Juliana Couras
Fernandes Silva

Uma Teoria sobre Redes Codificantes por Potenciais de Ação com Potenciais Pós-sinápticos Heterogéneos

A Theory of Spike Coding Networks with Heterogeneous Postsynaptic Potentials

Universidade de Aveiro
2021

Juliana Couras
Fernandes Silva

Uma Teoria sobre Redes Codificantes por Potenciais de Ação com Potenciais Pós-sinápticos Heterogéneos

A Theory of Spike Coding Networks with Heterogeneous Postsynaptic Potentials

"*Understand well as I may, my comprehension can only be an infinitesimal fraction of all I want to understand.*"

— Ada Lovelace

Universidade de Aveiro
2021

Juliana Couras Fernandes Silva

Uma Teoria sobre Redes Codificantes por Potenciais de Ação com Potenciais Pós-sinápticos Heterogéneos

A Theory of Spike Coding Networks with Heterogeneous Postsynaptic Potentials

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Computacional, realizada sob a orientação científica do Doutor Alexander Goltsev, Investigador Auxiliar do Departamento de Física da Universidade de Aveiro, e do Doutor Christian Machens, Investigador Principal no Laboratório de Neurociência Teórica do *Champalimaud Centre for the Unkown*.

Apoio financeiro do European Regional Development Fund (ERDF), através do Programa Operacional Regional de Lisboa.

Apoio financeiro da Fundação para a Ciência e a Tecnologia (FCT), através dos Fundos Nacionais Portugueses (PIDDAC).

Dedico este trabalho à minha irmã, na esperança que ela encontre inspiração naquilo que não percebe.

o júri / the jury

presidente / president

Prof. Doutor Fernão Rodrigues Vístulo de Abreu
Professor Auxiliar, Departamento de Física, Universidade de Aveiro

vogais / examiners committee

Prof.ª Doutora Fleur Zeldenrust
Assistant Professor, Radboud University (external examiner)

Prof. Doutor Christian Konrad Machens
Investigador Principal, Champalimaud Centre for the Unknown (supervisor)

agradecimentos / acknowledgements

I would like to thank the University of Aveiro for encouraging its students to pursue science "beyond borders", allowing them to participate in collaborative ways of doing science with other institutions. Moreover, I would like to thank my university supervisor Alexander Goltsev for supporting me in this endeavour of finding new challenges.

Importantly, I thank my hosting supervisor Christian K. Machens for his active scientific guidance and enduring support during this year. I also thank my second supervisor Nuno Calaim for his constant availability to discuss new problems and solutions. I thank the Machens Lab for welcoming me in this odd year and for all the fruitful meetings. Finally, I thank the Champalimaud Centre for the Unknown for having me as a master's student.

Esta dissertação marca o fim da aventura que foi fazer dois mestrados integrados em simultâneo. Assim, não poderia deixar de fazer aqui um balanço global dos últimos anos. Digo-vos três coisas. A resiliência nasce na individualidade, mas só se concretiza se for apoiada. Quem corre, ainda que por gosto, cansa-se e muito. Fazer dois cursos é um privilégio social que me foi possível graças a duas condições fundamentais: um ensino superior público de qualidade e um suporte inesgotável das pessoas maravilhosas à minha volta. Pelo primeiro, agradeço ao 25 de Abril, pelo segundo, agradeço:

Aos meus avós, pelo apoio incondicional. Se não fossem as constantes boleias e as entregas dos almoços na estação de comboios, fazer dois cursos em duas cidades diferentes seria consideravelmente mais difícil.

Aos meus pais, por não terem posto em causa as minhas ideias pouco convencionais e terem ido na onda.

À minha irmã, que é incrivelmente melhor do que eu era com a idade dela, por ser uma fonte constante de alegria e descontração. Algo especialmente importante nos dias mais stressantes.

Aos meus amigos de Medicina, porque por muita força de vontade de que haja, ninguém acorda todos os dias às 6h se não for para ir ter com pessoas especiais. Obrigada por serem essas pessoas.

Aos meus amigos de Engenharia, por terem sido sempre flexíveis em relação à calendarização de aulas e exames.

Aos meus amigos de sempre, por o continuarem a ser.

Palavras Chave

neurociência, redes codificantes por potenciais de ação, redes neuronais de potenciais de ação, redes recorrentes, redes de neurónios leaky integrate-and-fire, potenciais pós-sinápticos

Resumo

Modelar redes neuronais com princípios biologicamente plausíveis é um desafio para a neurociência teórica. De facto, há evidência crescente de que os tempos precisos dos potenciais de ação emitidos por um neurónio desempenham um papel crucial na computação neuronal. No entanto, construir redes neuronais funcionais que mimetizem a variabilidade de disparos encontrada *in vivo* não é uma tarefa trivial. Boerlin et al. sugeriu um modelo de redes *leaky integrate-and-fire* que, através de um balanço apertado entre excitação e inibição neuronal, conseguem construir uma estimativa de um sinal multi-dimensional em tempo real, usando a combinação ponderada de séries de potenciais de ação com variabilidade do tipo Poisson. Apesar destas plausabilidades biológicas, estas redes codificantes por potenciais de ação sustentam-se na propagação instantânea desta entidade biofísica. Uma vez que esta assunção não vai de encontro às escalas de tempo das sinapses observadas no cérebro, esta é uma limitação do modelo. Assim, tendo como objectivo construir uma rede codificante por potenciais de ação com potenciais pós-sinápticos biologicamente plasíveis, neste trabalho usamos o facto do modelo original destas redes permitir a reconstrução de sinais multi-dimensionais para transformar o problema de reconstrução preditiva num problema multi-dimensional no domínio temporal. Através desta transformação, emergem três propriedades que estas redes devem ter para se manterem funcionais: não codificar o presente; permitir heterogeneidade temporal; prever o futuro da estimativa da rede de acordo com a dinâmica do sinal original. Assim, introduzindo estas propriedades nas assunções originais de Boerlin et al., mostramos que é possível conceber uma rede codificante por potenciais de ação que reconstrua sinais multi-dimensionais sem a necessidade da comunicação instantânea dos mesmos.

Keywords neuroscience, spike coding networks, balanced networks, spiking neural networks, recurrent networks, leaky integrate-and-fire networks, postsynaptic potentials

Abstract Modeling biologically realistic neural networks is a challenge for neural theory. While there is increasing evidence that the precise times of spikes play a crucial role in neural computation, building spike neural networks that resemble the spiking variability encountered *in vivo* while computing some function is not a trivial task. Boerlin et al. suggested a framework of leaky integrate-and-fire networks that, through excitation-inhibition tight balance, can track high-dimensional signals while producing spike trains with Poisson-like statistics. Notwithstanding their biologically plausible features, the spike coding networks rely on the instantaneous propagation of spikes to ensure an optimal function. Given that such an assumption may not fit the slower timescales of the synapses encountered in the brain this is a limitation of the model. Thus, under the goal of deriving a model with biologically plausible postsynaptic potentials, in this work, we take advantage of the spike coding networks' ability to track high-dimensional signals to transform the problem of predictive tracking into a high-dimensional problem in the temporal domain. By doing so, we were able to get insights about the properties that such networks should have to be functional: no coding for the present time; temporal heterogeneity; prediction of the network's estimate according to the dynamics of the signal being tracked. Then, by deriving a network from the same assumptions as Boerlin et al. while enforcing these properties it was possible to build a spike coding network that tracks multi-dimensional signals without relying on instantaneous communication of spikes.

Contents

List of Figures

Glossary

ANN	Artificial Neural Network
SNN	Spiking Neural Network
RNN	Recurrent Neural Network
MNIST	Modified National Institute of Standards and Technology
GPU	Graphics Processing Unit
BCI	Brain-Computer Interface
LIF	Leaky Integrate-and-Fire
PSC	Postsynaptic Current
PSP	Postsynaptic Potential
EPSC	Excitatory Postsynaptic Current
IPSC	Inhibitory Postsynaptic Current
EPSP	Excitatory Postsynaptic Potential
IPSP	Inhibitory Postsynaptic Potential
COBA	Conductance-Based
CUBA	Current-Based
FORCE	First-Order Reduced and Controlled Error
SCN	Spike Coding Network
CV	Coefficient of Variation
ISI	Interspike Interval
hSCN	Heterogeneous Spike Coding Network
FR	Firing Rate
PCA	Principal Component Analysis
MT	Middle Temporal
V1	Visual Area 1 (Primary Visual Cortex)

CHAPTER 1

Introduction

The arts and sciences are avatars of human creativity.

— Mae Jemison

1.1 Motivation

The human brain is one of the most complex systems in nature. Throughout the years, the development of experimental techniques such as electrophysiology, multiphoton microscopy, and optogenetics allowed scientists to collect data either *in vitro* or *in vivo* from several areas of the brain and under diverse conditions. Interestingly, in 2011, Stevenson and Kording [1] analysed the advances in neural recording techniques over the past 50 years and postulated the equivalent of Moore's law for neural recording techniques – that the number of simultaneously recorded neurons doubles every 7 years. They keep tracking the progress in electrophysiology here and as of today, the doubling time is estimated to be 6.3 ± 0.2 years (figure 1.1).

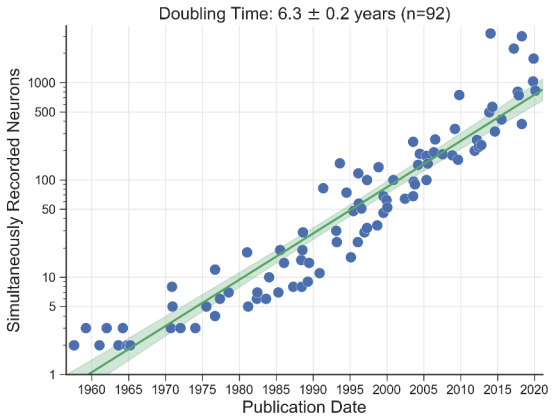

Figure 1.1: Progress of electrophysiology. Taken from [2].

However, despite the large amounts of data now available from different animal brains, they are not always meaningful enough to broader comprehension. For instance, for many years, anesthetized animals were used by neuroscientists in their research, which under-covered crucial aspects of neural

activity. It was only when technical advances made it possible to record from behaving animals in movement that neuroscientists started to understand how different states of motion could modulate sensory responses [3].

Currently, neuroscience is going through a shift in the paradigm that the brain might not be as noisy as once thought, i.e., that the trial-to-trial variability observed *in vivo* is not meaningfulness [4]. Rather, the precise times of spikes, with their sparsity and variability, might be crucial for the efficient processing of information. Thus, in order to go beyond the use of standard statistical models such as correlation or regression in order to extract meaning from the data, theoretical neuroscience aims at building brain-inspired models that are able to emulate the observed neural activity. In addition, having mathematical descriptions of neural data allow us to make experimentally testable predictions, thus helping us to guide experimental studies. In conclusion, neuroscience is a dynamic field where theory can be probed in a reasonable time frame as new techniques come along.

1.2 Rate versus Spiking Neurons

The neuron is the fundamental processing unit of neural circuits in the brain. Notwithstanding the heterogeneity of neurons in the cortex, a neuron can typically be described as having three functional compartments: dendrites, soma, and axon. In essence, the dendrites integrate the input signals received by the neuron, while the soma is the processing unit that performs non-linear computations and outputs an action potential if the input exceeds a threshold. Finally, the axon transmits the action potential to the connected neurons throughout a synapse [5]. These action potentials, also known as "spikes", are stereotyped electrical impulses whose duration is in the order of milliseconds, a timescale that contrasts with the timescales of the input perceptions or the output actions of the brain [4]. Although spikes are the mean of communication in neural circuits, there is a debate in neuroscience about whether the precise time of spikes are really necessary to encode information or if a proxy for the spiking frequency over a longer timescale is enough to reliably model a network of neurons. Given that the frequency of spiking was earlier correlated with stimuli strength [6], the classical framework of neural communication considers the firing rate, instead of the spike, as the unit of information in the brain [7]. Being a continuous variable, the firing rate is easily described mathematically, but at what functional cost? Note that it is always possible to compute firing rates having the spike times, but it is not possible to compute the precise spike times from the firing rates, i.e., many spike trains yield the same firing rate. Therefore, the key point of the debate "rate vs. spiking neurons" is whether firing rates are sufficient or not for information processing. In other words, it is expectable that firing rates correlate to information processing, but are they the functional unit of it? [7].

In vivo recordings have shown that the spike trains of a given neuron show a high spiking variability over trials under the same stimuli presentation [8]–[11] (figure 1.2), boosting the idea that information could not be readout from the precise time of spikes and reinforcing the need for a rate-based information unit. In spiking models, the neurons respond deterministically to injected current, i.e., they produce deterministic spike trains [12]. But when neurons are described by their firing rates $r(t)$, it is possible to generate a stochastic spike train given the deterministic rate. Then, spikes are interpreted as Poisson events occurring at rate $r(t)$ and the rate of a neuron depends on its presynaptic rates [7].

Despite this, rate-based neurons do not explain the cause of this observed variability. In fact, it has also been shown that single neurons recorded *in vitro* are robust to noisy perturbations [13], [14]. Thus, the spiking variability observed in *in vivo* recordings might be due to effects of network activity, attention, prior experience or other hidden variables not considered by the observer [15]. In other words, spiking variability might not be a problem to be solved by the use of statistical tools, but rather

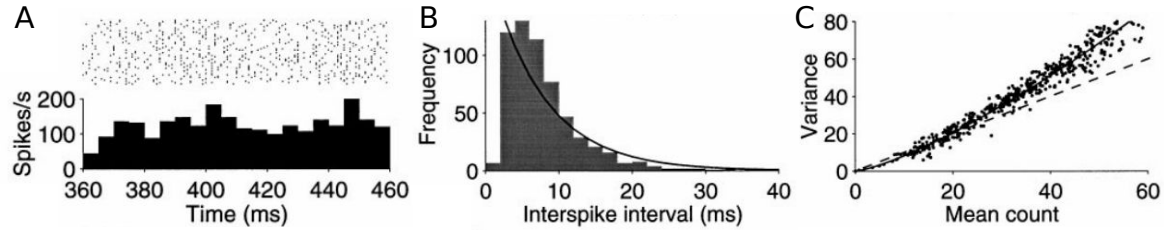

Figure 1.2: Spiking variability from *in vivo* recordings. Adapted from [9] Copyright 1998 Society for Neuroscience. **A**. Response variability of a Middle Temporal (MT) neuron. *Top*: Spike trains over different trials. *Bottom*: Firing rate. **B**. Interspike Interval (ISI) histogram. **C**. Variance of the spike count against the mean number of spikes. The dashed line is the expected line for a Poisson process.

a key feature of neural biophysics. If that is the case, rate-based models show a few limitations, given that they disregard the precise time of spikes. Furthermore, these models are especially sensitive to the synaptic weights of the network, with strong recurrent connections playing a dominant role in the network's activity [16].

While there are both experimental [17]–[19] and theoretical [4], [20] evidence that firing rates fail to capture all the necessary information about the stimuli to decode the output as well as the precise spike times, the rate-based neurons are still widely used due to their mathematical and computational feasibility. In fact, when it comes down to simulating large networks it is easier to approximate the average activity of a neural population by considering the average of its neurons' firing rates. Moreover, often scientists are interested in building networks that can learn some function and it is mathematically easier to derive learning rules using continuous variables. For instance, in Artificial Neural Networks (ANNs), the activation functions used are higher abstractions of a firing rate. Given these activation functions are differentiable, learning rules based on gradient descent may be easily derived, as backpropagation [21] in the case of a feedforward ANNs or backpropagation through time [22] in the case of a rate Recurrent Neural Network (RNN). The discontinuous dynamics of spiking neurons, along with the constraints of local Hebbian learning [23], hinder the derivation of biologically plausible learning rules.

1.3 Neural Dynamics of Spiking Neurons

Being a biological cell, the neuron has a phospholipid bilayer packed with set of ionic channels that keep a difference of charges between the intra and the extra-cellular medium. Thus, the membrane has a capacitance C_m that relates the voltage across the membrane and the amount of excess charge as a standard capacitor

$$Q = C_m V. \tag{1.1}$$

Although the electrical quantities measured in different sites of a neuron are not the same, the simplest neural dynamics models consider the neuron a single processing unit described by a unique membrane voltage V. That said, here, we focus on single-compartment models.

The synaptic input received by a neuron is translated into a current flowing through the ion membrane channels. Then, it is possible to describe the neuron's membrane by an electric circuit where the ion channels are just components of it (figure 1.3A). For many channels, the driving force is linearly proportional to the difference $V - E_i$, where E_i is the reversal potential for the ion. Moreover, considering that the i-th channel has a specific conductance, g_i, one can model the membrane current due to that channel as $g_i(V - E_i)$. Finally, the membrane current I_m due to all membrane channels is

written as

$$I_m = \sum_i g_i(V - E_i). \tag{1.2}$$

I_m is conventionally defined as positive-outwards. Extrinsic factors may also play a role, as the current injected into a neuron by an electrode. To this currents we call I_e, which are conventionally defined as positive-inwards [12]. Formally, we write the simplest neuronal dynamics as

$$C_m \frac{dV}{dt} = \frac{dQ}{dt} = -I_m + I_e. \tag{1.3}$$

The reader may already predict that while it may be straightforward to mathematically write the external current injected into a neuron, I_e, it is not trivial to formally write the membrane current I_m due to the variety and complexity of the ionic channels in the cell's membrane. In fact, there are many ways of writing this current, depending on the assumptions one is willing to make. That said, next we introduce how to mathematically write it.

1.3.1 Single-Neuron Models

There are two big groups of single neuron models: Leaky Integrate-and-Fire (LIF) models and voltage-dependent conductances models. While the firsts describe the membrane current as a leaky current, therefore portraying only the sub-threshold membrane potential, the seconds describe the membrane voltage as the sum of several non-linear terms reflecting a voltage-dependence conductance of the membrane ionic channels, thus being able to model the action potential (e.g. the Hodgkin and Huxley model [24]).

Given that in this thesis we focus on networks of LIF neurons, next we introduce the general dynamics of these models.

Leaky Integrate-and-Fire Models

In LIF models [25], the dynamics of the voltage during the action potential are not described biophysically. Rather, it is assumed that the action potential occurs at time t_i if the neuron reaches a spiking threshold V_{th} [26]. Following this, the voltage is reset to a value V_{reset} such that $V_{reset} < V_{th}$. Thus, only the sub-threshold membrane voltage is modeled and the crossing of the threshold defines the spiking time t_i

$$V(t_i) = V_{th} \quad \text{and} \quad \left.\frac{dV(t)}{dt}\right|_{t=t_i} > 0. \tag{1.4}$$

The circuit for a LIF model is a capacitor C_m in parallel with a resistor R_m driven by a current $I(t)$ (figure 1.3B). Formally, we write

$$I(t) = \frac{V(t)}{R_m} + C_m \frac{dV}{dt}. \tag{1.5}$$

Multiplying both members by R_m and introducing the membrane time constant $\tau_m = R_m C_m$, one gets

$$\tau_m \frac{dV}{dt} = -V(t) + R_m I(t), \tag{1.6}$$

which can be obtained from equation 1.3 considering that the membrane current is described by a single passive leakage term, $I_m(t) = I_{\text{leak}} = g_L(V(t) - E_L)$, where $E_L = 0$ is the resting potential, $R_m = 1/g_L$, and $I_e = I$.

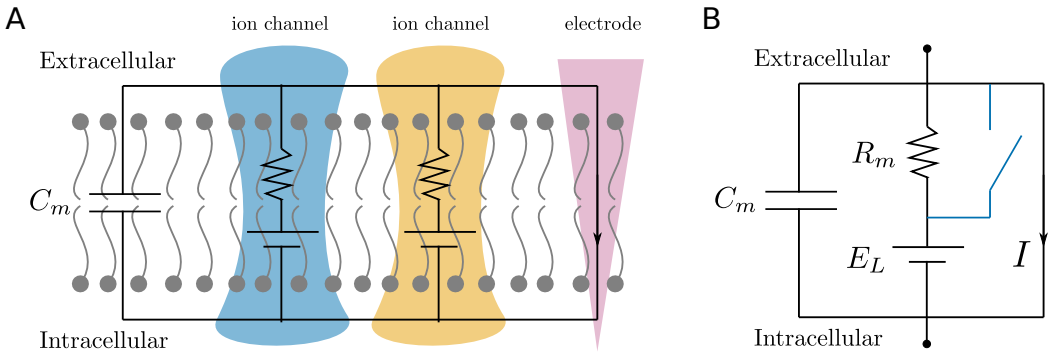

Figure 1.3: Electric circuits for the neuron's membrane. **A**. Generic circuit, representing the membrane capacitance in parallel with other components. Namely, the ion channels are represented by a resistor in series with a battery and an electrode injecting a constant current. **B**. The RC circuit with a switch representing the LIF model. When there is a spike the switch closes.

1.3.2 Network Models of LIF Neurons

Although the above equations are describing the dynamics of a single neuron, a network of LIF neurons may be constructed. In order to do that, one must enforce the communication between neurons. How do neurons communicate? Through synapses. A synapse is a chemical process in which the action potential reaches the presynaptic terminal activating voltage-dependent Ca^2+ channels. This yields the release of vesicles of neurotransmitters into the synaptic cleft. These neurotransmitters then bind to receptors in the postsynaptic membrane, which leads to the opening of ionic channels.

Formally, this amounts to adding a synaptic current, I_{syn}, to the model in equation 1.3. Then, generally, we write

$$C_m \frac{dV}{dt} = -I_{\text{leak}} - I_{\text{syn}} + I_e. \tag{1.7}$$

Synaptic Conductances

The opening of the ionic channels may be modeled by a physical quantity termed synaptic conductance, $g_{\text{syn}}(t)$. Given that the models for synaptic conductances depend on the concentration of the neurotransmitter in the synaptic cleft, it is expected that the conductance has a rising and a decaying phase, corresponding to the fast release of neurotransmitters and its consequent enzyme-mediated degradation. How fast this rising and decaying is depends on the channel mediating the conductance. For instance, AMPA channels mediate fast conductances (figure 1.4A), while GABA_B mediate slow ones (figure 1.4B). Moreover, depending on whether the synapse is excitatory or inhibitory, the Postsynaptic Currents (PSCs) may be either Excitatory Postsynaptic Currents (EPSCs) or Inhibitory Postsynaptic Currents (IPSCs).

Given the diversity of receptors able to mediate a synapse, different kernels may be used to describe the time evolution of their synaptic conductances for the time after a presynaptic spike at t_i [28]. Next, we present three kernels widely used to model $g_{\text{syn}}(t)$.

An instantaneous shift in the conductance may be modeled by a Dirac delta impulse, $\delta(t - t_i)$, (figure 1.5A). The exponential kernel may model a process with an instantaneous rise followed by a slow decay (figure 1.5B) and a double exponential is a more realistic waveform given it takes rising and decaying parameters (figure 1.5C).

Based on the nature of the synaptic models, we may divide them in Conductance-Based (COBA) and Current-Based (CUBA) networks [29], [30]. When the synaptic model is voltage-dependent, i.e, the synaptic current depends on the electric driving force, the network is COBA. On the other hand,

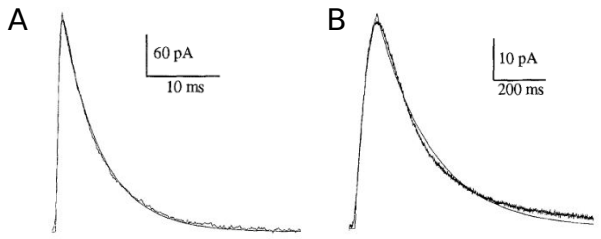

Figure 1.4: Averaged recordings of EPSC. Adapted from [27]. **A**. Averaged recordings of EPSC mediated by AMPA/kainate receptors in mossy-fiber synapses onto CA3 pyramidal neurons in hippocampal slices (wiggly line). And respective theoretical fit (smooth line), considering a first-order kinetic model. **B** Averaged recordings of EPSC mediated by $GABA_B$ receptors in dentate granule cells (wiggly line). And respective theoretical fit (smooth line), considering a first-order kinetic model.

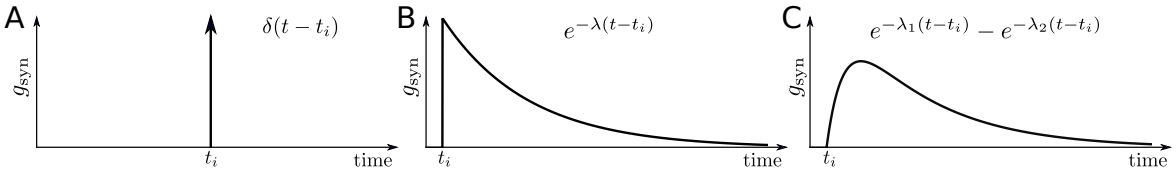

Figure 1.5: Synaptic conductances. **A**. Dirac delta impulse. **B**. Exponential kernel. **C**. Double exponential kernel.

when the synaptic model is voltage-independent, the network is CUBA. Mathematically, for the COBA model we write

$$I_{\text{syn}}(t) = g_{\text{syn}}(t)(V(t) - E_{\text{syn}}), \tag{1.8}$$

where $g_{\text{syn}}(t)$ is the synaptic conductance and E_{syn} is the reversal potential of the synapse. On the other hand, for the CUBA model we write

$$I_{\text{syn}}(t) = g_{\text{syn}}(t)(E_L - E_{\text{syn}}). \tag{1.9}$$

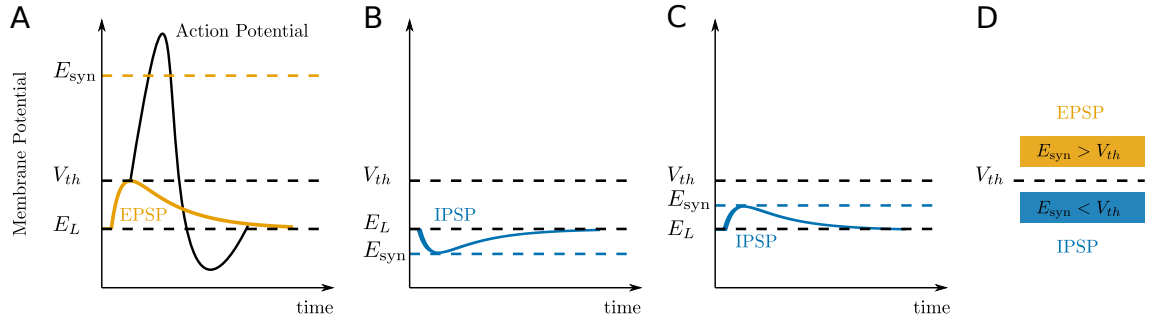

Figure 1.6: Excitatory Postsynaptic Potentials (EPSPs) and Inhibitory Postsynaptic Potentials (IPSPs).**A**. A EPSC depolarizing the membrane. **B**. A IPSC hyperpolarizing the membrane. **C**. A IPSC depolarizing the membrane.**D**. Simple rule for setting excitatory and inhibitory synapses.

Then, in a CUBA model, the current is just the conductance multiplied by a constant driving force, whereas in a COBA model this driving force depends on the current voltage of the membrane. While a CUBA excitatory synapse approximates well a COBA excitatory synapse, this is not the case for an inhibitory synapse [28]. In fact, the reversal potential of excitatory synapses are usually above the spiking threshold, while the reversal potential of inhibitory synapses are below it (figure 1.6D) [31]. This means that for an excitatory synapse the driving force is negative either in the COBA or in the CUBA model (figure 1.6A). In what comes to an inhibitory synapse, first consider the case in

which the reversal potential of an inhibitory synapse is between the resting potential and the spiking threshold, i.e., $E_L < E_{\text{syn}}^{\text{inh}} < V_{th}$ (figure 1.6C). Then, in the COBA model, if the current voltage of the membrane $V(t)$ is above the reversal potential $E_{\text{syn}}^{\text{inh}}$, the driving force is positive. On the other hand, if $V(t)$ is below the reversal potential, the driving force is negative. Whereas in the CUBA model, the sign of the driving force may only be negative. The only case an inhibitory COBA synapse will always match the sign of an inhibitory CUBA synapse is if the reversal potential is below the resting potential, i.e., $E_{\text{syn}}^{\text{inh}} < E_L$, yielding a positive driving force in either case (figure 1.6B).

Synaptic Connectivity

In the above section several models for a simple synapse between two neurons were presented. But a neural network has many neurons, with many synapses of different strengths and directions. Thus, another determinant of neural activity in a network is its connectivity. At the light of a hierarchical cortical organization, there are three main types of connectivity patterns [12]:

- **Feedforward** connectivity, which consists of connecting a region in an earlier stage of the processing pathway with a region in a later stage;
- **Recurrent** connectivity, which translates connections between neurons within the same hierarchical region;
- **Top-down** connectivity, which represents the connections between a region in a later stage of the processing pathway back to a region in an earlier stage.

The networks studied in this thesis rely on both feedforward and recurrent connections.

1.4 Overview of Spiking Neural Networks

In the previous sections we provided the mathematical fundamentals to build a network of biologically plausible spiking neurons. Next, we overview the generic features of spiking networks [32], ending with their applications in neuroscience and neuromorphic systems.

The emergence of mathematical tools to train Spiking Neural Networks (SNNs), along with the fact that increasing ANNs' accuracy demands an unsustainable high computational processing power [33] is pushing the research of SNNs further. In 2021, a meeting on "Spiking neural networks as universal function approximators" summarized the key advantages of SNNs [34]:

- Precise spiking time efficiently encodes information [35];
- Neural heterogeneity in the time domain adds computational value [36]–[38];
- Learning with biologically plausible rules, such as First-Order Reduced and Controlled Error (FORCE) training [39], gradients with respect to spike times [40], and surrogate gradients [41], allows the network to perform more brain-like tasks.

Currently, a drawback of SNNs is that there are not good benchmark datasets for spike-based inputs. Thus, when SNNs are trained in datasets often used as benchmarks for ANNs, like Modified National Institute of Standards and Technology (MNIST) [42], the first underperforms the second [43], [44].

However, the efficiency of ANNs relies on the von Neumann architecture, which implies a physical separation between the processing unit and memory. This means that when the data stream is too large, the communication between these two separated units turns into a bottleneck. Then, given this physical limitation, neuromorphic devices, originally proposed by Mead [45], are surging as low-power alternatives [46], [47]. The *modus operandi* of these devices rest on emulating SNNs in a physical substrate that, as the brain, co-localizes processing and memory units. Furthermore, neuromorphic devices are natively suited for parallelism, recurrence and stochastic computation [48]. Indeed, when

implemented in these devices, recurrent SNNs have outperformed their feedforward rate counterparts [49]. Moreover, the great feature of neuromorphic computation is that in these devices the simulation time matches the real time, something not possible with Graphics Processing Units (GPUs) [50]. This is a huge advantage for the computational neuroscience field, since one could simulate cortical microcircuits for hours and then compare the experimental activity with the simulated activity. Finally, neuromorphic devices are paving the way for Brain-Computer Interfaces (BCIs) where silicon and biological neurons interact [51].

1.5 Spike Coding Networks

The object of study of this thesis is a SNN called Spike Coding Network (SCN). The SCNs are LIF recurrent networks that were originally proposed by Boerlin, Machens, and Denève [52]. They provide an elegant framework for a network that is able to encode a high-dimensional input signal into a set of spike trains with the optimal number of spikes from which the output may then be decoded. These networks have been shown to be highly efficient and robust [53], while resembling experimentally observed properties, as the Poisson-like variability in the spike trains and a tight excitatory-inhibitory balance. Notwithstanding their biologically similarities, they rely on the instantaneous propagation of spikes and constrain neurons to homogeneous Postsynaptic Potentials (PSPs). In this thesis, we are going to study this biological implausibilities and try to handle them by creating a SCN with heterogeneous PSPs whose neurons do not communicate instantaneously while being efficient. To distinguish these networks from the classical SCNs we call them Heterogeneous Spike Coding Networks (hSCNs).

1.6 Goals and Thesis Outline

1.6.1 Goals

The main goal of this thesis is to understand how recurrent neural networks with spiking neurons can represent information reliably while fulfilling strong biological assumptions. Bellow are the specific objectives:

1. To derive and understand both the analytical and the geometric frameworks behind SCNs;
2. To identify the biological resemblances and the biologically implausibilities of SCNs;
3. To develop a model of a functional SCNs that does not rely on instantaneous communication and that introduces temporal heterogeneity;
4. To interpret the mechanisms of this new SCNs under the light of the geometric framework;
5. To provide experimentally testable predictions based on this new model.

1.6.2 Thesis Outline

The content of this thesis is organized into six chapters. The present chapter is a general introduction, comprising the motivation for this thesis, the theoretical foundations for its comprehension and broader applications of this work. Chapter 2 introduces both the analytical and geometric framework of SCNs. Moreover, it states both the advantages and the limitations of SCNs, introducing the problems this thesis aims to tackle, namely instantaneous propagation of spikes and temporal homogeneity. Chapter 3 provides insights about the new problems that arise in SCNs when using different postsynaptic kernels, including the necessity of predicting the future. Here, we also present two predictive scenarios to illustrate how differently the network behaves according to different assumptions. Chapter 4 frames the problem in a different angle, providing intuitions for the final solution on how to build hSCNs and

key aspects of their function. Specifically, we use a classical SCN in a discrete temporal domain to get insights about a hSCN. Chapter 5 provides a generalization of the proposed hSCNs for further heterogeneous networks in terms of postsynaptic potentials. Finally, chapter 6 presents an integrated discussion under the light of others' work, main conclusions, innovations, limitations and possible future research.

CHAPTER 2

Spike Coding Networks

The brain (...) is a prime example of a biological object too complex to be understood without mathematics.

— Grace Lindsay

Spike Coding Networks (SCNs), originally proposed by Boerlin, Machens, and Denève [52], are mathematical abstractions of neural dynamics derived from a single efficiency principle based on spikes for signal transduction. As a consequence of such derivation, several properties resembling neural biophysical features emerge. Namely, high spike variability, tight balance between excitation and inhibition and robustness [35], [54], [55]. Despite the biological plausibility of these networks regarding these features, they rely on the instantaneous propagation of spikes to fulfill their goal of tracking a signal by a synaptic integration of spike trains, which is biologically implausible.

That said, in this chapter, we start by deriving the mathematical framework of SCNs followed by the geometrical framework. Then, we present variations of these networks, either by adding spike costs or by considering non-optimal architectures. Finally, we dissect the biologically plausible properties of SCNs and their limitations.

2.1 Mathematical Principles of SCNs

A SCN (figure 2.1) is a recurrent network of Leaky Integrate-and-Fire (LIF) neurons whose function is to encode a given time-dependent signal, $\mathbf{x}(t)$, into spike trains from which an estimate of the signal may be decoded, $\hat{\mathbf{x}}(t)$. It is possible to build such a network relying only on two assumptions. The first assumption is that the estimate of the signal, $\hat{\mathbf{x}}(t)$, is obtained by a synaptic integration of the spike trains. That is, each neuron's spike train is convolved with a given kernel, in an analogy with a postsynaptic potential. Then the filtered spike trains are weighted and summed linearly, in an analogy with the integration in a dendritic tree. Formally, for a network with N neurons, this is written as the following linear summation

$$\hat{\mathbf{x}}(t) = \mathbf{D}\mathbf{r}(t), \tag{2.1}$$

where $\mathbf{r}(t) \in \mathbb{R}^{N\times 1}$ is a vector of N filtered spike trains, $\hat{\mathbf{x}}(t) \in \mathbb{R}^{M\times 1}$ is a vector of M readouts for a M-dimensional signal and $\mathbf{D} \in \mathbb{R}^{M\times N}$ is the decoding matrix where each column $\mathbf{D}_i$ is the decoding vector of the i-th neuron.

In fact, equation 2.1 is the solution of the dynamical equation of the system (see Attachment A for the derivation)

$$\dot{\hat{\mathbf{x}}}(t) = -\lambda_d \hat{\mathbf{x}}(t) + \mathbf{D}\mathbf{s}(t), \tag{2.2}$$

where λ_d is the filtering time constant and $\mathbf{s}(t) \in \mathbb{R}^{N\times 1}$ are the network's spike trains. These spike trains are given by $s_i(t) = \sum_k \delta(t - t_i^k)$, where t_i^k is the time of the k-th spike of the i-th neuron.

The second assumption is that a neuron only spikes when it minimizes the error between the signal $\mathbf{x}$ and the readout $\hat{\mathbf{x}}$. That is, the network minimizes, at every time point, the following loss function

$$L(t) = ||\mathbf{x}(t) - \hat{\mathbf{x}}(t)||_2^2, \tag{2.3}$$

where $||.||_2$ denotes the L2 norm. Then, the i-th neuron should fire a spike whenever this leads to a decrease in $L(t)$. This spiking rule for the i-th neuron is written as follows

$$L(t|i \text{ spikes}) < L(t|i \text{ silent}). \tag{2.4}$$

A spike by the i-th neuron amounts to adding a delta function to its spike train. Then, the integration of this delta function is equivalent to adding a postsynaptic potential, i.e., an exponential kernel, to the readout at the time of the spike. Hence, if the i-th neuron spikes at time t, we have, for the case of a decaying exponential kernel, the following update of the readout in the future time τ

$$\underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}} \text{ spike}} \rightarrow \underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}} \text{ silent}} + \underbrace{\mathbf{D}_i e^{-\lambda_d(\tau - t)} H(\tau - t)}_{\text{update}}, \tag{2.5}$$

where $H(t)$ is the heaviside step function. The next step to take is a "greedy" minimization of the loss at equation 2.3. This "greedy" approximation accounts only for the instantaneous change in the readout, that is, it only looks to the effect of the spike at the spike time t and discards its effect into the future. Then, at $\tau = t$ the spiking rule at 2.4 is written as (see Attachment A for the derivation)

$$\mathbf{D}_i^\top [\mathbf{x}(t) - \hat{\mathbf{x}}(t)] > \frac{\mathbf{D}_i^\top \mathbf{D}_i}{2}. \tag{2.6}$$

Since the i-th neuron spikes whenever its voltage V_i exceeds its threshold T_i, we can interpret the left-hand-side as the neuron's membrane potential

$$V_i(t) = \mathbf{D}_i^\top [\mathbf{x}(t) - \hat{\mathbf{x}}(t)], \tag{2.7}$$

and the right-hand-side as its threshold

$$T_i = \frac{\mathbf{D}_i^\top \mathbf{D}_i}{2}. \tag{2.8}$$

By taking the derivative of V_i we get

$$\dot{V}_i(t) = \mathbf{D}_i^\top [\dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t)], \tag{2.9}$$

where $\dot{\hat{\mathbf{x}}}$ is replaced by the dynamics equation 2.2, yielding

$$\begin{aligned} \dot{V}_i(t) &= \mathbf{D}_i^\top \dot{\mathbf{x}}(t) - \mathbf{D}_i^\top \left[-\lambda_d \hat{\mathbf{x}}(t) + \sum_k \mathbf{D}_k s_k(t) \right] \\ \dot{V}_i(t) &= \mathbf{D}_i^\top \dot{\mathbf{x}}(t) + \lambda_d \mathbf{D}_i^\top \hat{\mathbf{x}}(t) - \sum_k \mathbf{D}_i^\top \mathbf{D}_k s_k(t). \end{aligned} \tag{2.10}$$

Then, rewriting equation 2.7, we get $\mathbf{D}_i^\top \hat{\mathbf{x}}(t) = -V_i(t) + \mathbf{D}_i^\top \mathbf{x}(t)$. This term can then be replaced in the equation above to get

$$\dot{V}_i(t) = -\lambda_d V_i(t) + \mathbf{D}_i^\top [\dot{\mathbf{x}}(t) + \lambda_d \mathbf{x}(t)] - \sum_k \mathbf{D}_i^\top \mathbf{D}_k s_k(t), \tag{2.11}$$

which is the equation of a network of current-based, leaky integrate-and-fire neurons. Here, the term $\Omega_{ik} = -\mathbf{D}_i^\top \mathbf{D}_k$ can be interpreted as a lateral connection between neurons i and k in the network, also known as recurrent connectivity. When the voltage V_i reaches the threshold T_i, the self-connection $\Omega_{ii} = -\mathbf{D}_i^\top \mathbf{D}_i$ leads to a reset of the voltage to $V_{reset} = T + \Omega_{ii}$.

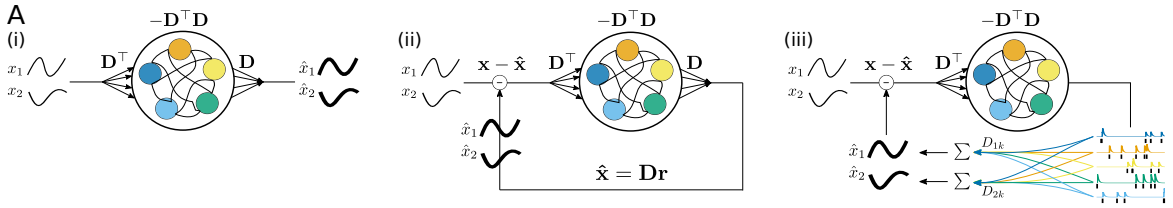

Figure 2.1: A SCN works like an autoencoder. (i) A SCN is a network with a feedforward matrix $\mathbf{D}^\top$ and a recurrent matrix $-\mathbf{D}^\top\mathbf{D}$ that is able to readout an input signal $\mathbf{x}$, yielding an estimate $\hat{\mathbf{x}}$. (ii) Same as (i), but unfolded to show that the readout is decoded from the network's spike trains, using the readout weights $\mathbf{D}$. (iii) Same as (ii), but further unfolded to show that the spike trains are first filtered with a postsynaptic potential. Then, these are weighted by the corresponding decoding weight and summed linearly to yield the signal's estimate $\hat{\mathbf{x}}$.

If we add some noise to equation 2.11 and write it in a vector form we obtain

$$\dot{\mathbf{V}}(t) = \underbrace{-\lambda_d \mathbf{V}(t)}_{\text{leak}} + \underbrace{\mathbf{D}^\top \mathbf{c}(t)}_{\text{input}} - \underbrace{\mathbf{D}^\top \mathbf{D}\mathbf{s}(t)}_{\substack{\text{fast recurrent} \\ \text{connections}}} + \underbrace{\sigma_v \eta(t)}_{\text{noise}}, \tag{2.12}$$

where $\mathbf{c}(t) = \dot{\mathbf{x}}(t) + \lambda_d \mathbf{x}(t)$ is the command input.

2.2 The Error Bounding Box

Besides the analytical framework derived above, there is also a geometric framework underlying SCNs. Calaim, Dehmelt, Gonçalves, *et al.* [53] introduced a geometric perspective about how the error between the signal $\mathbf{x}$ and the readout $\hat{\mathbf{x}}$ is bounded, i.e., the error bounding box. Moreover, Calaim, Dehmelt, Gonçalves, *et al.* [53] also provided a geometric interpretation of the spikes, the voltage, the thresholds and the perturbations that a SCN might undergo. Next we introduce some of the geometric features of SCNs.

Firstly, equation 2.5 tells us that whenever the i-th neuron spikes, the readout vector $\hat{\mathbf{x}}$ jumps along the direction of the decoding vector $\mathbf{D}_i$ (figure 2.2A). Thus, one neuron can only code for the readout along a single direction, predefined by its decoding vector $\mathbf{D}_i$, which may not be sufficient to bring $\hat{\mathbf{x}}$ close enough to $\mathbf{x}$. Adding another neuron may help on that (figure 2.2B). Also, note that each neuron defines a boundary between a spike and a no-spike region in the signal space and that the decoding vector $\mathbf{D}_i$ is perpendicular to this boundary. Now, if the network has several neurons with different decoding vectors, a larger set of directions may be encoded and the tracking improved (figure 2.2C and E). The bounding box is always centered in the signal $\mathbf{x}$ while the readout $\hat{\mathbf{x}}$ wanders within it. When it hits a boundary there is a spike, an instantaneous jump of the readout vector and finally it decays to 0, which amounts to the movement of the estimate towards the origin of the reference frame.

Under this geometric framework, the neuron's voltage, given by equation 2.7, is interpreted as the projection of the readout error onto the neuron's decoding vector (figure 2.2D). Then, each neuron does not have access to the global error, but rather to a local projection of it.

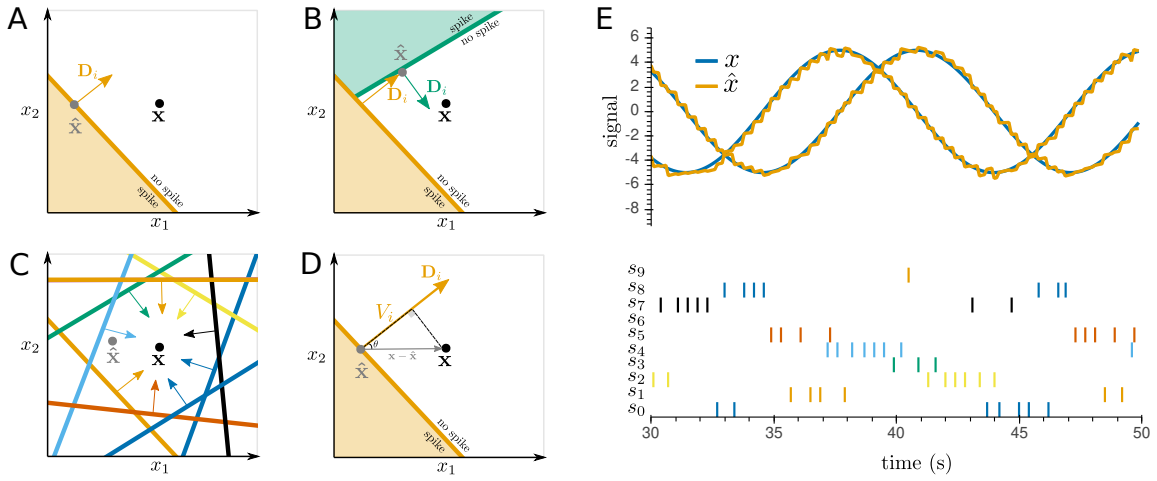

Figure 2.2: The error bounding-box. **A**. One neuron defines a spike/no-spike boundary. When the readout hits this boundary it jumps along a perpendicular direction to it, according to the decoding vector. One neuron is only able to correct the readout along one direction. **B**. Adding a second neuron with a different decoding vector allows the readout to jump closer to the signal. **C**. Adding several neurons defines a box that bounds the readout, allowing a correction along many directions. **D**. The neuron's voltage is the projection of the error vector along the neuron's decoding vector. **E**. Tracking of a 2D signal by a ten-neuron network with different decoding vectors. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron.

Although the above figure shows the geometric interpretation for a SCN tracking a 2D signal, the geometric framework may be generalized for higher-dimensional signals. In fact, if the M-dimensional decoders span the M-dimensional space properly, a box that bounds both $\mathbf{x}$ and $\hat{\mathbf{x}}$ emerges. The geometry of such bounding box is thus defined by both the dimensionality of the signal, M, and the number of neurons in the network, N. Then, in a 2D signal, the bounding box is a polygon where each neuron represents an edge, whereas in a 3D signal, the bounding box is a polyhedron where each neuron represents a face (figure 2.3).

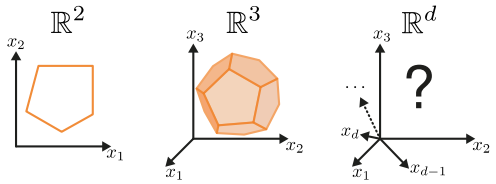

Figure 2.3: High-dimensional bounding boxes Taken from [53]. The dimensionality of the bounding box depends on the dimensionality of the signal $\mathbf{x}$.

2.3 Adding Spike Costs to SCNs

We started by postulating that SCNs greedily minimize the loss function at equation 2.3. However, in order to force the network to track the signal with the fewest spikes possible a linear cost term may be added. Furthermore, adding a quadratic cost term forces the network to distribute the spikes across neurons. Then, we write the new loss function

$$L(t) = ||\mathbf{x}(t) - \hat{\mathbf{x}}(t)||_2^2 + \mu||\mathbf{r}(t)||_2^2 + \nu||\mathbf{r}(t)||_1, \tag{2.13}$$

where $||.||_1$ denotes the L1 norm and μ and ν are control parameters. Then, the spiking rule becomes

$$\mathbf{D}_i^\top[\mathbf{x}(t) - \hat{\mathbf{x}}(t)] - \mu r_i(t) > \frac{1}{2}(\mathbf{D}_i^\top \mathbf{D}_i + \mu + \nu). \tag{2.14}$$

Again, we identify the time-dependent terms on the left hand-side as the voltages

$$V_i(t) = \mathbf{D}_i^\top[\mathbf{x}(t) - \hat{\mathbf{x}}(t)] - \mu r_i(t), \tag{2.15}$$

and the constant terms on the right hand-side as the thresholds

$$T_i = \frac{1}{2}(\mathbf{D}_i^\top \mathbf{D}_i + \mu + \nu). \tag{2.16}$$

An inspection of the voltage equation shows that when the i-th neuron has a high firing rate its voltage is pushed down, preventing the neuron from further hitting the threshold. Moreover, the threshold itself is higher due to the addition of μ and ν terms. Then, the cost terms enforce a sparse population response with low individual firing rates.

2.4 SCNs are Optimal Realizations of LIF Networks

A LIF network with recurrent and feedforward connections may be broadly described by the following dynamics [56]

$$\dot{\mathbf{V}}(t) = -\lambda_d \mathbf{V}(t) + \mathbf{F}\mathbf{c}(t) + \mathbf{\Omega}\mathbf{s}(t), \tag{2.17}$$

where $\mathbf{F} \in \mathbb{R}^{N\times M}$ is a matrix of feedforward weights and $\mathbf{\Omega} \in \mathbb{R}^{N\times N}$ is a matrix of recurrent weights (figure 2.4A). If these matrices are non-dependent on the decoding weights, then the architecture is non-optimal and there is an encoding-decoding mismatch. Then, in such network the voltage of the i-th neuron is given by

$$V_i(t) = \mathbf{F}_i^\top \mathbf{x}(t) + \mathbf{\Omega}_i^\top \mathbf{r}(t). \tag{2.18}$$

Enforcing an error-driven coding by the SCN allows the network to track the signal by creating a bounding box. Mathematically, to enforce an error-drive coding means that the local reconstruction error should be close to 0. Since the local reconstruction error is nothing less than the neuron's voltage, we write the error-driven coding condition as

$$V_i(t) = \mathbf{F}_i^\top \mathbf{x}(t) + \mathbf{\Omega}_i^\top \mathbf{r}(t) = \mathbf{F}_i^\top \mathbf{x}(t) - \mathbf{F}_i^\top \hat{\mathbf{x}}(t) \approx 0, \tag{2.19}$$

yielding the condition of $\mathbf{\Omega} = -\mathbf{F}\mathbf{D}$ (figure 2.4B).

Although the above condition is necessary to bound the readout, it is not the optimal architecture yet, since the feedforward vectors are not aligned with the decoding vectors (figure 2.4C). The optimal architecture is obtained when the goal of the SCN is to minimize a loss function. If such loss function is the one without costs, as written in equation 2.3, then we get that the optimal architecture follows

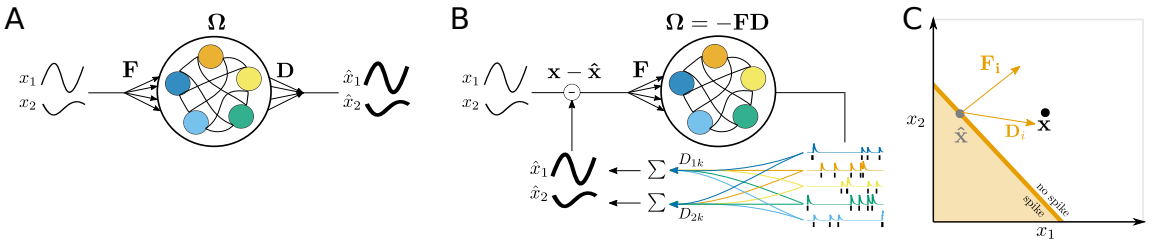

Figure 2.4: A generic LIF network with feedforward and recurrent matrices tracking a signal. A. This LIF network has a feedforward matrix $\mathbf{F}$ and a recurrent matrix $\mathbf{\Omega}$ that gets an input signal $\mathbf{x}$ and outputs a readout $\hat{\mathbf{x}}$. **B.** Unfolded network that follows an error-driven coding and thus has $\mathbf{\Omega} = -\mathbf{FD}$. **C.** A network with the recurrent matrix of B is non-optimal because the $\mathbf{F}_i$ vectors are not aligned with the $\mathbf{D}_i$ vectors. This leads to an encoding-decoding mismatch since the i-th neuron measures the error along $\mathbf{F}_i$ but then spikes along $\mathbf{D}_i$.

$$\mathbf{F} = \mathbf{D}^\top, \tag{2.20}$$

$$\mathbf{\Omega} = -\mathbf{D}^\top\mathbf{D}, \tag{2.21}$$

yielding equation 2.12.

On the other hand, if the loss function to be minimized has costs, as the one in equation 2.13, the optimal architecture becomes

$$\mathbf{F} = \mathbf{D}^\top, \tag{2.22}$$

$$\mathbf{\Omega} = -\mathbf{D}^\top\mathbf{D} - \mu\mathbf{I}. \tag{2.23}$$

Thus, the optimal architecture of a SCN depends on the loss function it is minimizing.

2.5 Biological Plausibility of SCNs

2.5.1 Spiking Variability

In vivo cortical cells have shown irregular firing in response to a sensory stimulus [8]–[11]. That is, trials recorded under the same conditions of the same stimulus presentation yield variable spike trains by a single neuron. This irregularity, expressed both in the number of spikes and its temporal distribution, can be described by a Poisson-like variability measured by the Coefficient of Variation (CV). The CV of a single spike train is given by the ratio between the standard deviation of the Interspike Intervals (ISIs) and their mean

$$\mathrm{CV} = \frac{\sigma_{\mathrm{ISI}}}{\mu_{\mathrm{ISI}}}. \tag{2.24}$$

If the spikes are generated by a Poisson process, then $\mathrm{CV} = 1$. If $\mathrm{CV} > 1$, then the spike train is less regular than a Poisson process, for a given firing, while if $\mathrm{CV} < 1$ it is more regular [57]. That said, neural spike trains have been shown to have $\mathrm{CV} \approx 1$ [8] (figure 2.5A).

Now, while it has been shown that balanced LIF networks may be chaotic and thus explain how deterministic neurons may generate Poisson-like statistics, they usually do not aim at computing any particular function [58], [59]. Interestingly, SCNs also yield spike trains with $\mathrm{CV} \approx 1$ [53] (figure 2.5) while aiming at tracking a signal efficiently.

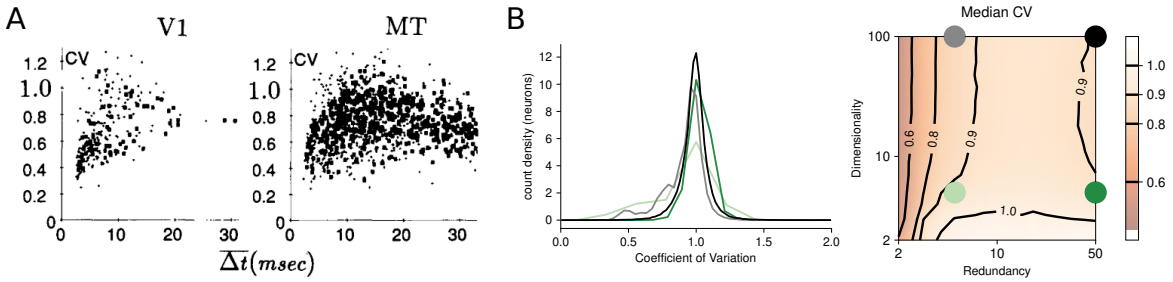

Figure 2.5: Poisson-like variability in neural data and SCNs. **A**. Taken from [8] Copyright 1993 Society for Neuroscience. The firing of almost all neurons in Visual Area 1 (Primary Visual Cortex) (V1) and Middle Temporal (MT) was consistent with a Poisson process (CV ≈ 1). **B**. Taken from [53]. *Left*: Distribution of CV for four different simulation conditions of redundancy (computed as N/M) and input dimensionality. *Right*: Median of CV distributions. Colors of the distribution on the left match the dots in the right plot. In SCNs, for low redundancies there are low CVs and for high redundancy we see CV ≈ 1.

2.5.2 E/I Tight Balance

In what comes the underling mechanisms of this neural variability, the hypothesis of balanced excitatory and inhibitory currents (E/I balance) is a strong candidate to explain the Poisson statistics [60]. This balance may be either loose or tight (figure 2.6A). In loosely balanced networks, excitation and inhibition balance each other on a slower time scale, remaining uncorrelated on a faster one [58], [61], [62]. In tight balanced networks, excitation and inhibition are coupled in time and inhibition tracks excitation on a fast timescale [63], [64].

Although the loosely balanced regime outputs spike trains with Poisson statistics, it does not explain other experimental observations, as the correlated synaptic inputs received by adjacent neurons and the correlation of the membrane potentials of identically tuned neurons [60]. On the other hand, under the tight balance hypothesis, each spike is a result of inhibition failing to track excitation rather than a noisy consequence of random voltage fluctuations.

In SCNs, a spike is fired with the purpose of decreasing the representation error. Biophysically, the depolarization of the neuron's membrane that causes this spike reflects an increase of the error. Since this is a depolarization, then it is mediated by excitatory currents. After a spike, the neuron immediately resets its voltage through an inhibitory instantaneous current (figure 2.6B). Then, the SCNs balance coding errors through a tight balance of excitation and inhibition [65].

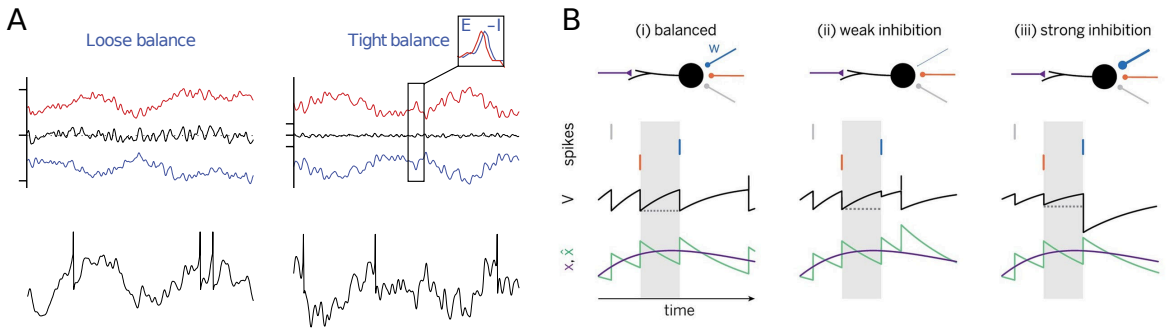

Figure 2.6: E/I balance. **A**. Either loose or tight balance produce highly variable spike trains. Taken from [60]. *Left*: Loose balance. *Right*: Tight balance. **B**. Taken from [56]. (i) In the optimal SCN an inhibitory spike balances perfectly the excitatory input and the signal is perfectly tracked. (ii) The input drive is not perfectly balanced, yielding a spike too early. (iii) The inhibition is stronger than the excitation and the spiking is delayed.

2.5.3 Robustness

Robust systems are able to keep a functional response while being subject to some internal or external perturbations. Many neural systems have been implied in robustness. For instance, the stomatogastric system in crustaceans has been implied in the maintenance of normal activity patterns despite large temperature changes [66]. And the hippocampus was robust against optogenetic silencing of place cells, by enforcing an alternative map [67].

Calaim, Dehmelt, Gonçalves, *et al.* [53] showed that SCNs are able to deal with a set of perturbations by adjusting the neuronal activity to fulfill its task (figure 2.7). In fact, SCNs are robust to either structural perturbations, as neural death or neural birth, and biophysical perturbations, as voltage or synaptic noise. These perturbations can be mapped onto the bounding box and while the bounding box remains a closed hyper-surface, the tracking holds and the network's performance is practically unaffected.

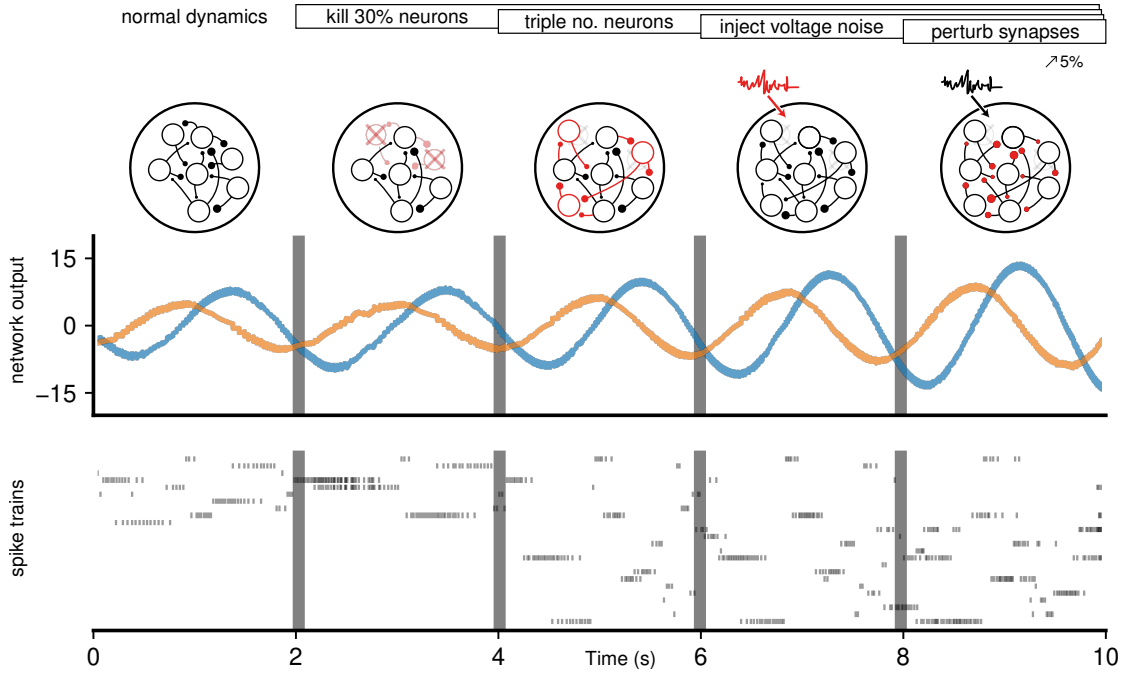

Figure 2.7: Robustness of SCNs. Taken from [53]. *Top*: Perturbations added along the simulation. *Middle*: Signal $x(t)$ and its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron.

Furthermore, robustness is a property observed ubiquitously in neural and, more broadly, biological systems. Therefore, it is important to acknowledge the underling principles of it. That said, Kitano [68] proposed four mechanisms that underline robustness: (1) system control, (2) alternative mechanisms, (3) modularity and (4) decoupling. Calaim, Dehmelt, Gonçalves, *et al.* [53] also encountered these mechanisms in SCNs [53]. In fact, lateral recurrent connectivity enforces negative feedback (1). When the decoding vectors span the full space, it introduces both heterogeneity and redundancy in the tracking (2). When $N >> M$ there are alternative sets of neurons that may be recruited to track the signal. Thus, SCNs are robust modules that adapt their neural codes in order to correct the perturbations instead of passing them onto downstream systems (3). Furthermore, the mechanisms underling the encoding (signal to spikes) are decoupled from the mechanisms underlying the decoding (spikes to readout) (4).

Finally, Kitano [68] also postulated that systems robust against certain perturbations are fragile

against others. That is, robustness comes along with some trade-offs. The SCNs are fragile against perturbations that shrink the bounding box.

2.6 Limitations of SCNs

2.6.1 Instantaneous Propagation of Spikes

The instantaneous propagation of spikes is mathematically evident when we inspect the LIF dynamics of the network in equation 2.12. Here, we see the postsynaptic currents coming from the recurrent connections are modeled as delta Dirac impulses, which biophysically translates into the instantaneous movement of ions through the ionic channels on the membrane of the postsynaptic neuron. The integration of these type of delta Dirac impulses leads to decaying exponential excitatory or inhibitory postsynaptic potentials.

Next, we describe a consequence of the instantaneous communication of spikes: the ping pong-effect. The ping pong effect occurs when a spike of the i-th neuron at time t causes an instantaneous spike of the j-th neuron, creating this bouncing of spikes that fails to track the signal efficiently.

To exemplify this effect, let us consider a two-neuron network tracking an 1D signal with oppositely tuned neurons, i.e., $D_0 = -1$ and $D_1 = +1$ (figure 2.8A). Note that since the only difference between the decoding weights is its sign, then, at every time point, $V_0(t) = -V_1(t)$. In what comes to the thresholds, although they may be defined by equation 2.8, one may also assign them other values or make them noisy. Then, the choice of the threshold may affect the tracking. Next, we will see how the instantaneous propagation of spikes leads to non-optimal spike trains when the thresholds are too low.

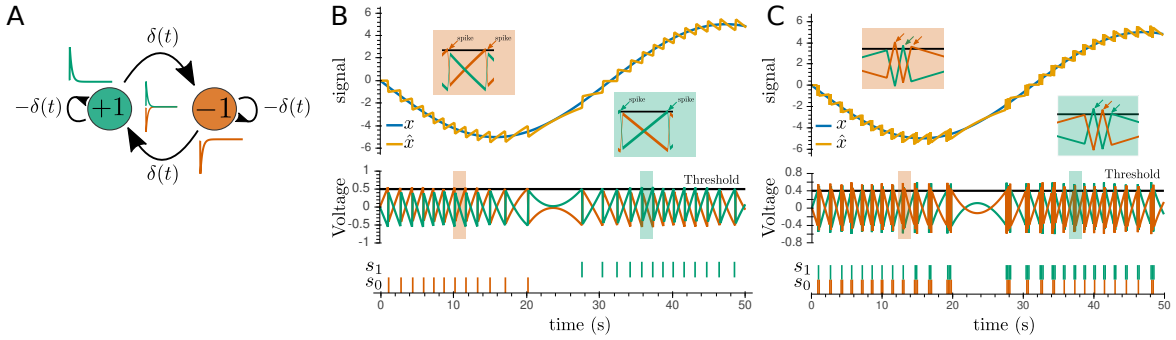

Figure 2.8: Effects of the instantaneous propagation of spikes . **A**. Diagram of the synaptic interactions between two neurons oppositely tuned ($D_0 = -1$ and $D_1 = +1$). **B**. Tracking of a 2D signal by two oppositely-tuned neurons without ping-pong. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$. *Middle*: Evolution of the voltage $V(t)$ for each neuron. *Bottom*: Spike trains $s(t)$ for each neuron. Note that the sign of the decoding vector spiking matches the sign of the signal being tracked. Shaded areas of the voltage are zoomed-in. **C**. Tracking of a 2D signal by two oppositely-tuned neurons with ping-pong. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$. *Middle*: Evolution of the voltage $V(t)$ for each neuron. *Bottom*: Spike trains $s(t)$ for each neuron. Shaded areas of the voltage are zoomed-in.

Firstly, mathematically, we write the instantaneous change in the neurons' voltages caused by a spike at t by the 0-th neuron as

$$\begin{aligned} V_0(t_+) &\to V_0(t_-) - D_0 D_0, \\ V_1(t_+) &\to V_1(t_-) - D_0 D_1, \end{aligned} \tag{2.25}$$

where $t_- = t_+ = t$, but we write it differently to note the difference of the value of the voltage V before (t_-) and after (t_+) a spike.

Then, according to equation 2.11, a spike by the first neuron at time t injects a current of $-D_0 D_0 \delta(t)$ into its own soma and a current of $-D_1 D_0 \delta(t) = D_0 D_0 \delta(t)$ into the other neuron's soma. This is

to say that each neuron excites the other while having an inhibitory autapse. Now, depending on the value of the threshold of the second neuron, the instantaneous excitation coming from the first neuron's spike may be sufficient for the second neuron to hit its threshold. If the threshold is high enough, this does not happen and the tracking is optimal (figure 2.8B). However, if the threshold is too small, the second neuron spikes and the ping-pong effect happens (figure 2.8C), i.e, both neurons yield several unnecessary spikes.

2.6.2 Synaptic Delays in SCNs

In the last section we showed how the simulation of SCNs relies on the instantaneous propagation of spikes and its consequences under certain conditions. Here, in an attempt to solve this issue, we introduce the synaptic delay θ, which delays both the lateral excitation or inhibition and the readout. The autapse of each neuron remains instantaneous in order to emulate the refractory period that follows a spike. Thus, equation 2.7 becomes

$$V_i(t) = \mathbf{D}_i^\top \mathbf{x}(t) - \sum_{k=1}^{N} \mathbf{D}_i^\top \mathbf{D}_k (r_k(t) \cdot \delta_{ik} + r_k(t-\theta) \cdot (1-\delta_{ik})), \tag{2.26}$$

where δ_{ik} is the Kronecker's delta.

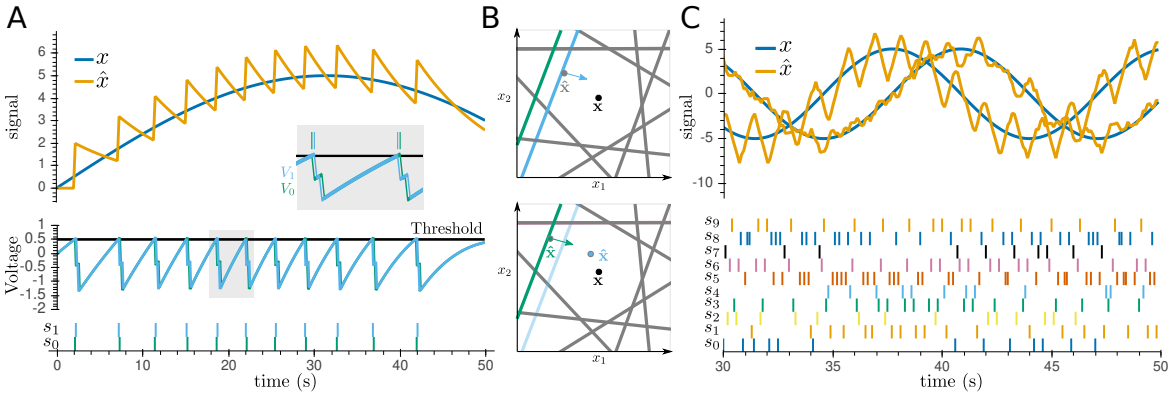

Figure 2.9: Synaptic delays in SCNs. **A**. Tracking of an 1D signal by a two-neuron network with identically tuned neurons. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$. *Middle*: Voltage $V(t)$ for each neuron. The shaded area is zoomed-in to illustrate that the delayed lateral inhibition allows the second neuron to spike. *Bottom*: Spike trains $s(t)$ for each neuron. **B**. In the geometric perspective, two identically tuned neurons are represented as parallel boundaries. *Top*: The readout $\hat{\mathbf{x}}$ hits the first boundary. This leads the spiking neuron to see a jump of the readout, but the rest of the network does not have access to this information yet. In the network's perspective, the readout did not jump and it is following its natural evolution. *Bottom*: Eventually, the readout hits the other neuron's boundary, yielding a spike. **C**. Tracking of a 2D signal by a ten-neuron network with synaptic delays. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron.

Considering now a two-neuron network with similarly tuned neurons tracking an 1D signal, the result is that for identically tuned decoders, delays lead to uninformed spikes that increase the tracking error (figure 2.9A). In this case, the lateral inhibition from the neuron that spikes first is delayed and thus the second neuron is allowed to spike. In the bounding box interpretation, two identically tuned neurons are represented as two parallel boundaries. When the readout $\hat{\mathbf{x}}$ hits the closest boundary the neuron spikes, but the instantaneous update of the voltage only occurs for the spiking neuron 2.9B). Thus, from the network perspective, the readout is still around the same position and following its natural evolution as if the first neuron had not spiked. This means that the readout may hit the parallel boundary behind the boundary that originated the first spike, leading to a new spike from another neuron. Comparing the tracking of a 2D signal by a SCN with (figure 2.9C) and without (figure 2.2E) synaptic delays, one qualitatively sees that the tracking with delays becomes highly

inefficient, since the SCN recruits more neurons, the neurons have higher firing rates and the tracking is worse, there being a great mismatch between the readout and the signal.

2.6.3 SCNs are Homogeneous Networks in the Time Domain

Previously we showed that SCNs could introduce heterogeneity by having different decoding vectors that correct the errors along different directions. However, when considering synaptic timescales, the network is homogeneous. In fact, equation 2.2 shows that this system is determined by one decaying constant, λ_d, that is shared by all neurons. This is to say that all synapses follow the same synaptic timescale. However, as stated in chapter 1, the heterogeneity of timescales is a biological feature that realistic Spiking Neural Networks (SNNs) should be able to implement.

2.6.4 Violation of Dale's Law

Dale's Law states that a neuron may either excite or inhibit all its post-synaptic neurons, but may never excite some and inhibit others [69]. In a SCN, a neuron inhibits similarly tuned neurons while exciting oppositely tuned neurons. Several solutions have been proposed to tackle this issue.

1. Boerlin, Machens, and Denève [52] proposed to create separate inhibitory and excitatory populations where the inhibitory population is tracking the readout of the excitatory population while the excitatory population is tracking the signal, an approach that further contributes to the E/I tight balance theory [65].
2. Mancoo, Keemink, and Machens [70] proposed to restrict the decoding weights $\mathbf{D}$ such that $\mathbf{D}_i^\top \mathbf{D}_j \geq 0, \forall_{i,j}$, i.e., all the neurons are inhibitory neurons. This option limits the set of signals a network is able to track.

2.6.5 A SCN is a Current-Based (CUBA) Model

As stated in chapter 1, Conductance-Based (COBA) models are more biologically realistic. However, as evidenced by equation 2.11, SCNs are CUBA models.

2.7 Conclusion

In this chapter, we showed that by following only two assumptions it is possible to derive efficient multidimensional LIF networks with biologically plausible emergent properties. Interestingly, these networks have CV values resembling a random process, while each spike is non-random, i.e., every spike has the efficient purpose of correcting an error. Furthermore, SCNs balance these coding errors through E/I tight balance while being robust to several noisy perturbations. Notwithstanding these strengths, SCNs also exhibit biologically unrealistic properties, as the instantaneous propagation of spikes and, consequently, homogeneous postsynatic potentials. In the following chapters we provide insights on how this instantaneous communication may be halted while keeping the networks functional and without losing their biological resemblances.

CHAPTER 3

Predicting the Future

I think neuroscience is obviously very esoteric, but I think there are aspects of it that can absolutely be brought down to the level of an interested 11, 12, 13 year-old easily.

— Mayim Bialik

In the last chapter, we showed that Spike Coding Networks (SCNs) perform a greedy minimization of a loss function. Specifically, the neuron corrects the present error by adding an exponential kernel to its filtered spike train. Given the instantaneous rise in the exponential kernel, it follows that a spike is instantaneously propagated through the network. Although this instantaneous communication of spikes ensures that the coding is efficient for optimal thresholds, i.e., minimizes the number of spikes needed for tracking the signal, it is biologically implausible.

Moreover, in chapter 1 we showed that there are several kernels able to model the postsynaptic current and, consequently, the postsynaptic potential. The exponential postsynaptic potentials used in SCNs represent the integration of delta Dirac postsynaptic currents, which are responsible for the instantaneous communication. However, this is not the most accurate model for biologically observed postsynaptic currents. Therefore, in this chapter we present an alternative framework based on the idea of slower postsynaptic currents and, consequently, slower postsynaptic potentials.

3.1 Slow Decoders

Let us assume that the dynamic of the network estimate $\hat{\mathbf{x}}$ is now given by

$$\dot{\hat{\mathbf{x}}}(t) = -a_d \hat{\mathbf{x}}(t) + a_d \mathbf{D}\mathbf{o}(t), \tag{3.1}$$

where a_d is the network's decaying constant and $\mathbf{o}(t) \in \mathbb{R}^{N\times 1}$, where each element is the i-th neuron's fast filtered spike train $o_i(t)$.

Again, the solution to equation 3.1 yields the network's readout as the linear summation of the neurons' filtered spike trains (see Attachment A for the derivation)

$$\hat{\mathbf{x}}(t) = \sum_{i=1}^{N} \mathbf{D}_i r_i, \tag{3.2}$$

where $r_i(t)$ is the slow filtered spike train for the i-th neuron.

In contrast with classical SCNs, in this framework each neuron has two dynamical variables, $o_i(t)$ and $r_i(t)$, describing its effect on the readout Mathematically, we write

$$\dot{o}_i(t) = -a_r^i o_i(t) + a_r^i s_i(t), \tag{3.3}$$

$$\dot{r}_i(t) = -a_d r_i(t) + a_d o_i(t). \tag{3.4}$$

Being the solutions, respectively,

$$o_i(t) = a_r^i e^{-a_r^i t} * s_i(t), \tag{3.5}$$

$$r_i(t) = a_r^i a_d e^{-a_d t} * e^{-a_r^i t} * s_i(t), \tag{3.6}$$

where we identify the fast filter as $e_i(t) = a_r^i e^{-a_r^i t} H(t)$ and the slow filter as $\alpha_i(t) = a_r^i a_d e^{-a_d t} * e^{-a_r^i t} H(t)$ (figure 3.1). Solving this convolution of exponentials, we get $\alpha_i(t) = \frac{a_r^i a_d}{a_d - a_r^i}(e^{-a_r^i t} - e^{-a_d t})H(t)$, a double exponential kernel. Being the kernels an analogy to postsynaptic potentials, then in this new definition of readout's dynamics we should get a more biologically plausible shape for the excitatory and the inhibitory postsynaptic potentials.

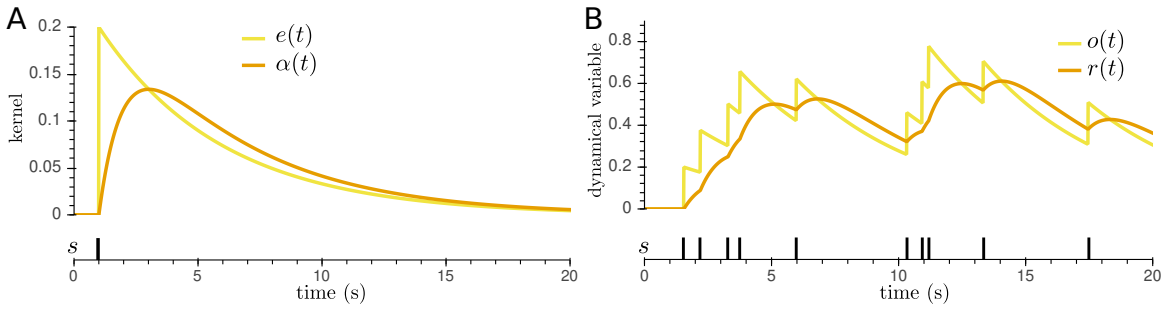

Figure 3.1: Fast and slow filters. **A**. The differences between an alpha and an exponential kernel. *Top:* Kernels *Bottom:* Spike train $s(t)$. **B**. *Top:* Fast $o(t)$ and slow $r(t)$ filtered spike trains. *Bottom:* Spike train $s(t)$.

Hence, if the i-th neuron spikes at time t its alpha-shaped kernel α_i is added onto the future of the readout, scaling its decoding vector $\mathbf{D}_i$ along time. Therefore, we have the following update in the network's estimate in the future time τ

$$\underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}}\text{ spike}} \rightarrow \underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}}\text{ silent}} + \underbrace{\mathbf{D}_i \alpha_i(\tau - t)H(\tau - t)}_{\text{update}}. \tag{3.7}$$

Now, we see that at the time of spike $\tau = t$ there is no change in the estimate $\hat{\mathbf{x}}$ since $\alpha_i(0) = 0$. Thus, the "greedy" instantaneous approximation is no longer an option and the i-th neuron is forced to predict the future of its estimate. In order to do that, we will see that the neuron must hold a set of assumptions about the network's behavior in the future it is looking into.

Moreover, the loss function to be minimized at a spiking decision time, t, is now an average over the prediction time window Δt

$$L(t) = \int_0^{\Delta t} ||\mathbf{x}'(t+\tau) - \hat{\mathbf{x}}'(t+\tau)||_2^2 d\tau, \tag{3.8}$$

which we call a predictive loss function. The quantities $\mathbf{x}'$ and $\hat{\mathbf{x}}'$ are the predictions used by the neuron for the (currently unknown) future signal and readout, respectively. What should these predictions be is what we are going to investigate.

Allora, under the second assumption of the SCNs, which states that the i-th neuron should only spike when it minimizes the predictive loss, i.e., spike if $L(t|i \text{ silent}) > L(t|i \text{ spikes})$, the resulting spiking rule now becomes

$$\int_0^{\Delta t} \alpha_i(\tau)\mathbf{D}_i^\top \left[\mathbf{x}'(t+\tau) - \hat{\mathbf{x}}'(t+\tau)\right] d\tau > \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} \alpha_i^2(\tau) d\tau. \tag{3.9}$$

Identifying the left hand-side of equation 3.9 as the i-th neuron's voltage V_i and the right hand-side as the i-th neuron's threshold T_i, we get

$$\begin{aligned} V_i(t) &= \int_0^{\Delta t} \alpha_i(\tau)\mathbf{D}_i^\top \left[\mathbf{x}'(t+\tau) - \hat{\mathbf{x}}'(t+\tau)\right] d\tau \\ T_i &= \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} \alpha_i^2(\tau) d\tau. \end{aligned} \tag{3.10}$$

Note that while the neuron's threshold depends only on the neuron's features, that is, its decoding weights and alpha kernel, the neuron's voltage depends on the predictions of both the input signal and the network's estimate. Thus, depending on the prediction of $\mathbf{x}$, $\hat{\mathbf{x}}$ and on the integration time window Δt, different spiking decisions may yield, which lead to a better or worse tracking of the signal. In the next sections, we will provide some insights about how these two constraints affect the tracking of the signal.

3.2 Prediction Boundaries

The decision of whether to spike or not depends on the prediction of the future of the estimate $\hat{\mathbf{x}}$ within a given time window starting at the decision time t. Here, we show that the prediction's domain is bounded by an upper and a lower limit, corresponding to two distinct biological scenarios that underline a set of assumptions. One the one hand, in the lower limit, when assessing the effect of a spike versus a no-spike, the i-th neuron makes two assumptions. First, it assumes that the readout evolves according to the past spikes yielded by the network. Second, it assumes that no other spike will occur, either by itself or other neuron in the network, within the integration time window.

On the other hand, the upper limit would correspond to the scenario in which there is a spike at each time step, within the integration time window, and the neuron spiking would be the one able to cause the biggest change on the estimate. The values attained are obviously dependent on the simulation time step, which forces the neuron to an absolute refractory period of the same duration as the discretization bin. Now, one can argue that the i-th neuron, that is the one making the prediction, may not have access the dynamics of every other neuron in the network or even its decoding vector. Thus, in order to fulfill the principle of Hebbian learning regarding local knowledge, we will assume that the upper limit of the i-th neuron's prediction is the scenario where the i-th neuron spikes at every future time step. These limits create a funnel shaped region that contains all possible predictions (figure 3.2).

It is worth mentioning that the predictive spiking rule presented at equation 3.9 is still a greedy spiking rule in the sense that it will lead to a spiking decision if a spike at the current time minimizes the loss in the future interval. However, it could be the case that delaying that spike leads to a further minimization of the full interval. Since placing a spike into the past or into the future would not be an online approach to this problem, the only way to ensure the neuron is doing the optimal tracking is if it is making the best prediction. For instance, in figure 3.3A, we see that if the neuron, at time t,

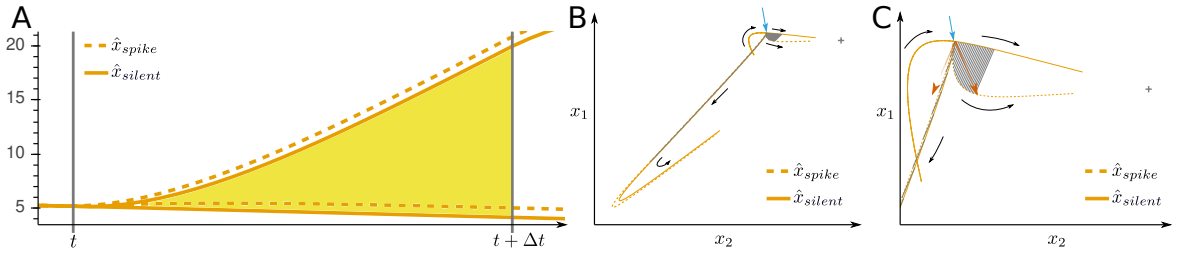

Figure 3.2: The neuron's prediction field. **A**. At time t the i-th the prediction of the i-th neuron regarding the future of the readout between the beginning (vertical line at t) and end (vertical line at $t + \Delta t$) of the integration window lies within the yellow region. **B**. Trajectories of the prediction boundaries for a 2D input. Black arrows represent the evolution in time. When the neuron hits a decision time point (blue arrow), it may predict the future according to the lower limit boundary, which keeps the estimate closer to the original point, or according to the upper boundary, which assumes future spikes of the i-th neuron, leading the estimate to a further point in space. The grey lines signal the difference between the spike and no-spike scenario for each prediction, up until the end of the preview time window. If we would let the signals evolve past this point, with no further spikes, they would decay to (0,0) (cross). In the 2D input case, the direction of the decoding vector also plays a role. **C**. Zoom-in of B. Although the direction of the decoding vector remains the same, the fact it has different starting points as the silent estimate evolves leads to different points at the corresponding spike estimate. Thus, the greatest effect of the spike is felt in another direction. The two dark orange arrows signal these directions of effect for each prediction boundary. The prediction field of the neuron is then given by the angle between these two directions.

predicts the future of the estimate according to the lower boundary limit, i.e., no further spikes are predicted within the integration window, the difference d between the voltage and the threshold is above 0, leading to the decision of spiking. On the other hand, if the neuron, at time t, predicts that there is a possibility of spiking at a future time t' within the integration window, then d is below the spiking threshold and the decision is not to fire at t, i.e., the spike is postponed.

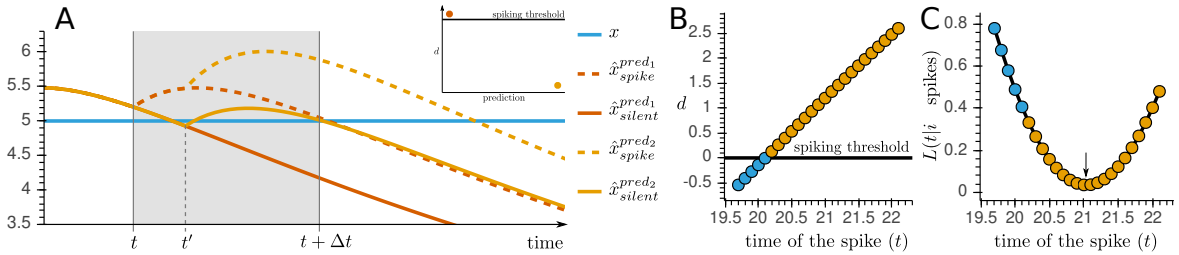

Figure 3.3: To postpone the spike is equivalent to predicting the future differently. **A**. At the time decision t, the neuron's voltage is above its threshold if the readout is being predicted according to $\hat{x}^{pred_1}$ and below it if the readout is being predicted according to $\hat{x}^{pred_2}$. $\hat{x}^{pred_1}$ is the prediction corresponding to the lower boundary limit, whereas $\hat{x}^{pred_2}$ is the prediction corresponding to having a spike at t'. **B**. d for a sequence of time points, considering the lower boundary limit prediction for every evaluation time. At every point the neuron evaluates its spiking rule, which is approaching the threshold. When it gets there, the neuron's greedy decision would be to spike. But should it spike at the first instant it hits the threshold? **C**. Evolution of $L(t|i$ spikes) for the same time points as B, considering the lower boundary limit prediction for every evaluation time. We see that the loss of spiking has a minimum (arrow), for a given integration window Δt, which corresponds to the optimal spike time to which the spike should be postponed in order to get a more accurate tracking of the signal.

Following an energy efficiency reasoning, the network only spikes when that decision leads to a minimization of the error. Then, although there are infinitely many predictions, the lower boundary is the extreme case where the neuron sees itself as the network's last resource. In other words, every time the neuron uses the lower boundary as a prediction and the spiking rule yields a no-spike as the best decision, this means that there is no error to correct from the last resource perspective. On the other hand, if the spiking rule yields a spike as the best decision, this means that there is a error to correct. However, instead of correcting it immediately in a greedy manner, the neuron should ask itself

if it can rely on the network for a bit longer in order to improve the tracking (figures 3.3B and C).

3.3 Spike's Effect into the Future

As stated in equation 3.7, the magnitude of the difference between the case of spiking or not is a scaled $\mathbf{D}_i$ where the scaling factor is time-dependent and its dynamics are ruled by an alpha kernel (figure 3.4). This time dependence leads to the consequence that spikes coming from the same neuron may have different global effects onto the readout depending on the dynamics of $\hat{\mathbf{x}}(t)$ at the moment of the spike. To understand this, consider the dynamics of an alpha function. The alpha function monotonically increases up to a global maximum and then it monotonically decreases. To the time the kernel takes to reach its global maximum we call the rising time h, such that $\alpha(h) = \max\{\alpha(t)\}$ (figure 3.4B). That said, the difference between a spike and a no-spike increases during h units of time and then it decreases until it vanishes due to the leaky effect to 0. Therefore, at the infinity we have $\lim_{t\to\infty} \alpha(t) = 0$ and the effect of spiking is barely felt by the network.

Now, if a spike occurs during the rising phase of the network's readout, its effect will be amplified by the past spikes (figure 3.4A), which have a slow cumulative effect due to the slow dynamics of the alpha kernel. On the other hand, a spike occurring at the descending phase of the network's estimate will have an attenuated effect onto the readout (figure 3.4C). That is, if there is a spike at time t, the value of the readout after the rising time, $\hat{\mathbf{x}}(t+h)$, will be shifted from the local maximum (figure 3.4C). As the readout descends progressively, the local maximum approximates the time point where the spike and the no-spike scenario differ the most, i.e., at $t+h$. When the readout reaches the baseline, i.e., when $\hat{\mathbf{x}} = \mathbf{0}$, then $\hat{\mathbf{x}}(t+h)$ is again a local maximum.

This is a major difference from the SCNs' framework, since the exponential kernel ensured that a spike by the i-th neuron always caused a jump of magnitude of $||\mathbf{D}_i||$ in the readout.

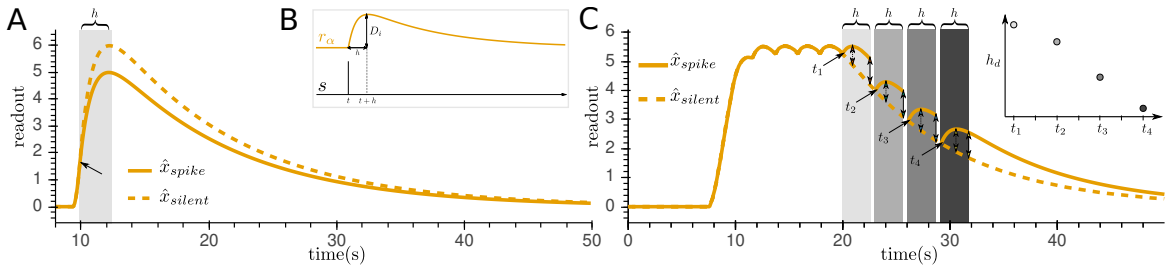

Figure 3.4: The effect of a slow spike in the readout. **A**. The peak of a spike in the ascending phase of the readout depends on the past spikes, which have a cumulative effect into the future. Here, the spike occurs when $\hat{x} = 1.6$ (arrow) and its maximum future effect is felt at $\hat{x} = 6$ due to the cumulative effect of past spikes. **B**. When the neuron's kernel is at the baseline, the rising time h is the time it takes for the firing rate to reach its peak. **C**. h is the time interval that it takes for the spike decision occurring at t_1, t_2, t_3 or t_4 (unidirectional arrows) to diverge the most from the silent decision. This distance is given by the length of $\mathbf{D}_i$. Here we illustrate the predictions at four decision time points. As the initial value of the readout decreases, the time interval h_d between the local maximum produced by the spike (dashed bidirectional arrows) and the end of the preview window (full line bidirectional arrows) decreases, indicating that the point of the maximum divergence of the spiking and no spiking hypothesis is approaching a local maximum of the readout.

3.4 Different Predictions Yield Different Network Behaviours

Here, we show the results for two alternative scenarios on how to predict the future of the estimate $\hat{\mathbf{x}}$ within a given time window starting at the decision time t.

Scenario I (figures 3.5 A and C) corresponds to the lower limit boundary described earlier. Then, for this scenario, the predictions follow the signal's and the estimate's true evolutions, considering no

further spikes. Accordingly, we write

$$\mathbf{x}'(t+\tau) = \mathbf{x}(t+\tau), \tag{3.11}$$

$$\hat{\mathbf{x}}'(t+\tau) = \mathbf{D}\mathbf{r}'(t+\tau), \tag{3.12}$$

where the j-th element of $\mathbf{r}'(t+\tau)$ is $r'_j(t+\tau) = \frac{a_r^j a_d}{a_d - a_r^j} \sum_{t_i^j < t} e^{-a_r^j(t-t_i^j)} - e^{-a_d(t-t_i^j)}$. Substituting these predictions in the predictive spiking rule in equation 3.9 we get that the i-th neuron's voltage is given by

$$V_i(t) = \int_0^{\Delta t} \alpha_i(\tau)\mathbf{D}_i^\top \left[\mathbf{x}(t+\tau) - \mathbf{D}\mathbf{r}'(t+\tau)\right] d\tau. \tag{3.13}$$

Let us define new variables for each component of the voltage. Namely,

$$\mathbf{X}_i(t) = \int_0^{\Delta t} \alpha_i(\tau)\mathbf{x}(t+\tau)d\tau, \tag{3.14}$$

$$\hat{\mathbf{X}}_i(t) = \int_0^{\Delta t} \alpha_i(\tau)\mathbf{D}\mathbf{r}'(t+\tau)d\tau,$$

where $\mathbf{X}_i(t) \in \mathbb{R}^{M\times 1}$ and $\hat{\mathbf{X}}_i(t) \in \mathbb{R}^{M\times 1}$. A simple interpretation of this is that the voltage is not the direct projection of the coding error onto the i-th neuron's decoding vector, but it is rather a projection of a weighted transformation of the predicted coding error. That weight is the i-th neuron's alpha kernel α_i. Given the dependence of these vectors on this kernel then, in a system with N neurons, there are N of each of these vectors.

Therefore, we rewrite the voltage of the i-th neuron as $V_i(t) = \mathbf{D}_i^\top \left[\mathbf{X}_i(t) - \hat{\mathbf{X}}_i(t)\right]$. Taking the derivative of this expression, we get the following neuron's dynamics (see Attachment B for the derivation)

$$\dot{V}_i(t) = -\underbrace{a_d V_i(t)}_{\text{leak}} + \underbrace{\mathbf{D}_i^\top\left[\dot{\mathbf{X}}_i(t) + a_d\mathbf{X}_i(t)\right]}_{\text{input}} - \underbrace{a_d\beta_i\mathbf{D}_i^\top\mathbf{D}^{\text{s}}{}_i\mathbf{o}(t)}_{\text{slow recurrent}} - \underbrace{\beta_i\mathbf{D}_i^\top\mathbf{D}^{\text{f}}{}_i\mathbf{s}(t)}_{\text{fast recurrent}}, \tag{3.15}$$

where $\mathbf{D}^{\text{s}}{}_i \in \mathbb{R}^{M\times N}$ and $\mathbf{D}^{\text{f}}{}_i \in \mathbb{R}^{M\times N}$. Again, given the dependence of these matrices on i, there is a different matrix for each neuron. Each column vector in $\mathbf{D}^{\text{s}}{}_i$ is given by $\mathbf{D}^{\text{s}}{}_{ij} = \mathbf{D}_j(\gamma_j - \gamma_{ij})$ and each column vector in $\mathbf{D}^{\text{f}}{}_i$ is given by $\mathbf{D}^{\text{f}}{}_{ij} = \mathbf{D}_j\beta_j(\gamma_j - \gamma_{ij} + \gamma_i - \gamma)$. These scalar values are defined in Attachment B.

In Scenario II (figures 3.5 B and D), the i-th neuron assumes that the estimate received at the decision time t, $\hat{\mathbf{x}}(t)$, is the new baseline for the network's readout. That is, in the case of a spike, the readout will decay to this new baseline instead of decaying to 0 whereas in the case of a no-spike the estimate holds to its current value. This assumption expects the network to maintain the current estimate during the integration time window. Mathematically, we write

$$\mathbf{x}'(t+\tau) = \mathbf{x}(t+\tau), \tag{3.16}$$

$$\hat{\mathbf{x}}'(t+\tau) = \hat{\mathbf{x}}(t). \tag{3.17}$$

Again, substituting these predictions in the predictive spiking rule in equation 3.9 we get that the i-th neuron's voltage is given by

$$V_i(t) = \int_0^{\Delta t} \alpha_i(\tau)\mathbf{D}_i^\top \left[\mathbf{x}(t+\tau) - \hat{\mathbf{x}}(t)\right] d\tau. \tag{3.18}$$

Now, we define

$$\mathbf{X}_i(t) = \int_0^{\Delta t} \alpha_i(\tau)\mathbf{x}(t+\tau)d\tau, \tag{3.19}$$
$$\hat{\mathbf{X}}_i(t) = \hat{\mathbf{x}}(t)\int_0^{\Delta t} \alpha_i(\tau)d\tau$$

and rewrite the voltage as $V_i(t) = \mathbf{D}_i^\top \left[\mathbf{X}_i(t) - \hat{\mathbf{X}}_i(t)\right]$. Taking its derivative, we get the neuron's dynamics

$$\dot{V}_i(t) = \underbrace{-a_d V_i(t)}_{\text{leak}} + \underbrace{\mathbf{D}_i^\top[\dot{\mathbf{X}}_i(t) + a_d\mathbf{X}_i(t)]}_{\text{input}} \underbrace{- a_d\mathbf{D}_i^\top\mathbf{D}\mathbf{o}(t)\int_0^{\Delta t}\alpha_i(\tau)d\tau}_{\text{slow recurrent}}. \tag{3.20}$$

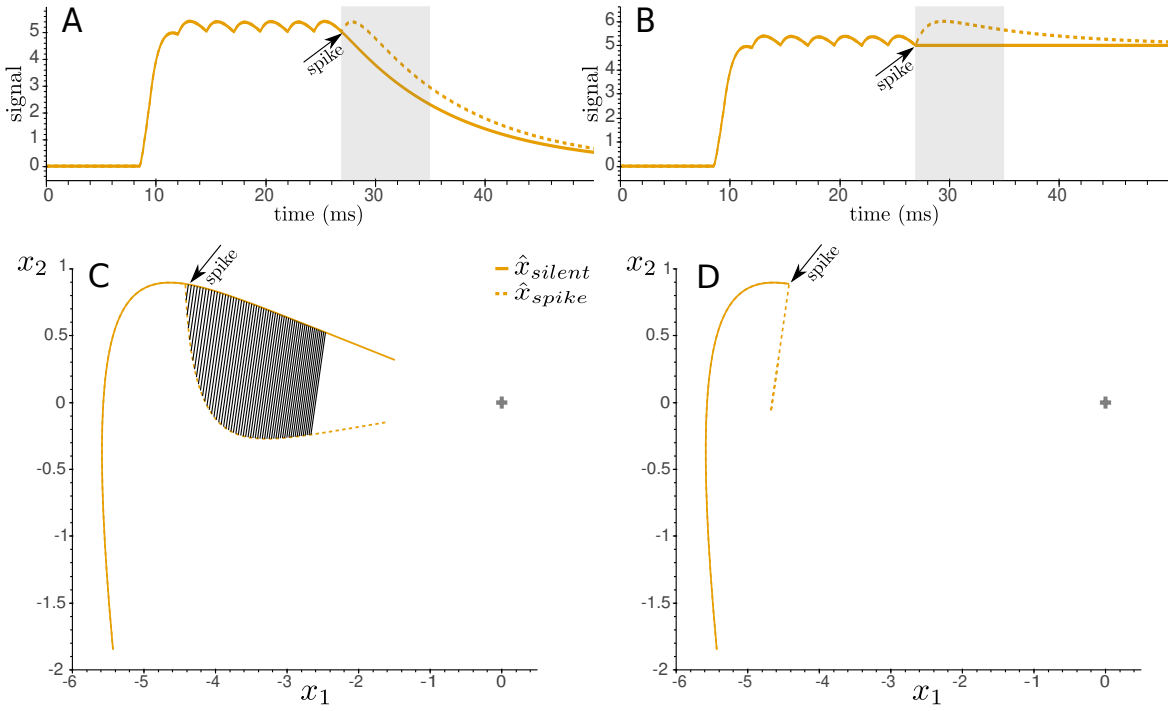

Figure 3.5: Diagram of the decaying of readout $\hat{x}$ within an integration time window (grey area). In every time step, the neuron has a decision to make: to spike or not to spike? Here, after some spikes had already been fired in the past, which cannot be changed, the neuron hits the current decision time (arrow) and, according to a defined spiking rule, makes its decision. **A**. 1D input signal considering scenario I. **B**. 1D input signal considering scenario II. **C**. 2D input signal considering scenario I. **D**. 2D input signal considering scenario II. (A and C) The neuron sees its future as if the spike at the given decision time is the only one within the integration window, which leads to the readout decaying to 0 (cross), considering the effect of past spikes. (B and D) The neuron is agnostic about the past and future spikes, only acknowledging the current estimate value, thus assuming that the readout must only remain constant or, in the case of a spike at the decision time, decay to the value of the readout at the time of the spike.

Next, we discuss the similarities and differences on tracking a signal by an one-neuron network according to scenario I or scenario II.

Intuitively, it is expectable that as the integration time window gets longer, the earlier in time the neuron sees the future changes in the signal and thus the earlier it can start coding for these changes. In fact, given that either in equation 3.15 or 3.20 the input term is the same, then in both cases the network "sees" the future of the signal in the same way. Therefore, this effect happens in both scenarios and to it we call the predictive power of the network (figures 3.6A and 3.7A). This

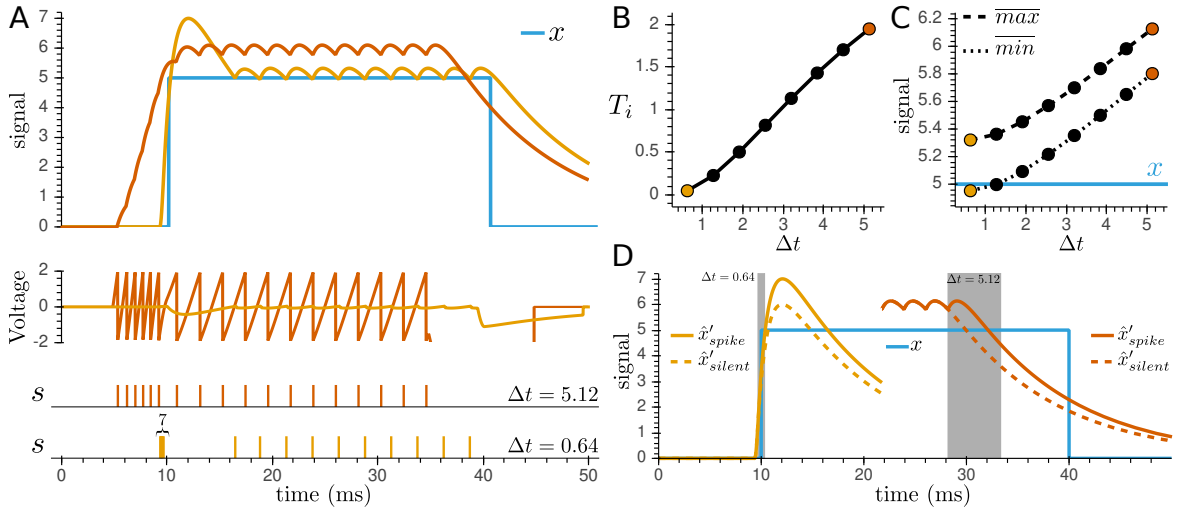

Figure 3.6: Tracking of a signal under scenario I A. *Top:* Estimate of an impulse signal by an one-neuron network (alpha kernel parameters: $a_r = 1$, $a_d = 0.1$), considering a long ($\Delta t = 2.56$ ms) and a short ($\Delta t = 0.64$ ms) integration time window. Note that the longer the integration window, the longer the predictive power of the network. *Middle:* Voltage for the long and short integration time window. Note that when V_i hits T_i it is reset to $-T_i$. *Bottom:* The neuron's spike train $s(t)$ for the long and short integration window. **B**. Dependence of the spiking threshold T_i on the integration time window Δt. **C**. Dependence of the average maximum and minimum of the readout on the integration time window Δt. **D**. Spike decisions that led to an over-estimation of the signal. If the estimate is at its rising phase and the integration window is too short, the neuron cannot see the further rise that a spike will cause, thus yielding the decision to spike. If the estimate is at its decaying phase and the integration window is too long, the neuron predicts a long decay on the network's estimate if it remains silent, thus yielding the decision to spike.

earlier coding of the signal avoids a high-frequency burst at the onset of the signal that occurs when the integration time window of the neuron is shorter. In fact, when the integration window is short, the neuron cannot see the onset of the signal up until the time point when it is about to happen. Moreover, the short integration window does not allow the neuron to see the great rise of the estimate caused by previous spikes (left panels in figures 3.6D and 3.7D). Note that this high-frequency burst is greater in a neuron predicting the future of the estimate according to scenario II. In this scenario, a spiking prediction during the rising phase of the readout under-estimates its true value, due to the new considered baseline, thus leading the neuron to consider it has to spike more than what it actually needs to. In what comes to the accuracy of the tracking during the constant portion of the input signal, we see that in scenario I, as the integration window increases, the signal is over-estimated (figure 3.6A), and the neuron is not able to bound the signal between its maximum and minimum estimates (figure 3.6C). This effect is explained by the fact that scenario I is the neuron's last resource. That is, since it assumes no further spike will occur during the integration window, then it must spike greedily as soon as it predicts a great mismatch between the effect of a spike versus a no-spike. Now, if the integration window is long, the readout prediction will have a long decay, leading the neuron to fire to prevent such an error (right panel in figure 3.6D). On the other hand, when predicting the future according to scenario II, the estimate's prediction is greater than the true estimate's evolution (right panel in figure 3.7D), due to the effects explained in the previous section, which leads the neuron to spike later and thus it slightly under-estimates the signal (figure 3.7C). It is also worth mentioning that while the right hand-side of equation 3.9 is the same for both scenarios, thus leading to the same dependency of the threshold T_i on the integration time window (figures 3.6B and 3.7B), this is not the case for the left hand-side, i.e, the neuron's voltage, as explicitly indicated in the dynamics equations 3.15 and 3.20. In fact, when looking into the evolution of the voltage during a steady signal, in scenario

I, whenever the voltage hits the threshold T_i, it is reset to $-T_i$ (middle panel in figure 3.6A). This reflects the instantaneous reset mathematically defined by the fast recurrent term in equation 3.15. Notwithstanding the fast recurrency, equation 3.15 also has a slow recurrent term that models the dynamics of the voltage after an instantaneous change. This slow modulation is better seen when the integration window is short (orange line in middle panel in figure 3.6A). On the other hand, in scenario II, whenever the voltage hits the threshold it is reset in a slower way, according to the slow recurrent term in equation 3.20. In this case, given that there is no fast recurrent term, there is no instantaneous reset of the voltage (middle panel in figure 3.7A). A higher level interpretation for this is the following: in scenario I, the neuron relies only on its present abilities only, thus it must correct the error as soon as it "sees" it. This greedy correction is followed by an instantaneous reset, which is the neuron saying "error corrected" (to the extent of the integration window). In scenario II, the neuron postpones the spike the longest it can, trusting that its future self may handle the job of correcting the error. Then, when it does spike, it does not reset itself instantaneously, which is actually a facilitating mechanism for further spikes. That is, if the neuron postponed the spike too much at least it should be able to burst in order to catch up with the signal when it realizes the mismatch is too great and a slow reset of the voltage instead of an instantaneous one facilitates just that.

In conclusion, scenario I shows that although the use of alpha-shaped postsynpatic potentials yield slow recurrencies, it is does not necessarily avoids fast recurrencies. Thus, depending on how the network is predicting its readout, the voltage shows different dynamics and the network yields different behaviours.

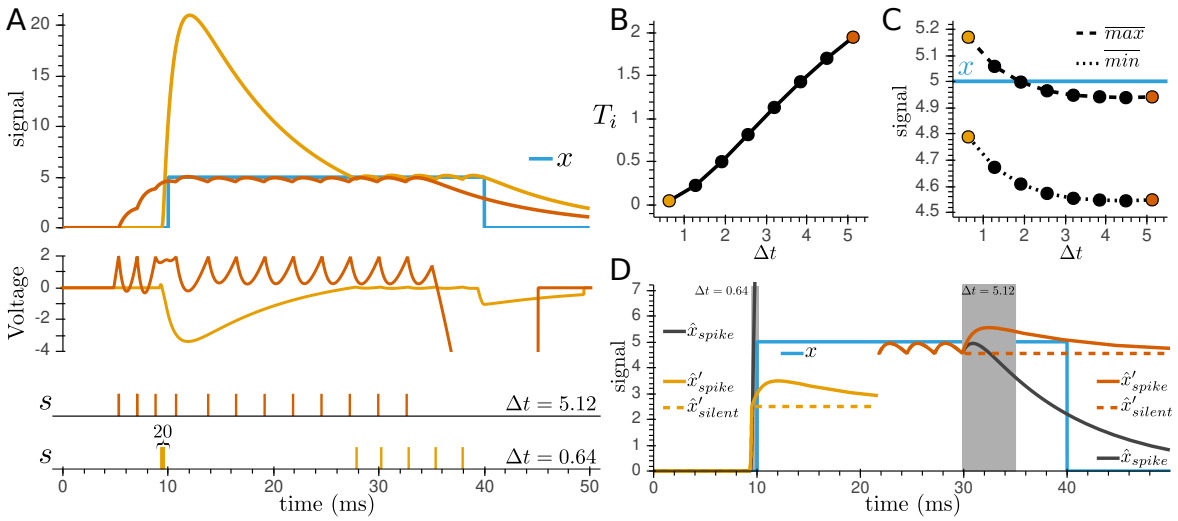

Figure 3.7: Tracking of a signal under scenario II. Same kernel parameters and integration windows as in figure 3.6. **A**. *Top:* Estimate of an impulse signal by an one-neuron network. *Middle:* Voltage for the long and short integration time window. *Bottom:* The neuron's spike train $s(t)$ for the long and short integration window. **B**. Dependence of the spiking threshold T_i on the integration time window Δt. **C**. Dependence of the average maximum and minimum of the readout on the integration time window Δt. **D**. Spike decisions that led to an over-estimation or under-estimation of the signal. Note that in this scenario the prediction in case of a spike ($\hat{x}'_{spike}$) is different from what actually happens if there is a spike ($\hat{x}_{spike}$), due to the new considered baseline. Thus, if the estimate is at its rising phase and the integration window is too short (left), the neuron not only cannot see the further rise that a predicted spike will cause, but it also does not see the great rise on the estimate already happening due to the past spikes that will be amplified by this new spike. If the estimate is at its decaying phase and the integration window is too long (right), the predicted effect of the spike is non-decaying, thus leading to a slight under-estimation.

3.5 Conclusion

The dynamic of slow decoders, namely the fact that at the time of the spike t, $\alpha(t) = 0$, forces the neuron to minimize a forward predictive loss function. Therefore, a network based on alpha kernels needs to predict the future of both the signal and its estimate in order to produce a spiking rule. There are infinitely many assumptions that a neuron may follow to predict these quantities, but some yield a better tracking than others. In the two alternative scenarios presented, the assumptions for the prediction of the signal were the same: the network assumes a perfect knowledge of the future, which although unrealistic, provides the perfect prediction in what comes to the signal. While it is trivial to identify the perfect prediction for the input signal, it is not so straightforward to identify the perfect prediction for the readout, i.e., the prediction that yields the optimal and most accurate tracking in the end. Thus, the question arises: how to choose the right assumptions to predict the future so that the tracking is the best possible while being robust to parameter changes and noise injection? In the following chapter we present a systematic approach to this question, hoping that it provides insights about how to build robust and biologically plausible Heterogeneous Spike Coding Networks (hSCNs).

CHAPTER 4

From Temporal Decoders to Heterogeneous Spike Coding Networks

I have always thought of theory in biology as disciplined dreaming. The discipline comes from the challenge of creatively marrying the rules of mathematics and physics with what is known of fundamental biological principles.

— Eve Marder

In the previous chapter, we showed that by using alpha kernels it is possible to get alpha-like postsynaptical potentials under the right assumptions on predicting the future of the network's estimate. In this chapter we develop a systematic approach to transform Spike Coding Networks (SCNs) into Heterogeneous Spike Coding Networks (hSCNs) through a set of properties that emerge when considering that a SCN is able to track a multidimensional signal in the discrete temporal domain. Moreover, we explore the features of these newly derived hSCNs geometrically, considering their advantage of allowing a heterogeneity of timescales.

4.1 Temporal Decoders in SCNs

Let us consider a M-dimensional signal where the dimensions represent discrete time. That is, each component m of the input signal is a forward time-shifted version of the previous one, i.e.: $x_m(t) = x_{m-1}(t + \Delta T)$ or, non-recursively, $x_m(t) = x_1(t + (m-1)\Delta T)$, where ΔT is the temporal shift. Then, we must distinguish the dimensionality in time from the dimensionality in space. The results in this chapter use an 1D input signal in space with M components in time.

A SCN tracks every temporal component at the same time (figure 4.1A), but the contribution of a spike by the i-th neuron to each component depends on its decoding weight along the corresponding component (figure 4.1B). In other words, if the dimensions represent time then a neuron may have a different weight for the present, the short-term and the long-term future. One constraint employed onto the decoders is that they may be either positive or negative along all time dimensions. That is, a neuron tracking a positive/negative signal in the present shall track a positive/negative signal in

the future (figures 4.1B and D). This is to avoid further violations of the Dale's law and to prevent a neuron from switching between being excitatory or inhibitory with time. Randomly generated decoding vectors that fulfill this hard constrain are able to track both the present (dimension 1) and the future (dimensions 2-5) of a 5D signal in time (figure 4.1A). However, when we inspect the decoder shapes along the five time dimensions (figures 4.1B and C) we see that given their randomness, they do not translate any specific dynamics. Thus, next we investigate if it is possible to track a signal in time using alpha-shaped decoders (figure 4.1E). That is, we aim at using slower decoders that follow the dynamics in equations 3.4, thus creating decoding vectors that code the future dynamically .

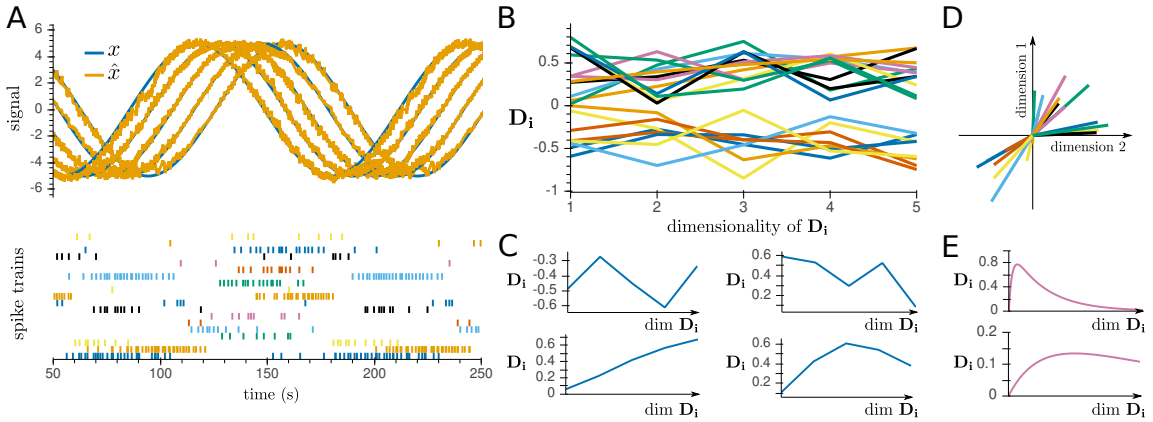

Figure 4.1: Random temporal decoders. **A**. Tracking of a 5D signal in time by a 20-neuron SCN. *Top*: Signal $x_m(t)$ dimensions and their estimates $\hat{x}_m(t)$. *Bottom*: Spike trains $s(t)$ for each neuron. **B**. Decoder weights for each neuron (each color is a neuron) along each temporal dimension. **C**. Decoder weights for 4 different neurons along each temporal dimension. *Top*: These time evolutions of the decoding weights do not resemble alpha kernels. *Bottom*: These time evolutions of the decoding weights resemble alpha kernels. **D**. The feature that time evolves forward constrains the temporal domain the decoders can span. Thus, only fully positive of fully negative quadrants are spanned. **E**. The temporal decoders could be approximated by alpha kernels with different parameters. *Top*: $a_d = 1$, $a_r = 10$. *Bottom*: $a_d = 1$, $a_r = 0.2$.

Our results show that it is indeed possible to use a SCN with alpha-shaped decoding vectors to track a signal in time (figure 4.2A), but this hard constraint also evinces an implicit feature of hSCNs: They do not code for the present, i.e., contrary to the classical SCNs, these networks are not aiming at correcting the instantaneous coding error at the time of the spike.

Recall that the key feature of the alpha kernels is that at the spike time t, $\alpha(t) = 0$. This implies that along the first dimension in time (the present), every neuron's decoding weight is 0 (figure 4.2B). Thus, when weighting the postsynaptic potential of every neuron along the first dimension, the result yielded is 0 (figure 4.2C). Then, despite the fact that we have three time components, since one is the present and thus is not being tracked, the problem resumes to a 2D problem in time where $x_2(t) = x_1(t + \Delta T)$ and $x_3(t) = x_1(t + 2\Delta T)$ are the dimensions tracked. Then, we may represent a bounding box in the 2D time-space (figure 4.2D).

The second emergent property of this approach is that a heterogeneity of decoding temporal vectors is required. That is, in order to close a box in the 2D time-space one needs a minimum of 3 neurons that originates 3 differently orientated edges. Since the orientation of those edges depends on the decoding vectors, then we need three different temporal decoding vectors. These different weights along dimension 2 (short-term future) and dimension 3 (long-term future) correspond to faster or slower alpha kernels. Furthermore, although 3 neurons, with a specific set of decoding vectors, may be sufficient to create a bounding box, it might not be sufficient to effectively track a specific signal. Indeed, if our signal is a sinusoid then a minimum of 4 neurons is required (figure 4.2A). With four neurons, a fast and a slow positive and a fast and a slow negative, a diamond-shaped temporal bounding box (figure

4.2D) is created. This bounding box then provides sufficient coding directions to track the ellipsoid trajectory of the signal in the time space (figure 4.2E).

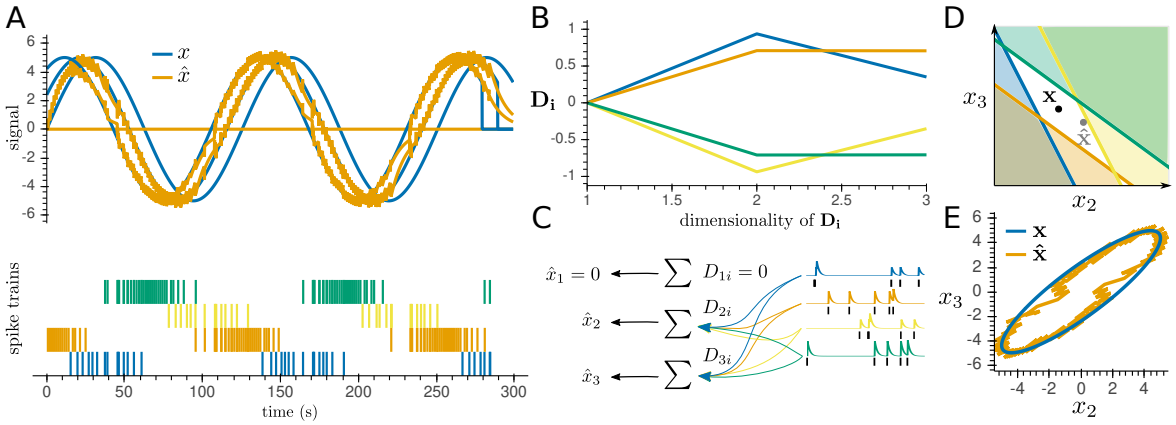

Figure 4.2: Temporal decoders as alpha kernels. **A**. Tracking of a 3D signal in time by a 4-neuron SCN. *Top*: Signal $x_m(t)$ dimensions and their estimates $\hat{x}_m(t)$. *Bottom*: Spike trains $s(t)$ for each neuron. **B**. Decoder weights for each neuron (each color is a neuron) along each temporal dimension. **C**. Each neuron's filtered spike train is multiplied by its decoding weight along each time dimension. Every decoding weight for the present has the value of 0, that is, the first dimension. Thus, the network fails to track the present (the curve $x_1(t)$ in A). **D**. Since the present is not tracked, the 3D problem in time is actually a 2D problem in time, where $x_2(t) = x_1(t + \Delta T)$ and $x_3(t) = x_1(t + 2\Delta T)$ are the dimensions tracked. **D**. Signal and readout trajectories in time.

The third emergent feature of this framework relates to the evolution of the readout in a spike or in a no-spike scenario. To study this, let us consider again a 5D time signal (figure 4.3A) tracked by a set of neurons with heterogeneous alpha-shaped decoding vectors (figure 4.3B).

Consider that while it is true that $x_m(t) = x_1(t + (m-1)\Delta T)$, it is not true that $\hat{x}_m(t) = \hat{x}_1(t + (m-1)\Delta T)$. Rather, since every component is tracked simultaneously and independently of each other, then each shifted component of the readout approximates, but is not equal to, the other in the corresponding time step: $\hat{x}_m(t) \approx \hat{x}_1(t + (m-1)\Delta T)$. Then, one may retrieve an average readout from all the four future components, in the matching time steps (figure 4.3B). Indeed, let us inspect the effect of a spike by the neuron whose temporal decoder is showed in figure 4.3D onto each component of the readout (figure 4.3E). Firstly, it is important to distinguish between the effect of a spike along the simulation time from the effect of a spike along the temporal dimensions. In fact, when the i-th neuron spikes, each component's readout jumps according to the decoding weight along that dimension, D_{mi}, and then decays as the simulation time goes by (left panel in figure 4.3E). However, the picture is different when we focus on the evolution from the perspective of the temporal dimensions (right panel in figure 4.3E). From this point of view, where we align chronologically the representation of each component in the spike time t, we see that the readout in the absence of a spike follows the dynamics of the signal $\mathbf{x}(t)$. Being the readout in the case of a spike related to the readout in the absence of a spike via equation 2.5, then obviously the readout in the case of a spike also follows the dynamics of the signal.

Now, in a hSCN, whose neurons decode the signal with slow decoders, the simulation time and the temporal dimensions should be aligned. Therefore, the observation that the silent prediction should follow the signal's dynamics is key to further develop the predictive framework.

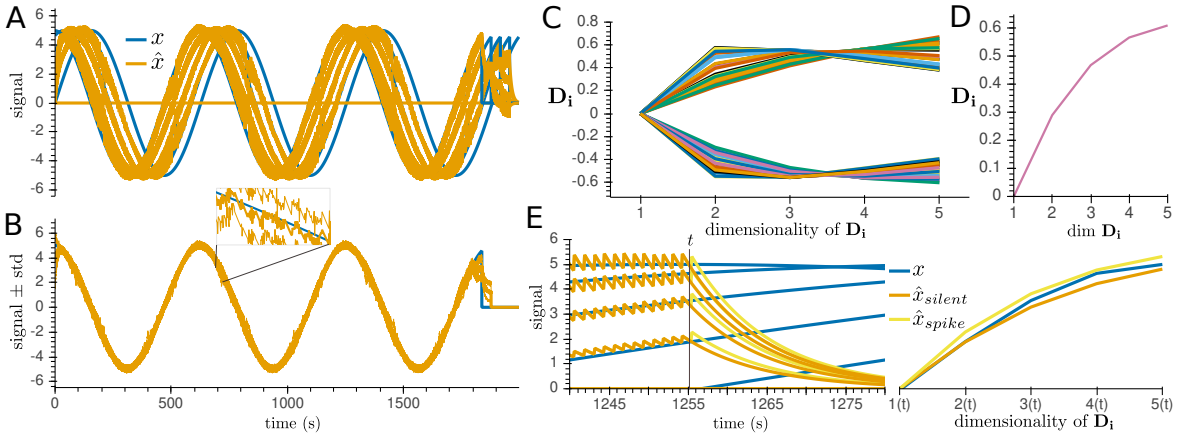

Figure 4.3: Temporal decoders provide insights about the predicted dynamics of the readout. **A**. Tracking of a 5D signal in time by a 50-neuron SCN. Signal $x(t)$ and its estimate $\hat{x}(t)$. **B**. Average of the signal $(\overline{\mathbf{x}}(t) \pm \sigma_x)$ and the readout $(\overline{\hat{\mathbf{x}}}(t) \pm \sigma_{\hat{x}})$ for each time step. **C**. Decoder weights for each neuron (each color is a neuron) along each temporal dimension. **D**. Decoder weight of the spiking neuron considered in E. **E**. What happens when the neuron with the decoder at C spikes? *Left*: Signal dimensions and their estimates with a spike at t or without it, plotted across the simulation time. *Right*: Signal $\mathbf{x}(t)$ and its estimate $\hat{\mathbf{x}}(t)$ with and without a spike plotted across each dimension.

4.2 Mathematical Principles of hSCNs

Here, we use the properties yielded by the use of temporal decoders in SCNs to build the framework for hSCNs (figure 4.4A).

Recall that the loss of a M-dimensional signal is written as

$$L(t) = ||\mathbf{x}(t) - \hat{\mathbf{x}}(t)||_2^2 = \sum_{m=1}^{M} (x_m(t) - \hat{x}_m(t))^2. \tag{4.1}$$

Since our input components are temporally entangled and the readout components approximate their future others on average, we may approximate the above loss function by

$$L(t) = \sum_{m=1}^{M} (x_1(t + (m-1)\Delta T) - \hat{x}_1(t + (m-1)\Delta T))^2. \tag{4.2}$$

Now, we want to transform the temporal dimensionality into the system's time. Thus, being time a continuous quantity, we start by transforming the above sum into an integral

$$L(t) = \int_0^{\Delta t} (x'(t+\tau) - \hat{x}'(t+\tau))^2 d\tau, \tag{4.3}$$

where x' and $\hat{x}'$ are the temporal component's values in a SCN and the predictions in a hSCN. This expression is then the 1D version of the predictive loss introduced in the previous chapter in equation 3.8.

Providing the insights about the temporal evolution of the signal's estimate explained in the previous section, the key to align the temporal dimensionality and the system's time is to enforce that the readout's prediction follows the same dynamics as the signal's, i.e, every time derivative of the predictions $\mathbf{x}'$ and $\hat{\mathbf{x}}'$ are equal to the corresponding time derivative of the true signal $\mathbf{x}$ (figure 4.4B). That is, using the notation of equation 3.8, we write, for a generic signal, the predictions for the signal and the readout using a Taylor expansion

$$\mathbf{x}'(t+\tau) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\tau + ... + \frac{d^n\mathbf{x}(t)}{dt}\frac{\tau^n}{n!} + ... \quad , \tag{4.4}$$

$$\hat{\mathbf{x}}'(t+\tau) = \hat{\mathbf{x}}(t) + \dot{\hat{\mathbf{x}}}(t)\tau + ... + \frac{d^n\mathbf{x}(t)}{dt}\frac{\tau^n}{n!} + ... \quad . \tag{4.5}$$

Substituting these predictions in the predictive spiking rule in equation 3.9 yields

$$\mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] \int_0^{\Delta t} \alpha_i(\tau)d\tau > \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} \alpha_i^2(\tau)d\tau. \tag{4.6}$$

That is, given that the network's estimate is being predicted as an analog signal ruled by the same differential equations as the input signal, the only difference between equation 4.4 and 4.5 is their current value at the decision time t. In other words, the assumption is that the network, on average, predicts its estimate as well as it predicts the input signal.

Therefore, the i-th neuron's voltage $V_i(t)$ and its spiking threshold T_i are

$$V_i(t) = \mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] \int_0^{\Delta t} \alpha_i(\tau)d\tau, \tag{4.7}$$

$$T_i = \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} \alpha_i^2(\tau)d\tau. \tag{4.8}$$

Then, is this case, the i-th neuron's voltage is still the projection of the coding error onto its decoding vector, but now scaled by a factor that is a proxy for the weight of the spike's effect into the future. Furthermore, a straightforward conclusion of this dynamics is that having different kernels translates into having different thresholds. Again, to merge the spiking rule with the network's dynamics we take the derivative of the voltage

$$\dot{V}_i(t) = -a_d V_i(t) + \mathbf{D}_i^\top \left[\dot{\mathbf{x}}(t) + a_d\mathbf{x}(t)\right] \int_0^{\Delta t} \alpha_i(\tau)d\tau - a_d\mathbf{D}_i^\top \mathbf{D}\mathbf{o}(t) \int_0^{\Delta t} \alpha_i(\tau)d\tau. \tag{4.9}$$

We, therefore, can write the hSCNs dynamics in vector form as

$$\dot{\mathbf{V}}(t) = -\underbrace{a_d\mathbf{V}(t)}_{\text{leak}} + \underbrace{\mathbf{A}\mathbf{D}^\top\mathbf{c}(t)}_{\text{input}} - \underbrace{a_d\mathbf{A}\mathbf{D}^\top\mathbf{D}\mathbf{o}(t)}_{\text{slow recurrent}}, \tag{4.10}$$

where $\mathbf{c}(t) = \dot{\mathbf{x}}(t) + a_d\mathbf{x}(t)$ is the input command and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with elements $A_{ii} = \int_0^{\Delta t} \alpha_i(t)dt$.

Again, inspecting equation 4.10, one recognizes the Leaky Integrate-and-Fire (LIF) model. Therefore, hSCNs are also LIF networks, but now with slow recurrent dynamics.

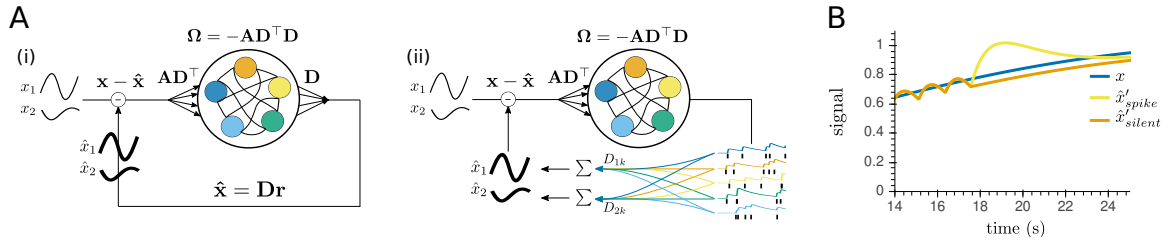

Figure 4.4: A hSCN is an autoencoder with a predictive time windows. **A**. (i) A hSCN is a network with a feedforward matrix $\mathbf{A}\mathbf{D}^\top$ and a recurrent matrix $-\mathbf{A}\mathbf{D}^\top\mathbf{D}$ that is able to readout an input signal $\mathbf{x}$, yielding an estimate $\hat{\mathbf{x}}$. (ii) Same as (i), but unfolded to show that the readout is decoded from the network's alpha-filtered spike trains, using the readout weights $\mathbf{D}$. **B**. In order to make a spiking decision, the neuron assumes that the evolution of the readout follows the signal's dynamics. Compare with figure 4.3E.

4.3.1 Slow Dynamics Shrink the Bounding Box

The integration of slow postsynaptic potentials, i.e., with longer timescales, implies that the neurons' voltages now have a slower dynamics as well. If the neuron's kernel is too slow it may happen that a single spike is not enough to push the readout beyond the spiking boundary. In order to understand this effect, recall that the slower the neuron's kernel, the more spread in time its effect and smaller its amplitude, a consequence of the filter's normalization, i.e., $\int_0^{+\infty} \alpha_i(\tau)d\tau = 1$. Therefore, the effect of the spike in the near future is small (figure 3.1). That said, in the settings of a very slow neuron, when then neuron hits its threshold, it bursts, i.e., it spikes multiple times in consecutive time steps, yielding sets of double, triple or more consecutive spikes (figure 4.5A). Neural bursting is a process that has been observed in several areas of the brain, namely in the cortex [71], [72], hippocampus [73], thalamus [74] and cerebellum [75].

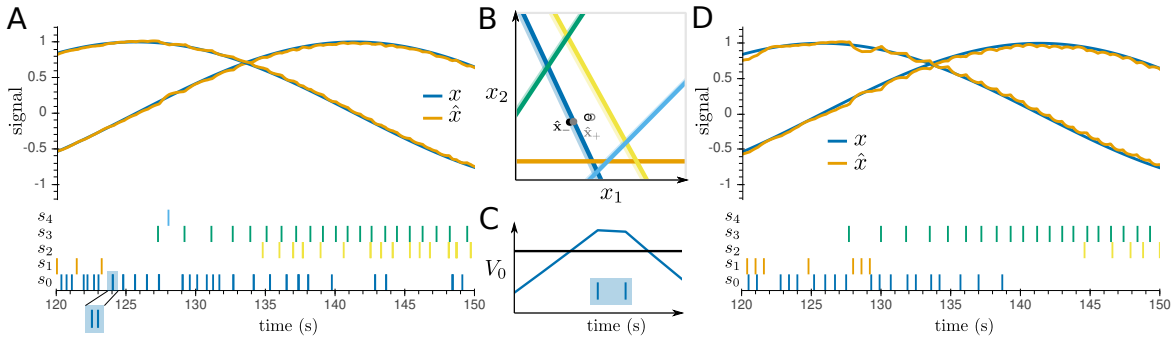

Figure 4.5: Neuron bursting in hSCN. **A**. Tracking of a 2D signal by a 4-neuron hSCN. *Top*: Signal $x(t)$ its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron. Note the double spike. **B**. Error bounding-box for two consecutive time steps. Lighter lines denote the bounding box before the first spike. Stronger lines represent the bounding box after the first spike. Grey dot is the readout before the first spike, $\hat{\mathbf{x}}_-$. Black dot is the readout after the first spike $\hat{\mathbf{x}}_+$. The rings signal the input signal $\mathbf{x}$ shift in consecutive time steps. Note that despite the shift on the bounding box and the readout after the first spike the last is still in the spiking region of the blue neuron. **C**. In the absence of an instantaneous reset, the neuron's voltage, $V_0(t)$, is above the threshold in two consecutive time steps. **D**. Tracking of a 2D signal by a 4-neurons hSCN with a refractory period. *Top*: Signal $x(t)$ its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron.

In the geometric perspective, we see that in consecutive time steps, the readout is in the spiking area (figure 4.5B), which means that the voltage is above the threshold (figure 4.5C).

This bursting may be halted by employing an absolute refractory period to the hSCN (figure 4.5D). Other solutions used in LIF models usually involve forcing spiking-adaptation frequency dynamics or adding spike costs, as we will do in the forthcoming section.

In order to show the shrinking effect of the bounding box by the addition of slow neurons, we constructed a hSCN with pairs of slow and fast neurons having the exact same decoding vector (figure 4.6). Furthermore, in order to study the effect of the kernel parameters independently of any other confounding factors, we used $\Delta t \to +\infty$ for every neuron. Given that our kernels are normalized, then $\int_0^{+\infty} \alpha_i(\tau)d\tau = 1$ for any neuron. In order words, every neuron has access to 100% of its time evolution.

The boundaries of a 2D bounding box of a hSCN are built considering the equality, for each neuron i at each time step t, $V_i(t) = T_i$:

$$\mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] = \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{+\infty} \alpha_i^2(\tau) d\tau \tag{4.11}$$

$$\hat{x}_2(t) = \left[\mathbf{D}_i^\top \mathbf{x}(t) - D_{1,i}\hat{x}_1(t) - \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{+\infty} \alpha_i^2(\tau) d\tau\right] / D_{2,i}.$$

Then, according to equation above, having the same decoding vectors, the two neurons within a pair form two parallel boundaries. The distance between these two boundaries depends on how different the kernels are, i.e., depends on the third term of the right hand-side of equation 4.12. This integral increases as the a_r parameter increases.

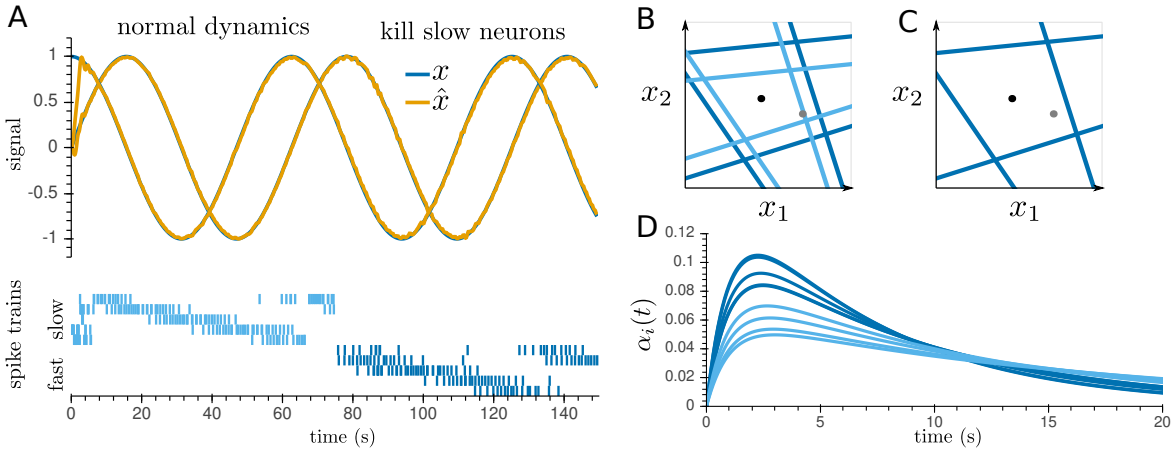

Figure 4.6: Slower neurons shrink the error bounding box. **A**. Tracking of a 2D signal in time by a 10-neuron hSCN. *Top*: Signal $x_m(t)$ dimensions and their estimates $\hat{x}_m(t)$. *Bottom*: Spike trains $s(t)$ for each neuron. Light blue spikes come from slow neurons, dark blue spikes come from fast neurons. **B**. Error bounding-box in the first half of the simulation. **C**. Error bounding-box in the second half of the simulation, after killing the slow neurons. **D**. Neurons' alpha kernels.

In other words, the threshold increases as the kernel gets faster. Calaim, Dehmelt, Gonçalves, *et al.* [53] studied the effect of perturbing the thresholds in SCNs, noting that increasing the threshold of a neuron amounts to pushing its boundary outwards of the bounding box, whereas decreasing it amounts to pushing the corresponding boundary inwards. Then, if the decoding vectors are normalized, faster kernels make the bounding box wider while slower kernels shrink it (figure 4.6B). That said, in a hSCN with pairs of slow and fast kernels with full access to their futures and no refractory periods, the slow kernels take over the tracking unless they are killed (figure 4.6A and C). Interestingly, the shrank bounding box created by the slow neurons now counter-balance the ping-pong effect - an intrinsic feature of shrank bounding boxes - with slow spiking dynamics. In other words, slow neurons structurally favour ping-pong, but dynamically avoid it.

4.4 Adding Spike Costs to hSCNs

Analogously to what was previously done in the SCNs, one may add costs to the act of spiking in hSCNs. The use of a slower decoder provides two dynamical variables for each neuron, both reflecting its firing activity, in a faster or slower scale. Since the derivation using one variable or the other are similar, here we show the dynamics of a hSCN subject to spike costs onto the most downstream variable, $o(t)$.

Then, the costly version of the predictive loss function is written as

$$L(t) = \int_0^{\Delta t} ||\mathbf{x}'(t+\tau) - \hat{\mathbf{x}}'(t+\tau)||_2^2 d\tau + \mu \int_0^{\Delta t} ||\mathbf{o}'(t+\tau)||_2^2 d\tau + \nu \int_0^{\Delta t} ||\mathbf{o}'(t+\tau)||_1 d\tau. \tag{4.12}$$

We already know how to write the signal's and the readout's predictions, but now we have to decide how to predict the evolution of the neuron's dynamical variable. This decision may be heuristically justified by considering the objective of the cost terms: to make the code sparse and the firing rates low. Then, the cost should penalize neurons which have high firing rates. This is to say that the future of the dynamical variables used in the loss function matters little in what comes to making a spiking decision. What really matters is their past. That is, if a neuron has fired a lot in the recent past, then it should have a higher cost to spike in the future. Since these dynamical variables are as high as the neural activity of the neuron they are describing, then their current value holds information about the recent firing activity. Then we make the assumption that $\mathbf{o}'(t+\tau) = \mathbf{o}(t)$.

Under that assumption, the predictive spiking rule becomes

$$\begin{aligned}&\mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] \int_0^{\Delta t} \alpha_i(\tau) d\tau - \mu o_i(t) \int_0^{\Delta t} e_i(\tau) d\tau > \\ &\frac{1}{2}\left[\mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} \alpha_i^2(\tau) d\tau + \mu \int_0^{\Delta t} e_i^2(\tau) d\tau + \nu \int_0^{\Delta t} e_i(\tau) d\tau\right],\end{aligned} \tag{4.13}$$

where $e_i(t) = a_r^i e^{-a_r^i t}$ is the filtering exponential kernel for the i-th neuron.

And the voltage's dynamics is given by

$$\begin{aligned}\dot{V}_i(t) = &- a_d V_i(t) + \mathbf{D}_i^\top \left[\dot{\mathbf{x}}(t) + a_d \mathbf{x}(t)\right] \int_0^{\Delta t} \alpha_i(\tau) d\tau - a_d \mathbf{D}_i^\top \mathbf{D}\mathbf{o}(t) \int_0^{\Delta t} \alpha_i(\tau) d\tau \\ &- \mu a_r^i \int_0^{\Delta t} e_i(\tau) d\tau \left(o_i(t) - s_i(t)\right).\end{aligned} \tag{4.14}$$

Again, considering $\Delta t \to +\infty$ and that the neurons came in pairs of fast and slow neurons with the same decoding vectors, we show the effects of each added norm term independently in the hSCN behaviour (figure 4.7).

The addition of only the L2 norm term rewrites equation 4.13 in the following spiking rule, $V_i(t) > T_i$

$$\mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] - \mu o_i(t) > \frac{1}{2}\left[\mathbf{D}_i^\top \mathbf{D}_i \int_0^{+\infty} \alpha_i^2(\tau) d\tau + \mu \int_0^{+\infty} e_i^2(\tau) d\tau\right]. \tag{4.15}$$

The main consequence of adding a L2 cost is to make the coding sparser throughout the neural population. Indeed, that is what happens in figure 4.7B, where the fast neurons become active. Given that now the i-th neuron's voltage depends on the current value of its variable o_i, which reflects the recent past spiking, the position of the boundaries in the bounding box change from one instant to another (bottom panel in figure 4.7B). Note that in the settings of normal dynamics (bottom panel in figure 4.7A), given that this is a periodic signal of period T, the bounding box is the same in two time points separated by T.

On the other hand, the addition of the L1 norm term rewrites equation 4.13 in the following spiking rule, $V_i(t) > T_i$

$$\mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] > \frac{1}{2}\left[\mathbf{D}_i^\top \mathbf{D}_i \int_0^{+\infty} \alpha_i^2(\tau) d\tau + \nu\right]. \tag{4.16}$$

Then, the main purpose of adding a L1 cost is to lower the individual firing rates, by increasing the threshold. We see this effect in figure 4.7C. Here, as in the normal dynamics, only the slow neurons are recruited, but their individual bursting is limited. Consequently, the tracking is harmed. From a geometric perspective, since there are no dependence of the neuron's voltages on their previous activity, the bounding box is the same in two time points separated by T (bottom panel in figure 4.7C). The effect of the cost increasing the threshold is seen by the fact that the bounding box is widened.

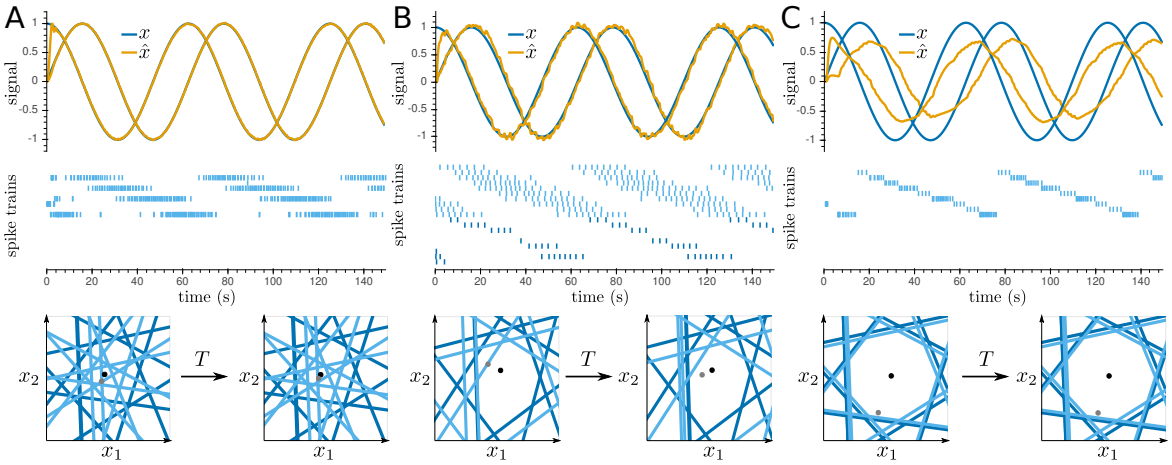

Figure 4.7: Costly hSCNs. **A**. Normal dynamics. **B**. With L2 cost. **B**. With L1 cost. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$.*Middle*: Spike trains $s(t)$ for each neuron. Light blue spikes come from slow neurons, dark blue spikes come from fast neurons. *Bottom*: Error bounding-box for two time points in the simulation separated by a period $T = 2\pi/\omega$. Simulations considering $\mu = \nu = 1$.

4.5 Robustness

As stated in chapter 2, robustness is a strength of SCNs. Given that hSCNs maintain the mechanisms responsible for robustness, namely, negative feedback, heterogeneity both in space and temporal domain, modularity and decoupling, then it is expected that these networks remain robust to noise. Indeed, hSCNs were shown to be robust against neural death, voltage noise and synaptic perturbations (figure 4.8).

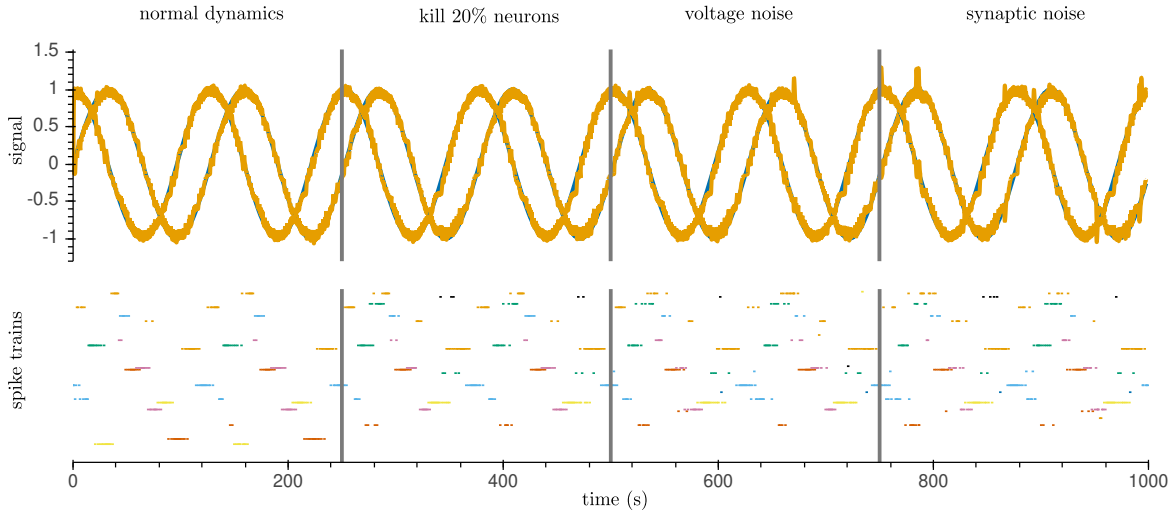

Figure 4.8: hSCNs are robust to several cumulative perturbations.

4.5.1 Temporal Heterogeneity in hSCNs

But robustness against noise is not the only robust feature of hSCNs. These networks are heterogeneous in the temporal domain and that is an advantage for tracking frequency modulated signals (figure 4.9).

Firstly, given that there is an a_r parameter per neuron, which allows us to build hSCNs with faster or slower dynamics, we have a structurally heterogeneous network in what comes to synaptic timescales,

as explored in the previous sections. But the a_r parameter is not the only source of heterogeneity. In fact, the use of finite integration windows enriches the behaviour of a hSCN.

In the previous sections we considered $\Delta t \to +\infty$, which implies, due to the normalization, that the temporal effect of the kernel is felt only through the i-th neuron's threshold T_i. If we consider a finite integration windows, then, according to the voltage equation in 4.8 even if the neuron's share the same decoding vectors, the way they percept the local error depends on the temporal scaling factor of the i-th neuron $\int_0^{\Delta t} \alpha_i(\tau)d\tau$.

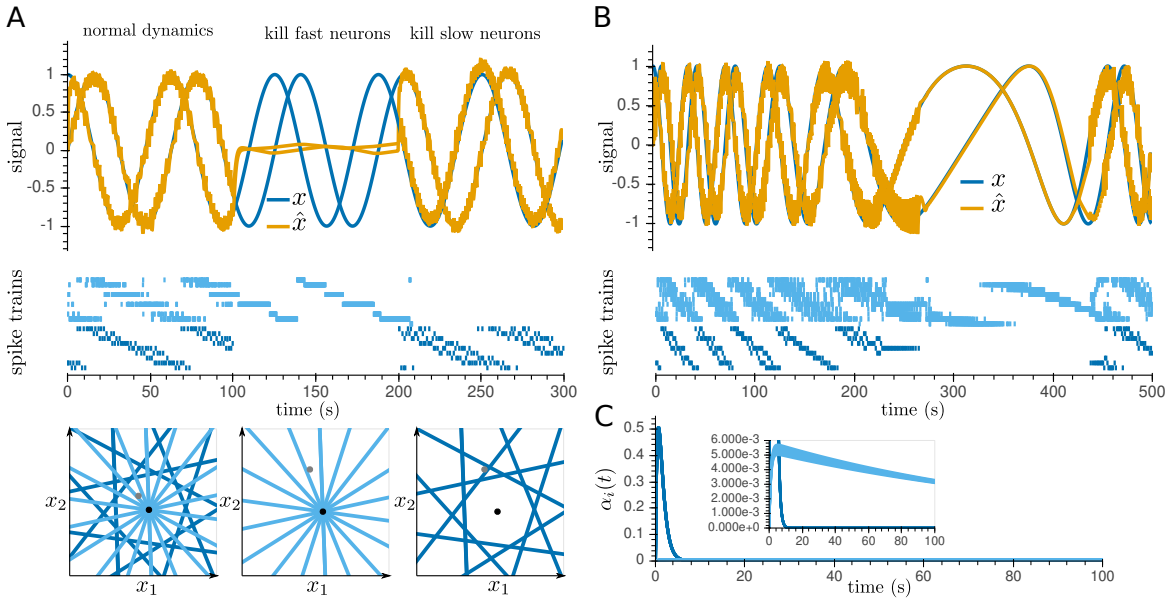

Figure 4.9: hSCNs are robust to signal changes in time. **A**. 20-neuron hSCN tracking a constant frequency oscillatory signal. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$.*Middle*: Spike trains $s(t)$ for each neuron. Light blue spikes come from slow neurons, dark blue spikes come from fast neurons. *Bottom*: Error bounding-box for each simulated condition. **B**. 20-neurons hSCN tracking a frequency modulated signal. *Top*: Signal $x(t)$ and its estimate $\hat{x}(t)$.*Middle*: Spike trains $s(t)$ for each neuron. **C**. Neurons' kernels used in B considering $\Delta t = 100$ s. Note the differences between the fast and the slow kernels.

Consider the example again of pairs of slow and fast neurons with the same decoding vectors. But now, the slow neurons are really slow and the fast ones are really fast. That is, for the same finite integration windows Δt, the fast kernel has had its full effect, i.e., $\int_0^{\Delta t} \alpha_f(\tau)d\tau \approx 1$ whereas the slow kernel is not even close to attaining its full effect, i.e., $\int_0^{\Delta t} \alpha_s(\tau)d\tau < 1$ (figure 4.9C). In other words, the fast neurons have access to 100% of its future evolution while the slow neurons have access to less than 100% of its future evolution. The result is a looser tracking of the signal while recruiting both fast and slow neurons (figure 4.9 A). In this case, the slow neurons are so slow that their bursting is not enough to accurately track the signal, i.e., the pushing of the readout into the bounding box that follows a spike is too slow. Then, the network is forced to recruit the fast neurons. In fact, when we kill the fast neurons, the tracking fails. On the other hand, the killing of the slow neurons barely affects the tracking. This means that this set of fast neurons have synaptic timescales that fit the tracking signal better. However, the addition of the slow neurons helps to tune the tracking. In the geometric perspective, we still have two bounding boxes, the fast and the slow one. Given that the slow bounding box is too shrank and the signal moves faster than this bounding box, the readout is almost always outside of it. The fast bounding box, however, is able to follow the signal's and the readout's dynamics, i.e, this bounding box moves faster. Then, the fast bounding box tracks the generality of the signal, leaving small corrections to the slowest neurons. Killing either set of slow or fast neurons amounts to deleting one of the bounding boxes.

This effect is especially useful in the scenario of a frequency modulated signal. In a SCN the frequency of the signal constrains the decaying constant λ_d. If the signal changes its frequency, given that SCNs are homogeneous networks in the temporal domain, the neurons sharing the same λ_d may not be able to keep tracking the signal due to saturation effects (depending on the simulation time step). However, in the settings of a hSCN redundant in space and heterogeneous in time, the tracking of a frequency modulated signal is successful (figure 4.9B). Furthermore, while the signal is too fast for the slow bounding box to track, the fast neurons are recruited, but as the signal gets slower and the slow bounding box is able to follow it, the slow neurons take over.

4.6 Benchmarking and Other Metrics

As stated in chapters 1 and 2, one of the strong features of SCNs is their spiking variability and low firing rates. In the last sections, the reader may have acquired the qualitative view that hSCNs share these features. In this section, we provide a quantitative view on this. In order to do so, we focus on the following measures: Coefficient of Variation (CV), as introduced in equation 2.24; the Firing Rate (FR), measured as the number of spikes divided by the total duration of the trial; and the coding error, measured, for each signal component, as $e_m(t) = (x_m(t) - \hat{x}_m(t))^2$. The proxy for the network performance is the global coding error, calculated as the average of all component's errors. Given the temporal of heterogeneity of hSCNs, we compute these metrics for four sets of the neuron-specific kernel parameter, i.e., the a_r^i parameter in equation 3.6. The lower the parameter, the slower the kernel. Furthermore, we compute the evolution of these quantities as a function of the integration window Δt (figure 4.10). These networks have no noise added.

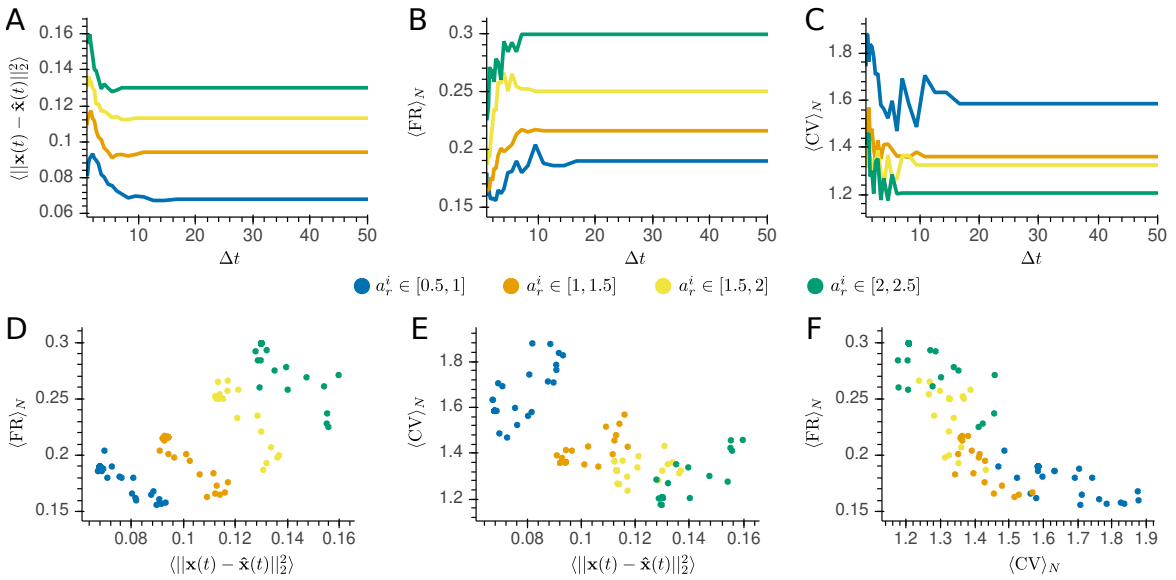

Figure 4.10: hSCNs metrics computed using a 2D sinusoidal signal with angular frequency $w = 0.1$ and period $T \approx 62.83$. Each color represents the set of the a_r^i parameters used in the simulation. **A**. The average of the coding error over the 2 components as a function of the integration window Δt. **B**. The average of the N individual firing rates, FR, as a function of the integration window Δt. **C**. The average of the N individual CV as a function of the integration window Δt. **D**. The relationship between the average of the individual FR and the average of the coding error. **E**. The relationship between the average of the individual CV and the average of the coding error. **F**. The relationship between the average of the individual FR and the average of the individual CV.

In a network with functionally fit kernels, i.e., kernels able to track the signal accurately, (figure 4.10A), the error tends to decrease for a network with slower neurons, which is explained by the shrank

bounding box formed by these neurons. Obviously, if the neurons get too slow, eventually the bounding box shrinks too much and the tracking is impaired, as explained previously. Moreover, the average FR of the network increases as the neurons get faster (figure 4.10B and D). Furthermore, the spiking variability is confirmed by the CV > 1, there being a tendency for this average value to increase as the kernels get slower (figure 4.10C, E and F). Finally, as the integration window increases, i.e., as $\Delta t \to +\infty$, these quantities converge.

4.7 Conclusion

In this chapter, we set the fundamental properties of a hSCN: no coding for the present; heterogeneity in the temporal domain; prediction of the readout according to the dynamics of the input signal. Then, by merging these three properties with the assumptions used to derive the classical SCNs it was possible to derive heterogeneous spike coding networks that do not rely on the instantaneous communication of spikes while keeping their function and biological realistic features, such as robustness and Poisson-like variability.

CHAPTER 5

Generalizing Heterogeneous Spike Coding Networks

At the theoretical level, we are somewhere between an Aristotelian and a Newtonian age, moving from description to quantification of intuitive views of how the brain works.

— Anna Kia Nobre

In the previous chapter we have shown how Spike Coding Networks (SCNs) are able to readout signals using slow decoders. To these networks we called Heterogeneous Spike Coding Network (hSCN). Here, we show that it is possible to generalize these networks using other kernels, being each kernel a realistic model for a Postsynaptic Current (PSC).

5.1 Postsynaptic Potentials as Convolutions

In chapter 1 we presented 3 alternative models for postsynaptic conductances. One such model, the instantaneous conductance modeled by Dirac impulses, is implausible. However, the heterogeneity of voltage-gated channels that mediate the transmission of spikes, require employing higher mathematical abstractions, which can result in slower or faster decoders. Here we show how one can generalize hSCNs to use other shapes for the postsynaptic potentials. For that, we generalize the use of multiple exponential filtering processes. Using this technique, we find in figure 5.1 that its main effect is a slower rise of the decoders.

Considering that $G^M(t)$ is the convolution of M exponential filters $g_j(t) = a_j e^{-a_j t}$, such that $G^M(t) = (g_1 * g_2 * ... * g_M)(t)$, then

$$G^M(t) = \sum_{i=1}^{M} \frac{a_i ... a_M}{\prod_{j=1, j\neq i}^{M} (a_j - a_i)} e^{-a_i t}. \tag{5.1}$$

The alpha kernel used in the previous chapter is thus equivalent to $G^2(t) = (g_1 * g_2)(t)$.

In this model, whenever a neuron adds a spike this is equivalent to adding a G^M kernel to the filtered spike trains. Then, we have the following dynamical equation for the network's readout

$$\dot{\hat{\mathbf{x}}}(t) = -a_M \hat{\mathbf{x}}(t) + a_M \mathbf{D}\mathbf{o}(t), \tag{5.2}$$

where $\mathbf{o}(t) \in \mathbb{R}^{N\times 1}$, being each element the i-th neuron's filtered spike $o_i(t) = (G^{M-1} * s_i)(t)$.

Similarly to the previous derivations, the solution to equation 5.2 yields the network's readout as the linear summation of the neurons' filtered spike trains

$$\hat{\mathbf{x}}(t) = \sum_{i=1}^{N} \mathbf{D}_i r_i, \tag{5.3}$$

where $r_i(t) = (G^M * s_i)(t)$.

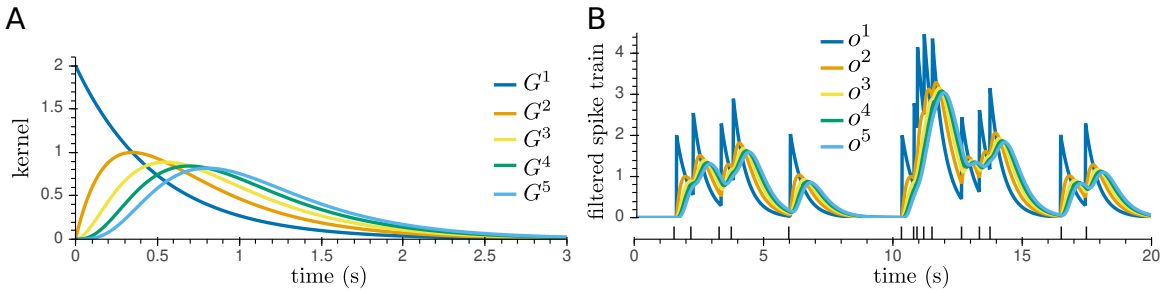

Figure 5.1: Convolutions of exponentials as PSCs. **A**. Shapes of G^1 to G^5 kernels. **B**. *Top:* The filtered spike trains according to the respective kernel, i.e. $o^M = (G^M * s)(t)$. *Bottom:* Spike train $s(t)$.

Then, whenever the i-th neuron spikes, we have the following change in the readout's future

$$\underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}}\text{ spike}} \rightarrow \underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}}\text{ silent}} + \underbrace{\mathbf{D}_i G_i^M(\tau - t)H(\tau - t)}_{\text{update}}. \tag{5.4}$$

Generalizing the predictive loss function and the predictive spiking rule of equations 3.8 and 3.9, respectively, and following all the mathematical steps used to derive the mathematical principles of hSCNs in chapter 4, we get the following voltage and threshold for the i-th neuron

$$V_i(t) = \mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] \int_0^{\Delta t} G_i^M(\tau) d\tau, \tag{5.5}$$

$$T_i = \frac{1}{2}\mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} (G_i^M)^2(\tau) d\tau. \tag{5.6}$$

Consequently, the dynamics follows

$$\dot{V}_i(t) = -\underbrace{a_M V_i(t)}_{\text{leak}} + \underbrace{\mathbf{D}_i^\top \left[\dot{\mathbf{x}}(t) + a_M \mathbf{x}(t)\right] \int_0^{\Delta t} G_i^M(\tau) d\tau}_{\text{input}} - \underbrace{a_M \mathbf{D}_i^\top \mathbf{D}\mathbf{o}(t) \int_0^{\Delta t} G_i^M(\tau) d\tau}_{\text{slow recurrent}}. \tag{5.7}$$

Using slower kernels implies delaying even more the effect of the spike, which has the effect of making the consecutive spikes even more frequent (figure 5.2).

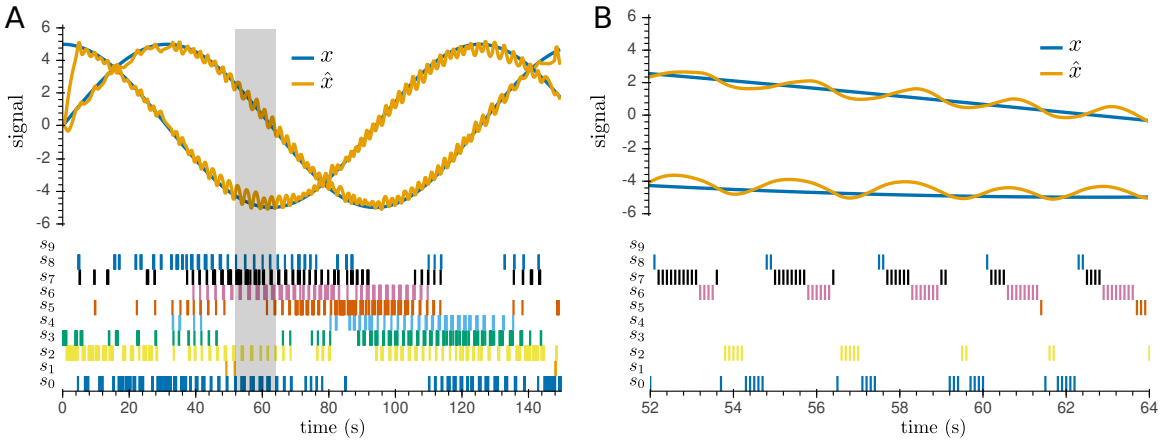

Figure 5.2: Neuron bursting in a slower hSCN. **A**. Tracking of a 2D signal by a 10-neuron hSCN with G^3 decoders. *Top*: Signal $x(t)$ its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron. **B**. Shaded area of A zoomed-in in order to see the multiple spikes. *Top*: Signal $x(t)$ its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron.

5.2 Postsynaptic Potentials as the Sum of Convolutions

While the dynamics in equation 5.7 provides the generalization for any decoder, it still constrains the reccurrency to only one type of postsynaptic currents, shaped by the G^{M-1} kernel. as given by the slow recurrent term in that equation. In this section we show that is possible to build hSCNs with several types of postsynaptic currents.

For that, we consider that the network's readout is a sum of dynamically different readouts. Biologically, these sub-readouts would correspond to the synaptic integration of the postsynaptic current along dynamically different tree branches, i.e., dendritic branches with different voltage-dependent channels. Mathematically, we write

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}^{\mathbf{2}}(t) + \hat{\mathbf{x}}^{\mathbf{3}}(t) + ... + \hat{\mathbf{x}}^{\mathbf{M}}(t) = \sum_i \mathbf{D}_i (G^2 * s_i)(t) + \sum_i \mathbf{D}_i (G^3 * s_i)(t) + ... + \sum_i \mathbf{D}_i (G^M * s_i)(t). \tag{5.8}$$

And the readout's dynamics follows the derivative of equation 5.8

$$\dot{\hat{\mathbf{x}}}(t) = \dot{\hat{\mathbf{x}}}^{\mathbf{2}}(t) + \dot{\hat{\mathbf{x}}}^{\mathbf{3}}(t) + ... + \dot{\hat{\mathbf{x}}}^{\mathbf{M}}(t). \tag{5.9}$$

Then, whenever there is a spike by the i-th neuron, the network's readout is updated according to the sum of the i-th neuron's kernel

$$\underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}} \text{ spike}} \rightarrow \underbrace{\hat{\mathbf{x}}(\tau)}_{\hat{\mathbf{x}} \text{ silent}} + \underbrace{\mathbf{D}_i \left(G^2(\tau - t) + G^3(\tau - t) + ... + G^M(\tau - t) \right) H(\tau - t)}_{\text{update}}. \tag{5.10}$$

Again, generalizing both the predictive loss function and the predictive spiking rule we get that the voltage and the threshold of the i-th neuron are given by

$$V_i(t) = \mathbf{D}_i^\top \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t) \right] \int_0^{\Delta t} G^2(\tau) + G^3(\tau) + ... + G^M(\tau) d\tau, \tag{5.11}$$

$$T_i = \frac{1}{2} \mathbf{D}_i^\top \mathbf{D}_i \int_0^{\Delta t} (G^2(\tau) + G^3(\tau) + ... + G^M(\tau))^2 d\tau. \tag{5.12}$$

Taking the derivative of the voltage, we get

$$\dot{V}_i(t) = \underbrace{-a_M V_i(t)}_{\text{leak}} + \underbrace{\beta \mathbf{D}_i^\top [\dot{\mathbf{x}}(t) + a_M \mathbf{x}(t)]}_{\text{input}} - \underbrace{a_2 \beta \mathbf{D}_i^\top \mathbf{D}(G^1 * \mathbf{s})(t)}_{\text{slow recurrent}} \tag{5.13}$$
$$+ \underbrace{(a_2 - a_3 - a_M)\beta \mathbf{D}_i^\top \mathbf{D}(G^2 * \mathbf{s})(t)}_{\text{even slower recurrent}} + ... + \underbrace{(a_{m-1} - a_m - a_M)\beta \mathbf{D}_i^\top \mathbf{D}(G^{m-1} * \mathbf{s})(t)}_{\text{getting slower}} + ... +$$
$$+ \underbrace{(a_{M-1} - a_M - a_M)\beta \mathbf{D}_i^\top \mathbf{D}(G^{M-1} * \mathbf{s})(t)}_{\text{slowest recurrent}},$$

where $\beta = \int_0^{\Delta t} G^2(\tau) + G^3(\tau) + ... + G^M(\tau) d\tau$.

A network built like this is an autoencoder with $M-1$ loops, where each loop represents a recurrent connection (figure 5.3).

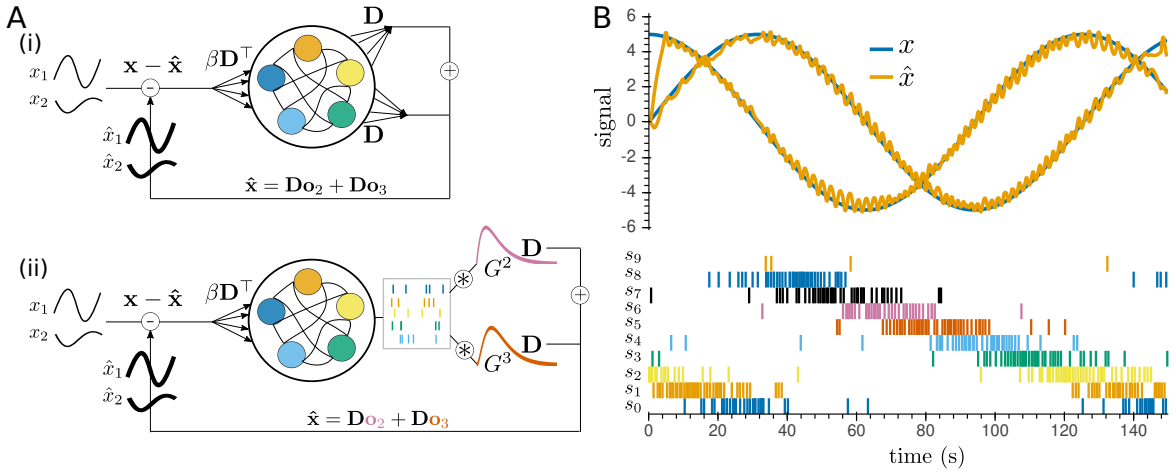

Figure 5.3: A hSCN with two loops. **A**. (i) A hSCN with several loops is a network with a feedforward vector $\beta \mathbf{D}^\top$ and a decoder vector $\mathbf{D}$ that is able to readout an input signal $\mathbf{x}$, yielding an estimate $\hat{\mathbf{x}}$. (ii) Same as (i), but unfolded to show that the readout is a sum of the spike trains filtered according to different kernels. **B**. Tracking of a 2D signal by a 10-neuron hSCN with spike trains filtered by G^3 and G^2. *Top*: Signal $x(t)$ its estimate $\hat{x}(t)$. *Bottom*: Spike trains $s(t)$ for each neuron.

5.3 Conclusion

A single neuron may have several voltage-gated channels that mediate the communication of the action potential. Our generalization shows that it is possible to incorporate several recurrent connections in hSCNs, each one modelling different dynamics, while keeping an accurate tracking of the signal.

CHAPTER 6

General Discussion

There are so many unanswered questions for generations of neuroscientists to answer—a wonderful but also sobering thought.

— Carla J. Shatz

Most of Leaky Integrate-and-Fire (LIF) networks model the Postsynaptic Currents (PSCs) homogeneously, i.e, they do not consider the different dynamics of the brain's PSCs given by the heterogeneity of voltage-dependent channels mediating neural synapses. However, the heterogeneity of neural biophysics has been shown to play an important role in the robustness of neural coding [76], [77] and in the encoding [78] and decoding [79] properties of neurons.

In this work, we used the LIF homogeneous networks proposed by Boerlin, Machens, and Denève [52], the Spike Coding Networks (SCNs), to derive a family of Heterogeneous Spike Coding Networks (hSCNs) that do not rely on the instantaneous communication of spikes and thus mimic the realistic delays encountered in PSCs. In order to do so, we used the same energy-efficiency reasoning behind the derivation of classical SCNs while enforcing three key properties that emerged while we were studying the behaviour of a SCN tracking a multi-dimensional signal in the discrete temporal domain: no coding for the present time; heterogeneous decoders; prediction of the network's estimate according to the dynamics of the input signal.

Then, we studied the properties of hSCNs using the geometrical framework provided by Calaim, Dehmelt, Gonçalves, *et al.* [53] for the standard SCNs. We found that neurons with longer timescales have lower spiking thresholds, which leads to neural bursting of the slower neurons. This may be overcome by adding a mathematical penalty to the act of spike, which shuffles the boundaries of the error bounding box, forcing the network to recruit other neurons. Furthermore, we showed that the heterogeneity of timescales merged with different integration time windows enables the network to recruit different subpopulations of neurons in a frequency-modulated signal. Moreover, likewise SCNs, hSCNs are robust to both structural and biophysical perturbations, while keeping low firing rates and high Coefficient of Variations (CVs). In addition, we observed that networks of neurons with faster alpha filters show higher firing rates and lower CVs than networks of neurons with longer alpha filters.

Finally, we showed that hSCNs can be further generalized to emulate other dynamics of the PSCs, allowing one to build networks with heterogeneous recurrent connections.

6.1 hSCNs and Other Published Work

Since Boerlin, Machens, and Denève [52] first proposed this spike coding framework, other scientists tried to tackle the problem of instantaneous propagation of spikes. During the development of this thesis we identified two public available solutions for this complex problem, which undoubtedly provided insights to this work.

Schwemmer, Fairhall, Denève, *et al.* [80] also started by assuming that the dynamics of the readout should instead be ruled by slower dynamics. As in this work, they employ alpha-shaped postsynaptic potentials, getting a predictive spiking rule. However, the way they deal with the problem of having to predict the future is solved by numerical approximations relying on the assumption of a very short integration window, which provides a heuristical solution rather than a systematic one. Questions about how short the integration window should be and the mechanism behind choosing it were left unanswered. However, this work provides valuable insights about how to include ionic conductances in the current-based model.

Zeldenrust, Gutkin, and Denève [81] aimed at building heterogeneous networks. Then, considering general filtering kernels, they reached a predictive spiking rule similar to our lower limit boundary scenario. And, like us, they acknowledged the problem that, under that scenario, to predict the future over a long integration window may lead to the over-estimation of the signal. Therefore, in order to deal with that, they constrained the integration window length to the filter size, which is also a heuristical approach to the predictive problem. Despite this, this work provided sharp insights about the performance of heterogeneous networks, namely that these networks are more robust against correlated noise. Moreover, this work suggested several predictions. Specifically, they predict that neurons sharing similar filters should have positive signal correlations and negative noise correlations and that neurons using bimodal filters couple stronger to the network activity.

6.2 Model Predictions

The big advantage of biologically plausible mathematical descriptions of neural activity is that they allow us to make predictions about experimentally recorded activity. SCNs already provide some key predictions. Next, we enumerate them.

1. **The decoding error should decrease with an increasing number of recorded neurons and it should be faster than predicted by a Poisson model.**

   Boerlin, Machens, and Denève [52] showed that as the number of neurons increases, the coding error decreases as $1\sqrt{N}$, as in a Poisson rate model. However, after a given threshold, the coding error scales as $1/N$. This could be tested using large multi-electrode arrays.

2. **Spiking statistics may correlate with signal dimensionality.**

   Calaim, Dehmelt, Gonçalves, *et al.* [53] showed that SCNs tracking higher dimensional signals show lower median firing rates. Thus, they propose that measuring firing statistics may inform the experimenter about the network's function.

3. **Partial inhibition should not affect the network's function.**

   By studying the robustness of SCNs, Calaim, Dehmelt, Gonçalves, *et al.* [53] showed that due to the redundancy of the decoding vectors and the neuron's recurrent connectivity, when a population of neurons is silenced, similar tuned neurons are recruited instantaneously to ensure the tracking. In fact, Trouche, Perestenko, Van De Ven, *et al.* [67] were able to specifically inhibit place cells in the hippocampus and, as a response, a population of neurons previously silent became active.

4. **Total inhibition should affect the network's function.**

   The study of Calaim, Dehmelt, Gonçalves, *et al.* [53] also showed that the inhibition of all similarly tuned neurons would result in a bias in the tracking of signals aligned with the decoding vectors of the inhibited neurons. Thus, functional impairment is expected if there is inhibition of all the redundant neurons required to perform some part of a given task.

In this work, the results of our model added a few more testable predictions. Namely,

1. **To look for experimental correlations between firing statistics and PSCs timescales.**

   In chapter 4 we showed that slower neurons in hSCNs seem to exhibit higher CVs and lower firing rates. Thus, a possible experimental follow-up is to look for this correlation in experimental data.
2. **To study the impact of slower and faster neurons in function.**

   In chapter 4 we showed that different populations of neurons are recruited according to the phase of the frequency-modulated signal being tracked. Thus, to look for the functional impact of neurons having slower and faster Excitatory Postsynaptic Potentials (EPSPs) or Inhibitory Postsynaptic Potentials (IPSPs) could be a line of research to follow. For instance, [82] showed that spastic motoneurons in patients who have had strokes have more prolonged EPSPs than the contralateral intact motoneurons.

6.3 Limitations of hSCNs

Although hSCNs emulate the Poisson-like variability observed in neural data while communicating spikes in a non-instantaneous manner, they still differ from neural circuits in some ways.

1. **Violation of Dale's Law**

   Like the standard framework of SCNs, hSCNs also violate Dale's law as a single neuron may simultaneously excite and inhibit other neurons according to their decoding vectors. One of the solutions proposed by Boerlin, Machens, and Denève [52] is to create separated excitatory and inhibitory populations, where the second is tracking the first. This is something that could be applied to hSCNs with the advantage that now one may specify different timescales for each population.
2. **A hSCN is a Current-Based (CUBA) Model**

   As stated in chapter 1, Conductance-Based (COBA) models are more biologically realistic. However, either SCNs or hSCNs are CUBA models.

6.4 General Comments and Future Work

The main challenge of this work was to frame the problem in order to get a systematic way to predict the network's estimate. The required properties to do that emerged when analysing a SCN in a high temporal domain. The fact that the classical SCN could be re-interpreted to further build a more biologically plausible model is an example of how such an elegant framework can under-cover a lot of insights.

In addition to the experimental suggestions made above, there are other theoretical works that could deepen the understanding of this model. Next, we enumerate them.

1. To test the hSCNs' behaviour against correlated noise.

2. Brendel, Bourdoukan, Vertechi, *et al.* [56] derived the learning conditions for a SCN to learn both the feedforward and recurrent connections. Given that hSCNs are inspired by a SCN tracking a multi-dimensional signal in the discrete time domain, it could be interesting to apply those learning rules to the same problem and see which insights it could provide to derive a learning framework for the kernel's timescales.
3. To use hSCNs to model specific experimental neural circuits, by identifying the signals they may be estimating, in order to compare the simulated activity with the neural data.
4. To apply dimensionality reduction techniques, such as Principal Component Analysis (PCA), to the model results to gain insights about how the population encodes information.

References

[1] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," in *Nature Neuroscience*, vol. 14, Nature Publishing Group, Feb. 2011, pp. 139–142. DOI: 10.1038/nn.2731. [Online]. Available: https://www.nature.com/articles/nn.2731.

[2] *Tracking Advances in Neural Recording | Statistical Neuroscience Lab*. [Online]. Available: https://stevenson.lab.uconn.edu/scaling/# (visited on 06/04/2021).

[3] C. M. Niell and M. P. Stryker, "Modulation of Visual Responses by Behavioral State in Mouse Visual Cortex," *Neuron*, vol. 65, no. 4, pp. 472–479, Feb. 2010, ISSN: 08966273. DOI: 10.1016/j.neuron.2010.01.033.

[4] L. F. Abbott, B. DePasquale, and R. M. Memmesheimer, "Building functional networks of spiking model neurons," *Nature Neuroscience*, vol. 19, no. 3, pp. 350–355, 2016, ISSN: 15461726. DOI: 10.1038/nn.4241.

[5] S. Ramón y Cajal, *Histologie du système nerveux de l'homme et des vertébrés*. Paris :Maloine, 1909, vol. v. 1, p. 1012, https://www.biodiversitylibrary.org/bibliography/48637. [Online]. Available: https://www.biodiversitylibrary.org/item/103261.

[6] E. D. Adrian and Y. Zotterman, "The impulses produced by sensory nerve-endings: Part II. The response of a Single End-Organ," *The Journal of Physiology*, vol. 61, no. 2, pp. 151–171, Apr. 1926, ISSN: 14697793. DOI: 10.1113/jphysiol.1926.sp002281. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1514782/.

[7] R. Brette, "Philosophy of the spike: Rate-based vs. Spike-based theories of the brain," *Frontiers in Systems Neuroscience*, vol. 9, no. November, pp. 1–14, 2015, ISSN: 16625137. DOI: 10.3389/fnsys.2015.00151.

[8] W. R. Softky and C. Koch, "The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs," *Journal of Neuroscience*, vol. 13, no. 1, pp. 334–350, 1993, ISSN: 02706474. DOI: 10.1523/jneurosci.13-01-00334.1993.

[9] M. N. Shadlen and W. T. Newsome, "The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding," Tech. Rep. 10, 1998, pp. 3870–3896. DOI: 10.1523/jneurosci.18-10-03870.1998.

[10] M. M. Churchland, B. M. Yu, S. I. Ryu, G. Santhanam, and K. V. Shenoy, "Neural variability in premotor cortex provides a signature of motor preparation," *Journal of Neuroscience*, vol. 26, no. 14, pp. 3697–3712, 2006, ISSN: 02706474. DOI: 10.1523/JNEUROSCI.3762-05.2006.

[11] A. Luczak, P. Barthó, and K. D. Harris, "Spontaneous Events Outline the Realm of Possible Sensory Responses in Neocortical Populations," *Neuron*, vol. 62, no. 3, pp. 413–425, 2009, ISSN: 08966273. DOI: 10.1016/j.neuron.2009.03.014.

[12] P. Dayan and L. Abbott, *Theoretical Neuroscience | The MIT Press*. 2001, pp. 24–34. [Online]. Available: https://mitpress.mit.edu/books/theoretical-neuroscience%0Ahttp://mitpress.mit.edu/books/theoretical-neuroscience.

[13] Z. F. Mainen and T. J. Seinowski, "Reliability of spike timing in neocortical neurons," *Science*, vol. 268, no. 5216, pp. 1503–1506, Jun. 1995, ISSN: 00368075. DOI: 10.1126/science.7770778. [Online]. Available: https://science.sciencemag.org/content/268/5216/1503%20https://science.sciencemag.org/content/268/5216/1503.abstract.

[14] E. Schneidman, B. Freedman, and I. Segev, "Ion Channel Stochasticity May Be Critical in Determining the Reliability and Precision of Spike Timing," *Neural Computation*, vol. 10, no. 7, pp. 1679–1703, Oct. 1998, ISSN: 08997667. DOI: 10.1162/089976698300017089. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/9744892/.

[15] A. Renart and C. K. Machens, "Variability in neural activity and behavior," *Current Opinion in Neurobiology*, vol. 25, pp. 211–220, 2014, ISSN: 18736882. DOI: 10.1016/j.conb.2014.02.013. [Online]. Available: http://dx.doi.org/10.1016/j.conb.2014.02.013.

[16] H. S. Seung, D. D. Lee, B. Y. Reis, and D. W. Tank, "Stability of the memory of eye position in a recurrent network of conductance-based model neurons," *Neuron*, vol. 26, no. 1, pp. 259–271, Apr. 2000, ISSN: 08966273. DOI: 10.1016/S0896-6273(00)81155-1.

[17] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, Feb. 2006, ISSN: 14764687. DOI: 10.1038/nature04485. [Online]. Available: https://www.nature.com/articles/nature04485.

[18] A. L. Jacobs, G. Fridman, R. M. Douglas, N. M. Alam, P. E. Latham, G. T. Prusky, and S. Nirenberg, "Ruling out and ruling in neural codes," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 14, pp. 5936–5941, 2009, ISSN: 00278424. DOI: 10.1073/pnas.0900573106. [Online]. Available: www.pnas.org/cgi/content/full/.

[19] D. F. Goodman, V. Benichoux, and R. Brette, "Decoding neural responses to temporal cues for sound localization," *eLife*, vol. 2013, no. 2, Dec. 2013, ISSN: 2050084X. DOI: 10.7554/eLife.01312.

[20] R. VanRullen, R. Guyonneau, and S. J. Thorpe, *Spike times make sense*, 2005. DOI: 10.1016/j.tins.2004.10.010. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/15626490/.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, ISSN: 00280836. DOI: 10.1038/323533a0. [Online]. Available: https://www.nature.com/articles/323533a0.

[22] P. J. Werbos, "Backpropagation Through Time: What It Does and How to Do It," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990, ISSN: 15582256. DOI: 10.1109/5.58337.

[23] D. O. Hebb, *The organization of behavior; a neuropsychological theory*. Wiley, 1949, p. 335.

[24] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, Aug. 1952, ISSN: 14697793. DOI: 10.1113/jphysiol.1952.sp004764. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/.

[25] L. Lapicque, "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation," *J. Physiol. Pathol. Gen*, vol. 9, pp. 620–635, 1907, ISSN: 0340-1200. [Online]. Available: https://www.scienceopen.com/document?vid=9cc07b80-d4bc-469f-a1e8-822bd3be8f1f%20http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Recherches+quantitatives+sur+l'excitation+?lectrique+des+nerfs+trait?e+comme+une+polarisation#0.

[26] W. Gerstner and W. M. Kistler, *Spiking Neuron Models*. Cambridge University Press, Aug. 2002. DOI: 10.1017/cbo9780511815706. [Online]. Available: https://www.cambridge.org/core/books/spiking-neuron-models/76A3FC77EC2D24CDD91E29EBB23ADB0B.

[27] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski, "Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism," *Journal of Computational Neuroscience*, vol. 1, no. 3, pp. 195–230, 1994, ISSN: 09295313. DOI: 10.1007/BF00961734.

[28] D. Sterratt, B. Graham, A. Gillies, and D. Willshaw, *Principles of Computational Modelling in Neuroscience*. Cambridge University Press, 2011, p. 404.

[29] T. P. Vogels and L. F. Abbott, "Signal propagation and logic gating in networks of integrate-and-fire neurons," *Journal of Neuroscience*, vol. 25, no. 46, pp. 10 786–10 795, Nov. 2005, ISSN: 02706474. DOI: 10.1523/JNEUROSCI.3508-05.2005. [Online]. Available: https://www.jneurosci.org/content/25/46/10786%20https://www.jneurosci.org/content/25/46/10786.abstract.

[30] S. Cavallari, S. Panzeri, and A. Mazzoni, "Comparison of the dynamics of neural interactions between current-based and conductance-based integrate-and-fire recurrent networks," *Frontiers in Neural Circuits*, vol. 8, no. MAR, 2014, ISSN: 16625110. DOI: 10.3389/fncir.2014.00012. [Online]. Available: www.frontiersin.org.

[31] D. Purves, G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. Lamantia, R. D. Mooney, M. L. Platt, and L. E. White, Eds., *Neuroscience*, 6th. Sinauer Associates, 2017, p. 960.

[32] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997, ISSN: 08936080. DOI: 10.1016/S0893-6080(97)00011-7.

[33] S. Davidson and S. B. Furber, "Comparison of Artificial and Spiking Neural Networks on Digital Hardware," *Frontiers in Neuroscience*, vol. 15, p. 651 141, Apr. 2021, ISSN: 1662-453X. DOI: 10.3389/fnins.2021.651141. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.651141/full.

[34] F. Zenke, S. M. Bohté, C. Clopath, I. M. Comşa, J. Göltz, W. Maass, T. Masquelier, R. Naud, E. O. Neftci, M. A. Petrovici, F. Scherr, and D. F. Goodman, "Visualizing a joint future of neuroscience and neuromorphic engineering," *Neuron*, vol. 109, no. 4, pp. 571–575, 2021, ISSN: 10974199. DOI: 10.1016/j.neuron.2021.01.009.

[35] S. Denève, A. Alemi, and R. Bourdoukan, *The Brain as an Efficient and Robust Adaptive Learner*, Jun. 2017. DOI: 10.1016/j.neuron.2017.05.016. arXiv: 1705.08031. [Online]. Available: http://dx.doi.org/10.1016/j.neuron.2017.05.016.

[36] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Communications*, vol. 11, no. 1, pp. 1–15, Dec. 2020, ISSN: 20411723. DOI: 10.1038/s41467-020-17236-y. [Online]. Available: https://doi.org/10.1038/s41467-020-17236-y.

[37] B. Yin, F. Corradi, and S. M. Bohté, "Effective and Efficient Computation with Multiple-timescale Spiking Recurrent Neural Networks," in *ACM International Conference Proceeding Series*, New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 1–8, ISBN: 9781450388511. DOI: 10.1145/3407197.3407225. [Online]. Available: https://dl.acm.org/doi/10.1145/3407197.3407225.

[38] I. M. Comsa, T. Fischbacher, K. Potempa, A. Gesmundo, L. Versari, and J. Alakuijala, "Temporal Coding in Spiking Neural Networks with Alpha Synaptic Function," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May, pp. 8529–8533, 2020, ISSN: 15206149. DOI: 10.1109/ICASSP40776.2020.9053856. arXiv: arXiv:1907.13223v1.

[39] W. Nicola and C. Clopath, "Supervised learning in spiking neural networks with FORCE training," *Nature Communications*, vol. 8, no. 1, Dec. 2017, ISSN: 20411723. DOI: 10.1038/s41467-017-01827-3. arXiv: 1609.02545.

[40] S. R. Kheradpisheh and T. Masquelier, "Temporal Backpropagation for Spiking Neural Networks with One Spike per Neuron," *International Journal of Neural Systems*, vol. 30, no. 6, Jun. 2020, ISSN: 17936462. DOI: 10.1142/S0129065720500276. arXiv: 1910.09495.

[41] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, Nov. 2019, ISSN: 15580792. DOI: 10.1109/MSP.2019.2931595.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, ISSN: 00189219. DOI: 10.1109/5.726791.

[43] M. Pfeiffer and T. Pfeil, "Deep Learning With Spiking Neurons: Opportunities and Challenges," *Frontiers in Neuroscience*, vol. 12, p. 774, Oct. 2018, ISSN: 1662-4548. DOI: 10.3389/fnins.2018.00774. [Online]. Available: www.frontiersin.org.

[44] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. M. McGinnity, "A review of learning in biologically plausible spiking neural networks," *Neural Networks*, vol. 122, pp. 253–272, Feb. 2020, ISSN: 18792782. DOI: 10.1016/j.neunet.2019.09.036.

[45] C. Mead, "Neuromorphic Electronic Systems," *Proceedings of the IEEE*, vol. 78, no. 10, p. 8, 1990.

[46] C. S. Thakur, J. Molin, G. Cauwenberghs, G. Indiveri, K. Kumar, N. Qiao, J. Schemmel, R. Wang, E. Chicca, J. O. Hasler, J. sun Seoa, S. Yu, Y. Cao, A. van Schaik, and R. Etienne-Cummings, *Large-Scale Neuromorphic Spiking Array Processors: A quest to mimic the brain*, May 2018. DOI: 10.3389/fnins.2018.00891. arXiv: 1805.08932. [Online]. Available: www.frontiersin.org.

[47] M. Davies, "Benchmarks for progress in neuromorphic computing," *Nature Machine Intelligence*, vol. 1, no. 9, pp. 386–388, Sep. 2019. DOI: 10.1038/s42256-019-0097-1. [Online]. Available: https://www.nature.com/articles/s42256-019-0097-1.

[48] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," *Proceedings of the IEEE*, 2021, ISSN: 15582256. DOI: 10.1109/JPROC.2021.3067593. [Online]. Available: https://doi.org/10.1109/JPROC.2021.3067593..

[49] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018, ISSN: 02721732. DOI: 10.1109/MM.2018.112130359.

[50] O. Rhodes, L. Peres, A. G. Rowley, A. Gait, L. A. Plana, C. Brenninkmeijer, and S. B. Furber, "Real-time cortical simulation on neuromorphic hardware," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 378, no. 2164, Feb. 2020, ISSN: 1364503X. DOI: 10.1098/rsta.2019.0160. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2019.0160.

[51] A. Serb, A. Corna, R. George, A. Khiat, F. Rocchi, M. Reato, M. Maschietto, C. Mayr, G. Indiveri, S. Vassanelli, and T. Prodromakis, "Memristive synapses connect brain and silicon spiking neurons," *Scientific Reports*, vol. 10, no. 1, pp. 1–7, 2020, ISSN: 20452322. DOI: 10.1038/s41598-020-58831-9.

[52] M. Boerlin, C. K. Machens, and S. Denève, "Predictive Coding of Dynamical Variables in Balanced Spiking Networks," *PLoS Computational Biology*, vol. 9, no. 11, p. 1 003 258, Nov. 2013, ISSN: 1553734X. DOI: 10.1371/journal.pcbi.1003258. [Online]. Available: www.ploscompbiol.org.

[53] N. Calaim, F. A. Dehmelt, P. J. Gonçalves, and C. K. Machens, "Robust coding with spiking networks: a geometric perspective," *bioRxiv*, p. 2020.06.15.148338, 2020. DOI: 10.1101/2020.06.15.148338.

[54] R. Bourdoukan, D. G. Barrett, C. K. MacHens, and S. Denève, "Learning optimal spike-based representations," *Advances in Neural Information Processing Systems*, vol. 3, no. c, pp. 2285–2293, 2012, ISSN: 10495258.

[55] R. Bourdoukan and S. Deneve, "Enforcing balance allows local supervised learning in spiking recurrent networks," *Advances in Neural Information Processing Systems*, vol. 2015-January, pp. 982–990, 2015, ISSN: 10495258.

[56] W. Brendel, R. Bourdoukan, P. Vertechi, C. K. Machens, and S. Denève, "Learning to represent signals spike by spike," *PLoS Computational Biology*, vol. 16, no. 3, pp. 1–23, 2020, ISSN: 15537358. DOI: 10.1371/journal.pcbi.1007692. arXiv: 1703.03777.

[57] *Neuronal Dynamics - a neuroscience textbook by Wulfram Gerstner, Werner M. Kistler, Richard Naud and Liam Paninski*. [Online]. Available: https://neuronaldynamics.epfl.ch/index.html (visited on 05/02/2021).

[58] N. Brunel, "Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons," *Journal of Computational Neuroscience*, vol. 8, no. 3, pp. 183–208, 2000, ISSN: 09295313. DOI: 10.1023/A:1008925309027. [Online]. Available: https://link.springer.com/article/10.1023/A:1008925309027.

[59] T. Tchumatchenko, A. Malyshev, T. Geisel, M. Volgushev, and F. Wolf, "Correlations and synchrony in threshold neuron models," *Physical Review Letters*, vol. 104, no. 5, pp. 2–6, 2010, ISSN: 00319007. DOI: 10.1103/PhysRevLett.104.058102. arXiv: 0810.2901.

[60] S. Denève and C. K. Machens, "Efficient codes and balanced networks," *Nature Neuroscience*, vol. 19, no. 3, pp. 375–382, 2016, ISSN: 15461726. DOI: 10.1038/nn.4243.

[61] C. Van Vreeswijk and H. Sompolinsky, "Chaos in neuronal networks with balanced excitatory and inhibitory activity," *Science*, vol. 274, no. 5293, pp. 1724–1726, Dec. 1996, ISSN: 00368075. DOI: 10.1126/science.274.5293.1724. [Online]. Available: https://science.sciencemag.org/content/274/5293/1724%20https://science.sciencemag.org/content/274/5293/1724.abstract.

[62] M. Xue, B. V. Atallah, and M. Scanziani, "Equalizing excitation-inhibition ratios across visual cortical neurons," *Nature*, vol. 511, no. 7511, pp. 596–600, 2014, ISSN: 14764687. DOI: 10.1038/nature13321.

[63] A. Renart, J. D. Rocha, P. Bartho, L. Hollender, N. Parga, A. Reyes, and K. D. Harris, "The Asynchronous State in Cortical Circuits," *Science*, vol. 327, no. January, pp. 587–591, 2010.

[64] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner, "Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks," *Science*, vol. 334, no. 6062, pp. 1569–1573, Dec. 2011, ISSN: 10959203. DOI: 10.1126/science.1211095. [Online]. Available: http://science.sciencemag.org/.

[65] D. G. Barrett, S. Denève, and C. K. Machens, "Optimal compensation for neuron loss," *eLife*, vol. 5, no. DECEMBER2016, Dec. 2016, ISSN: 2050084X. DOI: 10.7554/eLife.12454.

[66] T. O'Leary and E. Marder, "Temperature-Robust Neural Function from Activity-Dependent Ion Channel Regulation," *Current Biology*, vol. 26, no. 21, pp. 2935–2941, Nov. 2016, ISSN: 09609822. DOI: 10.1016/j.cub.2016.08.061. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27746024/.

[67] S. Trouche, P. V. Perestenko, G. M. Van De Ven, C. T. Bratley, C. G. McNamara, N. Campo-Urriza, S. L. Black, L. G. Reijmers, and D. Dupret, "Recoding a cocaine-place memory engram to a neutral engram in the hippocampus," *Nature Neuroscience*, vol. 19, no. 4, pp. 564–567, Mar. 2016, ISSN: 15461726. DOI: 10.1038/nn.4250. [Online]. Available: https://www.nature.com/articles/nn.4250.

[68] H. Kitano, *Biological robustness*, Nov. 2004. DOI: 10.1038/nrg1471. [Online]. Available: https://www.nature.com/articles/nrg1471.

[69] J. Eccles, "From electrical to chemical transmission in the central nervous system.," *Notes and records of the Royal Society of London*, vol. 30, no. 2, pp. 219–230, 1976, ISSN: 00359149. DOI: 10.1098/rsnr.1976.0015. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsnr.1976.0015.

[70] A. Mancoo, S. W. Keemink, and C. K. Machens, "Understanding spiking networks through convex optimization," *NeurIPS*, vol. 33, 2020.

[71] B. W. Connors and M. J. Gutnick, *Intrinsic firing patterns of diverse neocortical neurons*, 1990. DOI: 10.1016/0166-2236(90)90185-D. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/1691879/.

[72] C. M. Gray and D. A. McCormick, "Chattering cells: Superficial pyramidal neurons contributing to the generation of synchronous oscillations in the visual cortex," *Science*, vol. 274, no. 5284, pp. 109–113, 1996, ISSN: 00368075. DOI: 10.1126/science.274.5284.109. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/8810245/.

[73] H. Su, G. Alroy, E. D. Kirson, and Y. Yaari, "Extracellular calcium modulates persistent sodium current-dependent burst-firing in hippocampal pyramidal neurons," *Journal of Neuroscience*, vol. 21, no. 12, pp. 4173–4182, Jun. 2001, ISSN: 02706474. DOI: 10.1523/jneurosci.21-12-04173.2001. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/11404402/.

[74] N. A. Lesica and G. B. Stanley, "Encoding of natural scene movies by tonic and burst spikes in the lateral geniculate nucleus," *Journal of Neuroscience*, vol. 24, no. 47, pp. 10 731–10 740, Nov. 2004, ISSN: 02706474. DOI: 10.1523/JNEUROSCI.3059-04.2004. [Online]. Available: http://www.msg.ucsf.edu/.

[75] M. Womack and K. Khodakhah, "Active contribution of dendrites to the tonic and trimodal patterns of activity in cerebellar Purkinje neurons," *Journal of Neuroscience*, vol. 22, no. 24, pp. 10 603–10 612, Dec. 2002, ISSN: 02706474. DOI: 10.1523/jneurosci.22-24-10603.2002. [Online]. Available: https://www.jneurosci.org/content/22/24/10603%20https://www.jneurosci.org/content/22/24/10603.abstract.

[76] K. Padmanabhan and N. N. Urban, "Intrinsic biophysical diversity decorrelates neuronal firing while increasing information content," *Nature Neuroscience*, vol. 13, no. 10, pp. 1276–1282, Oct. 2010, ISSN: 10976256. DOI: 10.1038/nn.2630. [Online]. Available: https://www.nature.com/articles/nn.2630.

[77] S. J. Tripathy, K. Padmanabhan, R. C. Gerkin, and N. N. Urban, "Intermediate intrinsic diversity enhances neural population coding," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 20, pp. 8248–8253, 2013, ISSN: 00278424. DOI: 10.1073/pnas.1221214110.

[78] S. A. Prescott, Y. De Koninck, and T. J. Sejnowski, "Biophysical basis for three distinct dynamical mechanisms of action potential initiation," *PLoS Computational Biology*, vol. 4, no. 10, 2008, ISSN: 1553734X. DOI: 10.1371/journal.pcbi.1000198.

[79] C. K. Machens, R. Romo, and C. D. Brody, "Functional, but not anatomical, separation of "what"and "when"in prefrontal cortex," *Journal of Neuroscience*, vol. 30, no. 1, pp. 350–360, 2010, ISSN: 02706474. DOI: 10.1523/JNEUROSCI.3276-09.2010.

[80] M. A. Schwemmer, A. L. Fairhall, S. Denève, and E. T. Shea-Brown, "Constructing precisely computing networks with biophysical spiking neurons," *Journal of Neuroscience*, vol. 35, no. 28, pp. 10 112–10 134, Jul. 2015, ISSN: 15292401. DOI: 10.1523/JNEUROSCI.4951-14.2015. arXiv: 1411.3191. [Online]. Available: /pmc/articles/PMC6605339/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6605339/.

[81] F. Zeldenrust, B. Gutkin, and S. Denève, "Efficient and robust coding in heterogeneous recurrent networks," *PLoS Computational Biology*, vol. 17, no. 4, pp. 1–27, 2021, ISSN: 15537358. DOI: 10.1371/journal.pcbi.1008673.

[82] J. Son, X. Hu, N. L. Suresh, and W. Z. Rymer, "Prolonged time course of population excitatory postsynaptic potentials in motoneurons of chronic stroke survivors," *Journal of Neurophysiology*, vol. 122, no. 1, pp. 176–183, Jul. 2019, ISSN: 0022-3077. DOI: 10.1152/jn.00288.2018. [Online]. Available: https://www.physiology.org/doi/10.1152/jn.00288.2018.