MEAN-FIELD ANALYSIS FOR MODEL-BASED SPIKING NETWORKS

by

Valentin Paquin

B.S in Electrical Engineering, ENSEA (Cergy, FRANCE), 2015

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2018

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Valentin Paquin

It was defended on

May 30, 2018

and approved by

Zhi-Hong Mao, Ph. D., Associate Professor, Department of Electrical and Computer Engineering

Ahmed Dallal, Ph. D., Assistant Professor, Department of Electrical and Computer Engineering

Robert Kerestes, Ph. D., Assistant Professor, Department of Electrical and Computer Engineering

Thesis Advisor: Dr. Zhi-Hong Mao, Associate Professor, Department of Electrical and Computer Engineering

MEAN-FIELD ANALYSIS FOR MODEL-BASED SPIKING NETWORKS

Valentin Paquin, M.S.

University of Pittsburgh, 2018

The human brain is composed of millions of neurons, firing spikes according to their membrane potentials. The difficulty in studying the brain exists partly because of the randomness property of neurons firing in a network. To understand more about the dynamics of a neuron's firing rate, we choose to study a specific set of nonlinear dynamical equations that represent a neural network based on a spiking point of view with adaptation qualities. The dynamic membrane potential of a single neuron is a challenge to study since we can hardly know the number of spikes fired at a certain time. In this thesis, we use phase-plane analysis and more precisely mean-field analysis to address the random nature of the dynamic of model-based spiking networks. We find that the dynamics of neurons in a network offer exploitable and relevant information such as patterns of stable or unstable oscillations in certain circumstances.

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

I first would like to thank my advisor, Dr. Zhi-Hong Mao for his kindness, his help and for accompanying me into this new world of research. More than being a great advisor, Dr. Mao always treated me as a friend and colleague making me really comfortable entering in my new life in the United States. He also introduced me into his collaborators and friends that we regularly met into an excellent and interesting group of research where I learned so much.

Also, it means a lot to me to thank Eduardo Diniz, PhD student under Dr. Mao advisement who led me from the very beginning on every aspect of my study. It would have been impossible to learn as much as I did without him. He has been and will still be an inspiration and a model.

Then, I would like to thank Thomas Tang and Carine Sabouraud of the international relations team at ENSEA and Dr. Mahmoud El Nokali for giving me the opportunity to study abroad at the University of Pittsburgh.

I should thank all the students I met working with Dr. Mao and his collaborators as well. We practiced study groups all years long teaching each other on really captivating aspects of non-linear dynamical system theory.

Finally, this is meaningful to thank my fellow French Nathan Renaudie, Cedrine Rebrion, Blandine Russo, Lou Botherel and Sébastien Ollivier that have made my adventure a great experience.

1.0 INTRODUCTION

The brain has always been one of the greatest challenge concerning the understanding of a human being. Most of the advanced life we know works with a brain, from a simple brain for a mouse to a more complex tool for humans. Biologists and all kind of scientists have already made a lot of progress in the field and have given us a lot of knowledge, from the understanding of action potential with the Hodgkin-Huxley model [24] to large scale brain networks [7]. Most of the neurons are spiking neurons, that is they fires a spike or an action-potential when their membrane potential reaches a certain threshold. However, to study the dynamics of the brain, focusing on the dynamic of a single neuron is not sufficient since the variability of neurons is tremendous, in the number first but also, in the temporal distribution of spikes, making the response of a single neuron hard to cope with [42].

The brain's learning model is an inspiration for applications that want to be defined as intelligent. For this reason, neural networks have been and are still used and studied in many applications in the current world, from the classification algorithm, speech recognition, data exploitation, to self-driving car and robots. Many scientists came up with different neural network approaches, from firing-rates networks, to spiking networks, with different ways of training, with first the training of weights associated to each neuron in network or with the training of the exact spike timing studied in this thesis. Although, some of these approaches highlight such or such

cognitive function, only spiking network attempts to mimic the biological mechanisms generating behaviors of neurons living in our brain.

One specific spiking-based model constructed by M. Boerlin, C. K. Machens and S. Denève [6] have drawn our attention for its abilities and properties. In this thesis, we are not aiming to create an artificial neural network able to realize certain tasks, but we want to bring new unobserved dynamical properties to this neural network using tools at our disposal such as phase-plane analysis and specifically mean-field analysis.

This thesis aims to use a phase-plane analysis in order to study the dynamics of the spikes on a spike-based neural network. Our mathematical analysis is relevant when it helps to get a better understanding of a promising model. That is why, it is significant to review the main principles of neural network history to highlight the network we selected to work on. Therefore, this thesis will start with describing two different breakthroughs in neuroscience that has allowed the derivation of such a spiking-based network. The integrate and fire model [2], and the recurrent neural network [21] constitute the essential characteristics of a spiking neural network. Once the spike-based temporal neural network derived, it makes sense to apply nonlinear standard analyses and technics.

This thesis shows how mean field analysis is addressed to extract the random nature of spike's dynamics using some population measure tools. Pattern of stable oscillations are found for an individual neuron's spike timing, and mean field analysis's tool seen as the momentary state of the network at time t opens the door to new dynamic analyses. Some ideas for these new analyses will finally be presented to conclude this thesis. We expect, that our analytical result will offers interesting insights in the mechanisms behind the spike and their exact timing.

2.0 BACKGROUND

2.1 A NON-LINEAR DYNAMICAL NEURAL NETWORK MODEL

2.1.1 Leaky Integrate and Fire Model

To develop the main network of this thesis, we start from one of the simplest neural network model. That is the leaky integrate and fire neuron model (LIF), that we explore more deeply in the following section. First time introduced very early on the twentieth century by Lapicque, when neural action potentials mechanisms were far away from being known. However, only highlighted in 1999 thanks to Abbott [2], LIF neurons focus essentially on sensory neurons, that have the characteristic to convert a specific type of stimulus into an action potential assimilated to a spike.

A LIF neuron model is one of the simplest neuron model. To start, we model a neuron as a leaky integrator of its input $I(t)$. Neurons are described as ionic currents flowing through the cell membrane when neurotransmitters are released through a combination of channels and gates [25]. The following classic LIF equation shows the membrane potential v described at time t and conducted by a simple RC circuit [19]. The resistor membrane R and the capacitive integration of the input under a membrane potential timescale τ_m:

$$\tau_m \frac{dv(t)}{dt} = -v(t) + RI(t). \tag{2.1.1}$$

In this model, we don't define spikes explicitly, but instead we use an instantaneous reset to an initial membrane potential value v_r, when the membrane potential reaches a certain threshold v_{th}. Figure 1 precises the action potential of a neuron along the gated channels of the neuron.

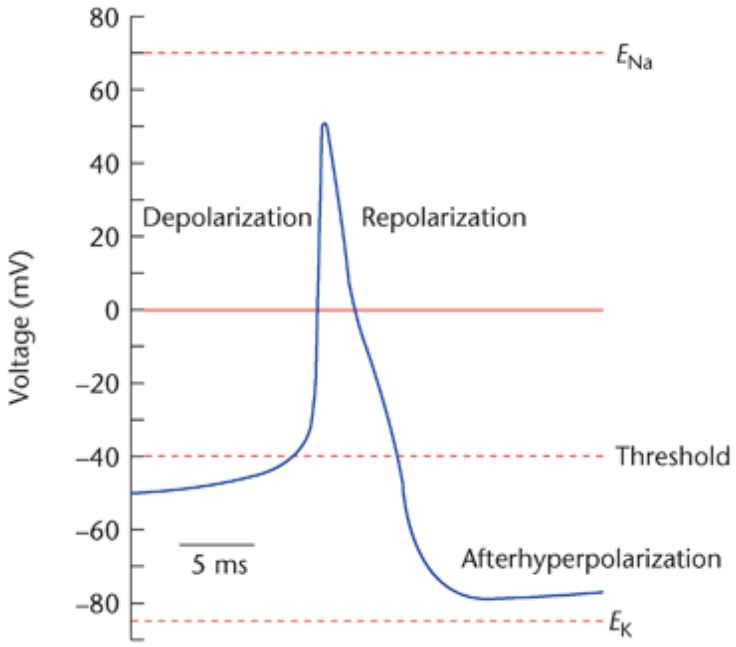


Figure 1: Action potential of a neuron associated with the equilibrium potentials for sodium $\boldsymbol{E_{Na^{2+}}}$ and potassium $\boldsymbol{E_{K^+}}$. Adopted from [17].

From the biophysics of a neuron, let's assume the neuron's membrane potential is first on an initial state known as the resting membrane potential. At this point, the different sodium $Na^{2+}$and potassium K^+ channels are all closed. But when a neuron fires a spike (action potential), the different gated channels open alternatively to depolarize, repolarize and hyperpolarize the membrane potentials. These different terms correspond to different parts of the action potential, see Figure 1, and to the changes in concentration gradient between sodium and potassium. However, when they are inactivated after the action potential, the neuron is prevented to spike again by the absolute refractory period Δ_{abs}. This period is applied immediately after

$v(t)$ hit v_{th} and prevent second stimulus to excite the neuron a second time. For further details on the biophysical details of the action potential of a neuron, see [15].

2.1.2 Recurrent Neural Network

Feedback plays an important role in neurons communication when in the brain, in many different ways, that are for short-term memory tasks [4], decision making [12] or more deeply in focus and attention [18]. There are defined as any network whose neurons send feedback signals to each other. Artificial recurrent neural networks (aRNNs) are an interesting alternative to feed-forward networks (FFNs) and convolutional neural networks (CNNs) for different reasons, particularly in machine learning. FFNs and CNNs are built on the principle that information only moves in only one direction from a layer including the input nodes to a layer that represent the output nodes and going through one or some hidden layers as showed on Figure 2. Because of this unique direction of information propagation, these two types of networks can't reproduce synapses as much as we would like.

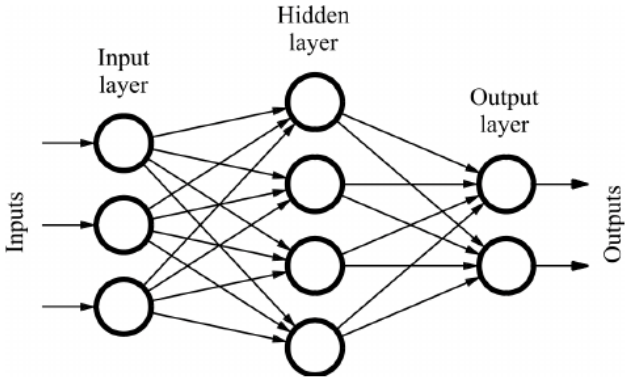


Figure 2: Example of a feed-forward neural network architecture. Adopted from [37].

Researchers came up with RNNs to answer the following biological rules of neural networks. A neuron is an input-output unit acting under a spiking rule, more than receiving its own

input, neurons also receive the collective output of other neurons in its neighborhood, through synaptic connections. Therefore, RNNs can be view as non-linear dynamical systems with two specific attributes that drawn our attention. As discussed above, RNNs are highly recurrent with feed-back loops at multiple spatial scales as biological circuits. Secondly, they are commonly known as really good when answering to a task with temporal structure, in other words, the system's dynamic is of the most importance. Mathematically, we can write them as systems of coupled ordinary differential equations (ODEs). Starting from scratch, we define:

$$\dot{\boldsymbol{x}} = F(\boldsymbol{x}) \tag{2.1.2}$$

where $\boldsymbol{x}$ is a N-dimensions state vector and F is the evolution function of the current state $\boldsymbol{x}$. In this case every state $\boldsymbol{x}_i$ is led by its ODE $\dot{\boldsymbol{x}}_i$ and where the derivative not only depends on its current state x_i but also on the current states of its neighbors. The network states the different interactions between neurons or synapses and associate to each of them a specific weight according to the importance of the information given by this particular interaction. We often say this weight is related to the strength of the synapse. We remind from last section that a neuron is an electrically excitable cell and that neural networks are built as electrical circuits [12]. Thus, the nonlinear function $\phi(x_i)$ is often introduced to define the relationship between the membrane potential and the electrical activity, respectively x_i and S_i, so that we have $S_i(t) = \phi(x_{\mathrm{i}}(\mathrm{t}))$. For instance, x_i might be related to the membrane potential of neuron i and S_i to its firing rate, that is a temporal average of the spike count over a time window $T = t + \Delta t$. Under realistic conditions, neuron's firing usually become smoother around the threshold and for that particular reason, the descriptive function $\phi(x)$ is often seen has a sigmoid shape function [43]. Therefore, we have the following constraints:

$$-1 \leq S_i(t) \leq 1 \quad \text{and} \quad -\infty < x_i(t) < \infty. \tag{2.1.3}$$

This kind of realizations is called rate-based network. However, the relationship between $x_i(t)$ and $S_i(t)$ and the interpretation of each variable are specific for each model. The majority of the RNNs models define a matrix J which is a $N * N$ matrix as the synaptic efficacy coupling the output of a presynaptic neuron j and the input of a postsynaptic neuron i. Thanks to these consideration, we describe the dynamics of a RNN using N first-order and coupled differential equations:

$$\dot{x}_i = -x_i + \sum_{j=1}^{N} J_{ij} S_j = F(x). \quad (2.1.4)$$

This classic RNN equation relates the membrane potential of a neuron on the left hand-side, the first term on the right hand-side represents a leak term reflecting the membrane passive nature [39]. Finally, the last term depicts synaptic currents going through neuron i and coming from all the other neurons in the network. Another thing we want to add when creating a recurrent network, is an external input to drive the network to mimic a certain feature chosen beforehand. Commonly, recurrent neural networks are used in natural language processing meaning then we introduce an external input to drive the network according to the RNN application we want to use that can be in speech or image recognition or ever image generation [20][21][27][41].

The more the input becomes complex and realistic, the more RNNs can bring us information and intuition on how the artificial black box operates. Finally, the main operation in building a RNN consists to simply reduce the complex dynamic of spiking neurons in order to study the derivation of its equations. Nevertheless, the rate-based interpretation using RNNs networks has been giving really good results when learning features and for long term dependencies. From their names, rate-based neural networks favor the information contained in the instantaneous firing rate (when Δt is considered really small) of a neuron instead of the exact spikes time.

From an artificial recurrent network point of view, the training process begins when all the parameters are chosen and the equations derived. Before training, the synaptic efficacy matrix representing the decoding weights is intuitively initialized using Gaussian random variables. Then to get the network to execute a specific task, the decoding weights need to be properly trained. There are many existing training methods for RNNs [29] that we won't introduce here since it is not of the most importance for this thesis. Recurrent neural networks have been a great breakthrough for neuroscientists [22]. Nevertheless, our goal is not to use a RNN in any application but to complete our understanding of its dynamics, that is we want to study the exact times of firing for neurons in the network and therefore focus on the study of spiking neural networks.

2.1.3 Spiking Neural Network

A third generation of neural networks called spiking neural network (SNN) have emerged and aims to bridge the gap between neuroscience and machine learning. Originally, a neuron reacts to an input according to a certain threshold. If the input's value is equal or get stronger than its threshold, the neuron responds with a linearly increasing number of spikes. After spiking, the membrane potential of the respective neuron gets back to a resting value. Using, the same notation as in the previous section and s_i as the spike counting variable for a specific neuron or also called spike train, the following set of equations describe a spiking network:

$$x(t) = F(x(t)) \tag{2.1.5}$$

$$x_i \geq x_{threshold} \tag{2.1.6}$$

$$x_i \rightarrow x_{rest} \tag{2.1.7}$$

$$s_i \rightarrow s_i + 1. \tag{2.1.8}$$

This model can easily incorporate biological details including voltage activation channels and gates in order to observe realistic spike generations as we can see in [19]. Besides, the concepts of artificial neuron and synaptic state as seen in section 2.1.2, spiking neural networks include time in its dynamics. SNNs use the idea of membrane potential developed above. The information transferred in SSNs are then transferred via the precise timing of spike or of a sequence of spikes. This type of network has the potential to work with large neural network [29]. Also, it allows us to study the dynamics of the spike generation for individual neurons and neurons in a network. However, since training methods are still not generalized for this kind of network, this is not the mainly used type of neural network. Indeed, we don't know any general supervised training methods for SNN yet. Spike trains are not differentiable, thus usual supervised learning algorithm are not good solutions anymore to train this kind network even if supervised learning algorithm can still be used in some cases [27][38]. However, more complicated training methods might still be a reason why SNNs are not well used in application. Another reason is just about the notoriety of firing rates or recurrent networks as convenient and universally understood for certain simple tasks.

Since researchers have already proven that SNN are computationally really powerful [28], studying the behavior of brain's neurons is relevant. SNNs allow us to analyze the different spike trains and the exact time at which a neuron in the network will fire. The precision of the spike times composing the spike train are then analyzed through phase-plane analysis in this thesis.

2.2 PHASE PLANE ANALYSIS

Phase plane analysis is useful to apply when studying the behavior of any system in time. Indeed, while studying the dynamics of a system, we don't need to know any closed-form solutions of the system to draw its phase portrait and to get an intuition of how the system behave in time [34]. Let's take the particular case of a 2-dimensional system of linear differential equations of the following form:

$$\dot{x} = Ax \tag{2.2.1}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}. \tag{2.2.2}$$

Its phase portrait is a representative set of the solutions for a system. In the context of a two-dimensional system, the phase plane or in this case, phase portrait describes the trajectory traced by every solution in time, on the plane $(x, y) = (x_1(t), x_2(t))$. To predict the different trajectories of a system's solutions, a classic study of eigenvalues and eigenvectors is needed. In this thesis, we want to develop a nonlinear phase plane analysis but to do so it is necessary to briefly remind the main idea behind linear system analysis, since the same technics would be use when working with nonlinearities.

2.2.1 Linear Systems

We first decided to generalize the phase-plane behavior of a linear system. Every linear system can be described in the form of equations (2.2.1, (2.2.2), in this case we have the following set of eigenvalues:

$$\lambda_{1,2} = \frac{tr(A) \pm \sqrt{tr(A)^2 - 4\,det(A)}}{2} \tag{2.2.3}$$

with A called the state matrix of the system, and $\lambda_{1,2}$, the eigenvalues. This system is written:

$$\begin{cases} \dot{x}_1 = a_{11}x_1 + a_{12}x_2 \\ \dot{x}_2 = a_{21}x_1 + a_{22}x_2 \end{cases} \quad (2.2.4)$$

when $det(A) \neq 0$, the system has exactly one solution at the origin but and when $det(A) = 0$, many different solutions. In both cases, there are different types of critical points and trajectories that can be observed. For clarity reasons, we are just going to develop the case where determinant of A is non-zero, but this can be expanded to critical points not located at the origin.

In the next sections, we are going to briefly explore the different dynamics of linear and nonlinear systems thanks to a general study of its ODEs.

2.2.1.1 Phase portraits of Linear Systems

In this section, we want to classify the different critical points and trajectories that you can find when applying the phase plane analysis theory and the eigenvalues and eigenvectors of system for an only critical point located at $(0,0)$.

- When eigenvalues are both distinct and real, $x = C_1 k_1 e^{\lambda_1 t} + C_2 k_2 e^{\lambda_2 t}$.
    - Either λ_1 and λ_2 are both positive or both negative.
        - $\lambda_1, \lambda_2 > 0$, Trajectories move from the critical point to infinite and we have unstable nodes that are also called Sources.
        - $\lambda_1, \lambda_2 < 0$, Trajectories move from infinite toward the critical point and we have asymptotically stable or stable nodes that are also called Sinks.

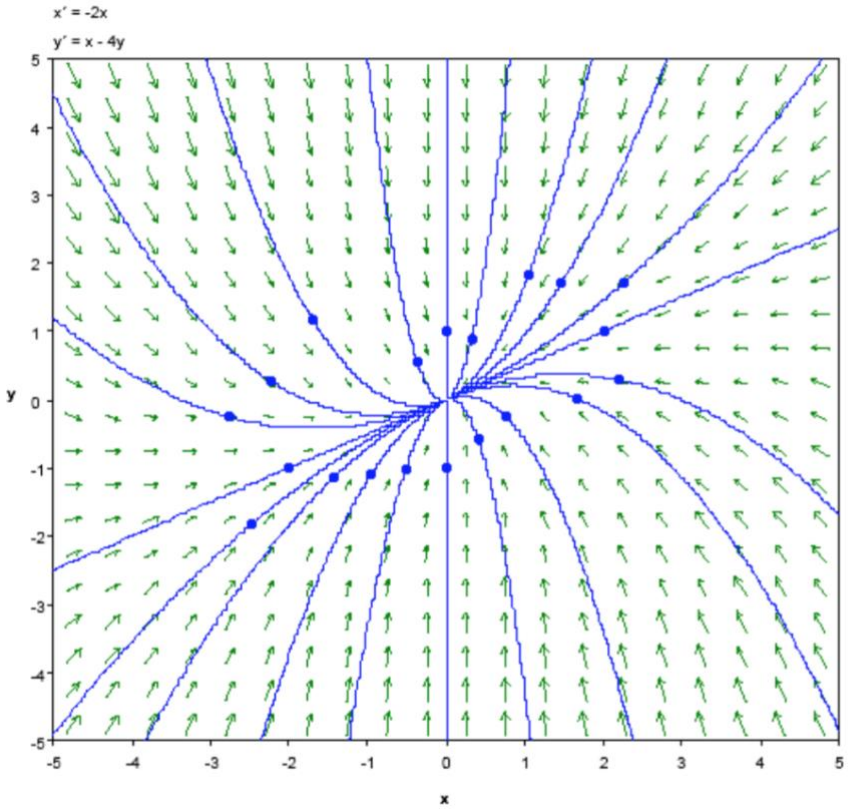


Figure 3: Represents the analysis from above. Stable or unstable node according to circumstances and eigenvalues values. Adopted from [45].

- λ_1 and λ_2 have opposite signs. There are two different trajectories to denote in this case. Either the eigenvalue is negative, $\lambda_1 < 0$, the associated eigenvector trajectory starts from infinite to converge toward the critical point, either the eigenvalue is positive, $\lambda_2 > 0$, the associated eigenvector trajectory starts from the critical point to move to infinite. All the other trajectories start from infinite, then get close to one of the previous eigenvector trajectory and move back to infinite by changing direction when getting close of the critical point. In this case, we have unstable nodes called Saddles.

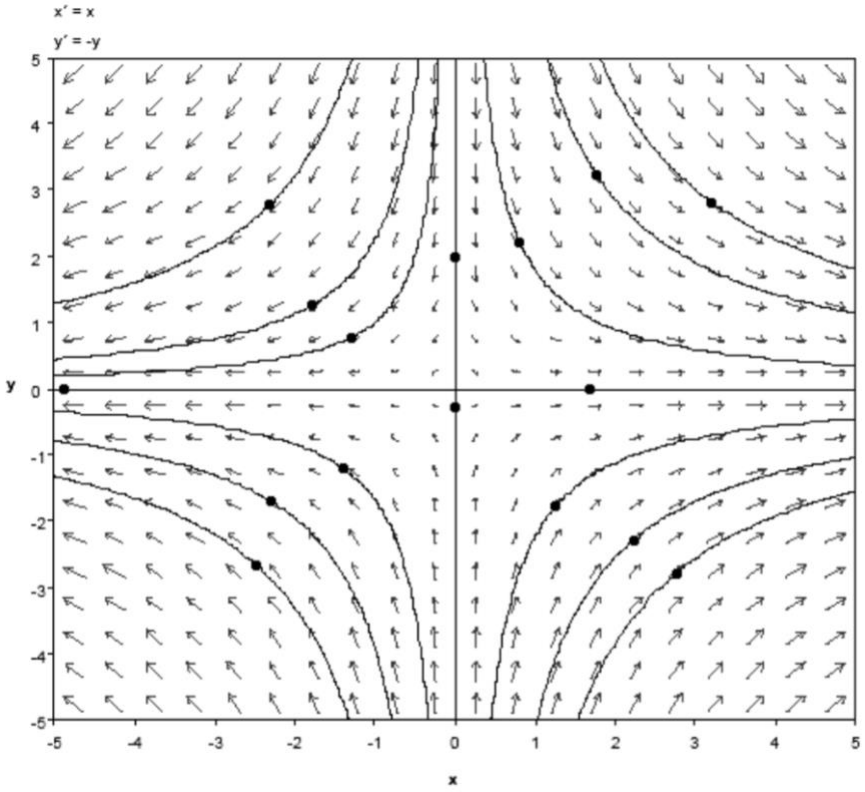


Figure 4: Represents the analysis from above. Saddle point representation, this is always unstable. Adopted from [45].

- When eigenvalues are repeated and real, $x = C_1k_1e^{\lambda t} + C_2k_2e^{\lambda t} = e^{\lambda t}(C_1k_1 + C_2k_2)$.
    - The vector $(C_1k_1 + C_2k_2)$ gives the direction of every nonzero solutions. So, the trajectories have 2 different movements possible.
        - Either $\lambda < 0$, the trajectory moves toward infinite. This gives us an asymptotically stable trajectory.
        - Either $\lambda > 0$, the trajectory converges toward the critical point, and we have an asymptotically unstable trajectory.

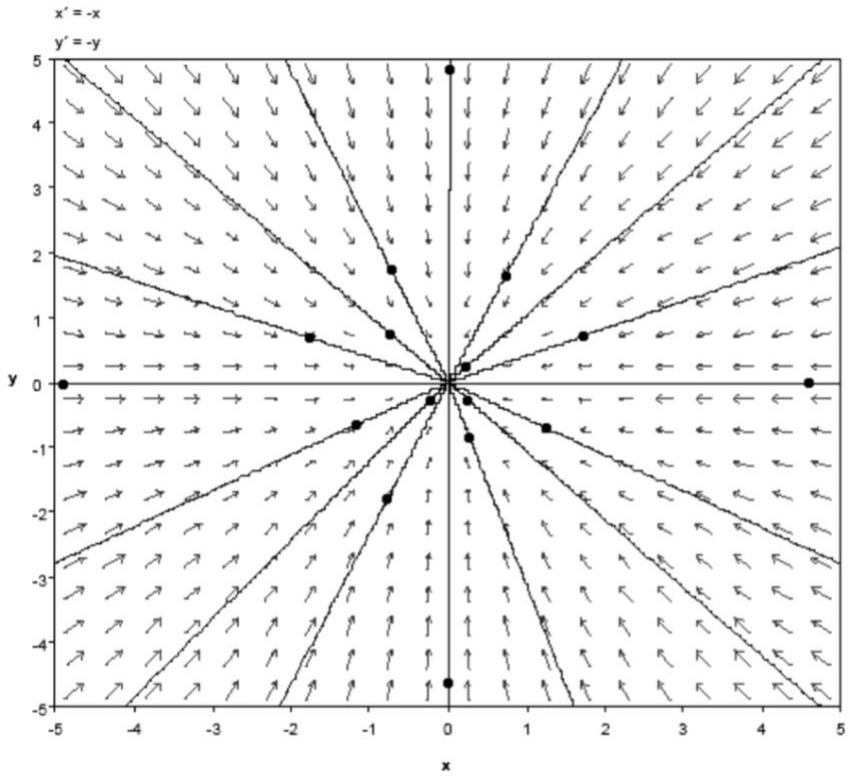


Figure 5: Represents the analysis from above. We call it a proper node or star point, unstable or asymptotically unstable according to eigenvalues. Adopted from [45].

- When eigenvalues are linearly associated to a same independent eigenvector k then the solution looks like, $x = C_1 k_1 e^{\lambda t} + C_2\left(kt\, e^{\lambda t} + \eta e^{\lambda t}\right)$.
  - Either $\lambda < 0$ and all trajectories converge toward the critical point and are asymptotically stable.
  - Either $\lambda > 0$ and all trajectories converge toward the critical point and are asymptotically unstable.

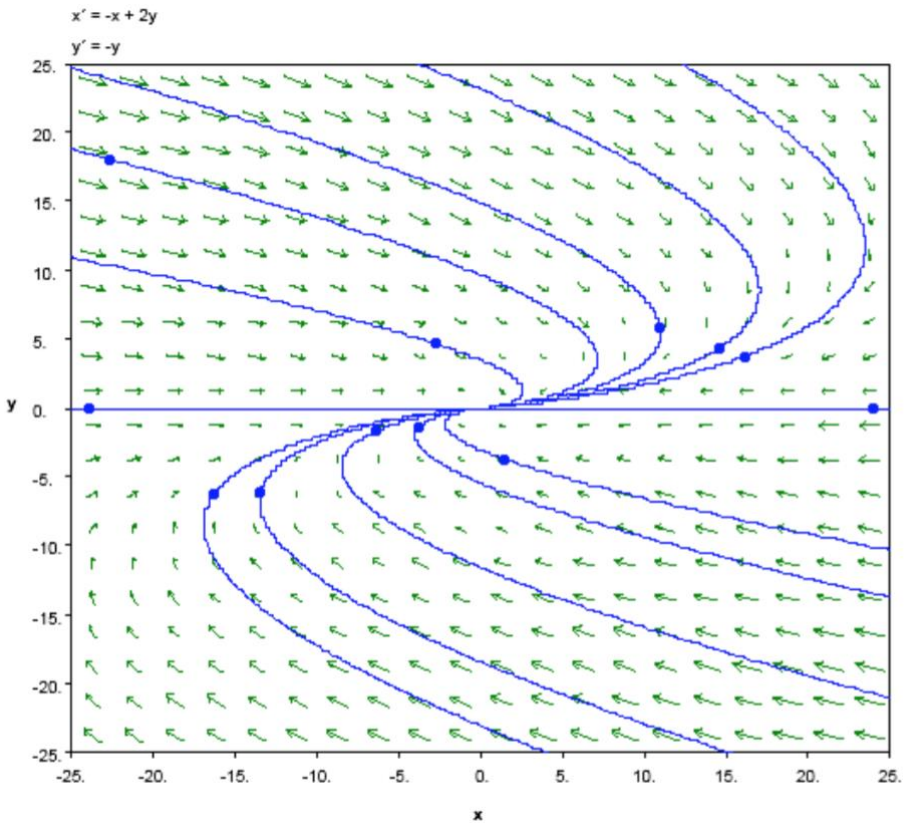


Figure 6: Represents the analysis from above. We call it an improper node unstable or asymptotically stable according to eigenvalues. Adopted from [45].

- When eigenvalues are complex and conjugated, $x = C_1 e^{\lambda t}(a\,cos(\mu t) - b\,sin(\mu t)) + C_2 e^{\lambda t}(a\,sin(\mu t) + b\,cos(\mu t))$.

    - λ has a zero-real part then the trajectories never converge to the critical point but never go to infinite as well. They either are constant, elliptical or orbits.

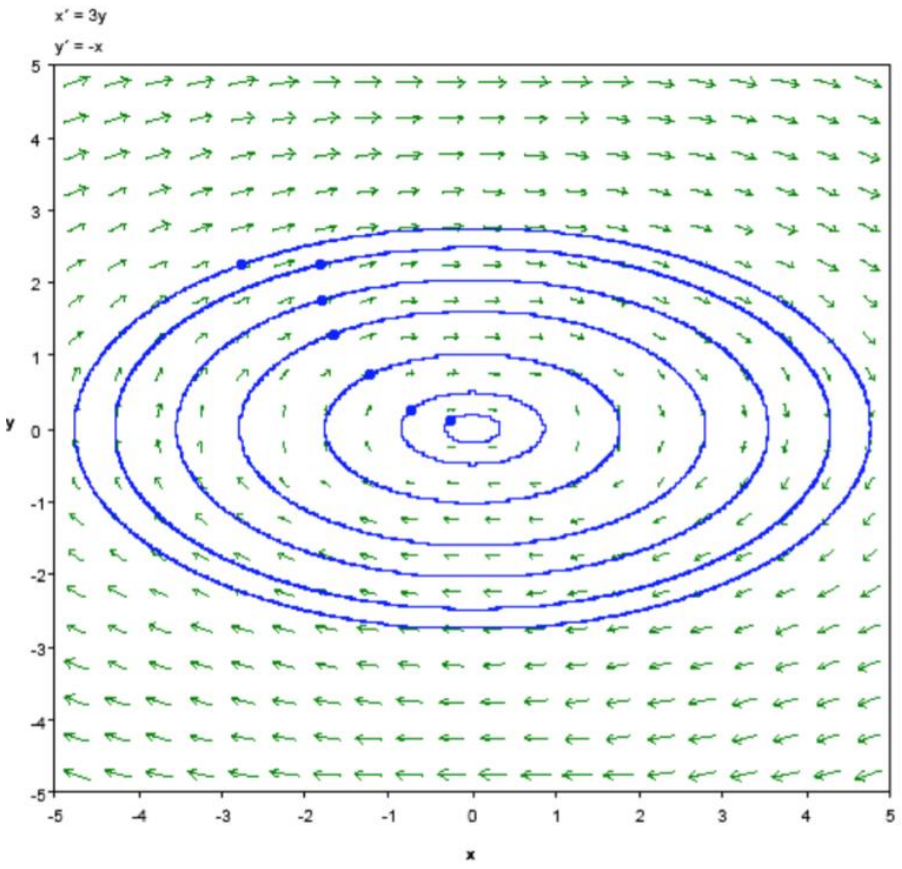


Figure 7: Represents the analysis from above. We call it a center, this is always stable. Adopted from [45].

- λ has a nonzero real part then the trajectories are still elliptic but the distance to the critical point or to the infinite grows or decays.
  - $Re(\lambda) > 0$, the trajectory goes to infinite in spirals and the trajectory is unstable.
  - $Re(\lambda) < 0$, the trajectory converges to the fixed point in spirals and the trajectory is asymptotically stable.

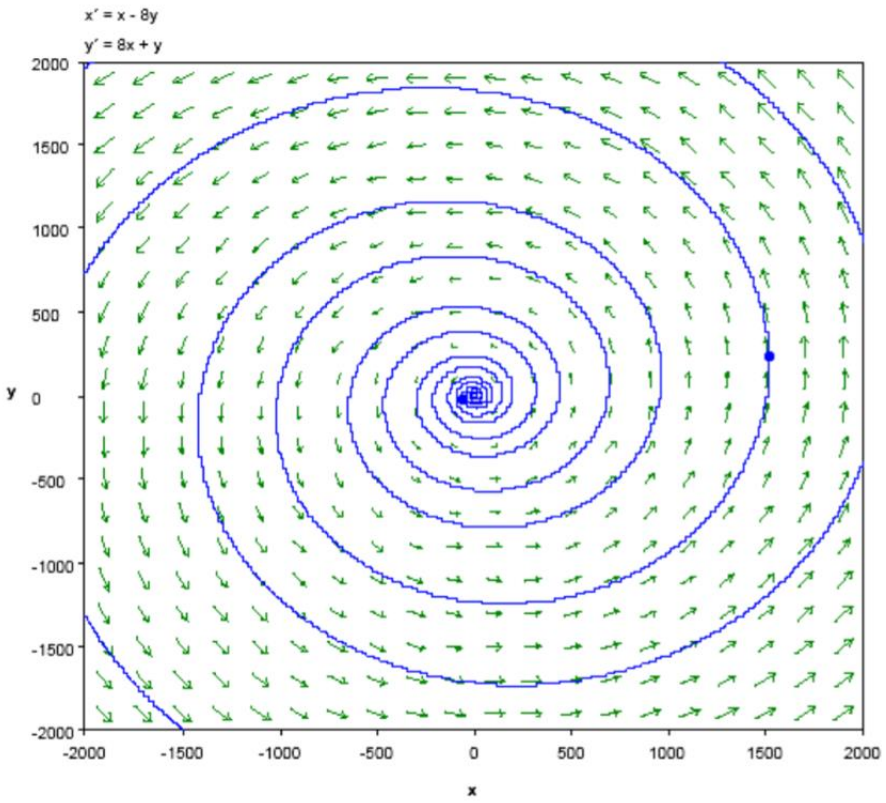


Figure 8: Represents the analysis from above. We call it a spiral, this is unstable or asymptotically stable according to eigenvalues. Adopted from [45].

To summarize the linear system analysis, it is a description of the different equilibriums obtained looking at the trace and the determinant of the state matrix A.

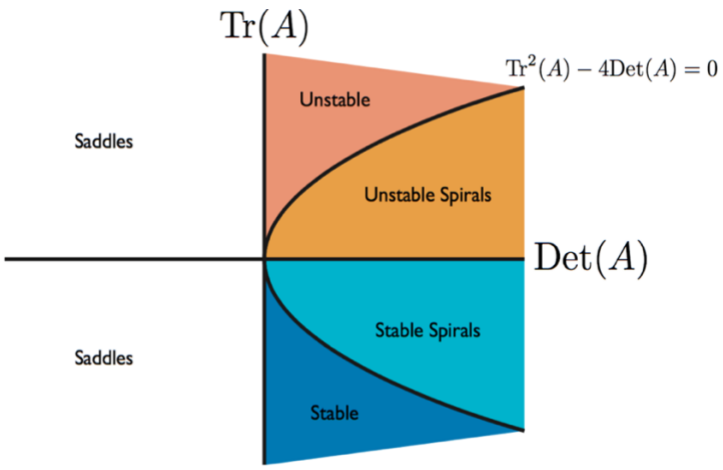


Figure 9: Types of nodes according to the trace and the determinant of the state matrix ***A*.** Adopted from [31].

2.2.1.2 Nonhomogeneous Linear Systems with Constant Coefficients

There is a last part we need to consider here, this is when the linear system is nonhomogeneous where b is a constant vector:

$$\dot{x} = Ax + b. \tag{2.2.5}$$

The system is then written:

$$\begin{cases} \dot{x}_1 = a_{11}x_1 + a_{12}x_2 + b_1 \\ \dot{x}_2 = a_{21}x_1 + a_{22}x_2 + b_2. \end{cases} \tag{2.2.6}$$

The critical point is no longer located at the origin but at the solution of this linear system. Then when $\dot{x_1} = 0$, and $\dot{x_2} = 0$. In order to find the A matrix, we first find out the new critical point (α, β) and then we wrote the new system using an update of $x_1$and x_2.

$$\dot{x'} = Ax' \tag{2.2.7}$$

$$\begin{cases} \dot{x'}_1 = x_1 - \alpha \\ \dot{x'}_2 = x_2 - \beta. \end{cases} \tag{2.2.8}$$

The two systems (2.2.5) and (2.2.7) have the same A matrix and then we can solve this equation as a homogeneous linear equation. With this whole analysis, we are now able to identify most of the simple trajectories possible around a critical point.

2.2.2 Nonlinear Systems

In the case of nonlinear system, there is no general analytic solution helping us to save any nonlinear system. In this case, a good alternative is to phase portraits. These are simple graphical tool used to visualize how solutions of a given set of differential equations behave in time. Using eigenvalues and eigenvector, we classify the different stabilities and the different equilibrium points we can find in a given system. Then, we can draw the shape and behavior of the trajectories starting from different initial conditions along time.

2.2.2.1 Phase portraits of Nonlinear Systems

$$\begin{cases} \dot{x}_1 = F(x_1(t), x_2(t)) \\ \dot{x}_2 = G\big(x_1(t), x_2(t)\big). \end{cases} \tag{2.2.9}$$

When $F(x)$ and $G(x)$ are two nonlinear functions of two variables. Recalling from above, a good start to study any system is to find out the equilibrium points and to describe the behavior of their trajectories. In the nonlinear case, the system can have from 0 to an infinite amount of critical points. The phase portrait is then, harder to interpret since every trajectory might be influenced by more than one critical point. When working with an ordinary nonlinear system, our best call is to give a look at the local behavior [26]. Indeed, for many types of critical points, it has been proven than close to an equilibrium point, a nonlinear system can qualitatively be determined by the behavior of the linear system [32]:

$$\dot{x} = Ax|_{x_0} \tag{2.2.10}$$

where A is called the linear part of F at a critical point x_0, and is defined as the Jacobian matrix applied to a critical point x_0.

$$A = J|_{x_0} = \begin{bmatrix} \frac{\partial F(x_1,x_2)}{\partial x_1}(x_0) & \frac{\partial F(x_1,x_2)}{\partial x_2}(x_0) \\ \frac{\partial G(x_1,x_2)}{\partial x_1}(x_0) & \frac{\partial G(x_1,x_2)}{\partial x_2}(x_0) \end{bmatrix}. \tag{2.2.11}$$

Thus, we apply a linearization of the system around every critical point. In other terms, we compute the Jacobian matrix for each critical point, and we write equation(2.2.9) as a linear form equation in (2.2.10). For instance, if $x_0 = (x_{1_0}, x_{2_0}) = (\alpha, \beta)$ depicts a specific critical point, then the linearization gives us the following expressions:

$$\begin{cases} \dot{x_1} = F(x_1,x_2) = F_{x_1}(\alpha,\beta)(x_1-\alpha) + F_{x_2}(\alpha,\beta)(x_2-\beta) \\ \dot{x_2} = G(x_1,x_2) = G_{x_1}(\alpha,\beta)(x_1-a) + G_{x_2}(\alpha,\beta)(x_2-\beta). \end{cases} \tag{2.2.12}$$

Then we compute the Jacobian matrix applied on (α, β). Finally, we just have to follow the method in the previous section to characterize all the different critical points for linear system.

$$A = J = \begin{bmatrix} \frac{\partial(\alpha,\beta)}{\partial x_1} & \frac{\partial F(\alpha,\beta)}{\partial x_2} \\ \frac{\partial G(\alpha,\beta)}{\partial x_1} & \frac{\partial G(\alpha,\beta)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} F_{x_1}(\alpha,\beta) & F_{x_2}(\alpha,\beta) \\ G_{x_1}(\alpha,\beta) & G_{x_2}(\alpha,\beta) \end{bmatrix}.$$

2.2.3 Limit Cycles

At this point, we studied many of the simplest different critical points for nonlinear system, we expect the trajectories of a critical point's neighbors to be similar to a the one of the critical point. Another important possibility which can influence how the trajectories look is if we find a trajectory tracing a closed curve C. In this case, points in the closed curve round around the curve for all t with a period T.

$$\begin{aligned} x_1(t+T) &= x_1(t) \\ x_2(t+T) &= x_2(t). \end{aligned} \tag{2.2.13}$$

If there is such an isolated closed curve, we call it C a limit cycle, it can be either stable when the neighbor trajectories spiral around and toward the limit cycle, unstable when they spiral away the limit cycle or even semi-stable when we find trajectories that spiral toward and other away of C.

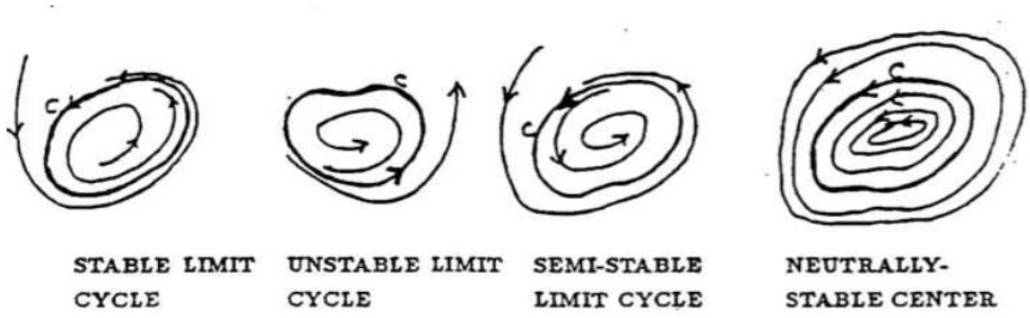


Figure 10: Different types of limit cycles. Adopted from [31].

A lot of periodic processes in nature can be described as stable limit cycles, so knowing how to find out and study limit cycle is necessary. Specifically, in this thesis, we aim to find a pattern of oscillations in the evolution of the membrane potential of postsynaptic neurons in a network. As much as limit cycles might be important to study the dynamics of nonlinear system, this is a current subject of research and the theory behind the analysis of limit cycle is not always really convenient. The Poincare-Bendixson criterion allows to prove the existence of a limit cycle in a two-dimensional space, and some other theorems can either prove the non-existence of limit cycles or talked about the number of limit cycles found in specific systems.

3.0 METHODS

3.1 COMPUTATION OF AN AUTONOMOUS SPIKING NEURAL NETWORK

In the context of neural network's principles previously discussed, we addressed the issue of understanding how discontinuous spikes and action potentials can describe continuous motions of our bodies. To answer this problematic, researchers are currently studying spiking networks. Our approach consists to take the phase plane analysis of a promising spiking network in order to bring some light on the spike dynamics. Our work focuses on the exact spike timing. To do so, we choose to work on the following spiking network based and built on exact spike timing.

3.1.1 Single Input Dimension

3.1.1.1 A Spiking Neural Network Defined as an Optimization Problem

The problem we are addressing is the following: we want to represent a simple one-dimensional signal $f(t)$ in the output activity of a spiking neural network. Then, we built a set of weights given by J describing synapses between neurons. This means that, neuron's spikes, answers to the input signal, are filtered and summed to define the output. Moreover, as previously stated, $f(\mathrm{t})$ is assumed to be decodable after synaptic integration. Let's define the response of neuron i $(i = 1,2 \dots, N)$ to the current striking its membrane potential to be the spike train:

$$\delta_i(t) = \sum_k \delta(t - t_i^k) \tag{3.1.1}$$

with $\delta(.)$, a Dirac function and $\{t_i^k\}$, the spike times of neuron i. From the spike train, we define the normalized synaptic current, $s_i(t)$ to be its filtered version using an exponential filter:

$$s_i(t) = \delta_i * e^{-\frac{t}{\tau_s}}. \tag{3.1.2}$$

We call, $e^{-\frac{t}{\tau_s}}$, the decaying exponential kernel with τ_s, the timescale of the filter. Now, we assume that neural membrane has capacitive properties that brings a temporal filtering of its input and gives the following dynamical equation for the filtered spike train:

$$\tau_s \frac{ds_i}{dt} = -s_i + \tau_s \delta_i. \tag{3.1.3}$$

Note that this equation depicts a model of simplified postsynaptic potential so that each presynaptic spike from neuron i causes the normalized synaptic current to increase instantaneously by 1 and decay exponentially to 0 with a time constant τ_s between spikes. The network output is defined as, $z(t)$, a weighted sum of the normalized synaptic currents.

$$z(t) = \sum_{i=1}^{N} \omega_i s_i(t). \tag{3.1.4}$$

Where ω_i is the fixed contribution of neuron i to the network's output. We remind that the goal of this experiment is to figure out a dynamical model to describe a neural network producing appropriate spike trains at appropriate times in order to provide an accurate representation of the input signal $f(t)$.

What should be an individual neuron's response to contribute to the collective cause? This is one of the question answered in [6], they came up with the idea to solve the classic credit-assignment problem for this network. An intuitive way to progress in the derivation of our network is to notice we can view this problematic as a simple optimization problem. It prevents to use any

gradient based method that we recall hard to use with SNNs. Therefore, let's define a loss function, as the cumulative mean squared decoding error:

$$E(to) = \int_{t_0}^{T} [f(t) - z(t)]^2 \, dt. \tag{3.1.5}$$

An important subject to keep in mind is that optimization is focused over the spike times. We want to find the minimal set of spike times so that $z(t)$ follows $f(t)$. Weights should stay fixed, they are a-priori chosen and always known. So, the following optimization problem is given:

$$\min_{\delta_1 \delta_2 \ldots, \delta_N} E(t_0). \tag{3.1.6}$$

The main idea is to use the previously stated assumption. Neuron i should fire a spike at time t if and only if its firing would reduce the decoding error at time t. This relates a greedy minimization of the cost function $E(t_0)$ using the previous specific spike rule. By tracking, the evolution of the cost term along the dynamic of the neural network,

$$\big(f(t) - z(t)\big)^2 < \big(f(t) - \bar{z}(t)\big)^2 \tag{3.1.7}$$

We find the dynamical equation for the output $z(t)$. Using past assumptions and definitions, we get:

$$\begin{aligned} \frac{dz}{dt} &= \sum_{i=1}^{N} \omega_i \frac{ds_i}{dt} \\ \tau_s \frac{dz}{dt} &= -z + \sum_{i=1}^{N} \tau_s \omega_i \, \delta_i \ . \end{aligned} \tag{3.1.8}$$

As in Eq. (3.1.3), the output z is increased instantaneously by ω_i for each spike of neuron i and then decreases exponentially to 0 with a constant τ_s between spikes. Eventually, the network derives his own spike rule where we respectively, define a membrane voltage v_i and a threshold T_i for each neuron i as in the leaky integrate and fire model,

$$E(t|neuron\ i\ spikes) < E(t|no\ spikes)$$

$$\begin{cases} v_i(t) = \omega_i\ (f - z) \\ T_i = \dfrac{\omega_i^2}{2}. \end{cases} \tag{3.1.9}$$

Neuron i only fires when $v_i(t)$ is larger than $\frac{\omega_i^2}{2}$ and the dynamical equation for the membrane potential $v_i(t)$ is computed:

$$\tau_s \frac{dv_i}{dt} = -v_i + \omega_i \left(\tau_s \dot{f} + f\right) + \sum_{j=1}^{N} \left(-\tau_s \omega_i \omega_j\right) \delta_j. \tag{3.1.10}$$

As an analogy with the dynamical equation of the output, our network implements a self-reset of its membrane potential after each spike. Indeed, the equation implies that when neuron i fires, it causes the membrane voltage to decrease instantaneously by ω_i^2, that mean to reset its membrane potential reset to a value $v_i = T_i - \omega_i^2 = -T_i$. We can, now, be sure that this network is derived from LIF neurons (2.1.1).

3.1.1.2 Starting from Driven Network to Autonomous System

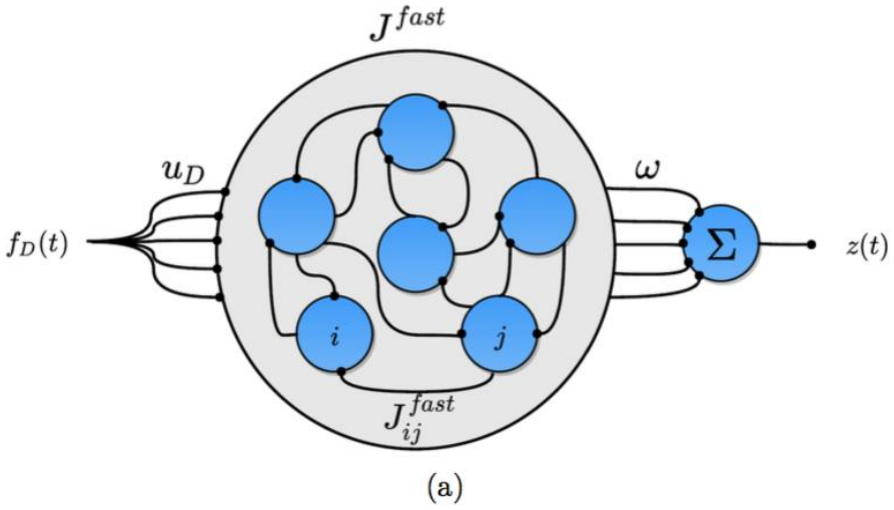
(a) Driven Network Computation

The goal of a spiking neural network is to reproduce a specific activity through its spiking dynamic. In that sense, when we need to evaluate the network, we make use of a driven function f_D, that will intrinsically depend of the original input f, to get the desired output. The mathematical development we used above, allowed to find a good set of decoding synapses $\left(-\tau_s \omega_i \omega_j\right)$ in order to get the right output. Now the ultimate goal for a SNN would be to be autonomous, meaning that the network doesn't need the original input f anymore, but needs a simple input signal to generate the driven function by itself and get to the desired output. To accomplish that, we need to compute

the set of recurrent connections that accomplishes the task with a reasonable degree of accuracy and by keeping the network dynamic stable.

Let's first introduce the function $f_D(t)$ as an input to drive the network to the desired output function. Figure 11 sets the model and call:

$f_D(t)$ through the weights $u_D(t)$, a set of synapses strength between all neurons of the model in a $N * N$ matrix, commonly named J^{Fast}, and finally call the fixed decoding ω. When running, an accurate representation $z(t)$ should be given as an output of the signal $f(t)$. Afterwards, to get the autonomous network, we want to observe neuron's behavior when the network track the error function thanks to the driven function. In other words, the driven function provides targets for building an autonomous network.



(a)

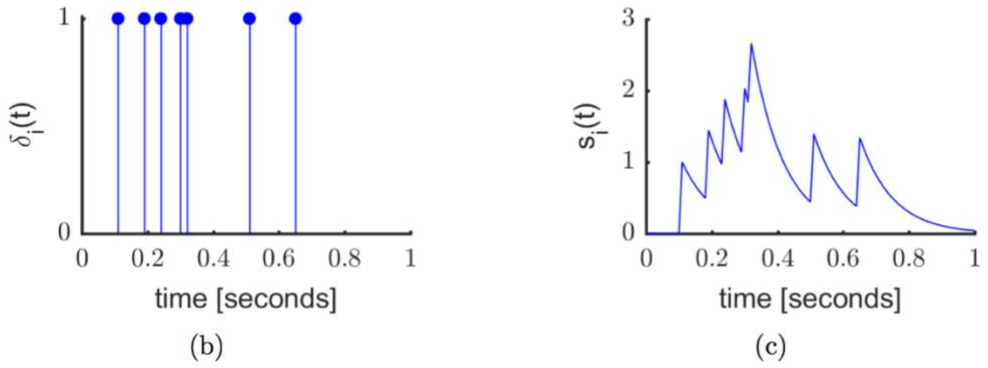


Figure 11: (a) Structure of the driven network. A defined input $\boldsymbol{f_D(t)}$ is provided to the network through weights $\boldsymbol{u_D}$. Neurons in the network are connected through synapses whose strengths are defined by the matrix $\boldsymbol{J^{Fast}}$. Specifically, the synaptic current generated in a post-synaptic neuron $\boldsymbol{i}$ by a presynaptic neuron $\boldsymbol{j}$ is given by the synaptic weight, $\boldsymbol{J_{ij}^{Fast}(t)}$ multiplied by the normalized synaptic current $\boldsymbol{s_i(t)}$. The output of the network is read out by summing, the normalized synaptic currents of the neurons with weights ω. (b) Example of a spike train from neuron $\boldsymbol{i}$, $\boldsymbol{\delta_i(t)}$ and (c) depicts the corresponding normalized synaptic current with $\boldsymbol{\tau_s = 100}$ msec.

The next goal is now to find good driven function to the network. Hopefully, using a high-pass filter, phase advanced version of the output, the driven network eventually treats the error pretty well:

$$given\ input: f_D(t) = \tau_s \dot{f} + f \tag{3.1.11}$$

$$Fast\ synaptic\ connection: J_{ij}^{fast} = -\tau_s \omega_i \omega_j \tag{3.1.12}$$

$$input\ weigths: u_D. \tag{3.1.13}$$

We now, appreciate the expressions of u_D and J_{ij}^{fast} in terms of ω. However, f_D is still in terms of the signal f. This auto-encoder receives the desired signal $f(t)$ as an input and represents it on the neuron's activity, then recovers $f(t)$ at the output, by decoding the spike trains.

(b) Autonomous Network Computation

For autonomous network, we need to find an input that can be self-generated by the network. Fortunately, we are able to find such an input when f is generated from a linear dynamic system, where A is the desired dynamic and $c(t)$ a control signal:

$$\tau_s = \dot{f} - f + Af + c(t). \tag{3.1.14}$$

Using this new form for f, we re-write our general dynamic equation for the membrane potential:

$$\tau_s \frac{dv_i}{dt} = -v_i + \sum_{j=1}^{N} \omega_i A \omega_j s_j \sum_{j=1}^{N} \left(-\tau_s \omega_i \omega_j\right)\delta_j + \omega_i c. \tag{3.1.15}$$

We have the same fast-synaptic connection than for a driven network, $J_{ij}^{fast} = -\tau_s \omega_i \omega_j$, we also have what we will call the slow synaptic connection $J_{ij}^{slow} = \omega_i A \omega_j$ since this new set of connection only applies on normalized synaptic currents which are, due to their decaying kernel of time constant τ_s, much slower than the dynamic of spike train tuned by J_{ij}^{fast}. Thanks to these two sets of connections, the network is able to predict the future trajectory of $f(t)$.

Even if $f(t)$ is a constant function, ($\dot{f} = 0$, $A = 1$, and $c(t) = 0$) then these two sets of connections still maintain activity in the network preventing $f(t)$ from decaying to zero. On Figure 12, we describe the system as an encoder of f, then implementing J^{Slow}, we make the system autonomous. Doing so we use a function f respecting: $\tau_s \dot{f} = -f + Af + c(t)$ as discussed above. This comes back to get the previous output tuned with the desired dynamic as an input.

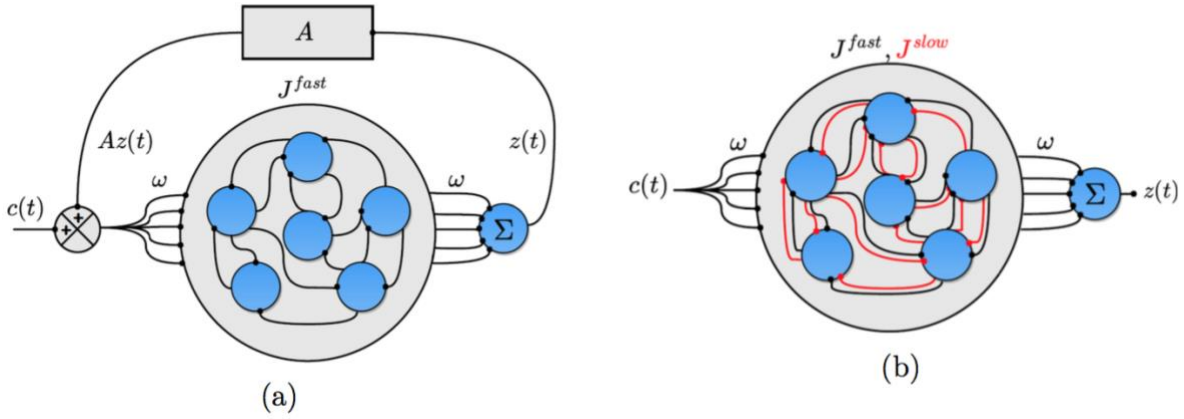


Figure 12: (a) Schematic highlighting the output feedback mechanism used to generate the input in terms of the output, assuming $\boldsymbol{f(t)}$ satisfy Eq. (3.1.14**)** (b) Schematic illustrating an equivalent network implemented using an extra set of synapses ($\boldsymbol{J^{Slow}}$) with strengths given by Eq.(3.1.15**)**. Red connections represent slow synaptic connections and black connections fast ones. $\boldsymbol{c(t)}$ is the control signal.

3.1.2 Multidimensional Input Function

So far, we described a way to create an autonomous network that can mimic a function $f(t)$ only when it is a scalar function. In this case, we can only describe a low-pass filter or an integrator. input and as an output to implement sensory responses and real biologically realistic scenario. The extension of the previous work is straightforward, since we follow exactly the same steps using now a vector $f(t)$, a matrix ω and a vector $z(t)$. The system is autonomous when $f_k(t)$ are generated by:

$$\tau_s \dot{f}_k = -f_k + \sum_{p=1}^{k} A_{kp} f_p + c_k(t). \qquad (3.1.16)$$

This gets us the following membrane voltage dynamical equation for an autonomous system, associated to its spiking rule:

$$\tau_s \frac{dv_i}{dt} = -v_i + \sum_{k=1}^{K} \omega_{ik} \sum_{p=1}^{K} A_{kp} \sum_{j=1}^{N} \omega_{pj} s_j - \tau_s \sum_{k=1}^{K} \omega_{ik} \sum_{j=1}^{N} \omega_{kj} \delta j - \tau_s \mu \delta_i + \sum_{k=1}^{K} \omega_{ik} ck. \quad (3.1.17)$$

Where the fast synapses are described by $J_{ij}^{fast} = -\tau_s \sum_{k=1}^{K} \omega_{ik} \omega_{kj} + \mu \delta j$ and the slow ones are described by $J_{ij}^{snow} = \sum_{k=1}^{K} \sum_{p=1}^{K} \omega_{ik} A_{kp} \omega_{pj}$.

This network can implement any linear dynamical system autonomously of the form: $\tau_s \dot{f} = -f + Af + c$ as long as A is chosen appropriately. In matrix notation, we have the following respectively spiking rule and dynamic:

$$\boldsymbol{T} = \frac{1}{2}(diag(\boldsymbol{\Omega}^{\mathbf{T}}\boldsymbol{\Omega}) + \mu \boldsymbol{I}) \quad (3.1.18)$$

$$\tau_s \dot{\boldsymbol{v}} = -\boldsymbol{v} + \boldsymbol{\Omega}^{\text{Slow}} \boldsymbol{s} + \boldsymbol{\Omega}^{\text{Fast}} \boldsymbol{\delta} + \boldsymbol{\Omega}^{\mathrm{T}} \boldsymbol{c} \quad (3.1.19)$$

$$\tau_s \dot{\boldsymbol{s}} = -\boldsymbol{s} + \tau_s \boldsymbol{\delta} \quad (3.1.20)$$

$$\boldsymbol{\Omega}^{\text{Fast}} = -\tau_s(\boldsymbol{\Omega}^{\mathbf{T}}\boldsymbol{\Omega} + \mu \boldsymbol{I}) \quad (3.1.21)$$

$$\boldsymbol{\Omega}^{\text{Slow}} = \boldsymbol{\Omega}^{\mathbf{T}} \mathbf{A} \boldsymbol{\Omega}. \quad (3.1.22)$$

3.1.2.1 Balanced neural network

Biologically speaking, neurons are not all the same, an important differentiation between two types of neurons can be made. Excitatory and inhibitory neurons have different neurotransmitters that bind different receptors. Although we still don't know all the different neurotransmitters and their roles, excitatory neurons are often associated with a positive weight since they trigger a positive change in the membrane potential when inhibitory neurons get a negative one for triggering a negative change [24].

Dale's law implies that a neuron only can release the same neurotransmitter to all of the other units. In this case, Dale's law is violated. Then to fix this issue, we make use of a balanced network separating the total population in two groups, one excitatory and another one inhibitory. When studying the dynamics of large-scale populations of LIF neurons, the balanced random network is commonly used, [9]. In this framework, we think network as balanced regime of excitatory and inhibitory neurons. Balanced networks have first been theoretically highlighted [13], then it has also been experimentally proven *in vivo* [23]. The collaboration or opposition, depending of the point of view, act to keep the average activity under the threshold and then keep the activity of the whole network stable. The balanced network provokes irregularities in spiking that can be found in a brain as well. More details on the dynamics of balanced network can be found in [9]. In this section, we derive a balanced network to comply with Dale's rule. Before this we need to assume what are the different population goals, we assume the excitatory population track the actual signal $x(t)$ and the inhibitory population tracks the estimate of the excitatory one.

(a) Inhibitory Membrane Potential

As mentioned above, the inhibitory population is tracking the estimation of the excitatory one z_E.

$$E_I(to) = \int_{t_0}^{T} [z_E(t) - z_I(t)]^2 \, dt. \tag{3.1.23}$$

Following the exact same steps described earlier, we get the following spike rule associated with the following membrane potential all determined for an inhibitory population.

$$\begin{cases} v_{I,i}(t) = \omega_{I,i} \, (z_E - z_I) \\ T_{I,i} = \dfrac{\omega_{I,i}^2}{2}. \end{cases} \tag{3.1.24}$$

$$\tau_s \frac{dv_{I,i}}{dt} = -v_{I,i} + \tau_s[\omega_{I,i} \sum_{j=1}^{N_E} (\omega_{E,j}) \delta_{E,i} - \omega_{I,i} \sum_{j=1}^{N_I} (\omega_{I,j}) \delta_{I,i}]. \tag{3.1.25}$$

The dynamic of the inhibitory membrane potential now obeys Dale's law since the input of the excitatory neurons depolarizes the voltage when the input of inhibitory neurons hyperpolarizes the voltage.

(b) Excitatory Membrane Potential

In a similar way, we now assume that excitatory neurons track the general input, the general error becomes:

$$E_E(t) = \int_{t_0}^{T} [f(t) - z_E(t)]^2 + \, dt. \tag{3.1.26}$$

The spike rule and the dynamic of an excitatory membrane potential is given in the following:

$$\begin{cases} v_{E,i}(t) \; = \; \omega_{E,i} \, (f - z_E) \\ \quad T_{E,i} = \dfrac{\omega_{E,i}^2 + \mu}{2}. \end{cases} \tag{3.1.27}$$

$$\tau_s \frac{dv_{E,i}}{dt} = \; - v_{E,i} \; + \omega_{E,i} \sum_{j=1}^{N_I} A\omega_{I,j} \, s_{I,i} \; - (\tau_s \; \omega_{E,i} \sum_{j=1}^{N_I} (\omega_{I,j})) \delta_{I,i} + \; \omega_{E,i} c_i. \tag{3.1.28}$$

In the spiking neural network developed above, the coding spike rule chosen leads to balance networks:

- When two neurons are said similar, meaning $\omega_i \omega_j > 0$, a spike from neuron i inhibit the similar neuron, instantaneously decreases its voltage membrane by an amount $\omega_i \omega_j$ and resets its own to T_i.
- However, when two neurons are not similar $\omega_i \omega_j < 0$, then a spike from neuron i exhibit the not similar neuron, instantaneously increases its voltage membrane by $\omega_i \omega_j$ and resets its own to T_i.

3.2 MEAN-FIELD ANALYSIS

3.2.1 From Current-Based to Conductance-Based Model

3.2.1.1 Introduction to Conductance-Based Model

In section 3.1, we chose a spiking neural network that is driven by current-based synapses only. However, biologically neurons modify their behavior and their firing rate through the opening or closing of different channels and gates. In 2016, a mapping between current-based and conductance-based synapses has been published [35]. Their results highlight the big realistic difference between both models. Current-based synapses substantially affect the network stability, mostly when the input described in the background synaptic activity is particularly noisy. An equivalent spiking model but as a conductance-based model would be the following [44]:

$$\frac{C_i dV_i(t)}{dt} = g_i^0 \left(V_i^0 - V_i(t)\right) + g_i^K \left(V_i^K - V_i(t) + \sum_\alpha g^\alpha(t)\left(V^\alpha - V_i(t)\right) + I_i(t) \right). \tag{3.2.1}$$

Where C_i is the capacity of the cell membrane, g_i^0 its passive conductance, V_i^0, the resting potential, g_i^K, the active potassium conductance that produce firing adaptation, g^α, the conductance of each input synapse α, V_i^K and V^α, the different equilibrium potentials associated with the respective conductance and $I_i(t)$, the input transmits to the cell.

As in the main model described in this thesis, when $V_i(t)$ reached a threshold V_i^{th}, the neuron fires a spike and its membrane potential directly reset to a reset value V_i^{reset} due to a repolarization of conductances:

$$\frac{dg^\alpha(t)}{dt} = -\frac{g^\alpha(t)}{\tau^\alpha} + \Delta g^\alpha \sum_k \delta(t - \Delta t - t_{k,j_\alpha}) \tag{3.2.2}$$

$$\frac{dg^K(t)}{dt} = -\frac{g_i^K(t)}{\tau^K} + \Delta g^K \sum_k \delta(t - t_{k,i}). \quad (3.2.3)$$

These two equations genuinely describe the polarization of conductances after a spike, a respective fixed amount is instantaneously added to the conductance over time when an exponential relaxation term associated to a specific time constant brings back the conductance to its resting state. Despite all these new conductance features, this simple integrate and fire model is still not enough to reproduce complex neuronal behavior but it allows us to include the basic features of a neuronal network and specifically the quality of adaptation in frequency of the network. In order to map both models, we first need to apply the different assumptions we made earlier to this model. For instance, $g^{\alpha}(t)$ that represents the set of decoding weights is chosen and a-priori fixed. Before mapping, let's divide Eq. (3.2.1) in two classes of neurons: excitatory and inhibitory. As in the main paper, subscripts F, G, will represent either excitatory or inhibitory population. Let's keep our notations simple adding a subscript x_F, with F that could be either E or I meaning either excitatory or inhibitory population. Also, the general set of equation has been rescaled as follow:

$$x = \frac{V - V_F^{reset}}{V_F^{th} - V_F^{reset}}. \quad (3.2.4)$$

The firing rates variable determined by the potassium conductance is rescaled as well:

$$y = \frac{gi^K}{\Delta g_i^K}. \quad (3.2.5)$$

After rescaling all parameters, here is the final dynamics of the membrane potential of this conductance-based model:

$$\begin{aligned}\frac{dx_{E,i}(t)}{dt} = &-x_{E,i}(t)[\omega_E^0 + \omega_E^E z_E^E + \omega_E^I z_E^I - \omega_E^S s_E] + \omega_E^K x_{E,i}^K(t) y_{E,i}(t) \\ &+ \omega_E^S x_{E,i}^E(t) s_E + \omega_E^0 x_{E,i}^0(t) + \omega_E^E z_E^E x_{E,i}^E(t) + \omega_E^I z_E^I x_{E,i}^I(t)\end{aligned} \quad (3.2.6)$$

$$\frac{dx_{I,i}(t)}{dt} = -x_{I,i}(t)[\omega_I^0 + \omega_I^E\, z_I^E + \omega_I^I z_I^I - \omega_I^S s_I] + \omega_I^K x_{I,i}^K(t) y_{I,i}(t) + \omega_I^S x_{I,i}^E(t) s_I + \omega_I^0 x_{I,i}^0(t) + \omega_I^E\, z_I^E x_{I,i}^E(t) + \omega_I^I z_I^I x_{I,i}^I(t) \quad (3.2.7)$$

$$\tau^K \frac{dy_{E,i}(t)}{dt} = -y_{E,i}(t) + \tau^K \sum_{j=1}^{N_E} \delta_E(t - t_i^j) \quad (3.2.8)$$

$$\tau^K \frac{dy_{I,i}(t)}{dt} = -y_{I,i}(t) + \tau^K \sum_{j=1}^{N_I} \delta_I\left(t - t_i^j\right). \quad (3.2.9)$$

Defining two variables, we re-write the dynamics:

$$A_{F,i}(t) = \omega_F^S x_{F,i}^E(t) s_F + \omega_F^0 x_{F,i}^0(t) + \omega_F^E\, z_F^E x_{F,i}^E(t) + \omega_F^I z_F^I x_{F,i}^I(t) \quad (3.2.10)$$

$$B_{F,i} = [\omega_F^0 + \omega_F^E\, z_F^E + \omega_F^I z_F^I - \omega_F^S s_F] \quad (3.2.11)$$

$$C_{F,i}(t) = \omega_F^K x_{F,i}^K(t). \quad (3.2.12)$$

Then we get:

$$\frac{dx_{F,i}(t)}{dt} = A_{F,i}(t) - B_F x_{F,i}(t) + C_{F,i}(t) y_{F,i}(t) \quad (3.2.13)$$

$$\tau^K \frac{dy_{F,i}(t)}{dt} = -y_{F,i}(t) + \tau^K \sum_{j=1}^{N_F} \delta_F\left(t - t_i^j\right). \quad (3.2.14)$$

3.2.2 Population Measure Tool

One of the main idea we exported is to compute the variable they called the density. As we will develop later in the result section of this thesis, this variable represents the probability of a neuron in the network to spike at a defined time. This mean field analysis's tool seen as the momentary state of the network at time t opens the door to new dynamic analyses,

$$\rho_F(x,t) = \frac{1}{N_F}\sum_i \delta(x - x_i(t)). \qquad (3.2.15)$$

4.0 RESULTS

4.1 ANALYSIS OF THE SPIKE-CODING NETWORK

After the previous chapters, we derived a network of leaky integrate-and-fire neurons governed by the following dynamical equations that we want to remind before applying a mean field analysis:

$$\tau_s \frac{dv_i}{dt} = -v_i + \sum_{j=1}^{N} \omega_i A \omega_j s_j + \sum_{j=1}^{N} \left(-\tau_s \omega_i \omega_j\right)\delta_j + \omega_i c \qquad (4.1.1)$$

$$\tau_s \frac{ds_i}{dt} = -s_i + \tau_s \delta_i \qquad (4.1.2)$$

where $J_{ij}^{slow} = \omega_i A \omega_j$ and $J_{ij}^{Fast} = -\tau_s \omega_i \omega_j$ are the recurrent connections between neurons $\delta_i(t) = \sum_k \delta\left(t - t_i^k\right)$ is the spike train of neuron i with spike times $\{t_i^k\}$. Figure 13 shows the computation of this model when following an input signal $c(t)$. Part (b) of the figure show the computed density used a measure of the probability of the population to spike at an instant $t + \Delta t$. For visual reasons, we decided to use an important bin, 25 steps per bin.

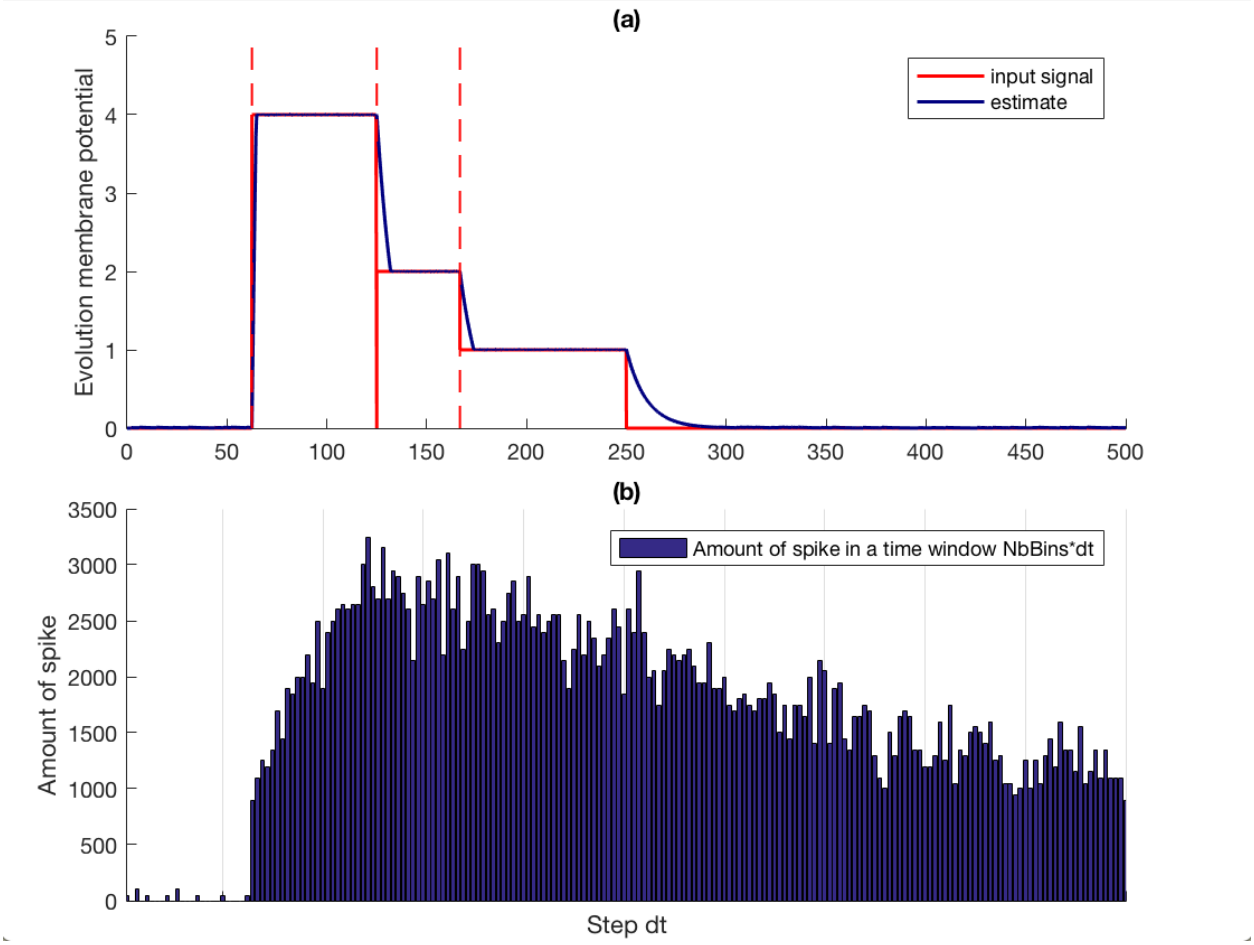


Figure 13: (a) Spiking neural network trained on spike times tracking a specific input signal. (b) Histogram that depicts the density $\boldsymbol{\rho}\ (\boldsymbol{v}, \boldsymbol{t})$ with a size bin of 25 steps per bin.

4.1.1 Periodic Behavior of the Isolated Neuron

For simplicity, let's start with the analysis of a single isolated neuron in this network. The network we provided in the previous sections contains an adaptive current $s(t)$ fed back to the neurons associated with a slow variable. This mechanism is also known as spike-triggered adaptation. Biologically, it depicts a more complex mechanism of opening or closing calcium gates in the neuron in order to monitor its own spike train. Thanks to this mechanism, it is possible to analyze

the dynamics of a single isolated neuron in the network. According to our previous work, we describe the dynamics of a single neuron using the following:

$$\tau_s \frac{dv(t)}{dt} = -v(t) + \alpha s(t) - \tau\omega^2 \sum_k \delta(t - t^k) + \omega c \tag{4.1.3}$$

$$\tau_s \frac{ds(t)}{dt} = -s(t) + \tau_s \sum_k \delta(t - t^k) \tag{4.1.4}$$

with α defined as $\omega^2 A_j$. As seen earlier when $t = t^k$, we suppose a neuron has fired and in this case following the leaky integrate-and fire model, the membrane voltage is reset to a certain value ($(v(t^k)^+) = -\vartheta = -\frac{\omega^2}{2}$) and the normalized synaptic current $s(t^k) = s^k$. Let's define the interspike interval value $T(s^k)$ as the amount of time required for the neuron to reach the threshold again after firing. By solving this system between spikes, we obtain the new set of solutions valid for $t^k < t \leq t^{k+1} = T(s^k)$:

$$v(t) = -\vartheta e^{-\frac{t-t^k}{\tau}} + \frac{\alpha s^k}{\tau}(t - t^k)\, e^{-\frac{t-t^k}{\tau}} + \omega \int_0^{t-t^k} e^{-\frac{s}{\tau}} c(t - s) ds \tag{4.1.5}$$

$$s(t) = s^k e^{-\frac{t-t^k}{\tau}}. \tag{4.1.6}$$

Proceeding, at time $t = t^{k+1}$, the neuron will fire again, and we have $v(t^{k+1}) = \vartheta = \frac{\omega^2}{2}$. We remember from Eq.(3.1.3) that when a neuron fire, the adaptation mechanism directly implies that feedback current increases by 1 as follows: $s(t^k) \to s(t^k) + 1$. Therefore, we can derive the following map for the adaptation current:

$$s^{k+1} = s(t)|_{t=t^{k+1}} + 1 = s^k e^{-\frac{T(s^k)}{\tau}} + 1. \tag{4.1.7}$$

Hence, we can draw a map in the phase-plane spanned by v and s where each of this solution describes a 'sub-trajectory' starting from the reset potential and finishing at the threshold potential,

let's respectively call them Γ^k. When the input c is big enough to drive the membrane potential towards the threshold, it makes the exponential decay to become more and more important when k increases. At some point when k goes to infinity, the overall trajectory converges to the limit cycle Γ^*. Furthermore, since each sub-trajectory Γ^k is uniquely determined by its starting current s^k, the feedback current also converges toward a certain value s^*.

$$\lim_{k\to\infty} \Gamma^k = \Gamma^* \tag{4.1.8}$$

$$\lim_{k\to\infty} s^k = s^*. \tag{4.1.9}$$

Therefore, the system has a stable limit cycle Γ^* solution, if only if, the map has a stable fixed point:

$$s^* = se^{-\frac{T^*}{\tau}} + 1 \tag{4.1.10}$$

where T^* denotes the limit cycle period as computed in (4.1.5) a*t* $v = \vartheta$

$$\vartheta = -\vartheta e^{-\frac{T^*}{\tau}} + \frac{\alpha s^k}{\tau}(T^*)\, e^{-(T^*)/\tau} + \omega \int_0^{T^*} e^{-\frac{s}{\tau}} c(t-s)ds. \tag{4.1.11}$$

Hence, the limit cycle can be parametrized by:

$$\begin{aligned} \Gamma^* : \qquad & s(t) = s^* e^{-\frac{t}{\tau}}, \\ & v(t) = -\vartheta e^{-\frac{t}{\tau}} + \frac{\alpha s^k}{\tau}\, te^{-\frac{t}{\tau}} + \omega \int_0^t e^{-\frac{s}{\tau}} c(t-s)ds, \\ & 0 \le t \le T^*. \end{aligned} \tag{4.1.12}$$

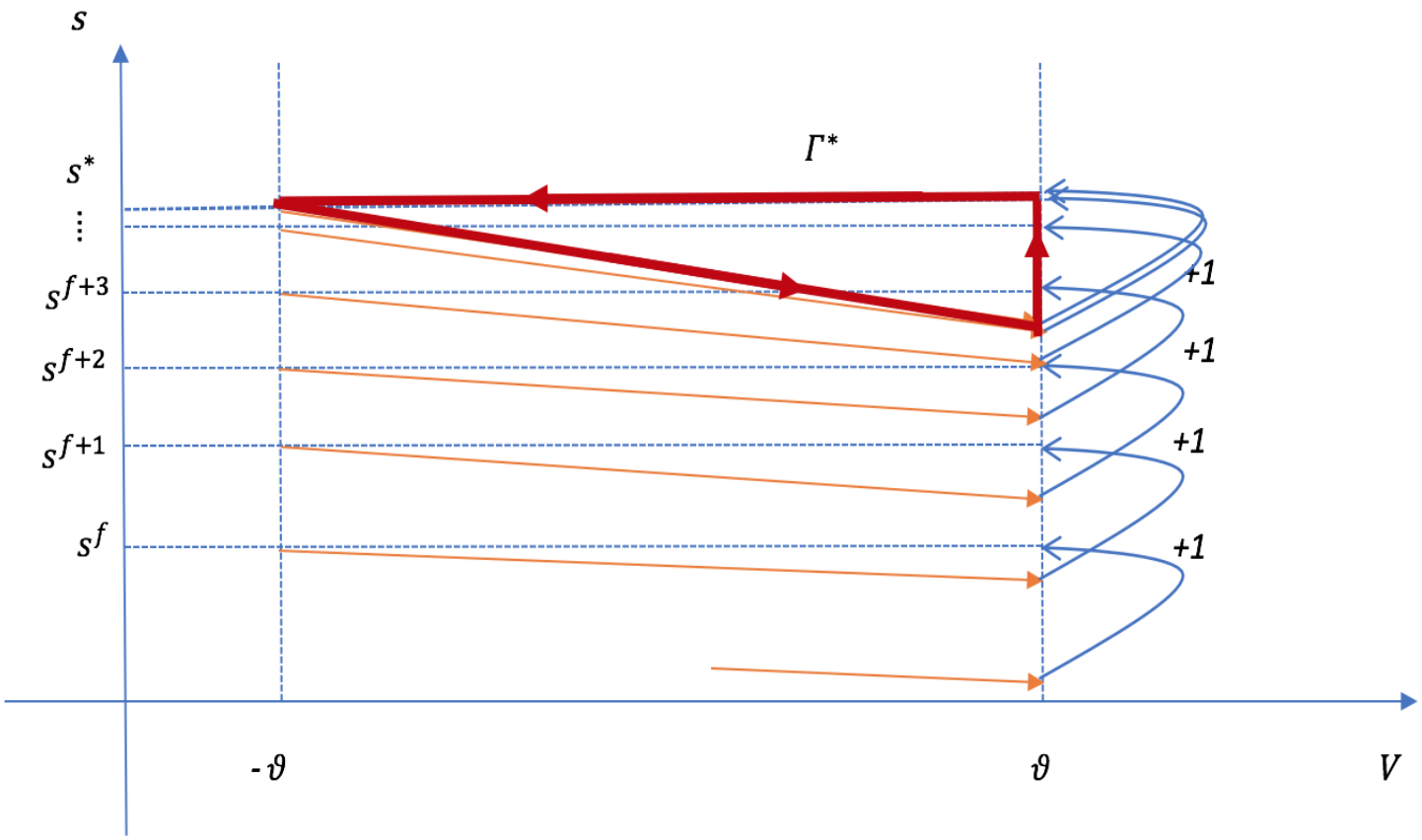


Figure 14: Solutions (4.1.5**)** visualized as 'sub-trajectories' in the phase plane. Each sub-trajectory is a curve $\boldsymbol{\Gamma^f}$(in orange and made straight for sake of simplicity and clarity) that starts at the reset potential and finishes at the threshold. At threshold, the phase point hops to the beginning of the next curve $\boldsymbol{\Gamma^{f+1}}$. In the limit $\boldsymbol{f \to \infty}$, the overall trajectory converges to the limit cycle $\boldsymbol{\Gamma^*}$(red curve).

4.1.2 Analysis of the Spike-Coding Network

In reality, it is not possible to isolate a neuron as done in the previous section. As discussed previously in this thesis, when a neuron fire, the spike affects other neuron's membrane potential. Therefore, a single neuron is bombarded with spikes from thousands presynaptic neurons in the network. The analysis reproduced above only consider the spike coming from the unique neuron itself. When adding the coupling terms, or synapses, accounting for neural interactions to the model, we lose the precedent method. The membrane potential of a neuron after it spiked can't be

predicted anymore. In the next section, we come to discuss an interesting approach to cope with this difficulty.

4.1.2.1 Population Activity

In this second section, we will make use of a deeper analysis. Suppose a particular neuron in the network fired a spike at time t_i^k, integration of the system (4.1.1)(4.1.2) states:

$$s_i(t) = s_i^k e^{-\frac{t-t_i^k}{\tau}} \tag{4.1.13}$$

$$\begin{aligned} v_i(t) = -\vartheta_i e^{-\frac{t-t_i^k}{\tau}} - \sum_{j=1}^{N}\sum_{k} \omega_i \omega_j \mathcal{H}\left(t - t_i^k\right) \\ + \sum_{j=1}^{N}\sum_{k} \frac{(\omega_i A \omega_j s_j^k)}{\tau} \sum_{k} \omega_i \omega_j \mathcal{H}\left(t - t_i^k\right)\left(t - t_j^k\right) e^{-\frac{t-t_j^k}{\tau}} \\ + \omega_i \int_0^{t-t_i^k} e^{-\frac{s}{\tau}} c(t-s) ds \end{aligned} \tag{4.1.14}$$

where $\mathcal{H}(t)$ denotes the Heaviside function, $\{\vartheta_i\}$ are the membrane thresholds, $\{t_i^k\}$ corresponds to the spike times and $\{s_i^k\}$ are the correspondent adaptation currents at these respective times. This set of solutions and precisely the double summation terms highlight the previous discussed issues. Indeed, the interspike times $\{T_i^k\}$ are random variables and the membrane potentials $v_i(t)$ can't be predicted. Therefore, it is not possible to conclude on the convergence to a limit cycle as discussed in section 4.1.1. However, we can compute the membrane potential density $\rho\ (v,t)$ that has been introduced in 3.2.2. In this case, $\rho\ (v,t)$ indicates the momentary state of the population as a whole. When the population is large enough, the number of neurons with membrane potential $v_0 < v(t) < v_0 + \Delta v$ is calculated as the density:

$$\lim_{N\to\infty}\left\{\frac{neurons\ with\ v_0 < v_i(t) \le v_0 + \Delta v}{N}\right\} = \int_{v_0}^{v_o+\Delta v} \rho(v,t)dv \qquad (4.1.15)$$

with the normalization condition:

$$\int_{-\infty}^{V} \rho(v,t)dv = 1. \qquad (4.1.16)$$

We remind that the output of spike-coding network is a weighted average of the synaptic currents (Eq. (3.1.4) and our interest in the collective behavior of the network. Consequently, it seems reasonable to describe the activity of a large network using a population measure rather than the spike trains of individual neurons. Furthermore, since we are interested in the stationary state $\rho\,(v,t) = \rho(v)$ (convergence to a limit cycle, for instance), the idea is to replace the double summation terms by an appropriate population measure based on the stationary density $\rho(v)$. One can guess a solution of replacement of these double summations terms and test such ansätz per simulation afterwards. If we assume that the density $\rho(v = \vartheta_j)$ represents the probability of neuron j hitting its firing threshold ϑ_j. Hence, the expected change of the membrane potential of the postsynaptic neuron i due to firing of the presynaptic neurons, is the average of the expected individual contributions $\omega_j\rho(\vartheta_j)$. Therefore, we replace the first double summation of Eq.(4.1.14) to get:

$$\sum_{j=1}^{N}\sum_{k} \omega_i\,\omega_j \mathcal{H}\left(t - t_i^k\right) \;\approx\; \omega_i \sum_{j=1}^{N} \omega_j \rho\left(\vartheta_j\right). \qquad (4.1.17)$$

The second assumption is more far-fetched. The double sum looks like the total amount of current entering neuron i at time t, so from the point of view of the postsynaptic neuron, it is as its membrane conductance is depending of time. Moreover, if we assume that the network reaches a stationary state, then currents from presynaptic neurons are a periodic function of voltage. For example, if the network converges to a limit cycle as in **Error! Reference source not found.**, then

the variable s can be parametrized using v. Therefore, we assume the pulse-based variable can be replaced by a conductance-based input which by definition depends on the voltage (see section 3.2.1).

$$\sum_{j=1}^{N}\sum_{k}\frac{(\omega_i A\omega_j s_j^k)}{\tau}\left(t-t_j^k\right)e^{-\frac{t-t_j^k}{\tau}} \approx \omega_i\sum_{j=1}^{N}A\omega_j\bar{g}_j\rho(\left(\vartheta_j-\bar{\vartheta}_j\right)). \qquad (4.1.18)$$

Where the precise form of $\bar{g}_j$ may be $\sum_{j=1}^{N}s_j^*\rho\left(\vartheta_j\right)$. In summary, we are using population metrics, such as the membrane potential density to approximate the stationary dynamics of a model neuron embedded in a population of leaky integrate-and-fire neurons with spike-triggered adaptation. We expect the dynamic of a postsynaptic neuron interacting with thousands of presynaptic neurons to be understandable and to provide limit cycles possibilities under certain circumstances. Specifically, we expect that the effective membrane time constant to change, that is we want that some inputs $c(t)$, those that cause individual neuron to have limit cycles behavior, change the period of oscillation T^* (see section 4.1.1).

5.0 CONCLUSION AND FURTHER WORK

Traditionally, analyses of neural networks activity are based on rate coding rather than spike temporal coding [40]. When considering large population of networks, a reason that make rate coding prevail over spike temporal coding was given by computational efficiency, since spike temporal coding needs the exact spike timing of every neuron, it was thought as computationally more efficient. Rate-coding networks are also extremely efficient for lots of application, often to describe properties of all types of sensory neurons and particularly for brain-machine interfaces [11].

However, this type of network can't take into account most of the recent experimental results in spike-based networks [8]. Despite a tremendous amount of spike trains and a high degree spike-train variability, powerful methods to construct spike-based networks have been recently derived [16]. More realistic and event-driven, spiking neural networks are finally found computationally more efficient if well implemented, since we need fewer neurons to accomplish the same task [29].

Evidences for spike-based networks include that an efficient coding spiking rule leads to balanced network [6]. An excitatory-inhibitory tight balance has numerous advantages. First the actual deviation between the input and desired output is improved. Furthermore, spike trains of individual neurons in spike-coding can be highly variable without affecting the collective output

accuracy thanks to adaptation mechanisms, that is if a neuron fails to emit a spike, neighbors quickly adjust their spiking to prevent an error [3].

Eventually the network modeled in this thesis gains in robustness when compared to alternative spike-coding network [5]. Nevertheless, it also has its own limitation, one could notice that robustness of the network comes from highly-recurrent property associated to fast connections. Unfortunately, it is not accounting for long range communication neither for a rich repertoire of states from synchronous to asynchronous as sparse balanced network can offer [9].

In this thesis, we suggested a new alternative for studying the dynamic of this type of spiking neural network, which stemmed from the increasing interest in the mechanisms behind spiking time generation and its implication in the whole dynamic of the brain [33]. As mentioned, the high amount of spike trains coming as input of a neuron makes the exact timing of a spike appears as a noise to the neuron itself and therefore is still really difficult to analyze. However, the use of the membrane potential density $\rho\ (v, t)$, an analytic tool indicating the momentary state of the population as a whole, answers this issue and points out a direction for the early stage of a phase plane analysis using the mean field theory. Indeed, this approach offers different perspectives for further work.

First, we started to explore the dynamics of the membrane potential density but some rigorous computation of its derivation should be made. Also, further analyses of its dynamics itself may led us to explore different critical points, if some [44]. This thesis finds that a single isolated neuron dynamic accesses a limit cycle when on a stationary state. Does the analysis of the density variable, that replace the random character of spike times in a large network, can bring new information on pattern of oscillations for the spike trains? For instance, could we observe any particular patterns of spikes when the network processes a specific task?

To go further, maybe, we could also adapt the work and the idea behind [5]. They discussed how neurons and networks adapt to an unexpected event, as the killing of some neurons for example. How does the density variable would react to the adaptation mechanism of the neural network? And what do the different patterns that we might have observed earlier, become along the adaptation process? What about after a stationary state is reached? This work only focuses on a theoretical idea but it now needs to be developed along real task-processing.

BIBLIOGRAPHY

[1] M. Abell, *Introductory Differential Equations: With Boundary Value Problems*, 3rd edition, Academic Press, 2009.

[2] L. Abbott, Lapicque's introduction of the integrate-and-fire model neuron, *Brain Research Bulletin*, vol. 50, no. 5-6, pp. 303-304, 1999.

[3] L. F. Abbott, B. DePasquale and R. M. Memmesheimer, Building functional networks of spiking model neurons, *Nature Neuroscience*, vol. 19, pp. 350-355, 2016.

[4] O. Barak, D. Sussillo, R. Romo, M. Tsoduks and L. F. Abbott, From fixed points to chaos: Three models of delayed discrimination, *Progress in Neurobiology*, vol.103, pp. 214-222, 2013.

[5] D. Barrett, S. Denève and C. Machens, Optimal compensation for neuron loss, *eLife.e121454*, vol. 5, 2016.

[6] M. Boerlin, C. Machens and S. Denève, Predictive Coding of Dynamical Variables in Balanced Spiking Networks, *PLoS Computational Biology e1003258*, vol. 9, no. 11, 2013.

[7] S. L. Bressler, V. Menon, Large-scale brain networks in cognition: emerging methods and principles, *Trends in Cognitive Sciences*, vol. 14, no. 6, pp. 277-290, 2010.

[8] R. Brette, Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain, *Frontiers in Systems Neuroscience*, vol. 9, 2015.

[9] N. Brunel, Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons, *Journal of Computational. Neuroscience*, vol. 8, pp. 182-208, 2000.

[10] G. Camera, A. Rauch, H. Lüscher, W. Senn and S. Fusi, Minimal Models of Adapted Neuronal Response to In Vivo–Like Input Currents, *Neural Computation*, vol. 16, no. 10, pp. 2101-2124, 2004.

[11] J. P. Cunningham, V. Gilja, S. I. Ryu, K. V. Shenoy, Methods for estimating neural firing rates, and their application to brain-machine interfaces, *Neural Network*, vol 22, no. 9, pp. 1235-1246, 2009.

[12] K. M. Dabrowski, D. J. Castaño and J. L. Tartar. Basic Neuron Model Electrical Equivalent Circuit: An Undergraduate Laboratory Exercise. *Journal of Undergraduate Neuroscience Education*, 2013.

[13] N. Dehgani, A. Peyrache, B. Telenczuk, M. L. V. Quyen, E. Halgren, S. S. Cash, N. G. Hatsopoulos and A. Destexhe, Dynamic Balance of Excitation and Inhibition in Human and Monkey Neocortex, *Scientific Reports*, vol. 6, 2016.

[14] B. DePasquale, Methods for Building Network Models of Neural Circuits, Columbia University, Thesis, 2016.

[15] M. DuBois, Action Potential: Biophysical and Cellular Context, Initiation, Phases, and Propagation (Cell biology research progress), *Nova Science Publishers Incorporated*, 2010.

[16] C. A. Eliasmith, A unified approach to building and controlling spiking attractor networks. *Neural Computation*, vol. 17, no. 6, 2005.

[17] K. S. Elmslie, Action Potential: Ionic Mechanisms, *Encyclopeia of Life Sciences*, 2010.

[18] K. Fukushima, Neural network model for selective attention in visual pattern recognition and associative recall, *Applied Optics*, vol. 26, no. 23, pp. 4985-4992, 1987.

[19] W. Gerstner et al., *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, Cambridge University Press, 2014.

[20] A. Graves, A. Mohamed and G. Hinton, Speech Recognition with Deep Recurrent Neural Networks, *arXiv:1303.5778*, 2018.

[21] K. Gregor, I. Danihelka, A. Graves, DRAW: A Recurrent Neural Network for Image Generation, *arXiv:1502.04623*, 2015.

[22] S. Grossberg, Recurrent neural networks, *Scholarpedia*, vol. 8, no. 2, pp. 1888, 2013.

[23] B. Haider, A. Duque, A. R. Hasenstaub, and D. A. McCormick, Neocortical Network Activity In Vivo Is Generated through a Dynamic Balance of Excitation and Inhibition, *Journal of Neuroscience*, vol. 26, no. 17, 2006.

[24] A. L. Hodgkin, A. F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, *Journal of Physiology*, vol. 117, no. 4, pp. 500-544, 1952.

[25] J. J. Hopfield, Neural networks and physical systems with emergent collective computation abilities, *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, 1982.

[26] E. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting (Computational neuroscience)*, MIT Press, 2006.

[27] H. Jaeger, A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, *GMD-Report*, vol. 159, 2002.

[28] W. Maass, Lower bounds for the computational power of networks of spiking neurons, *Neural Computation.*, vol. 8, pp. 1-40, 1996.

[29] W. Maass, Networks of spiking neurons: The third generation of neural network models, *Neural Networks*, vol. 10, no. 9, pp. 1659-1671, 1997.

[30] V. Mante, D. Sussillo, K. Shenoy and W. Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex, *Nature*, vol. 503, no. 7474, pp. 78-84, 2013.

[31] A. Mattuck, H. Miller, J. Orloff and J. Lewis. 18.03SC Differential Equations. *Massachusetts Institute of Technology: MIT*, 2011.

[32] R. Morgan, Linearization and Stability Analysis of Nonlinear Problems, *Rose-Hulman Undergraduate Mathematics Journal*, vol. 16, no. 2, 2015.

[33] S. Panzeri, R. A. A. Ince, M. E. Diamond, C. Kayser, Reading spike timing without a clock: intrinsic decoding of spike trains, *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 269, no. 1637, 2014.

[34] L. Perko, *Differential equations and dynamical Systems*, Springer, New York, 2001.

[35] A. Peterson, H. Meffin, M. Cook, D. Grayden, A. Burkitt and I. Mareels, are current-based synapses an accurate enough approximation? A homotopic mapping between current-based and conductance-based synapses in a neural field model of epilepsy, *arXiv:1510.0042*, 2018.

[36] D. Purves, G. J. Augustine, D. Fitzpatrick, et al, Excitatory and Inhibitory Postsynaptic Potentials, *Neuroscience*, 2nd edition, Sunderland, 2001.

[37] R. Quiza, Computationals methods and optimization, *Machining of hard materials*, pp. 177-208, 2011.

[38] R. D. Reed and R.J. Marks II, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, MIT Press, 1999.

[39] D. Ren, Sodium Leak Channels in Neuronal Excitability and Rhythmic Behaviors, *Neuron*, vol. 72, no. 6, pp. 899-911, 2011.

[40] F. Rosenblatt, The Perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, vol. 65, no. 6, 1958.

[41] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, Recent Advances in Recurrent Neural Networks, *arXiv:1801.0107*, 2018.

[42] M. N. Shadlen and W. T. Newsome, The Variable Discharge of Cortical Neurons: Implications for Connectivity, Computation, and Information Coding, *Journal of Neuroscience*, vol. 18, no. 10, pp. 3870-3896, 1998.

[43] D. Sussillo and O. Barak, Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks, *Neural Computation*, vol. 25, no. 3, pp. 626-649, 2013.

[44] A. Treves, Mean-field analysis of neuronal spike dynamics, *Network: Computation in Neural Systems*, vol. 4, no. 4, pp. 259-284, 1993.

[45] Z. S. Tseng, The Phase Plane Phase portraits, Course, 2008.