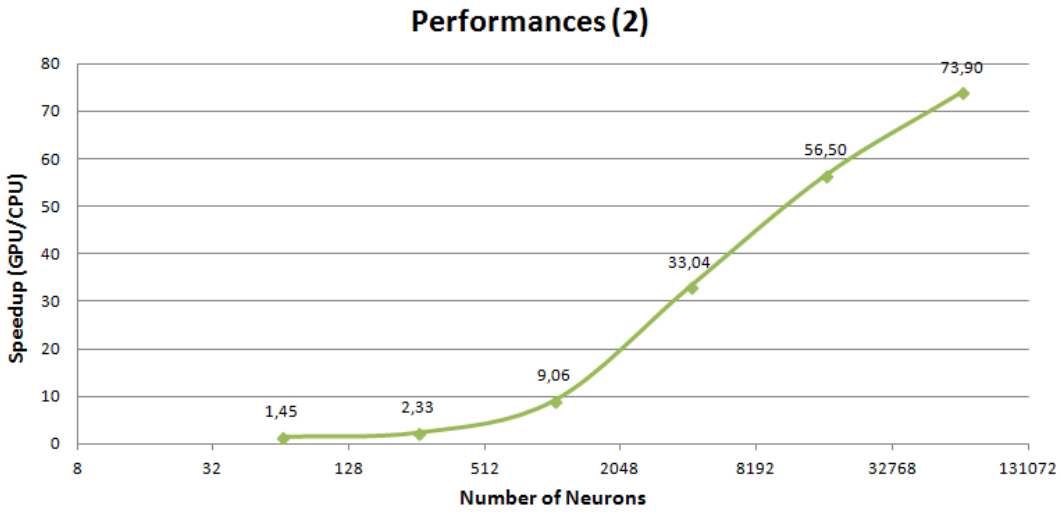


Figure 6.9: Illustration of the small-world topology used in our simulation. Green circles represent excitatory neurons, blue circles inhibitory neurons. Each neuron makes a fixed number of connections (288) with other neurons falling within a given distance: for example, the highlighted excitatory neuron (yellow circle) projects and receives connections to and from the neurons inside the red shaded region.

We created different networks with an increasing number of neurons (detatils in Table 6.2). The simulation was run on a Nvidia Fermi S2050 graphic card (448 core, GPU 1.15 GHz, RAM 2687 MB), an on a personal computer (CPU: Intel®*i5 -3470* 3.20 GHz (single core), RAM 8 GB DDR3). Performances of both the solutions are shown in Figure 6.10. It is clearly noticeable how performances are incredibly sped up exploiting the parallel architecture of a graphic processor.



(a) Comparison of the performances of a classical CPU implementation and of the parallel GPU implementation to simulate one time frame of the same network. Data are in milliseconds.



(b) Speedup of performances exploiting the to the hardware acceleration capabilities of the GPU.

Figure 6.10

# Somas	# Edges	# Nodes	# Synapses
65	14560	1625	18720
260	58240	6500	74880
1040	232960	26000	299520
4160	931840	104000	1198080
16640	3727360	416000	4792320
66560	14909440	1664000	19169280

Table 6.2: Different network configurations tested. The number of every type of entity is reported.

Chapter 7

Conclusions

We realized a flexible neuronal network simulation software, that closely mimics physiology, though maintaining good performances. In particular, also the different morphological aspects of various types of neurons can be cosidered.

It is, however, only the first step towards the realization of a more complete tools for the simulation of large-scale physiologic networks of spiking neurons. In the following pages, we will briefly comment our results, and have a look on future directions of this work.

Critical analysis

The scheme of Figure 6.8 corresponds to a forward Euler integration method, which is stable and precise only if very small time steps are used. More sophisticated integration techniques (which could allow to increase the time step, while maintaining if not increasing the same level of precision) would require to loop several times over the main cycle before advancing to the next time step of simulation. We wondered if it was possible to improve on the simulation with these more sophisticated tools adapting the updating and the looping logic, but keeping the same unchanged design. The answer was negative, and we are going to explain the reasons next.

The present software, as we said when we talk about the design, lacks the concept of "locality", because there are no "objects" that explicitly define and manage the behavior of high level entities (i.e. neurons). Rather, they (neurons) are implicitly defined by the interactions between the various lowest level build-

ing blocks (compartments and synapses), which are unaware of higher levels of abstractions. For this reason, all the intermediate steps, needed if more complex integration strategies are used, must be temporarily stored in global memory[1], with the following consequences:

1. since, in principle, additional data are required for each entity of the simulation, considerably much more memory should be used, the size of the network (number of neurons, compartments and synapses) being the same. It follows that, being the same the total amount of memory available on the graphic card, we are forced to simulate smaller networks.

2. a greater number of slow global memory access is required, probably reducing the overall performances of the simulator.

3. due to the large size of additional data to handle, there are important management issue to solve.

For all these reasons, we decided that changing the integration scheme was not worthwhile.

These considerations made us understand that further improvements are possible only changing the design, including higher level concepts, in order to organize data in such a way that the intermediate steps required by other integration schemes could be executed all together (meaning not simultaneously, but within a single function call, in order to exploit *shared memory*). We have been aware of this possibility from the design phase, but, if we had improved the design to optimize performances, we should also have thought how to instantiate a network according to the new data structures. As a matter of fact, the automatic creation of large-scale networks, problem not treated in this thesis, is not a trivial task, even in the case of the relatively simple data layout we adopted. For this reason, we preferred to realize a simpler but ready to use application, being completely aware of its possible limits, in order to proof the feasibility of our first idea (speeding up simulations using GPUs). Nevertheless, we are planning to improve considerably this software during a future expressly reserved project.

[1]It is not possible to exploit the faster *shared memory*, because compartments processed in the same block of threads are not guaranteed to belong to the same neuron.

Indeed we are already studying new algorithms and implementation strategies for this purpose. We are confident that a stable version will be available soon.

Future perspectives

We are evaluating to include also conductance-based models in future versions of this tools. As a matter of fact, being each ionic current explicitly described, these models allow to study the effect of pharmacological manipulations (i.e. blocking some ion channel) may have on networks' behavior. In the end of Chapter 4, we explained the main limitations about the use of this models in parallel simulations, the simulation of calcium dynamics being the most important. Recently, we found a work of 2008 (Pospischil *et al.* [2008]) where it has been shown that the most important firing patterns can be reproduced using only a limited subset of the known ionic current ($I_{Na,t}$, I_K, I_M, I_T and I_L) and without caring about calcium dynamics, at the cost that the current-frequency relations of simulated neurons are slightly different from real ones. This compromise could be the starting point to enhance our tool with the possibility to simulate even more realistic neuronal networks. We have already begun a new design phase to include these simpler conductance-based models in our simulators.

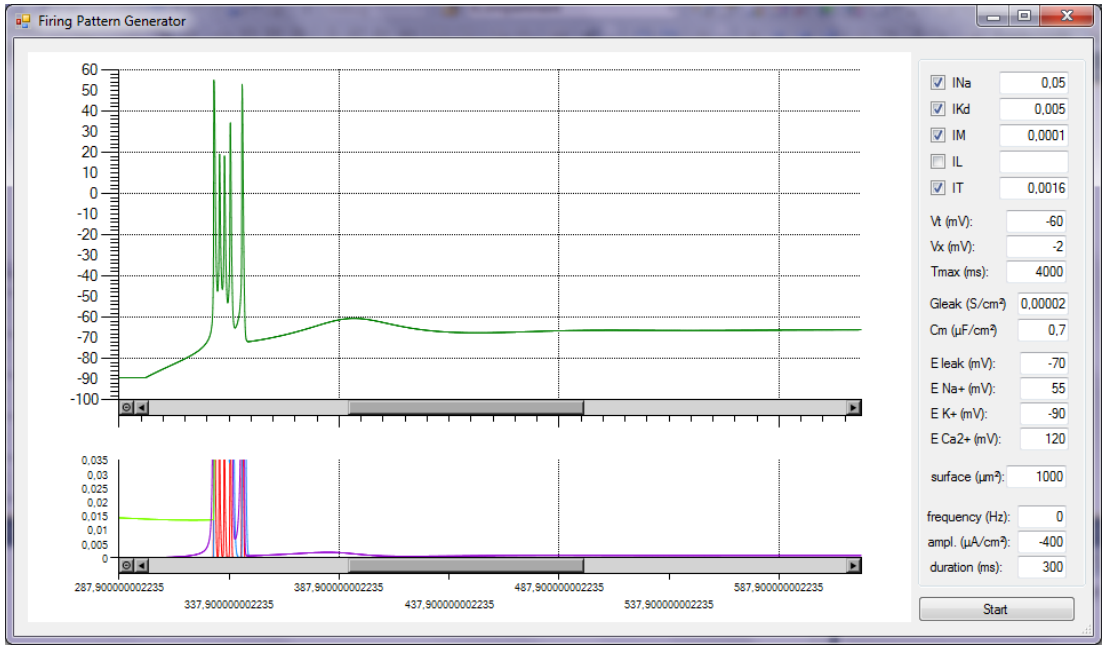


Figure 7.1: Screenshot of a demonstrative stub application to simulate conductance-based neurons. We developed this sample application to test our new design.

Appendix A

Kinetic schemes

We can suppose that a channel protein can assume several conformations, some of which allow ions transition across the protein, and others not. All together these conformations can be represented to the set of states $S_1 \dots S_n$. Conformations of the first type correspond to open states of the channel, the latter to closed or desensitized states. The gating process can be thought as a series of stochastic transitions between states ending with the opening of the channel.

Let $P(S_i, t)$ be the probability of being in state S_i at time t, and $P(S_i \to S_j)$ the transition probability from state S_i to state S_j. It is worth noting that $P(S_i \to S_j)$ might depend on the membrane potential V or the concentration of a ligand molecule $[L]$, according to the gating mechanism of the particular ion channel. Transitions between any couple of states S_i and S_j is described by:

$$S_i \underset{P(S_j \to S_i)}{\overset{P(S_i \to S_j)}{\rightleftharpoons}} S_j \tag{A.1}$$

The time evolution of the the probability of state S_i is described by the *Mater equation* (Colquhoun & Hawkes [1981]):

$$\frac{dP(S_i, t)}{dt} = \sum_{j=1}^{n} P(S_j, t) P(S_j \to S_i) - \sum_{j=1}^{n} P(S_i, t) P(S_i \to S_j) \tag{A.2}$$

The left term represents all the transition entering S_i, while the right term represents the contribution of all the transitions leaving it. The time evolution of the system depends only on its present state and it is entirely defined by knowl-

edge of the set of transition probabilities that, as we said, can be constant or dependent on some channel-extrinsic variable as the membrane potential or the concentration of some messenger molecule.

In the limit of large numbers of channels, the probability of being in a state S_i becomes the fraction of channels, noted s_i, in that particular state, and the transition probabilities form state S_i to state S_j become the rate constant s_{ij}, i.e. the number of channels that move from state S_i to S_j in the unit time. The transition probability between two states becomes a reaction:

$$S_i \underset{r_{ji}}{\overset{r_{ij}}{\rightleftharpoons}} S_j \tag{A.3}$$

and the master equation is rewritten as:

$$\dot{s_i} = \sum_{j=1}^{n} s_j \, r_{ji} - \sum_{j=1}^{n} s_i \, r_{ij} \tag{A.4}$$

which is a conventional kinetic equation describing the evolution of the system.

Appendix B

Ionic currents

- $I_{Na,t}$ Transient sodium current involved in action potential generation. Dominant in axons and cell bodies. Rapidly activates and rapidly inactivates.It is characterized by very fast kinetics: it activates within a few ms and is steeply voltage dependent with half maximal conductance at about $-8\,\mathrm{mV}$. Inactivation has a time constant of 10 – 20 ms.

- $I_{Na,p}$: Persistent and non-inactivating sodium current. Much smaller in amplitude than $I_{Na,t}$. Plays an interesting role in the neuron. Activated by depolarization bringing the membrane potential close to the action potential threshold. Markedly enhances the response to excitation and keep the cell moderately depolarized for extended periods.

- I_T: Low threshold "Transient" calcium current. Rapidly inactivates. Threshold is more negative than $-65\,\mathrm{mV}$. Rhythmic burst firing. Depolarization to $-60\,\mathrm{mV}$ inactivates this current and eliminates the bursting. Reactivated by repolarization.

- I_L: High threshold "Long-lasting" calcium current. Slowly inactivates. Threshold about $-20\,\mathrm{mV}$. Calcium spikes in dendrites. Involved in synaptic transmission.

- I_K: Potassium current activated by strong depolarization. "Delayed rectifier." Repolarizes the membrane after an action potential. Part of the Hodgkin-Huxley model. Common in the CNS and supplemented by other

currents in mammals. Activates at membrane potentials positive to $-40\,\mathrm{mV}$ and strengthens with depolarization. Slowly inactivates. Inactivation complete at about $10\,\mathrm{mV}$. Recovery from inactivation takes seconds. Also passes some other ions at low concentration.

- I_C Potassium current activated by calcium concentration increases within the cell (IL) and very sensitive to membrane potential depolarization. General category $I_{K_{Ca}}$. Plays a role in action potential repolarization and interspike interval. This current produces enhanced repolarization after each action potential. Inactivates quickly upon repolarization.

- I_{AHP}: Slow afterhyperpolarization potassium current (very slow). Sensitive to calcium concentration increases within the cell (IL) and a number of neurotransmitters, but insensitive to membrane potential. General category $I_{K_{Ca}}$. Supports slow adaptation of action potential discharge in the hippocampus and cortex.

- I_A: Transient, inactivating potassium current. Plays a role in action potential repolarization and in delaying onset of firing. Basically, the action potential is delayed until IA shuts down. Activates in response to membrane potentials positive to$-60\,\mathrm{mV}$, but then inactivates rapidly. Reactivates in response to repolarization. Kinetics resemble the fast voltage-dependent sodium inward current.

- I_M: Muscarinic potassium current. Activated by depolarization to about $-65\,\mathrm{mV}$. Noninactivating. Spike frequency adaptation. Quiets the cell after an initial spike.

- I_h: Depolarizing mixed cation (Na^+ and K^+) current activated by hyperpolarization. Rhythmic activities. Slow time course. May control the communication of synaptic inputs to the soma of cortical pyramidal cells.

Appendix C

Firing patterns

Neurons exhibit various types of steady-state responses when they are excited by a depolarizing current step (Connors & Gutnick [1990], Gray & McCormick [1996]):

- *regular spiking (RS)*: this firing pattern is the one most commonly exhibited by the neurons of the cortex, hence its definition of "regular" (Figure C.1). RS neurons respond with a train of action potentials when presented with prolonged stimuli of constant amplitude. Moreover, these neurons exhibit pronounced adaptation of the spike frequency. This firing pattern is almost only exhibited by excitatory pyramidal cells.

- *intrinsically bursting (IB)*: IB neurons are distinguished from RS neurons as their spikes tend to appear in a stereotyped clustered pattern, the burst, which is often the minimal response to a just-threshold intracellular stimulus (Figure C.1). Within a burst, each successive spike usually declines in amplitude. After the burst, the IB cells usually respond with a regular spike train, most of time exhibiting adaptation of the firing frequency. This particular firing pattern is usually found in layer V pyramidal cells.

- *fast spiking (FS)*: individual spikes produced by FS neurons are characterized by their very brief duration, usually lasting less than 0.5 ms. They exhibit little or no adaptation during prolonged intracellular current pulses. Indeed, when strongly stimulated, they can sustain spike frequencies of at

least 500 – 600 Hz for hundreds of milliseconds (Figure C.1). Most inhibitory neurons show this firing pattern.

- *low-threshold spiking (LTS)*: LTS neurons generate adapting trains of action potentials in response to depolarizing current injection, similar to the response of RS cortical neurons. In addition, they generate a burst of action potentials in response to injection of hyperpolarizing current pulses, due to the presence of T-type calcium channels. This behavior is also known as post inhibitory rebound. This firing pattern is shown by many cortical inhibitory neurons[1].

- *complex spiking*: complex spikes are slow, 1 – 3 Hz spikes, characterized by an initial prolonged large-amplitude spike, followed by a high-frequency burst of smaller-amplitude action potentials. This firing pattern is found in the cerebellar Purkinje cells[2].

Neurons can also be classified based on their sensitivity to the amplitude of the applied stimulation:

- *Class 1 excitability*: the strength of the applied stimulation is encoded in the firing rate of the cell. Class 1 excitable neurons have the ability to fire low-frequency spikes when the input is weak (but superthreshold). This kind of excitability is usually associated with pyramidal cells and other types of RS neurons.

- *Class 2 excitability*: if a neuron cannot fire low-frequency spike trains, it is said to be class 2 excitable. That is, it is either quiescent or it fires a train of spikes with a some relatively large frequency (about 40 Hz). It is straightforward that this type of neurons cannot encode the strength of the stimulation in their output. The most representative category of this type of excitability is provided by FS cells.

[1] A similar pattern is also exhibited by thalamic relay neurons, with the difference that they do not present adaptation and their burst are more powerful compared to ones of LTS cells.

[2] Even though the present work mainly focuses on neuronal network of the cerebral cortex, we mention them because, in the future, we intend to extend our software also to model cerebellar circuitry).

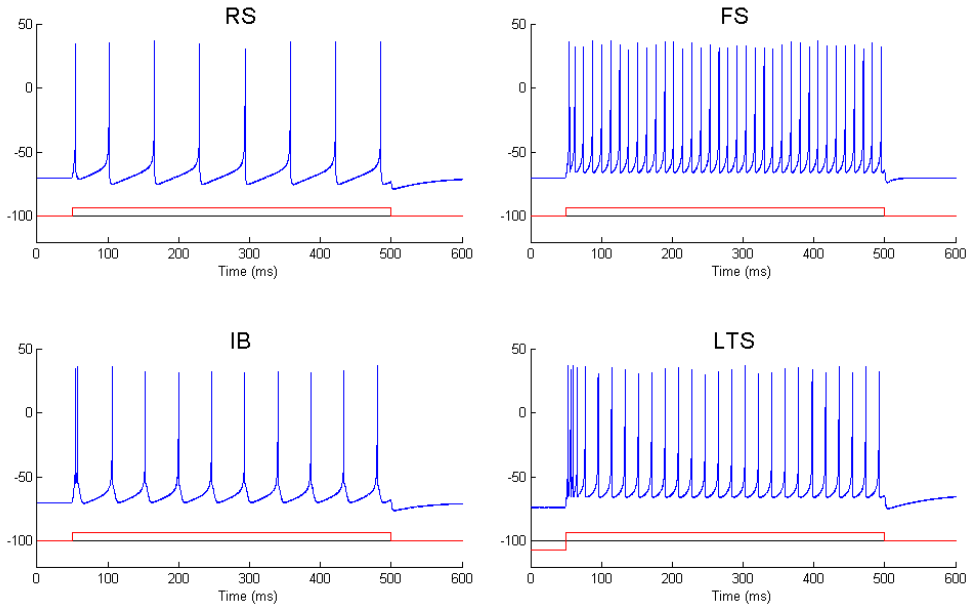


Figure C.1: Most important firing patterns found in neocortical neurons. Blue line are voltage traces, red lines the profile of the injected current. All current steps have the same amplitude.

For a complete review of all the firing patterns and computational properties of neurons, please refer to Izhikevich [2004].

Appendix D

Source code listings

We report the source code of the kernels of our simulator.

```
// some useful typedefs
typedef unsigned int uint;
typedef uint2 uint[2];

/*  KERNEL: ker_updateNodes
  PURPOSE: calculate the averaged potential for each node.
  INPUT:
    nodePotentials: potentials of the nodes
    edgePotentials: potentials of the edges
    incidentEdges:  contains the incident edges of all the nodes
    edgeType: type of the edges, needed to find their electrical parameters
    degree:   number of incident edges
    nNodes:   total number of nodes
    offset:   value needed to find the right info in incidentEdges
*/
__global__ void ker_updateNodes(
              float *nodePotentials,
              float *edgePotentials,
              uint *incidentEdges,
              char *edgeType,
              uint degree,
              uint nNodes,
              uint offset
            )
{
  // find the index of the node
  const uint nodeID = blockDim.x * blockIdx.x + threadIdx.x;
    // if it is greater than #nodes then exit
  if(nodeID >= nNodes) return;

  // perform an average of the potential of the incident edges
  double N = 0, D = 0; // help variables needed to calculate the average
  for(int j = 0; j < degree; j++){
    // find the reference to the j-th incident edge
    uint edgeID = incidentEdges[offset + nodeID + j * nNodes];
    // find th type of the edge
    char type = edgeType[edgeID];
    float G = c_edgeConductance[type];  // axial conductance of the edge
    float V = edgePotentials[edgeID];   // potential of the edge
    // perform a cumulative sum
```

```
    N += G * V;
    D += G;
  }
  // calculate and save the averaged potential of the node
  nodePotentials[nodeID] = N / D;
};

/* KERNEL: ker_updateSomaNodes
  PURPOSE: calculate the dendritic current of each soma
  INPUT:
    somaPotentials: potentials of the somas
    edgePotentials: potentials of the edges
    somaCurrent:  dendritic current
    incidentEdges:  contains the incident edges of all the somas
    edgeType: type of the edges, needed to find their electrical parameters
    degree:   number of incident edges
    nSoma:    total number of somas
    offset:   value needed to find the right info in incidentEdges
*/
__global__ void ker_updateSomaNodes(
                float *somaPotentials,
                float *edgePotentials,
                float *somaCurrent,
                uint *incidentEdges,
                char *edgeType,
                uint degree,
                uint nSomas,
                uint offset
              )
{
  // find the index of the soma
  const uint somaID = blockDim.x * blockIdx.x + threadIdx.x;
    // if it is greater than #somas then exit
  if(somaID >= nSomas) return;

  Vs = somaPotentials[somaID];
  // calculate the amount of current flowing out from the soma
  // towards the dendrites
  double I = 0; // total current
  for(int j = 0; j < degree; j++){
    // find the reference to the j-th incident edge
    uint edgeID = incidentEdges[offset + nodeID + j * nNodes];
    // find th type of the edge
    char type = edgeType[edgeID];
    float G = c_edgeConductance[type];  // axial conductance of the edge
    float V = edgePotentials[edgeID];   // potential of the edge
    // perform a cumulative sum of all the currents
    I += G * (Vs - V);
  }
  // calculate and save the total dendritic current
  somaCurrent[somaID] = I;
};

/* KERNEL: ker_updateEdges
  PURPOSE: update the potential of the edges
  INPUT:
    potentials:   potentials all the objects of the simulation
    edgePotentials: potentials of the edges
    edgeAdjacency:  nodes adjacent to the edges
    edgeSynapses: contains the info on synaptic connections
    synTransmitter: transmitter concentrations
    edgeType: type of the edges, needed to find their electrical parameters
```

```
    nEdges:   total number of edges
*/
__global__ void ker_updateEdges(
                float *potentials,
                float *edgePotentials,
                            uint2 *edgeAdjacency,
                            uint2 *edgeSynapses,
                float *synTransmitter,
                char *edgeType,
                            uint nEdges
                        )
{
    // find the index of the edge
  const long edgeID = blockDim.x * blockIdx.x + threadIdx.x;
    // if it is greater than #edges then exit
  if(edgeID >= nEdges) return;

  // find the electrical properties of the edge
  char type = edgType[edgeID]; // find the type of the edge
  float C = c_edgeCapacity[type];    // mem. capacity of the edge
  float Gl = c_edgeLeakCond[type];   // mem. leak cond. of the edge
  float El = c_edgeLeakPot[type];    // leak rev. potential
  float Ga = c_edgeConductance[type]; // axial conductance of the edge
  float V  = edgePotentials[edgeID];  // potential of the edge

  // calculate current from adjacent nodes
  float Iden = 0;
  uint2 nodeIDs = edgeAdjacency[edgeID];
  Iden += Ga * (V - potentials[nodeIDs[0]]); // current from node 1
  Iden += Ga * (V - potentials[nodeIDs[1]]); // current from node 2

  // calculate current from synapses
  float Isyn = dev_synCurrent(edgeID, V, type, edgeSynapses,
          synTransmitter, nEdges, N_SYNAPSES_PER_EDGE);

  // update the potential of the edge
  edgePotentials[edgeID] = V + DELTA_T * (Gl * (El - V) - Iden - Isyn) / C;
};

/*  KERNEL: ker_updateSomas
  PURPOSE: update the state of the somas
  INPUT:
    somaPotentials: potentials all the objects of the simulation
    somaState:    recovery variables of the somas
    somaCurrent:  dendritic currents of the somas
    somaSynapses: contains the info on synaptic connections
    synTransmitter: transmitter concentrations
    somaSpikeTimes: firing time of each soma
    somaType: type of the soma, needed to find electrical parameters
    nSoma:    total number of edges
    iter:   current time step
*/
__global__ void ker_updateSomas(
                            float *somaPotential,
                            float *somaState,
                            float *somaCurrent,
                            uint2 *somaSynapses,
                float *synTransmitter,
                uint *somaSpikeTimes,
                char *somaType,
                            uint nSoma,
                uint iter
```

```
                        )
{
    // find the index of the soma
  const long somaID = blockDim.x * blockIdx.x + threadIdx.x;
    // if it is greater than #somas then exit
  if(somaID >= nSomas) return;

  char type = somaType[somaID];    // type of the soma
  float V = somaPotential[somaID];  // potential of the soma
  float w = somaState[somaID];     // recovery var. of the soma

  // calculate the total current
  float I = 0;
  I -= somaCurrent[somaID];
  I -= dev_synCurrent(somaID, V, type, somaSynapses,
        synTransmitter, nSoma, N_SYNAPSES_PER_SOMA);

  // update the model and detect a possible spike
  int spike = dev_updateAdEx(type, I, &V, &w);
  if (spike) {
    // if a spike occurred, save the current time step
    somaSpikeTimes[somaID] = iter;
  }

  // save the updated values
  somaPotential[somaID] = V;
  somaState[somaID] = w;
};

/*  KERNEL: ker_updateTransmitter
  PURPOSE: update transmitter concentrations
  INPUT:
    synTransmitter: transmitter concentrations
    somaSpikeTimes: firing time of each soma
    somaType: type of the soma, needed to find electrical parameters
    nSoma:    total number of edges
    iter:   current time step
*/
__global__ void ker_updateTransmitter(
                float *synTransmitter,
                uint *somaSpikeTimes,
                char *somaType,
                uint nSomas,
                uint iter
              )
{
  // find the index of the soma
  const long somaID = blockDim.x * blockIdx.x + threadIdx.x;
    // if it is greater than #somas then exit
  if(somaID >= nSomas) return;

  char type = somaType[somaID];
  float Trel = c_somaReleaseDuration[type];    // duration of the release

  float T;
  // compare the current time with the time of the last spike
  if (iter - somaSpikeTimes[somaID] < Trel / DELTA_T){
    T = c_somaMaxConcentration[type];
  }
  else {
    T = 0;
  }
```

```
    // save the concentration of transmitter
    synTransmitter[somaID] = T;
};
```

References

Aldrich, R., Corey, D. & Stevens, C. (1983). A reinterpretation of mammalian sodium channels gating based on single channel recordings. *Nature*, **306**, 436–441. 39

Badel, L., Lefort, S., Brette, R., Petersen, C., Gerstner, W. & Richardson, M. (2008). Dynamic i-v curves are reliable predictors of naturalistic pyramidal-neuron voltage traces. *J. Neurophysiol.*, **99**, 656–666. 32

Benazilla, F. (1985). Gating of sodium and potassium channels. *J. Membrane Biol.*, **87**, 97–111. 39

Borg-Graham, L. (1991). *Cellular and Molecular Neurobiology: A Practical Approach*, chap. Modeling the nonlinear conductances of excitable membranes. Oxford University Press, New York, NY. 42

Brette, R. & Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.*, **94**, 3637–3642. 31

Brette, R. & Goodman, D. (2011). Vectorised algorithms for spiking neural network simulation. *Neural Comput.*, **23**, 1503–1535. 78

Colquhoun, D. & Hawkes, A. (1981). On the stochastic properties of single ion channels. *Proc. R. Soc. Lond.*, **211**, 205–235. 93

Connors, B. & Gutnick, M. (1990). Intrinsic firing patterns of diverse neocortical neurons. *Trends Neurosci.*, **13**, 99 – 104. 97

D'angelo, E. & Peres, A., eds. (2012). *Fisiologia*. Edi-ermes, Milan, IT. 5

Destexhe, A. (1998). *Computational models of molecular and cellular interactions*, chap. Kinetic models of membrane excitability and synaptic interactions. 70

Destexhe, A. & Huguenard, J. (2000). Nonlinear thermodynamic models of voltage-dependent currents. *J. Comput. Neurosci.*, **9**, 259–270. 40, 41, 45, 46

Destexhe, A. & Sejnowski, T. (1995). G protein activation kinetics and spillover of gamma-aminobutyric acid may account for differences between inhibitory responses in the hippocampus and thalamus. *Proc. Natl. Acad. Sci. USA*, **92**, 9515–9519. 58

Destexhe, A., Mainen, Z. & Sejnowski, T. (1994). Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism. *J. Comput. Neurosci.*, **1**, 195–230. 21, 53, 55, 70, 72

Destexhe, A., Mainen, Z. & Sejnowski, T. (1998). *Methods in neuronal modeling*, chap. Kinetic models of synaptic transmission. MIT Press, Cambridge, MA. 56, 57, 70

Eyring, H. (1935). The activated complex and the absolute rate of chemical reactions. *Chemical Reviews*, **17**, 65–77. 41

Fitzhugh, R. (1961). Impulses and physiological states in models of nerve membrane. *Biophys. J.*, **1**, 445466. 29

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 73

Gray, C. & McCormick, D. (1996). Chattering cells: superficial pyramidal neurons contributing to the generation of synchronous oscillations in the visual cortex. *Science*, **274**, 109–113. 97

Hartshorne, B.K.R., Talvenheimo, J., Catterall, W. & Montal, M. (1986). Sodium channels in planar lipid bilayers. channel gating kinetics of purified sodium channels modified by batrachotoxin. *The Journal of General Physiology*, **88**, 1–23. 43

Hestrin., S. (1993). Different glutamate receptor channels mediate fast excitatory synaptic currents in inhibitory and excitatory cortical neurons. *Neuron*, **11**, 1083–1091. 54

Hindmarsh, J.L. & Rose, R.M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proc. R. Soc. Lond. B*, **221**, 87–102. 29

Hodgkin, A. & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, **117**, 500 – 544. 37, 42

Izhikevich, E. (2003). Simple model of spiking neurons. *IEEE T. Neural Networks*, **14**, 1569 – 1572. 30

Izhikevich, E. (2004). Which model to use for cortical spiking neurons? *IEEE T. Neural Networks*, **15**, 1063 –1070. 29, 30, 99

Izhikevich, E. (2008). *Dynamical systems in neuroscience: the geometry of excitability and bursting*. MIT Press, Cambridge, MA. 51

Izhikevich, E. & Edelman, G. (2008). Large-scale model of mammalian thalamocortical systems. *P. Natl. Acad. Sci.*, **105**, 3593–3598. 1, 3

Jahr, C. & Stevens, C. (1990). Voltage dependence of nmda-activated macroscopic conductances predicted by single-channel kinetics. *J. Neurosci.*, **10**, 3178–3182. 56

Nageswaran, J., Dutt, N., Krichmar, J., Nicolau, A. & Veidenbaum, A. (2009). A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural Networks*, **22**, 791 – 800. 2

Naud, R., Marcille, N., Clopath, C. & Gerstner, W. (2008). Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, **99**, 335–347. 31

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frgnac, Y., Markram, H. & Destexhe, A. (2008). Minimal hodgkinhuxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, **99**, 427–441. 91

Rall, W. (1962). Electrophysiology of a dendritic neuron model. *Biophys J.*, **2**. 47

Reuveni, I., Friedman, A., Amitai, Y. & Gutnick, M. (1993). Stepwise repolarization from ca2+ plateaus in neocortical pyramidal cells: evidence for nonhomogeneous distribution of hva ca2+ channels in dendrites. *J. Neurosci*, **13**, 46094621. 40

Rushton, W. (1951). Theory of the effects of fibre size in medullated nerve. *J. Physiol.*, **115**, 101–122. 19

Standley, C., Norris, T., Ramsey, R. & Usherwood, P. (1993). Gating kinetics of the quisqualate-sensitive glutamate receptor of locust muscle studied using agonist concentration jumps and computer simulations. *Biophys. J.*, **65**, 13791386. 55

Thomson, A. & Bannister, A. (2003). Interlaminar connections in the neocortex. *Cereb. Cortex*, **13**, 5–14. 1

Traub, R. & Miles, R., eds. (1991). *Neuronal networks of the Hippocampus*. Cambridge University Press, Cambridge. 40

Yamada, W. & Zucker, R. (1992). Time course of transmitter release calculated from simulations of a calcium diffusion model. *Biophys. J.*, **61**, 671 – 682. 53

Yamada, W., Koch, C. & Adams, P. (1989). In C. C. Koch & I. Segev, eds., *Methods in neuronal modeling*, chap. Multiple channels and calcium dynamics, 97–133, MIT Press, Cambridge, MA, USA. 37, 40

Yudanov, D., Shaaban, M., Melton, R. & Reznik, L. (2010). Gpu-based simulation of spiking neural networks with real-time performance amp; high accuracy. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, 1 –8. 2