

UNIVERSITY OF MANCHESTER

Models of Neurons and Neuronal Networks

by

Wonkwon Lee

BSc in Computer Science and Mathematics
Dr Eva M. Navarro-Lpez

Department of Computer Science

May 2018

UNIVERSITY OF MANCHESTER

Abstract

Department of Computer Science

BSc in Computer Science and Mathematics

by Wonkwon Lee

Despite its size, a neuron is a very complicated entity; without sufficient biological and mathematical background knowledge, simulations of the neuronal network is out of reach. Therefore, the aim of the thesis is to implement networks of biologically accurate neuron models within computationally feasible cost and to simulate and analyse various electrophysiological behaviours including firing patterns of spiking neurons and the emergence of synchronisation. As a result, several mathematical models of single neurons and networks of neurons were explored and simulated. In addition to this, the synchronisation property of a network of neurons was analysed in the presence and absence of the background stimuli.

Contents

Abstract	i
List of Figures	iv
List of Tables	vi
1 Introduction	1
1.1 Motivation	1
1.2 Scope of the problem	1
1.3 Objectives	2
1.4 Research methodology	2
1.5 Summary of the results	3
1.6 Outline of the thesis	3
2 Neurons and Models of Neuron	4
2.1 Introduction	4
2.2 What is a neuron	4
2.3 Conductance-based neuron model	5
2.4 Integrate-and-fire models	7
2.4.1 Leaky integrate-and-fire model	8
2.4.2 Quadratic integrate-and-fire model	8
2.4.3 Adaptive Exponential integrate-and-fire model	10
3 Models of Neuronal Networks	15
3.1 Introduction	15
3.2 Implementation of neuronal network	15
3.3 Modelling network connections	16
3.3.1 Fully connected homogeneous network	16
3.3.2 Randomly connected network	17
3.4 Modelling interacting populations of neurons	18
4 Simulation Results and Analysis	21
4.1 Introduction	21
4.2 Software tools	21
4.3 Emergence of Synchronisation	23

Contents

5 Conclusion and Reflection	26
Acknowledgements	28
Bibliography	29

List of Figures

1	Structure of a basic biological neuron [1].	5
2	Hodgkin-Huxley Model based on giant squid axon [2].	6
3	Multiple firing patterns in neurons with a step current [3].	11
4	AdEx neuron model simulation with a step input current Top: Weak input current, no spike. Bottom: Strong input current, regular spiking after initial bursting.	12
5	Simulation of firing patterns of AdEx neuron model. A: Regular spiking. Middle : Bursting. Bottom: Fast spiking.	13
6	Python simulation of AdEx neuron firing patterns. Top: Regular spiking. Middle : Bursting. Bottom: Fast spiking.	14
7	Schematic representation of the neuronal network where neurons are connected with a probability p [4].	16
8	Structure of Brunel network: an excitatory population, an inhibitory population, and an external population are interacting [3].	18
9	Simulation of Brunel network of sparsely connected LIF neurons. Top: A network of 100 excitatory neurons and 25 inhibitory neurons. Bottom: A network of 2000 excitatory neurons and 100 inhibitory neurons.	20
10	Simulation of Brunel network of sparsely connected QIF neurons. Top: A network of 100 excitatory neurons and 25 inhibitory neurons. Bottom: A network of 2000 excitatory neurons and 100 inhibitory neurons.	20
11	Python GUI design for Neuronal Network Simulation.	22

List of Tables

- 12 Simulation of Brunel network of sparsely connected LIF neurons with re-
current feedback. Initial membrane potential of each neuron is randomly
selected. **Top:** Simulation for whole simulation time. **Middle:** Initial
50ms of simulation. **Bottom:** Last 50ms of simulation. 24

13 Simulation of Brunel network of sparsely connected LIF neurons in the
absence of recurrent feedback. Initial membrane potential of each neu-
ron is randomly selected. **Top:** Simulation for whole simulation time.
Middle: Initial 50ms of simulation. **Bottom:** Last 50ms of simulation. . 25

List of Tables

1	Parameters for the Hodgkin-Huxley equations [3].	7
2	Parameters for the QIF model	9
3	Parameters for Izhikevich's QIF model [5]	9
4	Parameters for AdEx model [6]	10
5	Parameters for multiple firing patterns in neurons [6]	11
6	Parameters for Brunel's recurrent network model [7].	19

Chapter 1

Introduction

1.1 Motivation

The synergy of integrating computer science with biological neuron model has led neuroscience to a non-conventional dimension: computational neuroscience. Computational neuroscience is the field of study in which interdisciplinary approaches from electrical engineering, computer science, physics, and even more are used collectively to investigate how the nervous system processes information. The mathematical models are closely intertwined with the biological models, so researchers are capable of translating biological facts into mathematical truths and vice versa.

The aim of the thesis is to implement networks of biologically accurate neuron models within the computationally feasible cost, and thus simulate and analyse firing patterns and the synchronisation across networks.

1.2 Scope of the problem

As the broad title indicates, this project was not limited to specific phenomena in the brain; in fact, the project could take as many directions and complexity as needed. The focus of this research was to study various nonlinear physiological aspects found in networks of neurons and to identify the synchronisation patterns in a complex network. For this purpose, research was conducted into simulating neuronal networks and analysing different behaviours along with using an existing Python simulator program.

The research began with investigating different neuron models; each of them was customised to highlight specific behavioural traits or to remove such variable factors. After investigating neuron models, several approaches concerned with constructing a network

of neurons were investigated; e.g. how to connect the neuron models together, or how to construct a synapse model? Then network models of thousands of neurons in both excitatory and inhibitory states were implemented. To facilitate simulation and dynamical analysis of networks, a GUI network simulator was implemented. Finally differing dynamical behaviours in networks were studied; the author concentrated on investigating varying firing patterns and synchronisation properties seen in a complex network.

1.3 Objectives

The research was aimed to achieve the following objectives:

- To understand the mathematical models of integrate-and-fire neurons.
- To understand the populations of neurons.
- To analyse multiple dynamical behaviours of the neuronal network model along with Python GUI simulator.
- To investigate the network connectivity and coupling schemes.
- To analyse the synchronisation of network and varying electrophysiological behaviours within the brain.

This project sought to effectively extend Wulfram Gerstner and Romain Brette's mathematical neuron model, to simulate how neuronal connection influences the synchronisation of networks and firing patterns of models.

1.4 Research methodology

To achieve the above objectives, the research was developed in the following three phases:

1. The first phase was to investigate neuron dynamics by probing and simulating mathematical models of a single neuron. Different integrate-and-fire neuron models were studied to reproduce varying dynamical behaviours of a biological neuron, especially firing patterns.
2. The next phase was to extend this project to the populations of neurons by exploring and simulating mathematical models of networks of neurons. Primarily the focus was to investigate dynamical behaviours of neuronal networks observed to regulate the connection strength between neurons to form a complex network.

3. The final phase was to extend this project to further investigate the synchronisation properties in networks of neurons and to explore various firing patterns by integrating a dynamical synapse model to the network.

In order to gain a deeper understanding of single neuronal behaviours and behaviours in neural networks, the author has studied through academic books, research papers, and other supplementary reading. The mathematical background for the research was studied through [8], and the biological understanding was supported by a book *Neuronal Dynamics* written by Wulfram Gerstner and Werner M. Kistler [3].

1.5 Summary of the results

The results described in this thesis can be separated into finding the three main simulations:

1. Simulation of single neurons: different neuron models and neuronal behaviours were explored.
2. Simulation of a network of neurons: network connectivity and neuronal population activities were explored
3. Simulation of a recurrent network of neurons: the emergence of synchronisation was explored.

1.6 Outline of the thesis

The remainder of this thesis explains the research done as follows. Chapter 2 introduces neurons, the biological and mathematical models of neurons explored, and the simulations carried out to study single neuron behaviour. Chapter 3 discusses challenges in the implementation of network connectivity and interacting populations of neurons. Finally, Chapter 4 discusses the main results of the paper, through analysis of the network simulations.

Chapter 2

Neurons and Models of Neuron

2.1 Introduction

This chapter will introduce biological background behind a neuron and further discuss a variety of mathematical neuron models. The purpose of this research is to investigate different models of a neuron and to seek the most appropriate model to be implemented as a network of neurons.

2.2 What is a neuron

A nerve cell, or neuron, is the most significant concept in neuroscience. The human nervous system contains approximately 100 billion neurons and these unique cells transmit information in the form of electrical and chemical signals over long distances. Each neuron receives electrical signals via specialised connections called synapses. Neurons are known to be connected with approximately 10^{14} - 10^{15} synapses. These biological neurons communicate by generating and propagating electrical pulses.

A typical neuron receives input from more than 10,000 other neurons through connected synapses. The inputs produce electrical trans-membrane currents which change the membrane potential of the cell. Large synaptic currents produce significant postsynaptic potentials which are amplified and lead to the generation of electrical pulse called an action potential, or spike. In other words, a spike is the result of abrupt and transient changes in membrane voltage. Neuronal spiking is the means of communication between neurons, and neuron fires as a consequence of incoming spikes from other neurons [9].

A single biological neuron consists of three functionally distinct parts, called dendrites, soma, and axon; see Figure 1. The dendrites perform as an input device that accumulates

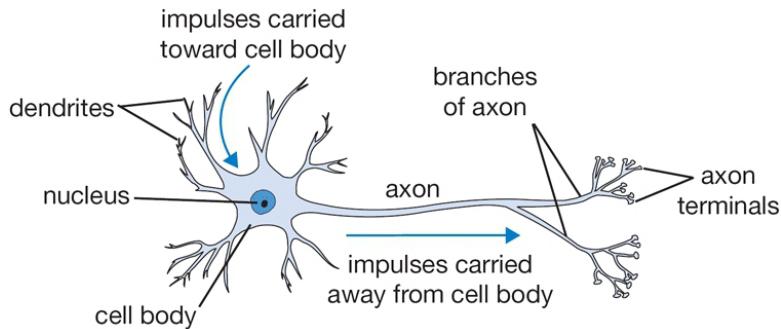


FIGURE 1: Structure of a basic biological neuron [1].

signals from other neurons and transmits them to the soma. The soma is the central processing unit: If the total input exceeds the firing threshold, then an output signal is generated. The output signal is then carried by the axon and is transmitted to other neurons via the synapses [10]. It is common to refer to the sending neuron as the presynaptic neuron and to the receiving neuron as the postsynaptic neuron.

When a spike input arrives, the potential changes and slowly decays back to the resting potential. There are two different types of neurons: If the change is positive i.e. increasing the membrane potential of the postsynaptic neuron, it is categorised as excitatory and if the change is negative i.e. decreasing the membrane potential of the postsynaptic neuron, it is inhibitory. A typical action potential or spike is a short pulse of $1 - 2\text{ms}$ duration with an amplitude of about 100mV . At rest, the cell membrane potential is around -65mV [10].

Mathematically, a neuron model can be described as a dynamical system: a system in which a set of time-dependent variables represents the state of the system over a period of time. There are various models of neurons with varying complexities for distinct purposes from combining utmost variables to make the dynamical system biologically more accurate to limiting the number of variables to concentrate on specific dynamical behaviours in the system.

2.3 Conductance-based neuron model

Electrical activity in neurons is generated and propagated via ionic currents. If an input current $I(t)$ is injected into the neuron, it either continues charging on the capacitor or leaks through the channels in the cell membrane. The active ion transmembrane transport creates differing concentrations. The difference in ion concentrations builds electrochemical gradients across the cell membranes [10]. Most ionic currents across the cell membranes involve with sodium (Na^+), potassium (K^+), calcium (Ca^-) and

chloride (Cl^-) ions. The extracellular medium has high concentrations of Na^+ , Cl^- , and a relatively high concentration of Ca^+ , while the intercellular medium has high concentration of K^+ [9].

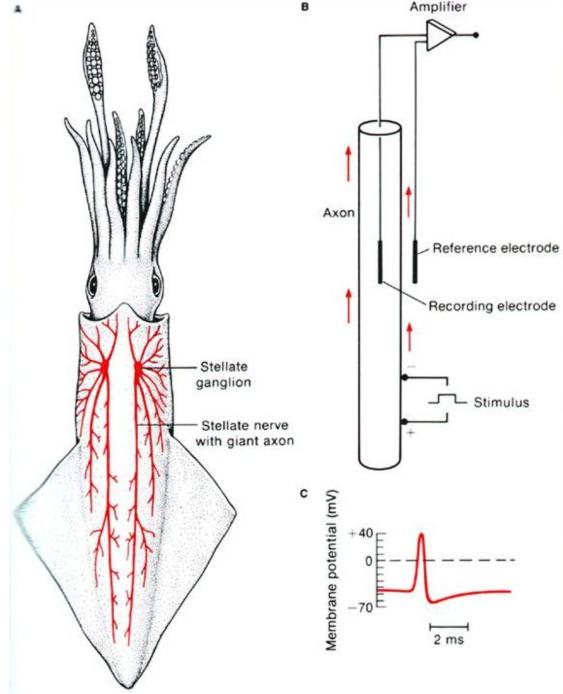


FIGURE 2: Hodgkin-Huxley Model based on giant squid axon [2].

One of the most renowned neuron models in computational neuroscience is the **Hodgkin-Huxley neuron model**, or conductance-based model. The Hodgkin-Huxley model of the squid giant axon in Figure 2 is a four-dimensional dynamical system described by the membrane potential, v , and gating variables n, m, h , respectively for K^+ current, Na^+ current and the leak current, carried mainly by Cl^- ions.

$$\begin{aligned} C \frac{dv}{dt} &= I_{\text{stim}} - I_{\text{ionic}}, \\ I_{\text{ionic}} &= g_{\text{Na}} m^3 h (v - E_{\text{Na}}) + g_{\text{K}} n^4 (v - E_{\text{K}}) + g_L (v - E_L). \end{aligned} \quad (1)$$

where

$$\begin{aligned} \frac{dn}{dt} &= \alpha_n(v)(1-n) - \beta_n(v)n, \\ \frac{dm}{dt} &= \alpha_m(v)(1-m) - \beta_m(v)m, \\ \frac{dh}{dt} &= \alpha_h(v)(1-h) - \beta_h(v)h \end{aligned} \quad (2)$$

C represents the membrane capacitance, v the membrane potential, K^+ , Na^+ ionic currents, and the leak current, I_L . E_{Na} , E_K , E_L represent equilibrium potentials, g_{Na} , g_{Na} conductance, and n, m, h gating variables with respect to v . The constants α_n , α_m , α_h are experimental values derived from the results of Hodgkin and Huxley experiment. Later studies reveals that these constants depend on the type of neuron in study [11].

Parameter (x)	$E_x(\text{mV})$	$g_x(\text{mS/cm}^2)$
Na	55	40
K	-77	35
L	-65	0.3
Parameter (x)	$\alpha_x(\text{ms}^{-1})$	$\beta_x(\text{ms}^{-1})$
n	$0.02(u - 25) / [1 - e^{-(u-25)/9}]$	$-0.002(u - 25) / [1 - e^{(u-25)/9}]$
m	$0.182(u + 35) / [1 - e^{-(u+35)/9}]$	$-0.124(u + 35) / [1 - e^{(u+35)/9}]$
h	$0.25 e^{-(u+90)/12}$	$0.25 e^{(u+62)/6} / e^{(u+90)/12}$

TABLE 1: Parameters for the Hodgkin-Huxley equations [3].

Despite the Hodgkin-Huxley neuron model is capable of reproducing various neuronal behaviours, it is computationally overly expensive on an extension to a network of neurons. Moreover, not only it is practically impossible to construct a neuron model with all variables but also it is challenging to efficiently analyse a system with excessive variables. In contrast to physiologically accurate but computationally costly neuron models like the HodgkinHuxley model, the integrate-and-fire neuron models limit the number of confounding variables to narrow down the scope of studies. From now on, different models of the integrate-and-fire neuron will be introduced.

2.4 Integrate-and-fire models

Detailed conductance-based neuron models can reproduce varying electrophysiological dynamics to a high degree of accuracy, but these models are difficult to analyse due to their intrinsic complexities. Thus, simple integrate-and-fire neuron models are widely used for studies of neural coding, memory, and network dynamics [10]. Complex integrate-and-fire neuron models are capable of expanding and reproducing a wide variety of different neuron spiking behaviours. The term integrate-and-fire describes how these neuron models perform. When an input current is applied to the system, the membrane potential, v , increases with time t until it reaches a threshold. At the threshold voltage, the neuron *fires*, and then resets to its resting voltage. Several integrate-and-fire models will be introduced starting from linear leaky integrate-and-fire neuron to complex adaptive exponential integrate-and-fire model in order of increasing complexity.

2.4.1 Leaky integrate-and-fire model

The basic circuit of a Leaky Integrate-and-Fire neuron model (LIF) is composed of a capacitor C in parallel with a resistor R driven by a current $I(t)$. The driving current is the sum of two currents, I_R and I_C . Then the driving current can be calculated by using the definition of capacity and the Ohm's law:

$$I(t) = \frac{v(t)}{R} + C \frac{dv}{dt} \quad (3)$$

Multiplying Equation 3 by R and introducing the time constant $\tau_m = RC$ yields the standard form

$$\tau_m \frac{dv}{dt} = -v(t) + RI(t), \quad (4)$$

$$\text{if } v > v_{thres}, \text{ then } v \leftarrow v_{reset} \quad (5)$$

Equation 4 describes the membrane potential, v , at time t with respect to the membrane time constant τ_m and the membrane resistance R . The leaky integrate-and-fire neuron model is typically incorporated with an absolute refractory period. If v reaches the threshold at time t , the system interrupts the dynamics during an absolute refractory period Δ_{abs} and restart the integration at time $t + \Delta_{abs}$ [3]. The leaky integrate-and-fire neuron model is a linear function model and therefore is not appropriate to fit a biological neuron behaviour.

2.4.2 Quadratic integrate-and-fire model

The Quadratic Integrate-and-Fire neuron model (QIF) is a commonly used nonlinear neuron model. The nonlinear curve of the quadratic integrate-and-fire model fits better to the Hodgkin-Huxley neuron model computationally less costly. The QIF model is derived from using the bifurcation theory and normal reduction form [5].

$$\tau_m \frac{dv}{dt} = v^2(t) + RI(t), \quad (6)$$

$$\text{if } v > v_{thres}, \text{ then } v \leftarrow v_{reset} \quad (7)$$

Although the quadratic integrate-and-fire model is computationally more efficient than the Hodgkin-Huxley model to be used in a large scale network, the model cannot produce

Parameter	Description	Value
v_{thres}	Firing threshold	$-30mV$
v_{reset}	Reset voltage	$-65mV$
τ_m	Membrane time constant	$12ms$
R	Membrane resistance	$20M\Omega$
I	Injected current	$1nA$

TABLE 2: Parameters for the QIF model

various electrophysiological behaviours. Hence, Eugene M. Izhikevich further extended the quadratic integrate-and-fire model to a two-dimensional system with an auxiliary recovering equation.

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \quad (8)$$

$$\frac{du}{dt} = a(bv - u) \quad (9)$$

$$\text{if } v > v_{thres}, \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (10)$$

Izhikevich's two-dimensional system describes how the membrane potential v and the recovery variable u evolves over time t in response to injected current I . The quadratic system introduces four parameters, a, b, c, d . The part $0.04v^2 + 5v + 140$ is obtained from mimicking the initial spike dynamics of a cortical neuron.

Parameter	Description	Value
v_{thres}	Firing threshold	$30mV$
a	Time scale of u	0.02
b	Sensitivity of u to v	0.2
c	Reset value of u	$-65mV$
d	Reset value of v	2

TABLE 3: Parameters for Izhikevich's QIF model [5]

The variable v represents the membrane potential of the neuron, u represents a membrane recovery variable, which represents the activation of K^+ ions and inactivation of Na^+ ions in the membrane of a neuron [5]. After the spike reaches its peak of $30mV$, the membrane voltage and the recovery variable are reset to new values according to the equation 10.

The implementation of recovery variable u further improves the integrate-and-fire model

to reproduce various firing patterns; i.e. Izhikevich's model combines the biological plausibility of Hodgkin-Huxley type dynamics and the computational efficiency of integrate-and-fire neurons [5].

2.4.3 Adaptive Exponential integrate-and-fire model

The Adaptive Exponential Integrate-and-Fire neuron model (AdEx) is firstly proposed by Wulfram Gerstner and Romain Brette [12]. Further developing the complexity of Izhikevich's two-dimensional system, the adaptive exponential integrate-and-fire model is able to predict 96% of the spike times ($\pm 2ms$) of a regular spiking Hodgkin-Huxley model [12].

$$C \frac{dv}{dt} = -g_L(v - E_L) + g_L \Delta_T \exp\left(\frac{v - v_{thres}}{\Delta_T}\right) - w + I \quad (11)$$

$$\tau_w \frac{dw}{dt} = a(v - E_L) - w \quad (12)$$

$$\text{if } v > v_{thres}, \text{ then } \begin{cases} v \leftarrow E_L \\ w \leftarrow w + b \end{cases} \quad (13)$$

The AdEx model is a spiking neuron model with an additional recover variable. Equation 11 and 12 describe the dynamics of the membrane potential v and the adaptation variable w over a period of time t . The auxiliary reset condition 13 describes how the spike resets when the membrane potential reaches the spike threshold v_{thres} .

Parameter	Description	Value
C	Membrane Capacitance	$281pF$
g_L	Leak Conductance	$30nS$
E_L	Leak reversal potential	$-70.6mV$
v_{thres}	Firing threshold	$-50.4mV$
Δ_T	Slope factor	$1mV$
τ_w	Adaptation time constant	$144ms$
a	Sub-threshold adaptation parameter	$4ns$
b	Spike triggered adaptation parameter	$0.0805nA$

TABLE 4: Parameters for AdEx model [6]

The parameters in the table 4 can be separated into two compartments: scaling parameters and bifurcation parameters [6]. The scaling parameters are responsible for scaling the time axis, and the five parameters are following: the total capacitance C , the total leak conductance g_L , the leak reversal potential E_L , firing threshold v_{thres} , and the

threshold slope factor Δ_T . The bifurcation parameters are: the adaptation time constant τ_w , the sub-threshold adaptation parameter a , and the spike-triggered adaptation parameter b [6].

The exponential term, $\exp(\frac{v-v_{thres}}{\Delta_T})$, describes the process of spike generation and the increasing action potential. The adaptive exponential integrate-and-fire neuron model is capable of describing known firing patterns, e.g., adapting, bursting, delayed spike initiation, initial bursting, fast spiking, and regular spiking [13]. Figure 3 shows a variety of firing patterns, of which parameters are stated in the table 5 respectively.

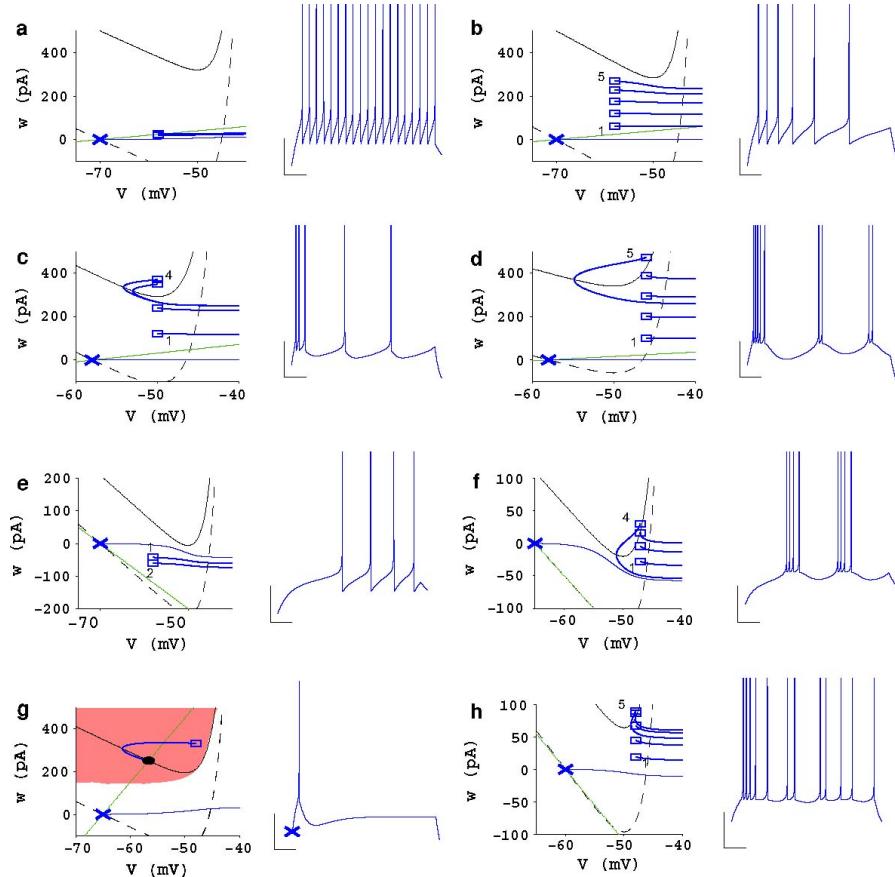


FIGURE 3: Multiple firing patterns in neurons with a step current [3].

Type	C (pF)	g_L (nS)	τ_w (ms)	a (nS)	b (nA)	v_{reset} (mV)
Tonic	200	10	30	2	0	-58
Adapting	200	12	300	2	0.06	-58
Initial Burst	130	18	150	4	0.12	-50
Bursting	200	10	120	2	0.1	-46
Irregular	100	12	130	-11	0.030	-48
Transient	100	10	90	-10	0.030	-47
Delayed Burst	200	12	300	-6	0	-58

TABLE 5: Parameters for multiple firing patterns in neurons [6]

Figure 4 shows the output from the Python simulation of the adaptive exponential integrate-and-fire neuron model with a step current with low and high amplitude. In top figure no spike is produced due to weak input current; in fact, the driving current needs to exceed certain amplitude to stimulate regular spiking of the neuron. On the other hands, the bottom figure exhibits a pattern that the spikes are regularly spaced (*regular spiking*) after a short period of initial higher spiking frequency *initial bursting*.

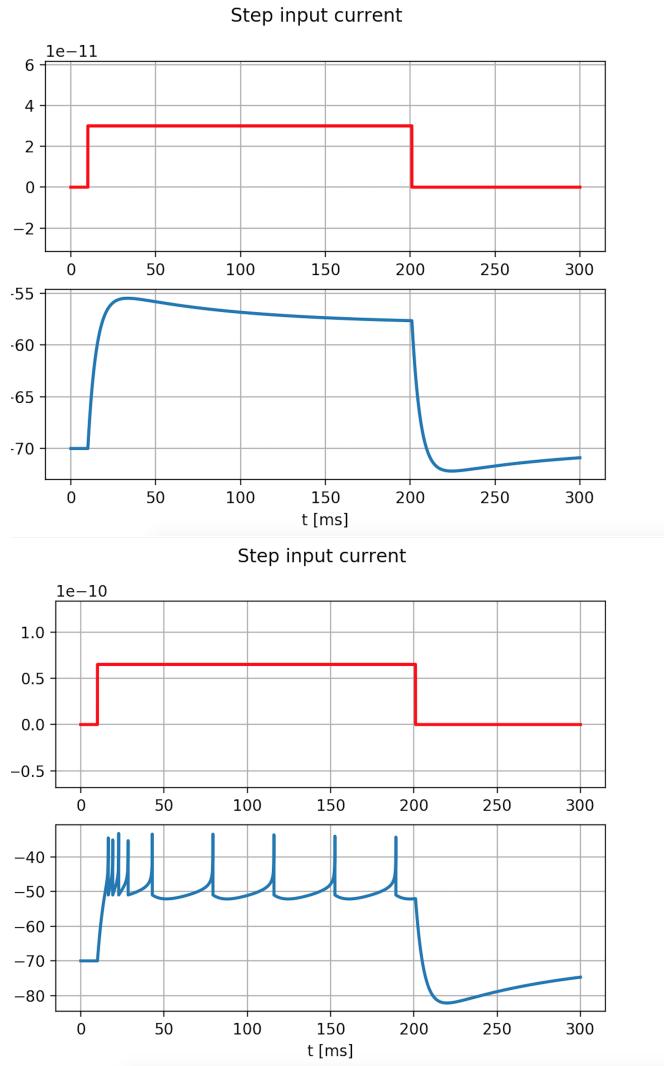


FIGURE 4: AdEx neuron model simulation with a step input current **Top:** Weak input current, no spike. **Bottom:** Strong input current, regular spiking after initial bursting.

The adaptive exponential integrate-and-fire neuron model is able to reproduce varying firing patterns that have been observed experimentally. Figure 5 presents three typical spiking patterns of cortical neurons: regular spiking, bursting, and fast spiking. The parameter values for each spiking pattern are described in the table 5.

Figure 6 is a continuation of Figure 5; the results of the simulation are displayed via

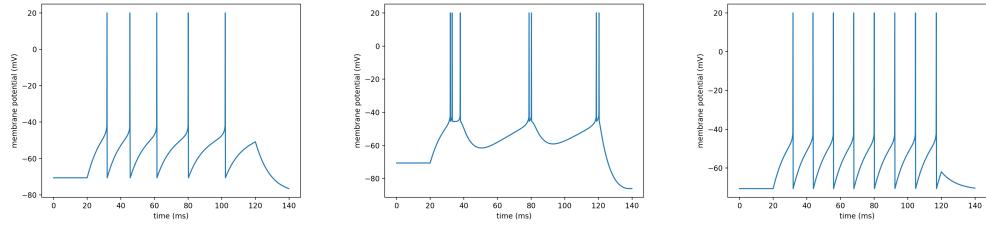


FIGURE 5: Simulation of firing patterns of AdEx neuron model. **A:** Regular spiking. **Middle :** Bursting. **Bottom:** Fast spiking.

Python GUI simulator. In addition to voltage trace and raster plot, Python GUI simulator provides phase plane representation and adaptation current graph. Phase plane representation of firing patterns in Figure 3 resembles that of firing patterns in simulation.



FIGURE 6: Python simulation of AdEx neuron firing patterns. **Top:** Regular spiking. **Middle :** Bursting. **Bottom:** Fast spiking.

Chapter 3

Models of Neuronal Networks

3.1 Introduction

Neurons are organised in populations of units in diverse areas of the brain. Each of these groups of neurons forms a complex neural network and function collectively. By definition, a biological neural network is a series of interconnected neurons of which activation (firing threshold) defines a recognisable linear pathway. Simply, a neural pathway is a path formed by neurons connected together to send electrical signals from one brain region to another. Recent neuroscientists view large-scale brain activities, including cognition, as a result of the actions of neurons and the patterns of connections among the populations of neurons [14].

The populations of neurons are much more influential and insightful dynamical systems than neurons as individuals. This chapter will introduce the properties of populations of spiking neurons with respect to network connectivity and background input.

3.2 Implementation of neuronal network

As discussed in the previous chapter, the conductance-based neuron model is relatively more computationally expensive than the integrate-and-fire neuron model; indeed, the simplicity of the integrate-and-fire neuron model is the key to implement networks of neurons. The implementation of the neuronal networks was carried out using Python, by extending an existing library **Brian2**. Brian2 is an open-source spiking neural network simulator, designed to provide high flexibility and easy extensibility.

To construct the network of spiking neurons, the author had the following questions in mind:

1. How to model neuron-to-neuron connectivity?.
2. How to model the interaction of populations of neurons?

3.3 Modelling network connections

Due to the limited experimental data, the biologically plausible connectivity between cortical neurons of different types or of the same type still remains unknown. To construct and simulate a biologically accurate network of neurons, providing proper synaptic strength is essential. Figure 7 represents how neurons are inter-connected in the neuronal network according to a connection probability of $p < 1$ and $p = 1$. Two network connectivity schemes were explored: full connectivity and random coupling.

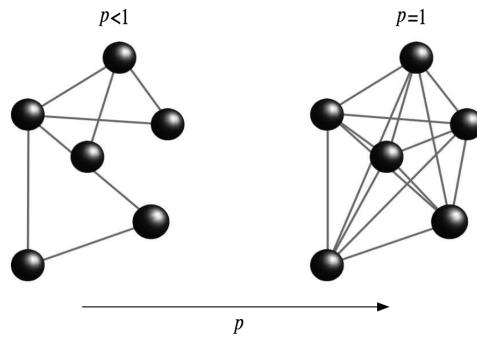


FIGURE 7: Schematic representation of the neuronal network where neurons are connected with a probability p [4].

3.3.1 Fully connected homogeneous network

The simplest coupling scheme is all-to-all connectivity within a population. By homogeneous, all neurons $1 \leq i \leq N$ are identical and receive the same external input $I_i^{ext}(t) = I^{ext}(t)$; the connection strength between the neurons is linearly scaled by following coupling function [10]:

$$w_{ij} = \frac{J_0}{N} \quad (14)$$

where N is the number of neurons and J_0 is a parameter indicating neuron coupling state: for $J_0 > 0$ excitatory all-to-all coupling, for $J_0 < 0$ inhibitory all-to-all coupling, and if $J_0 = 0$ all neurons are independent [10]. The fully connected homogeneous network is easy to build, so any integrate-and-fire neuron models can be implemented as neuronal networks, even the linear leaky integrate-and-fire neuron model as well.

In the case of leaky integrate-and-fire model, the dynamics are described in equation 4. Then the dynamics of i^{th} single component neuron are described as:

$$\tau_m \frac{dv_i}{dt} = -v_i(t) + RI_i(t) \quad (15)$$

combined with a reset condition: if $v_i > v_{thresh}$, then integration restarts at v_{reset} . A fully connected homogeneous network implies that a neuron is coupled to all other neurons and to itself with identical coupling strength. Then the input current of i^{th} neuron $I_i(t)$ is a sum of the external driving current and synaptic coupling described as:

$$I_i = \sum_{j=1}^N \sum_f w_{ij} \alpha(t - t_j^{(f)}) + I^{\text{ext}}(t) \quad (16)$$

where N represents the number of identical neurons, $w_{ij} = w_0$ is the identical coupling strength, and $I^{\text{ext}}(t)$ represents external stimuli input. In a fully connected homogeneous network, the total input current $I(t)$ is identical for all neurons and is a sum of all input currents ranging from I_1 to I_N and external input I^{ext} :

$$I(t) = w_0 N \int_0^\infty \alpha(s) A(t-s) ds + I^{\text{ext}}(t) \quad (17)$$

Constructing a fully connected homogeneous network is much easier than the next step; the challenge is to model a randomly connected network and to provide proper weight (*connection*) between nodes (*neurons*) together in the network.

3.3.2 Randomly connected network

Experimentally the connection probability of a cortical neuron coupled to another neuron is approximately 10 percent, but varies around; in simulations once the connection probability p is fixed, the i^{th} postsynaptic neuron has the number of presynaptic neuron partners $C = pN$ [3]. Then the strength of connection can be described as:

$$C = pN, \quad (18)$$

$$w_{ij} = \frac{J_0}{C} = \frac{J_0}{pN} \quad (19)$$

Equation 19 is extended from the equation 14, where C is the number of presynaptic coupled neurons, p is the connection probability, and J_0 is neuron coupling parameter.

The linearity of the coupling function indicates the number of presynaptic neurons per neuron is proportional to the size of the population N . Likewise the number of presynaptic excitatory neurons is $C_E = p * N_E$; the number of presynaptic inhibitory neurons is $C_I = p * N_I$.

In the implementation of a network of spiking neurons, the author attempted to solve problems regarding the network connectivity and the heterogeneous interacting populations of neurons. By far, methods to design sparsely connected network were explored and now methods to construct interaction of multiple populations will be discussed.

3.4 Modelling interacting populations of neurons

A computational neuroscientist Nicolas Brunel proposed a sparsely connected recurrent network of leaky integrate-and-fire neurons consisting of two populations [7]. The network consists of N_E excitatory neurons and N_I inhibitory neurons with a ratio of 4 : 1. Each neuron receives C_E connections from the excitatory population, $C_I = C_E/4$ connections from the inhibitory population, and C_E connections from external population; See Figure 8. Here external population refers to the number of presynaptic excitatory neurons. Simply speaking, the Brunel network is a network of leaky integrate-and-fire models, of which an excitatory (inhibitory) population is coupled to itself and to an inhibitory (exhibitory) population and receives external input from a third population [3].

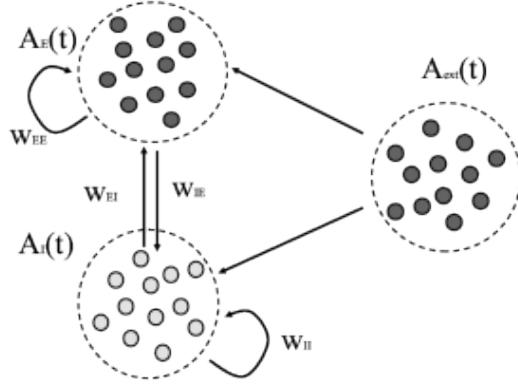


FIGURE 8: Structure of Brunel network: an excitatory population, an inhibitory population, and an external population are interacting [3].

The mean input to a single neuron in the Brunel network is described as:

$$v(t) = \tau_m w_0 C_E [1 - g/4] A(t - \Delta) + \tau_m w_0 C_E \nu^{\text{extern}}, \quad (20)$$

Parameter	Description	Value
τ_m	Membrane time constant	20ms
w_0	Synaptic strength	0.1mV
g	Relative inhibitory strength	4
Δ	Synaptic delay	1.5ms
v_{thresh}	Firing threshold	+20mV
v_{reset}	Reset voltage	+10mV

TABLE 6: Parameters for Brunel's recurrent network model [7].

where τ_m represents the membrane time constant, w_0 represents the synaptic strength of excitatory population, g represents the relative strength of inhibitory neurons in a constant value, Δ represents the synaptic delay, and ν^{extern} represents the time-dependent background firing rate.

In the Brunel network, the firing threshold v_{thresh} and the reset voltage v_{reset} are 20mV and 10mV above arbitrary resting potential respectively. The arbitrary equilibrium potential was set at 0mV, for the convenience of simulations. The synaptic strength of the inhibitory population is derived from the multiplication of the synaptic strength of the excitatory population and the relative inhibitory strength $J_{inhib} = -g * w_0$. Considering the size of excitatory is four times that of the inhibitory population, the amount of excitatory inputs is four times the inhibitory inputs; then the total amount of inhibition and that of excitation are balanced if and only if the relative strength of inhibitory population is four times stronger i.e. $g = 4$ [3].

Simply the amount of total external stimuli to a neuron can be calculated by following: $\tau_m w_{extern} p N_{extern} \nu^{\text{extern}}$. The number of external population depends on the number of excitatory population and connection probability as $N_{extern} = p * N_E = C_E$. The coupling strength of external population is equal to the strength of excitatory population $w_{extern} = w_0$. Then the total amount of background input to a single neuron can be rewritten as following: $\tau_m w_0 C_E \nu^{\text{extern}}$.

The simulations of the Brunel network are described in Figure 9 and Figure 10; Figure 9 was extended from the linear leaky integrate-and-fire neuron model and Figure 10 from nonlinear Quadratic integrate-and-fire neuron model. Both of the networks demonstrate strong synchronisation patterns in the beginning, but the synchronisation fades out over time. Next chapter will further extend the Brunel network simulation for a better understanding of synchronisation.

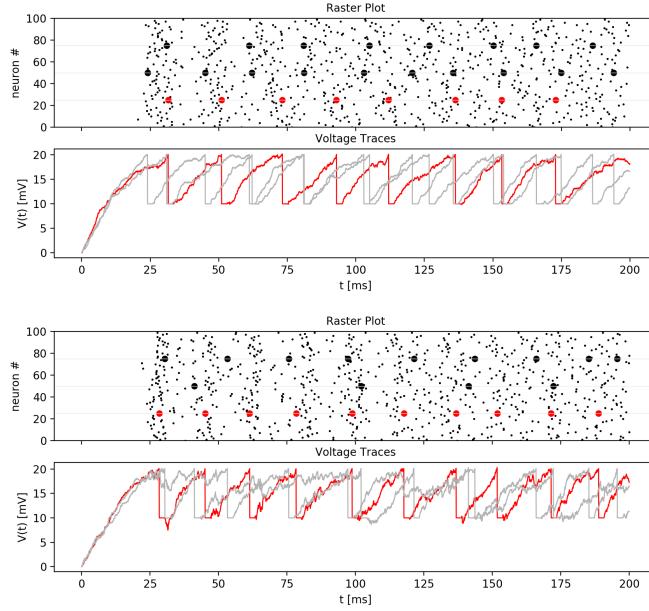


FIGURE 9: Simulation of Brunel network of sparsely connected LIF neurons. **Top:** A network of 100 excitatory neurons and 25 inhibitory neurons. **Bottom:** A network of 2000 excitatory neurons and 100 inhibitory neurons.

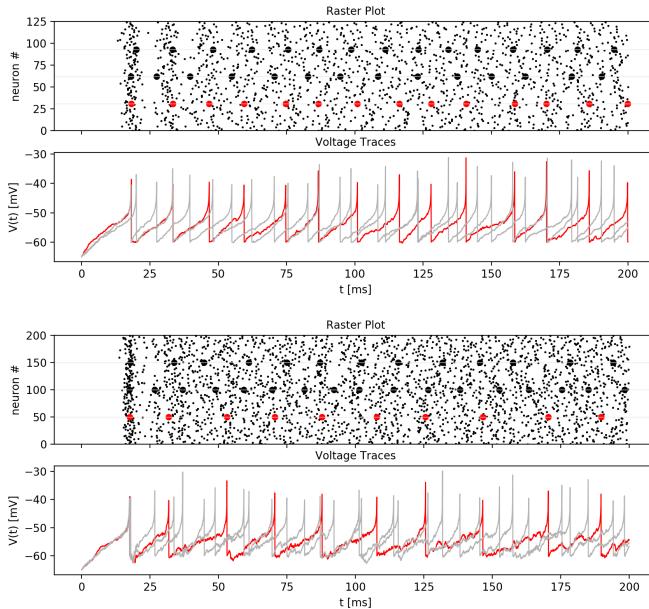


FIGURE 10: Simulation of Brunel network of sparsely connected QIF neurons. **Top:** A network of 100 excitatory neurons and 25 inhibitory neurons. **Bottom:** A network of 2000 excitatory neurons and 100 inhibitory neurons.

Chapter 4

Simulation Results and Analysis

4.1 Introduction

This chapter will firstly introduce software tools used to simulate neuronal networks. Then the results of network simulation will be introduced and be analysed. The simulations of neuronal networks in this chapter continue the simulations of the recurrent network in Chapter 3.

4.2 Software tools

To successfully implement the network of different neuron models and to analyse various electrophysiological behaviours observed in simulations, the author used an open-source Python simulator **Brian2**. Brian2 is designed to maximise flexibility and simplicity and to minimise users implementation time. Most importantly users can define any models of neurons, simply by writing their differential equations in ordinary mathematical form.

In addition to Brian2, the author extended example models from the book *Neuronal Dynamics* written by Wulfram Gerstner and Werner M. Kistler [3]. Neurodynex not only contains several pre-written Python codes, including the adaptive exponential integrate-and-fire model or leaky integrate-and-fire model but also supports convenient plotting tools. The author modified the plotting tools and integrated them into the GUI simulator.

The implementation of the GUI simulator was supported by a GUI widgets toolkit **PyQT**, one of the most popular cross-platform GUI library. The completed Python GUI network simulator is seen in Figure 11. The simulator is capable of simulating networks of three different neuron models and displaying raster plot and voltage trace for all three

models. Additionally, the neuronal network simulator supports phase plane representation and adaptation current graph, in case of the adaptive exponential integrate-and-fire model being used.

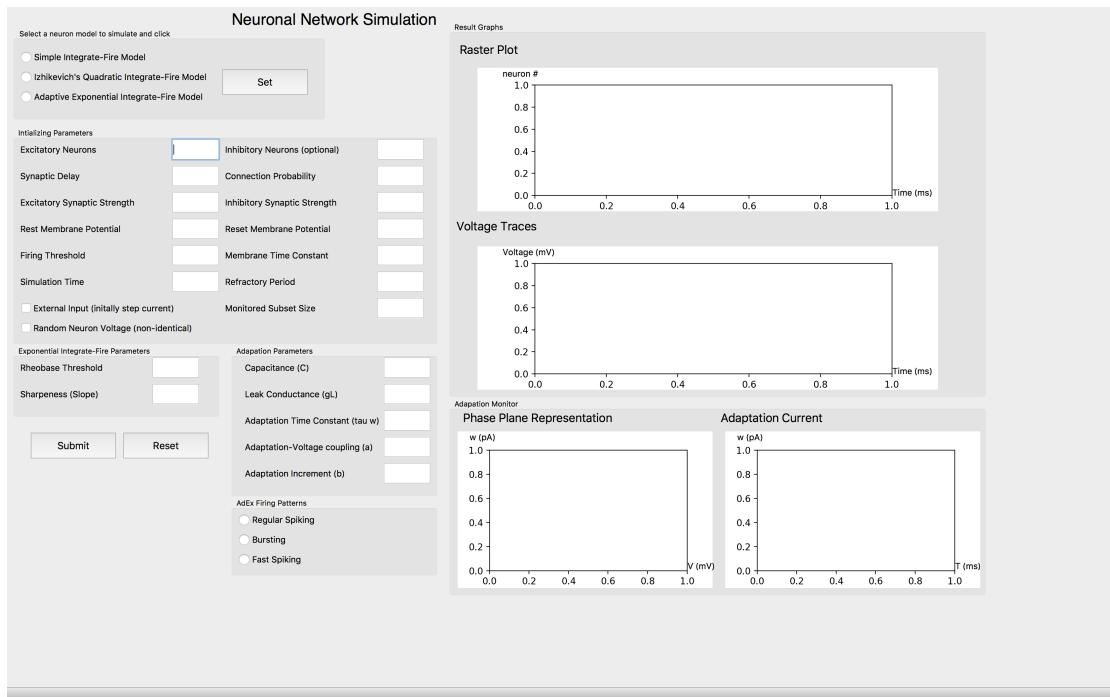


FIGURE 11: Python GUI design for Neuronal Network Simulation.

4.3 Emergence of Synchronisation

The Brunel network of 2500 leaky integrate-and-fire neurons was constructed due to the simplicity of linear neuron models. The ratio of the excitatory population and the inhibitory population is 4:1, respectively 2000 excitatory neurons and 500 inhibitory neurons. The simulation was run for a period of time $t = 300s$, with external stimuli rate of $14Hz$ and relative inhibitory strength $g = 2.5$. Consider the amount of inhibitory inputs balances with the excitatory inputs at relative inhibitory strength is four times ($g = 4$), it is predictable that the amount of total excitatory inputs exceed the amount of inhibitory, and thus the network will tend to *excite*. Additionally neurons are connected via synaptic weight of $p = 0.1$.

To simulate the emergence of synchronisation, each neuron in the network was initialised with random membrane potential in a range of resting potential and firing threshold.

Figure 12 shows the results of the Brunel network simulation in three periods of time. As seen in the middle figure, each spike train has different starting points, indicating each neuron has random membrane potential. Also, neurons in the raster plot seem scattered at first, and each spike train has different starting points, indicating each neuron has random membrane potential. Evidently, neurons are not synchronised at the beginning of this simulation. In contrast to the initial $50ms$ of simulation, the last $50ms$ of simulation demonstrates a strong synchronisation of neurons. The synchronisation of neurons emerges approximately at $t = 30ms$.

Figure 13 shows the results of the Brunel network simulation in the absence of recurrent feedback. Same parameter values were used except the synaptic strength of the excitatory population w_0 is set to be 0 to disable the periodic feedback. Here the emergence of synchronisation appears after an extended period; indeed synchronisation pattern is not recognisable in the first $50ms$ of simulation and starts to emerge after the halfway of simulation time past. In conclusion, the synchronisation of the neurons is not a residue of shared initial conditions but emerges slowly over time [3].

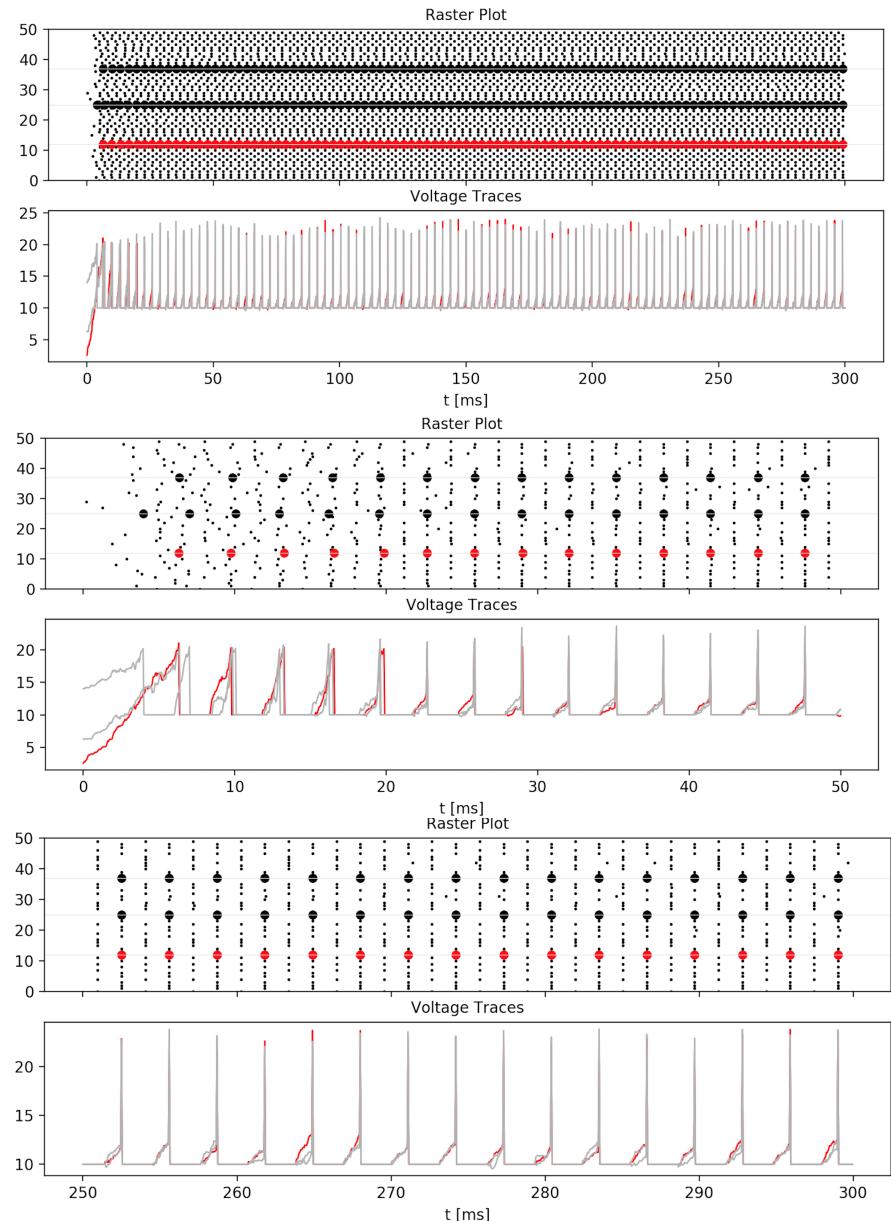


FIGURE 12: Simulation of Brunel network of sparsely connected LIF neurons with recurrent feedback. Initial membrane potential of each neuron is randomly selected.
Top: Simulation for whole simulation time. **Middle:** Initial 50ms of simulation.
Bottom: Last 50ms of simulation.

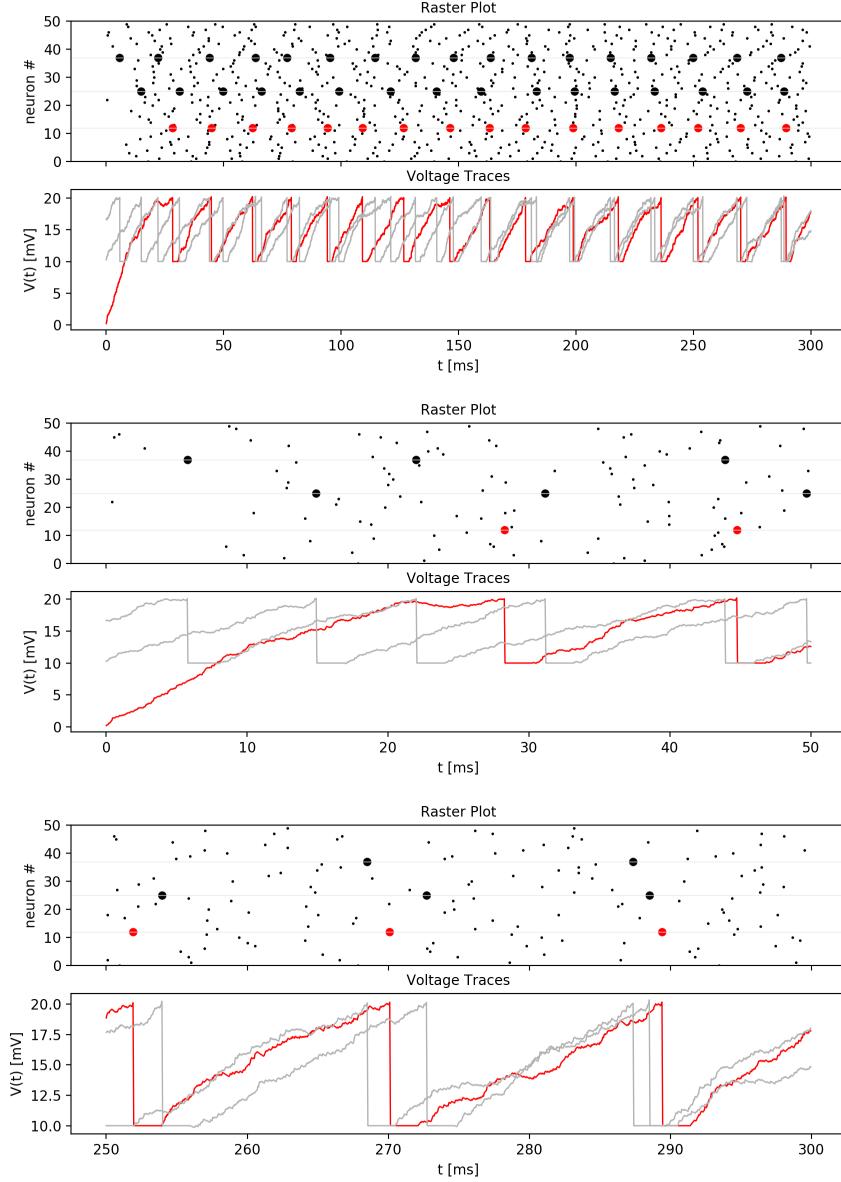


FIGURE 13: Simulation of Brunel network of sparsely connected LIF neurons in the absence of recurrent feedback. Initial membrane potential of each neuron is randomly selected. **Top:** Simulation for whole simulation time. **Middle:** Initial 50ms of simulation. **Bottom:** Last 50ms of simulation.

Chapter 5

Conclusion and Reflection

This research was dedicated to studying different dynamical behaviours of single neurons and neuronal networks and analysing varied electrophysiological behaviours observed.

The research started from gaining insight into biological and mathematical understandings of the neuron. On the simulations of the two-dimensional integrate-and-fire neuron model, varying electrophysiological behaviours were observed. This thesis presented four types of neuron models: the famous Hodgkin-Huxley model, leaky integrate-and-fire model, Izhikevich's quadratic integrate-and-fire model, and the adaptive exponential integrate-and-fire model. Of the four models, the adaptive exponential integrate-and-fire model was specifically investigated to explore various electrophysiological behaviours due to its computational speed and biologically relevance to the Hodgkin-Huxley model. Throughout the simulations of the adaptive exponential integrate-and-fire model, three typical firing patterns of spiking neurons were simulated and analysed.

In the construction of the neuronal network, I investigated the network coupling and the interaction of multiple populations of neurons. Properties of fully connected homogeneous network and random network were presented with a brief description of coupling function. The interactions of populations of neurons were studied by implementing a sparsely connected recurrent network of the Brunel. The recurrent network has an excitatory population, an inhibitory population, and an external population, which gives background stimuli.

The simulations of the Brunel's recurrent network show the emergence of synchronisation in the network. The neurons in the recurrent network are synchronised after a short period of time, while the neurons in the non-recurrent network need an extended time to be synchronised. Notably, the neurons can synchronise even in the absence of recurrent

feedback. That is to say, the synchronisation of neurons can emerge solely by the external input.

On reflection, I had difficulties in implementation interacting population of neurons. There were many ambiguities due to unclear explanations and abstruse mathematical formulas. The challenges in understanding and implementation lead to a significant amount of time spending on investigating research papers and lecture notes. The other challenge was in implementation of the Python GUI simulator. I did not have any prior experience with Python nor GUI before this project. Thus, a considerable amount of time was spent to learn Python, to investigate open-source scientific libraries, and to implement a simulator. I believe the experience of manipulating and extending scientific packages, such as Numpy, Brian2, and Neurodynex, will be highly beneficial in my future academic career.

Acknowledgements

I would like to thank my supervisor, Dr Eva M. Navarro-Lpez, for her patient guidance and invaluable advice throughout this project. Also, I would like to thank my parents for their unconditional love and consistent supports.

Bibliography

- [1] Cs231n convolutional neural networks for visual recognition. <http://cs231n.github.io/neural-networks-1/>. Last Accessed: 24-04-2018.
- [2] Matilde Marcolli. Geometry of neuroscience. <http://www.its.caltech.edu/~matilde/GeomNeuroClass.html>. Last Accessed: 24-04-2018.
- [3] R. Naud W. Gerstner, W.M. Kistler and L. Paninski. *Neuronal Dynamics*. Cambridge University Press, July 2014.
- [4] F.S.Borges., P.R.Protachevicz, and et al. E.L.Lameu. Synchronised firing patterns in a random network of adaptive exponential integrate-and-fire neuron model. *Neural Netw*, 90:1–7, 2017.
- [5] Eugene M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.
- [6] N. Marcille W. Gerstner, R. Naud and C. Clopath. Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, 99:335–347, November 2008.
- [7] N. Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8:183–208, 2000.
- [8] J.K. Aggarwal. *Notes on Nonlinear Systems*. Van Nostrand Reinhold Co., 1972.
- [9] Eugene M. Izhikevich. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press Cambridge, 2007.
- [10] W. Gerstner and W.M. Kistler. *Spiking Neuron Models Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [11] Imogen Gough. Models for neurons and neuronal networks. *University of Manchester*, April 2016.

- [12] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neuroscience*, 94(5):3637–3642, November 2005.
- [13] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model. *Scholarpedia*, 4(6):8427, November 2009.
- [14] Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, 2009.
- [15] C. van Vreeswijk N. Fourcaud-Trocme, D. Hansel and N. Brunel. How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of neuroscience*, 23:11628–11640, 2003.