# Slow dynamics in structured neural network models

THÈSE N$^{O}$ 9157 (2018)

PRÉSENTÉE LE 19 DÉCEMBRE 2018

À LA FACULTÉ DES SCIENCES DE LA VIE
LABORATOIRE DE CALCUL NEUROMIMÉTIQUE (IC/SV)
PROGRAMME DOCTORAL EN NEUROSCIENCES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Samuel Pavio MUSCINELLI

acceptée sur proposition du jury:

Prof. C. Petersen, président du jury
Prof. W. Gerstner, directeur de thèse
Prof. H. Sompolinsky, rapporteur
Dr S. Ostojic, rapporteur
Prof. M. Wyart, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2018

to my brothers, Eugenio and Giosuè

# Acknowledgments

Let me start by thanking my advisor Wulfram Gerstner, for having given me the chance of pursuing my PhD in a lively and intellectually stimulating group. I should also thank him for the intellectual freedom and trust that he granted me, teaching me how to think critically and independently. I am profoundly in debt to Johanni Brea, who mentored me since the moment he arrived in the lab; our endless discussions about the most various topics have deeply influenced the way I think about the brain and about science in general. I would also like to warmly thank Tilo Schwalger, who introduced me to the beauty of stochastic processes and who always encouraged me to address fascinating theoretical questions. I would like to thank all the other people I have collaborated with, in particular Chiara Gastaldi. This thesis was considerably improved thanks to the careful reading and to the detailed feedback of the three jury experts: Haim Sompolinsky, Srdjan Ostojic and Matthieu Wyart, to which I am sincerely grateful. I am in debt to all the scientists with whom I discussed my projects during the years of my PhD and that gave great feedback: Maurizio Mattia, Nicolas Brunel, Brent Doiron, Moritz Helias, Sandro Romani, Francesca Mastrogiuseppe, Jean-Pascal Pfister and Felix Schürmann.

This thesis is easier to read thanks to the proofreading effort of Valentin Schmutz, Noé Gallice and Bernd Illing. A major improvement was also due to the feedback of Vasiliki Liakoni, to which I am grateful both for her scientific and psychological help. Her restless rigor in giving feedback has been extremely helpful (and sometimes scary) anytime I had to write a scientific text. A special thank goes to all the past and present members of the LCN, that have contributed to transform a corridor into the place I have constantly been looking forward to get to. In roughly chronological order, I would like to thank Lorric Ziegler, Felipe Gherard, Moritz Deger, Kerstin Preuschoff, Mohammadjavad Faraji, Hesam Setareh, David Kastner, Aditya Gilra and Martin Barry. Great scientific discussions are on the daily agenda at the LCN. For those I particularly thank Laureline Logiaco, Friedemann Zenke, Christian Pozzorini, Carlos Stein and Chris Stock. Science happens during coffee breaks some people say, and for an uncountable number of them I want to thank Marco Lehmann. For the great social time I particularly thank, in addition to the above, Tomas van Pottelbergh, Olivia Gozel, Dane Corneil, Ho Ling Li, and Wilem Wybo. I should thank Alex Seeholzer, who among many other things taught me how to properly use a computer and, incidentally, also how to make movies. I am grateful to Florian Colombo for having contributed to keep music in my life, together with all the old guard of the Oche: Giuseppe Peronato, Xavier Mettan, Thomas Simonet, Lila Ashanti, Emmanuel Walter and Liz Speyer. For contributing to keep creativity and art as part of my daily life, I would like to thank all members of The Catalyst, that would be too many to mention one by one, as well as Miranda Kreković and

our whole Exposure team.
I am grateful my parents, Giuseppe and Patrizia, for never doubting my choices and for the support during the whole path that lead to my Ph.D.. Big thanks to Teresa Dormi, Andrea Alberti, Jacopo Guadagni and Pietro Mercati, for the effort they put in maintaining our small group alive despite the distance. I should also thank Veronica Baldi, for the years of support. A warm thanks goes also to Davide Poderini who, when I manage to reach him, is always a mine of interesting discussion, scientific and non. I also want to thank Ria Pribadi, who was very supportive during my writing of this thesis. Finally, I would like to thank my brothers, Eugenio and Giosuè, who make me looking forward to visit my hometown, and who make me a bit younger every time I visit them.

*Lausanne, 19 November 2018* S. P. M.

# Abstract

Humans and some other animals are able to perform tasks that require coordination of movements across multiple temporal scales, ranging from hundreds of milliseconds to several seconds. The fast timescale at which neurons naturally operate, on the order of tens of milliseconds, is well-suited to support motor control of rapid movements. In contrast, to coordinate movements on the order of seconds, a neural network should produce reliable dynamics on a similarly "slow" timescale. Neurons and synapses exhibit biophysical mechanisms whose timescales range from tens of milliseconds to hours, which suggests a possible role of these mechanisms in producing slow reliable dynamics. However, how such mechanisms influence network dynamics is not yet understood. An alternative approach to achieve slow dynamics in a neural network consists in modifying its connectivity structure. Still, the limitations of this approach and in particular to what degree the weights require fine-tuning, remain unclear. Understanding how both the single neuron mechanisms and the connectivity structure might influence the network dynamics to produce slow timescales is the main goal of this thesis.

We first consider the possibility of obtaining slow dynamics in binary networks by tuning their connectivity. It is known that binary networks can produce sequential dynamics. However, if the sequences consist of random patterns, the typical length of the longest sequence that can be produced grows linearly with the number of units. Here, we show that we can overcome this limitation by carefully designing the sequence structure. More precisely, we obtain a constructive proof that allows to obtain sequences whose length scales exponentially with the number of units. To achieve this however, one needs to exponentially fine-tune the connectivity matrix.

Next, we focus on the interaction between single neuron mechanisms and recurrent dynamics. Particular attention is dedicated to adaptation, which is known to have a broad range of timescales and is therefore particularly interesting for the subject of this thesis. We study the dynamics of a random network with adaptation using mean-field techniques, and we show that the network can enter a state of resonant chaos. Interestingly, the resonance frequency of this state is independent of the connectivity strength and depends only on the properties of the single neuron model. The approach used to study networks with adaptation can also be applied when considering linear rate units with an arbitrary number of auxiliary variables. Based on a qualitative analysis of the mean-field theory for a random network whose neurons are described by a $D$-dimensional rate model, we conclude that the statistics of the chaotic dynamics are strongly influenced by the single neuron model under investigation.

Using a reservoir computing approach, we show preliminary evidence that slow adaptation can be beneficial when performing tasks that require slow timescales. The positive impact of adaptation

on the network performance is particularly strong in the presence of noise. Finally, we propose a network architecture in which the slowing-down effect due to adaptation is combined with a hierarchical structure, with the purpose of efficiently generate sequences that require multiple, hierarchically organized timescales.

**Keywords:** Binary networks; chaotic dynamics; adaptation; nonlinear dynamics; sequence generation.

# Compendio

Alcuni animali sono in grado di eseguire azioni che richiedono la coordinazione di movimenti su più scale temporali, che variano da centinaia di millisecondi fino a diversi secondi. La scala temporale alla quale i neuroni agiscono, nell'ordine di decine di millisecondi, ben si adatta al controllo dei movimenti più rapidi. Tuttavia, al fine di coordinare movimenti su una scala temporale nell'ordine dei secondi, una rete neurale deve poter produrre una dinamica caratterizzata da una scala temporale adeguatamente "lenta". I neuroni e le sinapsi possiedono meccanismi biofisici che agiscono su scale temporali che variano da decine di millisecondi fino a diverse ore. Ciò suggerisce che tali meccanismi biofisici possano giocare un ruolo importante nella produzione di dinamica lenta. Tuttavia, non è ancora ben chiaro come questi meccanismi biofisici influenzino la dinamica della rete. Modicare la struttura della connettività di una rete neurale al fine di rallentarne la dinamica costituisci un approccio alternativo per la produzione di dinamica lenta. Ciononostante, non è chiaro quali siano le limitazioni di questo approccio e in particolare quanta precisione sia richiesta nella regolazione dei pesi sinaptici. Lo scopo principale di questa tesi è lo studio di come i meccanismi biofisici dei singoli neuroni e la struttura della connettività influenzino la dinamica della rete, risultando in una dinamica lenta.

Iniziamo col considerare la produzione di dinamica lenta mediante la regolazione dei pesi sinaptici in reti di neuroni binari. È noto che le reti binarie possono produrre dinamica sequenziale. Tuttavia, se la sequenze in oggetto sono aleatorie, la massima lunghezza tipica ottenibile scala linearmente con il numero di neuroni. In questa tesi mostriamo che possiamo superare questo limite grazie ad un'attenta scelta della struttura della sequenza. Più precisamente, proponiamo una dimostrazione costruttiva che permette di produrre sequenze la cui lunghezza scala esponenzialmente con il numero di neuroni. Tuttavia, ciò richiede una precisione esponenziale nella regolazione dei pesi sinaptici.

Nella seconda parte di questa tesi ci concentriamo invece sull'effetto dei meccanismi presenti nei singoli neuroni sulla dinamica della rete ricorrente. Particolare attenzione è dedicata all'adattamento neurale, un meccanismo che agisce su più scale temporali e quindi particolarmente interessante per l'oggetto di questa tesi. Il nostro studio della dinamica di una rete con connessioni aleatorie e adattamento, sviluppato utilizzando tecniche di campo medio, dimostra come la rete possa entrare in uno stato di caos risonante. Tale frequenza di risonanza è indipendente dall'intensità dei pesi sinaptici, in quanto dipende solamente dalle proprietà del modello di singolo neurone utilizzato. L'approccio utilizzato per studiare la rete con adattamento può essere utilizzato anche per studiare una rete i cui neuroni sono modellizzati con un numero arbitrario di variabili ausiliarie. Basandoci su un'analisi qualitativa della teoria di campo medio, derivata nel

caso di neuroni lineari $D$-dimensionali, concludiamo che le proprietà statistiche della dinamica caotica sono fortemente influenzate dalle caratteristiche dei modelli di singolo neurone presi in considerazione.
L'adattamento neurale può migliorare le prestazioni della rete neurale in situazioni che richiedono scale temporali "lente", come mostriamo in risultati preliminari ottenuti con tecniche di reservoir computing. Proponiamo infine un'architettura di rete in cui l'effetto dell'adattamento è combinato con una struttura gerarchica, con lo scopo di poter generare in maniera efficiente sequenze che richiedono scale temporali multiple e con un'organizzazione gerarchica.
**Parole chiave:** Reti binarie; dinamica caotica; adattamento neurale; dinamica nonlineare; generazione di sequenze.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Our brain is able to coordinate movements across multiple temporal scales with an impressive reliability. Similarly, the cerebral cortex also exhibits complex, multi-scale dynamics, that might be suited to support such complex behaviors. This is particularly striking when considering that neurons, the building blocks of our brain, operate considerably faster than behavior. It is therefore of crucial importance to understand how slow dynamics emerge in neural networks.

We address this question from a modeling perspective, by focusing on *structured* neural network models. Theoretical studies often assume *unstructured* network models, e.g. by choosing the connectivity to be entirely random or by using neuron models that capture only the basic features of real neurons. This allows to obtain theoretical insights that do not depend on specific assumptions about the connectivity or about the neuron model under consideration. However, obtaining reliable dynamics across multiple timescales in unstructured networks has been proven to be difficult. In this thesis in contrast, we investigate the influence that structure has on dynamics. By "structure" we refer either to nonrandom connectivity or to the use of neuron models equipped with additional biophysical mechanisms, whose influence on the network dynamics is not clear yet. Our hypothesis is that this additional structure, that might have developed either because of evolution or because of learning, could facilitate the emergence of multi-scale dynamics.

In this introduction, I first review some fundamental neuroscience notions that will be used throughout this thesis, with particular attention to the relevant timescales at which different phenomena take place. Then, I discuss binary neuron models, focusing on the limitations that they have in producing slow sequential dynamics. I then move to rate neuron models and discuss the general mechanism that allows to slow down the dynamics in the vicinity of attractor states. Finally, I introduce the tools that are necessary to study the dynamics of chaotic rate networks and discuss the emergence of slow dynamics in the chaotic state.

## 1.1 Timescales of fundamental processes

Throughout this thesis, I will frequently use the word *timescale*. When referring to a physical system, its timescale can be loosely defined as the time it takes for that system to change *significantly*. This definition is rather qualitative, since the meaning of the word *significantly* depends on which features of the system we are interested in. In linear (or linearized) dynamical systems, a more rigorous definition can be given and the details underlying this statement will be reviewed in 1.1.4. However, since the systems we are interested in are often nonlinear, I will refer to timescales using the qualitative definition above, while trying to be more rigorous in the cases where this is possible.

Most of the tasks we encounter during behavior are characterized by timescales on the order of seconds. If sufficiently trained, humans are able to coordinate motor commands across multiple scales, from hundreds of milliseconds to several seconds, with an impressive precision (think about musicians or professional sportsmen). The same considerations hold true for our ability to integrate sensory information over time. Neurons in contrast, appear to operate on much faster timescales, on the order of tens of milliseconds. As we will see in this section, this is only one part of the story. Looking closer at fast neuronal dynamics, we see that they are in fact influenced by many biophysical mechanisms that exhibit different timescales, some of which comparable to behavioral ones. I will now briefly review such mechanisms, while in parallel introducing some fundamental concepts of computational neuroscience.

### 1.1.1 Single neurons and synapses

Neurons are the fundamental units of the brain and communicate with each other via short electrical pulses called action potentials, or *spikes*. As many other cells, they are enveloped in a lipid membrane, that can be thought as an electrical capacitor on which charged ions accumulate due to the difference in potential between the inside and the outside of the cell. Such potential difference is maintained by ion pumps, special proteins that sit in the cell membrane and consume energy to *pump* ions against the electrical gradient. The neuronal membrane has a small leakage, so that a small amount of ions flows in or out until an equilibrium between the electrical force and the concentration pressure is reached. For a typical neuron, this equilibrium is achieved when the interior of the cell is at a voltage $u_{\text{rest}} \sim -70$ mV. We can describe these properties of the neuronal membrane as an electrical circuit, in which a capacitor (the cell membrane) is in parallel with a resistor (the leak of the membrane) (Fig. 1.1A, Gerstner et al. (2014))

$$\dot{u} = -\frac{1}{RC}\left(u(t) - u_{\text{rest}}\right) + \frac{1}{C}I(t) \quad . \tag{1.1}$$

$I(t)$ is any additional electric current that goes through the membrane, and it can result for example from the input from other neurons or from an experimental injection. The parameter $\tau_m := RC$ sets the timescale of the passive membrane, which for a typical neuron is around 10 ms, and it is responsible for the fast dynamics mentioned above. If for example a constant current

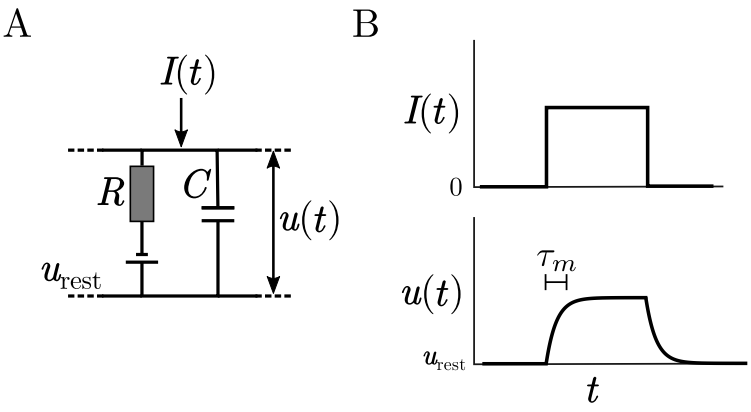



Figure 1.1 – **Passive membrane behavior. A:** Electrical circuit representation of the neuron passive membrane. **B:** Evolution of the membrane potential $u(t)$ (bottom), in the presence of a step current (top). All panels were redrawn from Gerstner et al. (2014).

is injected, the voltage reaches a new stationary value in a typical time that is on the same order as $\tau_m$ (Fig. 1.1B).

If a neuron receives enough input so that its internal voltage reaches a critical value, nonlinear effects trigger a sudden depolarization of roughly 100 mV, before the membrane potential returns to its resting value. Whenever this happens we say that a spike is emitted or, in other words, the neuron *fires*. This critical value is known as *spike threshold* and for a typical neuron is about $u_{\text{th}} \sim -50$ mV. The biophysics underlying the spike generation process is quite complex, and it was first described quantitatively by Hodgkin and Huxley (1952). For the purpose of this thesis, it is important to point out that spikes usually have a stereotypical shape, with a duration of roughly 1 ms. For this reason, spikes are often modeled as *events* and the details of their time course are neglected. After the spike, the membrane potential goes back to a value close to the resting potential. This type of spike-generation mechanism can be easily added to the passive membrane Eq. (1.1)

$$\begin{aligned} &\tau_m \dot{u} = -u(t) + u_{\text{rest}} + RI(t) \\ &\text{when} \quad u(t) > u_{\text{th}} \quad \text{then} \quad u(t) \rightarrow u_r \quad , \end{aligned} \tag{1.2}$$

where $u_r$ is the reset potential that we consider as a fixed parameter. Eq. (1.2) defines the leaky integrate and fire model (LIF, Lapicque (1907)).

Action potentials are typically generated at the initial part of the axon, along which they then propagate until they reach the pre-synaptic terminals of the *synapse*, the site where the electrical activity is transmitted to other neurons. There, a cascade of biophysical mechanisms triggers the release of neurotransmitters, which diffuse outside the cell and that activate specific receptors on the post-synaptic terminal of the synapse. There exist several neurotransmitters, such as glutamate or GABA ($\gamma$-aminobutyric acid). Importantly, each neuron can usually release only one type of neurotransmitter. Based on the neurotransmitter they release, neurons are distinguished in mainly two classes: excitatory (releasing glutamate) or inhibitory (releasing GABA). Other types of neurons present in the brain, such as dopaminergic neurons, are not considered in this thesis.

Neurons collect input from many others (on the order of 10000) thanks to the dendritic tree,

an arborization of the cell membrane that allows to have multiple contact points with axons coming from other neurons. On the post-synaptic terminal of a synapse, we find receptors for both glutamate, the most important ones being AMPA ($\alpha$-amino-3-hydroxy-5-methyl-4isoxalone propionic acid) and NMDA (N-methyl-D-aspartate), and for GABA, the most important being GABAa and GABAb. Here we will not detail the different properties of these receptors, but we will mention two important distinctions: first, the opening of glutamate receptors drives the influx of positive ions which depolarize the cell, while when GABA receptors open there is an influx of negative ions (or an outflux of positive ions), which hyperpolarize the cell. Second, different types of receptors are responsible for currents with different timescales. For example, the timescale of the AMPA-associated current is on the order of 10 ms, i.e. comparable to the passive membrane time constant, while the timescale of the NMDA-associated current is much slower, i.e. on the order of 100 ms (Hestrin et al. (1990)). Similarly, GABAa receptors are fast, while GABAb receptors are much slower (Bowery et al. (2002)).

The simplest way to introduce synaptic interactions in the LIF model is to add to Eq. (1.2) a current

$$I_{\text{syn}} = \sum_{\hat{t}_j} \epsilon(t - \hat{t}_j) \quad , \tag{1.3}$$

where the sum runs over all the pre-synaptic spike times $\hat{t}_j$. Each pre-synaptic spike causes a post-synaptic current whose typical time course is given by $\epsilon(s)$. Its exact shape depends on the synaptic model we consider, and one common choice is to set it to a single exponentially-decaying function

$$\epsilon(s) = w \cdot \tau_{\text{syn}} \cdot \exp\left(-\frac{s}{\tau_{\text{syn}}}\right) \quad , \tag{1.4}$$

where $w$ is the synaptic strength (which can be positive or negative depending on the pre-synaptic neuron type) and $\tau_{\text{syn}}$ is the timescale of the decay of the post-synaptic current.

The LIF model as described by Eq. (1.2) is a memoryless device, in the sense that once a spike is emitted, the voltage is reset to $u_r$ and any trace of previous spikes is lost. Real neurons in contrast, exhibit multiple history-dependent mechanisms such as refractoriness and adaptation. More precisely, for a few milliseconds after a neuron emits a spike, it becomes less likely for it to emit another one. This effect is usually described as resulting from three different phenomena. First, the *absolute refractoriness* is the time interval after the spike is emitted during which it is *impossible* to spike again. In threshold models like the LIF, it is important to include this period to account for the finite length of the action potential. After the absolute refractory period, the neuron might enter a state of *relative* refractoriness, in which a spike can be elicited but this is harder than when the neuron is at rest. We can account for this effect by introducing a spike-triggered hyperpolarizing current. Importantly, this effect is short-term, in that it depends only on the last spike and not on the full spike-history. Finally, *spike-frequency adaptation* (SFA) can also be modeled as spike-triggered hyperpolarizing currents. In contrast to relative refractoriness

however, the SFA current accumulate over spikes, which allows it to have a longer-lasting effect and which leads to interesting consequences from the sensory coding perspective. Due to the abundant use of adaptation in the models presented in this thesis, in the next section I provide some additional details on the mechanisms that underlie it and on the approaches towards its inclusion in a modeling framework.

### 1.1.2 Single neuron adaptation

The word *adaptation* is used to describe a broad class of phenomena that are thought to be responsible for how a neuron *adapts* to the statistics of the incoming stimuli. For example, if a step current is injected into a neuron, it emits a series of spikes whose frequency decreases with time (Benda and Herz (2003)). This phenomenon is known as spike-frequency adaptation (SFA). Moreover, even when the injected current is not strong enough to cause a spike, many neurons undergo subthreshold adaptation due to the presence of hyperpolarizing subthreshold currents, mediated by voltage sensitive ion channels (Benda and Herz (2003)). Both these phenomena represent a form of negative feedback, that might be useful to maintain the neuron in an optimal response regime.

The biophysics underlying SFA is rich and I will not go into extensive details in this thesis. However, since we are interested in the typical timescale at which different mechanisms take place, it is important to mention a few striking properties of SFA. It was shown in several studies that a single spike can have a significant effect on the dynamics of the neuron even seconds after it was emitted (La Camera et al. (2004); Lundstrom et al. (2008); Pozzorini et al. (2013)). However, concluding that SFA has a timescale of seconds would not be accurate, since its effect is also significant at shorter timescales. In fact, SFA has been shown to be scale-free, i.e. it is not possible to find a unique timescale at which SFA has a significant effect (Lundstrom et al. (2008)).

From the modeling perspective, the LIF model can be extended to include SFA in two ways: via a spike-triggered hyperpolarizing current or via a moving-threshold mechanism (Gerstner et al. (2014)). Using the first solution, Eq. (1.2) becomes

$$\begin{aligned} &\tau_m \dot{u} = -u(t) + u_{\text{rest}} - \sum_{t_f} \eta(t - t_f) + RI(t) \\ &\text{when} \quad u(t) > u_{\text{th}} \quad \text{then} \quad u(t) \rightarrow u_r \quad , \end{aligned} \tag{1.5}$$

where $\eta(s)$ is the time course of the spike-triggered current, in voltage units. Using a single exponential for such time course, albeit common, is at odds with the scale-free behavior of adaptation (Lundstrom et al. (2008)), since exponentials require to fix a timescale. Interestingly, when the profile of such currents is fitted to data, the outcome is a power law-shaped current, which is inherently scale-free. Notice that adaptation is obtained if $\eta(s)$ is positive. However, neurons that exhibit facilitation, i.e. an increase of the firing rate in response to a sustained stimulus, have also been observed (Edman et al.; Benda and Herz (2003)). The model in Eq. (1.5)

can account for facilitation if $\eta(s)$ is set to be negative.

Similarly, turning the fixed threshold of Eq. (1.2) into a dynamic one provides another solution to realize SFA:

$$\begin{aligned} &\tau_m \dot{u} = -u(t) + u_{\text{rest}} + RI(t) \\ &u_{\text{th}} = u_{\text{th}}^0 + \sum_{t_f} \eta_{\text{th}}(t - t_f) \\ &\text{if} \quad u = u_{\text{th}} \quad \text{then} \quad u \to u_r \quad . \end{aligned} \tag{1.6}$$

Again, $\eta_{\text{th}}(s)$ is the time course of the effect of one spike on the neuronal threshold, for which the same considerations about the timescale apply. These two solutions are similar but not equivalent, in that in the first solution the spike-triggered current enters the potential via the additional membrane filtering, while in the second solution the threshold dynamics does not get filtered. It is worth mentioning that to fit the behavior of real neurons it is advantageous to include both mechanisms (Pozzorini et al. (2013)).

Interestingly, a power-law current can be approximated by a sum of exponential currents having different timescales. This approach has the advantage of being easier to treat analytically, since exponentials are solutions of linear differential equations. In this thesis I will only consider models with a single exponential adaptation current, and only briefly mention the effect of multiple adaptation timescales. When doing this, we hope that the results obtained for this simple type of adaptation would be easier to generalize to multiple-exponentials adaptation than it would be to directly consider a power-law adaptation. In the rest of this thesis, I will implement exponential adaptation by using auxiliary variables that obey linear differential equation. This solution is completely equivalently to the implementation of Eq. (1.5). The equations for the LIF with adaptation read

$$\begin{aligned} &\tau_m \dot{u} = -u(t) + u_{\text{rest}} - a + RI(t) \\ &\tau_a \dot{a} = -a(t) + c \cdot S(t) \\ &\text{if} \quad u = u_{\text{th}} \quad \text{then} \quad u \to u_r \quad , \end{aligned} \tag{1.7}$$

where $\tau_a$ sets the adaptation time scale, $c$ is a parameter that controls the adaptation strength and $S(t) = \sum_{t_f} \delta(t - t_f)$ is the spike train of the neuron, i.e. a sum of Dirac $\delta$-functions with support at the spike times.

A type of adaptation distinct from SFA is subthreshold adaptation, which does not depend on the spike history but only on the subthreshold voltage. The LIF model can be extended to include subthreshold adaptation by using an auxiliary variable

$$\begin{aligned} &\tau_m \dot{u} = -u(t) + u_{\text{rest}} - a(t) + RI(t) \\ &\tau_a \dot{a}(t) = -a + c \cdot u(t) \quad . \end{aligned} \tag{1.8}$$

The parameters have the same interpretation as in Eq. (1.7), but the mechanism is different since

in this case the adaptation variable is a filtered version of the membrane potential and not of the spike train. Notice that, as for SFA, $c$ can be both positive and negative, the latter corresponding to a facilitating current. If the adaptation effect is strong enough, such a neuron model exhibits resonant behavior, and for this reason it is called resonate and fire (Izhikevich (2001); Richardson et al. (2003)). This resonant behavior is similar to the one that we observe in the rate model considered in chapter 3.

### 1.1.3 The timescales of synaptic plasticity

The electrical efficacy of synapses is not fixed, but it changes through time in response to the electrical activity of the pre- and post-synaptic neurons. This phenomenon is called *synaptic plasticity* and is considered to be the neural correlate of learning and memory (Hebb (1949); Martin et al. (2000); Hayashi-Takagi et al. (2015)). In this thesis, I neglect the consequences of synaptic plasticity for the dynamics of neural networks and I only marginally discuss the possibility of plastic synapses in chapter 6. However, to give a complete overview of the repertoire of timescales present in the brain, I briefly review here the main plasticity mechanisms.

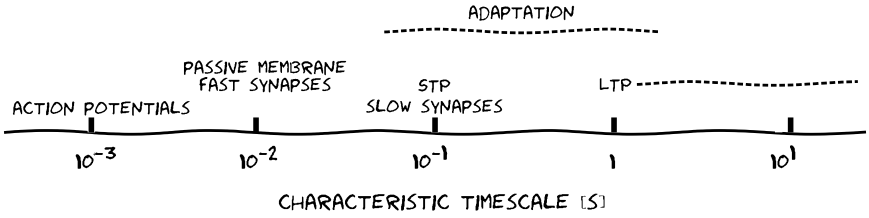
On a timescale of roughly hundreds of milliseconds, synapses undergo *short-term plasticity* (STP), whose dynamics depends mostly on the pre-synaptic firing. Short-term *facilitation* leads to a transient increase of the synaptic efficacy in response to a series of pre-synaptic spikes. On the other hand, short-term *depression* corresponds to a transient decrease of the synaptic efficacy. Whether a synapse exhibits one or the other is thought to depend on the interplay between the probability of release of synaptic vesicles and on the available pool of releasable vesicles (Zucker and Regehr (2002)). Since STP leads to transient changes in the efficacies, it is usually not directly related to learning and memory. However, it has been proposed to play an important role for the stability of attractor states in realistic neural network models (Mongillo et al. (2005); Sussillo and Maass (2007)) and to be a crucial ingredient for a metabolically efficient implementation of working memory (Mongillo et al. (2008)).

Long-term potentiation (LTP) and long-term depression (LTD) of synaptic efficacies take place on much longer timescales (from tens of seconds to hours) and they are considered to be the main mechanism by which information is stored in synapses. In contrast to STP, the induction of LTP and LTD usually requires the proximity in time of both pre- and post-synaptic firing. Two components can be distinguished: early-LTP (LTD) can be elicited with relatively few spikes and it has typical timescales ranging from tens of seconds to minutes. From the theoretical perspective, early-LTP is usually understood in terms of learning rules (i.e. equations according to which synaptic efficacies are modified) based on spike-timing dependent plasticity (STDP) (Gerstner et al. (1996)). Such learning rules assume that the modification of synaptic efficacies does not depend merely on the mean activity of the pre- and post- synaptic neurons, but on the precise timing of their spikes. Such rules do not only provide an excellent fit to experimental data (Pfister and Gerstner (2006); Clopath et al. (2010)), but they are also believed to support optimal learning of exact spike timing (Pfister et al. (2006); Gütig and Sompolinsky (2006)).

The synaptic changes induced by few episodes of pre-post pairing do not last long enough to support stable memories, since they usually decay with a timescale of roughly ten minutes (Bliss and Collingridge (1993); Frey and Morris (1997)). On the other hand, if such pairing episodes are repeated and strong enough, changes can be lasting for the whole duration of the recording, i.e. up to several hours (Frey and Morris (1997, 1998)). This slow component is called late-LTP and the phenomenon of inducing such long-lasting changes is called synaptic *consolidation*. At the theoretical level, understanding the outcome of experimental protocols seems to require the introduction of complex dynamics and of multiple auxiliary variables (Clopath et al. (2008); Ziegler et al. (2015)). To summarize, synaptic plasticity provides a large repertoire of mechanisms and consequently of timescales. Such richness is believed to have beneficial consequences for memory capacity (Fusi et al. (2005); Benna and Fusi (2016))

In a collaboration with Chiara Gastaldi (Gastaldi et al. (2018)), we studied a bistable model of synaptic dynamics that features two timescales, and we found that the ratio of the two timescales influences the sensitivity of the synapse to stimulation protocols. Being quite disconnected from the rest of the thesis, this work is not included here but a more detailed summary of it is contained in appendix A.

We have seen that the building blocks of biological brains, neurons and synapses, have timescales that essentially tile the full range necessary for behavior. We can summarize these mechanisms in a sketch:




However, it is not clear how such timescales are reflected in the network dynamics. Even in the case in which they are, a further question arises, namely how they can be exploited in order to solve tasks. These questions will be the main motivation for chapters 3–6. In the next section, I will briefly review linear system theory, focusing on their characteristic timescales. The reader familiar with this topic can skip the next section.

In the rest of the introduction, we introduce the concept of binary neurons and rate neurons, simplifications aimed to ease the study of network dynamics. Thanks to these theoretical tools, we discuss the possibility of obtaining slow dynamics as a result of the connectivity of the network, rather than emerging from single neuron properties.

### 1.1.4 Timescales of linear systems

In this section, we do a detour from neuroscience to briefly review linear system theory and how for such systems the concept of timescale can be better defined. The material presented here is textbook knowledge, and can be found in any book on linear system or ordinary differential equations theory. I particularly enjoyed the book by Hespanha (2009).

Consider a homogeneous time-invariant system

$$\begin{aligned} &\dot{x} = \mathrm{A}x \\ &x(t_0) = x_0 \in \mathbb{R}^n \quad . \end{aligned} \tag{1.9}$$

The solution to this problem is known to be

$$x(t) = e^{\mathrm{A}(t-t_0)} \cdot x_0 \quad , \tag{1.10}$$

where the matrix exponential is defined by

$$e^{\mathrm{A}} = \sum_{k=0}^{\infty} \frac{1}{k!} \mathrm{A}^k \quad . \tag{1.11}$$

If the matrix A is diagonalizable, the matrix exponential can be easily computed

$$e^{\mathrm{A}t} = \mathrm{P} \begin{pmatrix} e^{\lambda_1 t} & 0 & \dots & 0 \\ 0 & e^{\lambda_2 t} & \dots & 0 \\ 0 & 0 & \dots & e^{\lambda_n t} \end{pmatrix} \mathrm{P}^{-1} \tag{1.12}$$

where P can be constructed by having the eigenvectors of A as columns. If the matrix A is normal, i.e. if $\mathrm{A}\mathrm{A}^T = \mathrm{A}^T\mathrm{A}$, then its eigenvectors form an *orthonormal* basis. This implies that a trajectory of the system can be described as a sum of independent components, the eigenmodes, each of which has an exponential profile, with a timescale given by the real part of the associated eigenvalue. Notice that if the eigenvalue also has a nonzero imaginary part, each mode exhibits exponentially-modulated oscillations controlled by two timescales, the one of the oscillations and the one of the exponential envelope, given by the imaginary and the real part of the eigenvalues, respectively. If the matrix A is not normal, then the eigenvectors are not orthogonal. This complicates the analysis of the timescales and can lead to a transient behavior whose duration exceeds the one predicted by the eigenvalues (Trefethen and Embree (2005)).

Notice that a nonlinear system can be linearized around a fixed point, which gives a good approximation of the system behavior in the vicinity of that point. For this reason, we sometimes refer to the timescales of a nonlinear system in the vicinity of a fixed point as the ones of the corresponding linearized system.

## 1.2 Binary networks and sequential dynamics

As we have seen in section 1.1, the output of a neuron is an *all-or-none* process, i.e. the neuron is either spiking or not spiking. This motivates the description of neurons with binary variables, that take value $\sigma = +1$ when the neuron is firing or $\sigma = 0$ when the neuron is silent, an idea that dates back to McCulloch and Pitts (1943). Binary neuron models are often treated in discrete time, which implies the choice of a time step $\Delta t$. The interpretation of the two possible states of the variable depends on the time step that we consider. For example, if $\Delta t = 1$ ms, we interpret the state $\sigma = +1$ as a spike emitted by the neuron, while if $\Delta t = 500$ ms, the same state should be rather interpreted as a period of sustained spiking activity (Gerstner et al. (2014)).

A common choice for the dynamics of a network of $N$ binary neurons is given by the update rule (Amit (1989))

$$\sigma_i(t+1) = \Theta\left(\sum_{j=1}^{N} J_{ij}\sigma_j(t) - T_i\right) \quad , \quad i = 1, \dots, N \tag{1.13}$$

where $T_i$ represents the threshold of neuron $i$, $\Theta(\cdot)$ is the Heaviside function and $J_{ij}$ represents the strength of the synaptic connection from neuron $j$ to neuron $i$. Intuitively, the dynamics in Eq. (1.13) indicates that if the total input into a neuron, given by $\sum_{j=1}^{N} J_{ij}\sigma_j(t)$, is larger than a threshold $T_i$, then the neuron is active. Neurons in the network can be updated either *synchronously* or *asynchronously*. In the synchronous version of the update rule, the next state is computed for all neurons in parallel based on the full state of the network at the previous time step. In the asynchronous version instead, neurons are updated one at a time, usually in a random order.

Alternatively, binary neurons can be described by a variable that takes value $S \in \{+1, -1\}$. The two descriptions can be mapped onto each other by the transformation

$$S = 2\sigma - 1 \quad , \tag{1.14}$$

from which we can find the update rule for the $S$ variables, equivalent to Eq. (1.13)

$$S_i(t+1) = \text{sign}\left(\frac{1}{2}\sum_{j=1}^{N} J_{ij}S_j + \frac{1}{2}\sum_{j=1}^{N} J_{ij} - T_i\right) \quad , \tag{1.15}$$

where the $\text{sign}(\cdot)$ is defined with the convention that $\text{sign}(0) = 1$. If the mean input drives neurons close to their thresholds and there are approximately as many neurons active as non-active, we have that $\frac{1}{2}\sum_{j=1}^{N} J_{ij} \simeq T_i$. This situation has the advantage that neurons are most sensitive to changes in their input (Amit (1989)). In what follows, I will consider the $S \in \{+1, -1\}$ convention and I will neglect the biases, i.e. $\frac{1}{2}\sum_{j=1}^{N} J_{ij} - T_i \simeq 0$.

The Hopfield model (Hopfield (1982)) is one of the most influential models in computational neuroscience and it was proposed as a possible mechanism for associative memory and for

content-based memory addressing. It was originally defined for a network following asynchronous dynamics with the update rule given by Eq. (1.15), and it allows to "store" in the network a set of $P$ patterns, defined by $\boldsymbol{\xi^\mu} = (\xi_1^\mu, \ldots, \xi_N^\mu)$, with $\mu = 1, \ldots, P$. To "store" these patterns, one has to set the connections to

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \quad , \tag{1.16}$$

which results in a symmetric connectivity matrix. With this choice of weights the patterns become fixed points of the dynamics in Eq. (1.15). In this case, the network can be initialized in a corrupted version of the pattern and, when let free to evolve, it will converge to the stored pattern which is most similar to the initial state. This mechanism can be seen as a form of error correction or of associative memory. There are however limitations in the number of patterns one can store. For the case of random uncorrelated patterns in which $+1$s and $-1$s are drawn independently and with the same probability, the capacity, i.e. the number of maximum patterns that can be faithfully retrieved, is known to scale linearly with $N$ (see Hertz et al. (1991) for a textbook treatment of the topic).

We have seen how to induce fixed points in the dynamics of a binary network by following Hopfield's prescription for the weights. In the next section I will focus on how to modify this prescription to obtain sequential activity and highlight the problem of slow timescales in binary networks.

### 1.2.1 Sequential activation in binary networks

Manifestly, the only intrinsic timescale of a single binary neuron is given by $\Delta t$, the time step that separates consecutive updates. Thanks to network interactions however, slower timescales can be produced. If the synaptic connections are allowed to be asymmetric, the network can enter reliable cycles (Amit (1989)).The period of such cycles can be considered an effective timescale of the system. To clarify this concept, consider a neuron which is downstream of a binary network. If the network is in a cycle of period $T$, the connections to the downstream neuron can be set so that the latter is active only when the network is in a specific state. In this way, the downstream neuron will be active only every $T$ time steps. This represents a straightforward way to exploit long cycles to obtain long effective timescales.

Hopfield (1982) already proposed to add a set of asymmetric weights on top of the ones defined by Eq. (1.16), to produce transitions between patterns. We will refer to these weights as *transition weights*, and they are given by

$$J_{ij}^T = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^\mu \quad . \tag{1.17}$$

The idea is that these connections would drive the transition from pattern $\mu$ to pattern $\mu+1$, so that each pattern becomes effectively a metastable state. However, this mechanism works only for very short sequences, while for longer ones it suffers from instabilities (Hopfield (1982)). The reason for this failure is that if the transition weights are not strong enough then the patterns are stable, while if they are too strong then the patterns cannot be correctly retrieved (Sompolinsky and Kanter (1986)).

This problem was addressed by adding a second timescale to the system at the level of the synapses (Sompolinsky and Kanter (1986); Kleinfeld (1986)). More precisely, if one postulates that the synapses associated to the transition weights have a slower dynamics than the ones encoding the patterns, one can increase the transition weight strength without harming the stability of the patterns on a short timescale. This leads to stable sequential activation both with asynchronous or synchronous dynamics (Fig. 1.2). From the perspective of this thesis, this approach represents a direct way to exploit an intrinsic slow timescale of the system (the slow synaptic timescale) to slow down the dynamics of the full network. Slow mechanisms at the synapse level exist in real neurons (see section 1.1) and good candidates to implement this type of mechanism could be NMDA receptors or short-term facilitation. In a different approach, Buhmann and Schulten (1987) proposed to exploit stochasticity to "escape" the stable fixed points given by the patterns and transition to the next one. While this solution avoids the introduction of intrinsic slow timescales, it results in quite irregular sequences when $N$ is not very large.

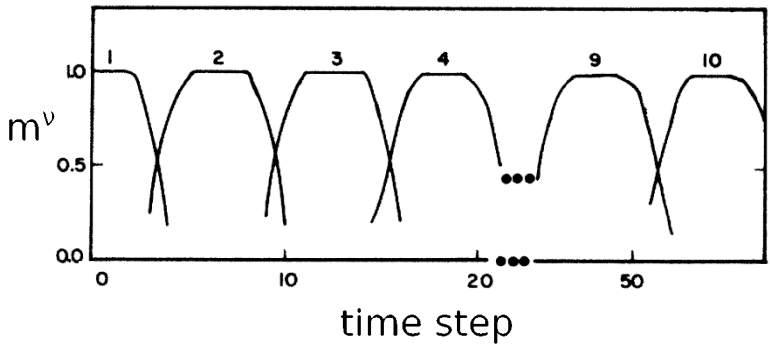



Figure 1.2 – **Sequential activation in binary networks.** The overlap $m^\nu(t) := N^{-1}\sum_{i=1}^{N}\xi_i^\nu S_i(t)$ of the network with different patterns is plotted against time steps. Transition weights are chosen to integrate input with a step-function kernel. Figure adapted from Sompolinsky and Kanter (1986).

Can we get slow timescales in a deterministic binary network without any intrinsic slow mechanism? For simplicity, consider synchronous dynamics. In this case, using only a set of transition weights given by Eq. (1.17) without any pattern-encoding weights, results in a sequential reactivation during which every pattern is visited for only one time step. As we have seen, the length of the sequence that the network produces can be seen as an effective timescale of the system. Therefore, to obtain slow timescales with this strategy the network has to produce long sequences. However, as for the Hopfield model, the number of patterns that can be visited in a sequence is

limited, and so is the effective timescale of the network. Using the result of Gardner and Derrida (1988), we conclude that the maximal expected length of the orbit, when considering random patterns, scales linearly with $N$. In chapter 2, we try to overcome the limitation of linear scaling by assuming that the patterns are not randomly chosen, but highly structured.

### 1.2.2 Activity propagation in networks of spiking neurons

Abstract binary models of sequential activation share conceptual similarities with spiking neuron models of activity propagation. Synfire chains (Diesmann et al. (1999)) allow fast propagation of activity across feed-forwardly connected populations of spiking neurons. The timescale of the activity propagation is set by the refractory period (Kistler and Gerstner (2002)) and it is therefore on the order of few milliseconds. The robustness and the speed of the activity propagation generated by synfire chains makes them appealing for some applications, such as song generation in songbirds (Hahnloser et al. (2002)). However, the large amount of neurons required to have stable propagation makes the use of such models for long timescale generation (i.e. long chains of activity propagation) rather unlikely. To overcome this limitation, in a recent model Setareh et al. (2018) proposed to use assembly dynamics in combination with a slow fatigue mechanism to obtain slow activity propagation, that could be more suited to support some behavioral tasks. Similarly to Sompolinsky and Kanter (1986) and Kleinfeld (1986), this model highlights the possibility to slow down the effective network dynamics exploiting an intrinsic slow mechanism.

## 1.3 Slow dynamics in networks of rate neurons

We have seen that spiking neuron models can be simplified by describing neurons only in terms of whether they are active or not. This leads to binary neuron models, that we described in the previous section. In a different approach, we can describe single neurons not in terms of single spikes, but in terms of spike rates, i.e. discarding the information about the exact spike timing. The spike rate is the expected number of spikes at a certain time, and it can be interpreted in multiple ways. In one way, it can be seen as the spike count of a single neuron on a single trial over a certain time window. To obtain a reliable value however, this interpretation requires long time windows which leads to information loss at shorter timescales. Alternatively, the firing rate can be interpreted as an average over a homogeneous population of neurons, which does not require a time average. Finally, it can represent the average spike count of a neuron over trials, which allows to consider smaller time windows. In this interpretation, the neuron rate is similar to the peri-stimulus time histogram (PSTH) measured experimentally. The reason to study firing rate models is twofold: first, we are interested in slow dynamics and for this reason we can discard the information about the exact spike timing. Second, models of recurrent networks of rate units are simpler to study analytically while still exhibiting interesting dynamics, as we will detail in section 1.4.

Systematic reductions of spiking models to rate models started with Wilson and Cowan (1972) and it consists in deriving, through some approximations or some heuristics, an equation (differential or integral) that describes the evolution of the firing rate in time. The classical approach consists of first determining the stationary input-output transformation (or gain function), that gives the output rate in response to a certain input current. Then, one needs to determine the transient behavior by which the stationary state is reached. Several techniques have been proposed to tackle this problem, using integral equations (Gerstner (2000)), linear-nonlinear Poisson models (Aviel and Gerstner (2006); Ostojic (2011)) or multi-dimensional rate models (Mattia and Del Giudice (2002); Schaffer et al. (2013)). In this introduction however, we will focus only on phenomenological rate models of the form

$$\tau_y \dot{y}(t) = -y(t) + I(t) \quad , \tag{1.18}$$

where $y$ is an intermediate variable, from which the rate $r$ can be obtained by applying an appropriate gain function $g$, i.e. $r = g(y)$. The timescale $\tau_y$ of the intermediate variable is typically chosen to be on the same order as the membrane timescale $\tau_m$.

### 1.3.1 Slowness in rate models: Attractor dynamics

Imagine a minimal "network" consisting only of one neuron connected to itself. Substituting the self-connection for the current in Eq. (1.18), we have

$$\tau_y \dot{y}(t) = -y(t) + w_r g(y(t)) \quad , \tag{1.19}$$

where $w_r$ is the strength of the self-connection. If the neuron is quiet in the absence of input, i.e. $g(0) = 0$, the point $y = 0$ is a fixed point of the system. By linearizing the dynamics, we find that the fixed point is stable if $g'(0)\, w_r < 1$. The effective timescale of the system in the stable regime behaves as

$$\tau_{\text{eff}} = \frac{\tau_y}{1 - g'(0)\, w_r} \quad . \tag{1.20}$$

If $g'(0)\, w_r \to 1^-$, the system becomes infinitely slow. This result is also true for other fixed points of Eq. (1.19): by fine-tuning the self-connection strength, one can slow down the relaxation dynamics to the fixed point.

This mechanism can be generalized to a network of $N$ rate neurons. We can write the analogous of Eq. (1.19) for a network of $N$ units as

$$\tau_y \dot{y_i}(t) = -y_i(t) + \sum_{j=1}^{N} w_{ij}\, g(y_j(t)) \quad , \tag{1.21}$$

where $w_{ij}$ is the synaptic strength from neuron $j$ to neuron $i$. A stable fixed point in a neural network is also called *attractor*, since the dynamics attract the network to this state, and it is analogous to a Hopfield pattern (see section 1.2). If $y_i^0$ are the components of the attractor state, stability theory requires that all the eigenvalues of the matrix $\mathscr{J}$, whose elements are

$$\mathscr{J}_{ij} = -\delta_{ij} + w_{ij} g'\left(y_j^0\right) \quad , \tag{1.22}$$

have negative real parts. If this condition is satisfied and the connectivity matrix is normal, then the slowest effective timescale of the system is given by

$$\tau_{\text{eff}} = \frac{\tau_y}{|\text{Re}(\lambda_{\max})|} \quad , \tag{1.23}$$

where $\lambda_{\max}$ is the eigenvalue of $\mathscr{J}$ with the largest real part (see section 1.1.4). Again, tuning the connectivity allows to slow down the dynamics arbitrarily in the vicinity of an attractor.

Attractor models in networks of rate or spiking neurons have a long history (see Amit (1989) for a nice introduction to the topic) and have been proposed as models of working memory (Brunel and Wang (2001); Mongillo et al. (2003)). Some studies looked in particular at slow dynamics while approaching an attractor state. For example, the derivative feedback approach proposed by Lim and Goldman (2013) allows to vary the effective speed at which the network relaxes to a fixed point, while requiring less fine-tuning of the weights than in the typical attractor picture. Similarly, line attractor models (Amari (1977); Ben-Yishai et al. (1995); Compte (2000)) are characterized by a connectivity structure that induces a stable one-dimensional manifold in the dynamics. Due to heterogeneity in the network or to noise, such models can exhibit slow drift (Seeholzer et al. (2018)), which could provide an interesting alternative approach to the generation of slow dynamics.

## 1.4 Chaotic dynamics in random rate networks

In the absence of external input or noise, a network of rate neuron models is a (typically high-dimensional) deterministic dynamical system. As such, it can asymptotically be either at a fixed point (attractor state), in a periodic (or quasi-periodic) orbit, or in a chaotic state. The first proof of the existence of a chaotic phase in random rate networks was given by Sompolinsky et al. (1988), for balanced networks, using dynamic mean-field theory (DMFT). It was later shown that a transition to chaos appears quite generally in many different network models, that are not necessarily balanced (Kadmon and Sompolinsky (2015)). For simplicity however, in this introduction we will discuss the emergence of chaotic dynamics in the balanced case.

In randomly connected networks of spiking neurons, a global balance of excitation and inhibition can be dynamically regulated (van Vreeswijk and Sompolinsky (1996); Brunel (2000)). If the inhibitory feedback is stronger than the excitatory one, the network exhibits an *asynchronous irregular* state, characterized by very low average firing rates (Brunel (2000)). In this state, the spike times are essentially chaotic (van Vreeswijk and Sompolinsky (1996)), which renders them largely unpredictable. This motivates the description of the balanced state in terms of firing rates, which are more reliable quantities.

Since in the balanced state the mean firing rate is relatively stable, it is convenient to describe the system in terms of the deviations from the mean values (*effective* rates), i.e. $r_i(t) - r_0$ (see Hennequin (2013) for a simple introduction to this topic). We can rewrite the network Eq. (1.21) as

$$\tau_x \dot{x}_i(t) = -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) \quad , \tag{1.24}$$

where $\phi$ is the effective gain function of a neuron embedded in a network in the balanced state, and it transforms deviations of the intermediate variable $x$ into deviations from the mean rate. If a neuron is momentarily firing at a lower rate than average, then the effect on the other units should be negative, i.e. $\phi(x) < 0$ if $x < 0$. Analogously, we should choose $\phi$ such that $\phi(x) > 0$ if $x > 0$. Finally, $\phi$ is bounded from below by $-r_0$, since the rate cannot be negative. A widely used choice consists in setting $\phi(x) = \tanh(x)$ (Sompolinsky et al. (1988)), which is amenable to theoretical analysis due to its symmetry. A more realistic choice, used by Rajan et al. (2010), is

$$\phi(x) = \begin{cases} r_0 \tanh\left(\frac{x}{r_0}\right) & \text{for} \quad x \leq 0 \\ (2 - r_0)\tanh\left(\frac{x}{2-r_0}\right) & \text{for} \quad x > 0 \end{cases} \quad , \tag{1.25}$$

which provides a larger range of firing rates above the average than below.

In the next section I briefly review the fundamental steps and assumptions needed to study the network dynamics using DMFT.

### 1.4.1 Dynamic mean-field theory

Consider a fully-connected neural network described by Eq. (1.24), in which we assume that the synaptic weights are i.i.d. sampled from a normal distribution, i.e. $J_{ij} \sim \mathcal{N}\left(0, g^2/N\right)$. Using a distribution with zero mean is necessary in order to have balance between excitation and inhibition. Scaling the variance with $1/N$, allows to have the input variance independent of $N$. In this way, both the mean input and its fluctuations remain finite when $N \to \infty$ (Sompolinsky et al. (1988)). From the physics point of view, this type of neural network is an example of a *disordered system*, in which the quenched disorder is represented by the randomly chosen synaptic weights.

Since $\phi(0) = 0$, the network has a fixed point at $x_i = 0 \ \forall i = 1, \ldots, N$. In the $N \to \infty$ limit, the eigenvalues of the connectivity matrix J obtained by the sampling process described above are known to lie uniformly on a disk in the complex plane, centered in zero and of radius $g$ (Girko (1985)). If for simplicity we assume that $\phi'(0) = 1$, the zero fixed point is therefore stable for $g < 1$. In this regime, the network relaxes to the attractor at zero with a timescale that becomes slower as $g \to 1^-$, as we expect from the discussion of section 1.3 (Fig. 1.3A).

How does the network behave above the instability, i.e. for $g > 1$? Empirically, one observes that the network enters a state of irregular fluctuations that are self-sustained (Fig. 1.3B). To understand the dynamics of the network in this regime, one needs to resort to mean-field techniques. The crucial step in the derivation of Sompolinsky et al. (1988) is to understand that, in the limit of $N \to \infty$ and when averaged over the quenched disorder, neurons become independent. This result can be justified using statistical physics techniques based on the path-integral description of disordered systems (Schücker et al. (2016a); Crisanti and Sompolinsky (2018)). Thanks to the independence of the neurons, one can approximate the input to each neuron with a Gaussian process $\eta(t)$ and replace the average over initial conditions, neurons and weight realizations with an average over the statistics of $\eta$. The dynamics of each unit can be seen as a realization of the following stochastic differential equation

$$\tau_x \dot{x}(t) = -x(t) + \eta(t) \quad , \tag{1.26}$$

with $\langle \eta(t) \rangle = 0$ and $\langle \eta(t)\eta(s) \rangle = g^2 \langle \phi(x(t))\phi(x(s)) \rangle$, which means that the statistics of $\eta$ need to be self-consistently matched to the ones of $x$. For this reason, solving Eq. (1.26), i.e. finding the statistics of $x$, is hard. However, Sompolinsky et al. (1988) were able to find the set of possible solutions and to conclude that the only one consistent with the stability analysis requires that the autocorrelation of $x$ goes to zero as the time interval goes to infinity. This behavior of the mean field variable $x$ corresponds to chaotic dynamics in the microscopic network.

In contrast to the case discussed here, neural networks in the brain are not fully connected. To be consistent with this observation, one could set each weight independently to be nonzero with a probability $p = K/N$, so that each neuron receives on average input from $K$ pre-synaptic units. To make this setup tractable, all the nonzero connections are usually chosen to have the same value. Notice that in this case, to be in the balanced regime one would need to consider at least two

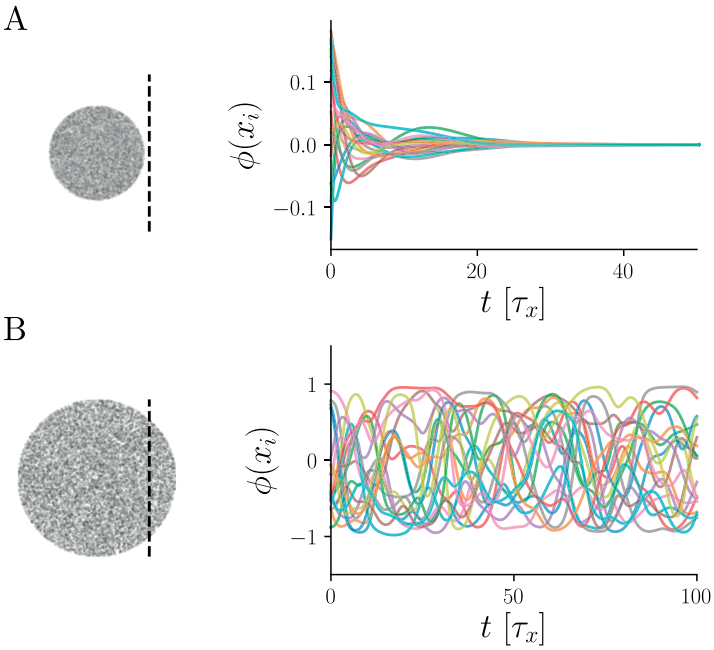



Figure 1.3 – **Dynamical regimes of a random network. A:** Eigenvalue spectrum for $g < g_c$ (left), where the dashed line indicates the imaginary axis. On the right, the evolution of a subset of rates over time, for the same value of $g$. **B:** Same as A, but for $g > g_c$. In this regime, the fluctuations are self-sustained and rather slow.

interacting populations, one inhibitory and one excitatory (Brunel (2000)). If $K$ is large, one can approximate the effect of this connectivity by a fully-connected Gaussian connectivity, the mean and variance of which can be matched to have the same statistics as for the sparse connectivity (Kadmon and Sompolinsky (2015)). For small $K$, one can simplify the system by assuming that each neuron receives *exactly* $K$ inputs, which again allows to use DMFT tools (Mastrogiuseppe and Ostojic (2017)).

In the chaotic state, the network exhibits rich dynamics with multiple timescales. One straightforward way to quantify the effective timescale of the network is to compute (or measure) the decay time of the autocorrelation, which gives an indication of how long a typical $x_i$ variable remains correlated with itself. As it is typical of critical systems, the correlation time diverges as $g \to 1^+$. In contrast to the slowing down happening below the criticality ($g \to 1^-$), in this case the slow dynamics is self-sustained and therefore does not end. A more sophisticated approach to quantify the timescale of the network consists in computing the duration of a memory trace induced by an external stimulus, based on optimal readout theory (Toyoizumi and Abbott (2011); Schücker et al. (2016b)). This approach leads to the conclusion that the network has a longer memory above the criticality than below, which suggests an interesting functional role of the chaotic state (Toyoizumi and Abbott (2011)).

The rate model considered in this introduction is highly simplified. Yet, how network dynamics are modified when considering different rate models has not been explored. In chapter 3 and 4 we address exactly this question, for the case of rate models with adaptation and for general linear rate models.

### 1.4.2 Training chaotic networks

Thanks to their rich dynamics, chaotic networks have been proposed to constitute an ideal substrate for learning. The problem of modifying the recurrent weights according to a desired network dynamics is indeed notoriously hard to solve. The back-propagation-through-time algorithm (Rumelhart et al. (1986)), suffers from instabilities due to the problem of vanishing and exploding gradients (Hochreiter and Schmidhuber (1997)). In machine learning this issue is addressed by using long-short term memory networks (Hochreiter and Schmidhuber (1997)), that do not however have an obvious counterpart in the brain. Moreover, it is unclear if it is possible to implement the back-propagation algorithm with local learning rules (Marblestone et al. (2016)).

The idea of *reservoir computing* (Maass et al. (2002); Jaeger and Haas (2004)) is to exploit the dynamical richness of a randomly connected recurrent neural network to linearly read out interesting temporal patterns. By "linear read out" we mean that the network output $z$ depends only on a linear combination of the outputs of the units, i.e. $z(t) = \sum_{i=1}^{N} w_i^{RO} \phi(x_i(t))$. The main advantage of this approach is that only the readout weights $w_i^{RO}$ are modified, which makes the learning process more stable and amenable to be performed using local learning rules.

To increase the robustness of the learned trajectories, one can introduce feedback from the output to the network (Sussillo and Abbott (2009)). These feedback connections also modifies the dynamics of the network, since they constitute a rank-one perturbation of the connectivity matrix. The dynamics of random networks with low-rank perturbations has been recently studied using mean-field techniques (Mastrogiuseppe and Ostojic (2018)), which allowed to construct networks able to solve interesting tasks while maintaining the typical variability of chaotic networks. Finally, recently proposed strategies to learn the full connectivity of the recurrent network (Gilra and Gerstner (2017); DePasquale et al. (2018)) represent a promising approach toward a biologically plausible learning algorithm for complex tasks.

## 1.5 Thesis contribution

This thesis summarizes the research I performed during my Ph.D. at EPFL, from 2013 to 2018, in the laboratory of Prof. Wulfram Gerstner. The main goal of my work was to understand to which extend the presence of a build-in structure impacts the capability of a recurrent neural network to exhibit slow timescales. "Structure" is intended in a broad sense: it can refer either to a non-random network connectivity, or to additional properties of the neuron model under consideration. This questions are motivated by the necessity of producing (or being sensitive to) slow dynamics during most behavioral tasks that involve a sequence of actions.

In chapter 2, we investigate the generation of periodic orbits by recurrent networks of binary neurons which evolve synchronously, in discrete time and in a purely deterministic fashion. Our main result is a constructive proof which allows to design weight matrices in such a way that the corresponding network evolves along a maximally-long orbit. In other words, a network of $N$ units visits all possible $2^N$ states before visiting one state twice, so that the length of the orbit scales exponentially with the number of units. This result is obtained by considering highly structured sequences of patterns, and has the downside of requiring an exponential fine-tuning of the weights. An interesting feature of the resulting orbit is that it naturally exhibits a hierarchy of timescales, which can be advantageous in producing both fast and slow sequences.

In chapter 3, we shift our attention to the interaction between the properties of single neurons and the recurrent network dynamics. In particular, our goal was to understand how slow intrinsic mechanisms, such as rate adaptation, influence the dynamics of recurrent networks. We show that adaptation stabilizes the dynamics of a random network, by increasing the critical connectivity strength at which the transition to chaos occurs. The network with adaptation also exhibits a new chaotic phase, that we call "resonant chaos", in which the statistics of the dynamics are dominated by a specific resonance frequency. Strikingly, this resonance frequency does not depend on the connectivity strength and can be predicted purely based on single neuron properties. The recurrent connections interact with the adaptation mechanism by increasing the coherence of the oscillatory behavior typical of the single neuron. On the other hand, this can decrease the correlation time by increasing the amount of correlation at short time lags.

In chapter 4, we generalize the theoretical approach introduced in chapter 3 to networks of multi-dimensional rate units. Multi-dimensional rate models are of theoretical and practical interest because they can be used both to better capture the transient behavior of a population of spiking neurons and to include additional mechanisms such as adaptation or refractoriness. We find that a transition to chaos occurs consistently across different models, at values of the connectivity strength that depend on the model parameters. Consistently with the findings of the previous chapter, the properties of the single neuron model, and in particular its linear response function, are predictive of the qualitative features of the recurrent network dynamics in the chaotic phase.

In the two final chapters, we divert from the study of network dynamics to consider possible roles

of adaptation in learning slow and complex sequential tasks. Both chapters present preliminary results, that were obtained in parallel to the study of the dynamics, and that require further investigation. In chapter 5, we probe the network with adaptation described in chapter 3 in learning tasks requiring slow timescales. To focus on the intrinsic benefits deriving from the introduction of adaptation, we use a reservoir learning approach and find that adaptation has in general a positive effect on the performance of the network, with only minor negative consequences on the performance on fast tasks. In chapter 6, we propose a hierarchical network architecture that allows to produce sequences that have a hierarchical structure, while representing them in a compact way. This architecture might be suited to perform biologically plausible learning of sequence generation and has similarities to the way humans seem to learn such tasks.

My specific contribution to the different projects is explicitly stated at the end of every chapter. In appendix A, I list and briefly summarize additional publications to which I did not contribute as first author.

# 2 Exponentially long orbits in binary networks

This chapter presents the paper (Muscinelli et al. (2017))

*Exponentially Long Orbits in Hopfield Neural Networks.* Samuel P. Muscinelli, Wulfram Gerstner and Johanni Brea. Neural Computation 2017 29:2, 458-484

## 2.1 Introduction

Humans and some animals can learn complex sequential behavior, such as dancing, singing, playing a musical instrument or writing. These complex sequential behaviors require precise coordination of many muscles on the timescale of seconds or minutes. That the brain achieves this coordination is remarkable, in particular, given that typical processes on a neuronal level, like action potentials or synaptic transmission, operate on a timescale of milliseconds.

To introduce a neuronal mechanism that could underlie such computations, we give an operational definition of sequence: *a sequence is a map from a ordered set of indices to a set of sequence elements.* We can take for example the natural numbers as the ordered index set and lowercase roman letters as the sequence elements. An example of a map is $1 \mapsto s, 2 \mapsto e, 3 \mapsto q, 4 \mapsto u, 5 \mapsto e, 6 \mapsto n, 7 \mapsto c, 8 \mapsto e$. A putative neuronal mechanism uses a recurrent network of neurons to represent the ordered set of indices and a group of readout neurons to represent the set of sequence elements (see Figure 2.1A). Each neuronal activity pattern in the index network encodes an index and the ordering is established by the autonomous dynamics. Neurons in the index network are recurrently connected to each other such that when the network is initialized in a particular state, the activity patterns evolve through a fixed sequence. The activity in readout neurons could encode motor commands that lead to a specific coactivation of muscles. To produce complex movements, it is sufficient to learn a map from index patterns to motor commands such that the first motor command is activated by the first index pattern and so forth.

It has been hypothesized that songbirds use this mechanism to learn songs (Fee et al. (2004)). For example, zebra finches produce songs that consist of motifs (sequences), each defined by a

specific ordering of sounds (elements). The activity in premotor area RA is highly correlated with the vocalization of single sounds and can thus be seen as encoding sequence elements. Neurons in RA receive input from brain area HVC. Most of the neurons in HVC that project to RA are active only once during a motif and the time of activity is locked relative to the onset of the motif itself (Hahnloser et al. (2002)). This observation leads to the hypothesis that neurons in HVC form a recurrent neural network that produces a chain-like activity pattern, where one group of neurons excites the next group of neurons and so forth (see Figure 2.1B). This can be seen as implementing the index network, where an index is associated to the activity of a particular group of neurons. In this way, each neuron is active only once during a sequence.

The main limitation of reading out from a chain-like activity is the maximal length of the sequence that can be generated in the recurrent network. Indeed, with each neuron in the recurrent network being active only once during a sequence, the length of learnable sequences is severely limited. The maximal length scales *linearly* with the number of neurons. If each recurrently connected neuron would be allowed to spike more than once, one would expect that the recurrent network could generate much longer sequences. Here we focus on intrinsically generated sequential activity that allows to overcome the linear scaling limit.

Models of recurrent neural networks come in different flavors. We can distinguish between discrete and continuous temporal dynamics, between deterministic and stochastic updates and between binary (spiking) and real-valued (rate-based) signal transmission. Each flavor comes with its own ways to overcome the linear scaling limit.

In systems with an infinite state space, which is typically the case for models with continuous temporal dynamics, a better scaling behavior is possible by exploiting the chaotic regime. Under specific conditions, transients in random networks of coupled oscillators (Zumdieck et al. (2004)) have been shown to scale exponentially with the number of units. A similar phenomenon can also be observed in spiking networks (Zillmer et al. (2009)). Rate-based networks were shown to be useful to implement the index network (Sussillo and Abbott (2009); Laje and Buonomano (2013)). In this case each index corresponds to a certain configuration in the state space and the order is determined by the intrinsic dynamics of the network.

The linear scaling limit can also be overcome in rate-based networks without relying on chaotic trajectories. One remarkable example is the coding strategy of grid cells, where the combination of cells with different (real-valued) periods leads to a representation capability that is exponential in the number of units (Fiete et al. (2008); Sreenivasan and Fiete (2011); Mathis et al. (2012)). Although grid cells code for space, a translation of the same mechanism to the temporal domain could be possible (Gorchetchnikov and Grossberg (2007); Eichenbaum (2014)).

Here we consider discrete dynamics with binary signal transmission, which does not allow to make use of the chaotic regime, since the state space is finite. More specifically, we study Hopfield neural networks with synchronous update and asymmetric weights. The dynamics of these networks converges usually to a limit cycle with a short period or to a fixed point. Indeed,

sequence generation in a Hopfield network can be related to linear separability in perceptron learning (Gardner and Derrida (1988); Brea et al. (2013)). This implies that the expectation of having an admissible sequence made of random patterns goes to zero when its length is larger than $2N$, where $N$ is the number of units. Therefore, using random patterns does not lead to any significant advantage with respect to the activity chain approach.

However, there are examples of very long sequences that can be generated with such networks. Distinct subnetworks could for example produce activity chains of different lengths. A network of 10 units produces a periodic orbit of length $T = 2 \cdot 3 \cdot 5 = 30$ steps, if it is divided into subnetworks of 2, 3 and 5 units with each subnetwork generating an activity chain of corresponding length. Generally, combinations of chains of co-prime length yield a very fast growth of the sequence length. This idea is related to the already mentioned coding strategy of grid cells (see e.g. Fiete et al. (2008)).

The occurrence of long periodic orbits in Hopfield networks raises the question: what are the longest sequences that such a network can generate? Here we prove that for each network size, it is possible to find weights such that the dynamics generates an orbit of maximal length. Moreover, our proof provides an algorithm to construct the weight matrix. In contrast to the network with chains of co-prime lengths, this network produces orbits of length $T = 2^N$ and it cannot be split into distinct subnetworks. Finally, we show that this networks is surprisingly robust to dynamical noise, and that small perturbations of the optimal weights lead to networks that are likely to produce nonmaximal but long orbits.

## 2.2 Results

We consider a recurrent neural network of $N$ binary neurons, whose state at time $t$ is specified by the single neuron activities

$$\xi_i(t) \in \{1, -1\}, \quad i \in \{1, \dots, N\} \quad . \tag{2.1}$$

Such a network has $2^N$ possible states, corresponding to all possible $N$-tuples made of 1 and $-1$. Geometrically, the network states correspond to the $2^N$ vertices of an $N$-dimensional hyper-cube. The set of all possible network states is called *state space*.

Time is treated as discrete and the network dynamics is synchronous, i.e. all neurons update their state at every time step. The update rule is

$$\xi_i(t) = \operatorname{sign}\left(\sum_{j=1}^{N} w_{ij}\xi_j(t-1)\right) \quad t \in \mathbb{N}, \tag{2.2}$$

where $\operatorname{sign}(\cdot)$ is the sign operator with the convention that $\operatorname{sign}(0) = 1$. Every neuron updates its state based on the status of the full network at the previous time step. The influence of neuron $j$ on neuron $i$ is weighted by $w_{ij} \in \mathbb{R}$ . Since the system is deterministic and there are only $2^N$

different network states, the dynamics in Eq.(2.2) can only lead to a fixed point or to a periodic orbit. We define the *length* $T$ of a periodic orbit as its smallest period. The maximal length of a periodic orbit is equal to $2^N$.

A specific $N$-dimensional sequence of length $T$

$$X = \begin{pmatrix} x_1(1) & \dots & x_1(T) \\ \vdots & & \vdots \\ x_N(1) & \dots & x_N(T) \end{pmatrix} \quad , \tag{2.3}$$

where $x_i(t) \in \{-1, +1\}$, is a *periodic orbit* of the system if we can find weights $w_{ij}$ such that the dynamics in Eq.(2.2) leads to that sequence, for a certain set of initial conditions, i.e.

$$\exists\ t_0 : \quad \xi_i(t_0 + kT + t) = x_i(t) \quad \forall k \in \mathbb{N} \quad . \tag{2.4}$$

Here and in the remainder we consider $i \in \{1, \dots, N\}$ and $t \in \{1, \dots, T\}$, unless differently stated. *The main result of this paper is the proof of the existence of a maximal length orbit for arbitrary* $N$. First, we present a *necessary* condition for a sequence to be an orbit of maximal length. Then, we present an iterative method to construct a maximal length orbit, for which we can find the weights explicitly. In the main text we only give the intuition of the mechanism, while the formal proof is given in the appendix.

## Maximal length orbits need reflection symmetry

In this section, we prove a necessary condition that sequences have to satisfy in order to be maximal-length orbits for the dynamics in Eq.(2.2). We notice that if the dynamics in Eq.(2.2) produces a sequence $X$, then

$$x_i(t) \sum_{j=1}^{N} w_{ij} x_j(t-1) > 0 \quad , \tag{2.5}$$

since Eq.(2.2) implies that $x_i(t)$ and $\sum_{j=1}^{N} w_{ij} x_j(t-1)$ have the same sign. We use in Eq.(2.5) and in the following the convention that $x_i(0) = x_i(T)$, $\forall i \in \{1, \dots, N\}$. The converse is also true, i.e. if a sequence satisfies Eq.(2.5) then the dynamics in Eq.(2.2) admits it as an orbit. We will refer to Eq.(2.5) as the condition of linear separability, in analogy with the geometrical concept (Elizondo (2006); Hertz et al. (1991)). The formulation in Eq.(2.5) allows us to prove the following lemma.

**Lemma 1.** *If there exists a set of weights such that a N-dimensional sequence of length* $T = 2^N$*, with the property that* $x_i(t) \neq x_i(t')$ *if* $t \neq t'$*, satisfies Eq.(2.5) for* $t \in \{1, \dots, 2^N\}$ *and* $i \in \{1, \dots, N\}$*, then*

$$x_i(t) = -x_i\left(t + 2^{N-1}\right), \quad t \in \{1, \dots, 2^{N-1}\} \quad , \tag{2.6}$$

*which means that the second half of the sequence should be the sign-inverted copy of the first*

*half.*

*Proof.* The sequence covers the whole state space, therefore it exists a $\tau$ for which $x_i(t+\tau-1) = -x_i(t-1)$ for all $i \in \{1,\dots,N\}$. Since the sequence is linearly separable,

$$0 < x_i(t+\tau)\sum_{j=1}^{N} w_{ij}x_j(t+\tau-1) = x_i(t+\tau)\sum_{j=1}^{N} w_{ij}\cdot\left(-x_j(t-1)\right) \quad . \tag{2.7}$$

The comparison with the linear separability condition, Eq.(2.5), at time $t$ implies

$$x_i(t+\tau) = -x_i(t) \quad , \tag{2.8}$$

i.e. also the state at time $t+\tau$ is the reflection of the state at time $t$. The argument can be iterated, implying that $x(t+\tau+1) = -x_i(t+1)$ and so on, until the whole state space is covered. Iterating the argument above $\tau$ times we get

$$x_i(t+2\tau) = -x(t+\tau) = x_i(t) \quad , \tag{2.9}$$

therefore $\tau$ should be equal to the half of the length of the sequence. □

Sequences that satisfy the hypothesis of lemma 1 will be referred to as maximal-length orbits. Lemma 1 illustrates a *necessary* condition that a maximal-length sequence needs to satisfy in order to be linearly separable, that is, implementable in a recurrent network. However, the condition is not sufficient and one could construct maximal-length sequences that have the reflection symmetry but are not linearly separable.

### Existence of maximal length period orbit

In this section we illustrate a recursive procedure that allows us to construct linearly separable sequences of maximal length. The procedure is inspired by lemma 1. Suppose we have a sequence of maximal length for a network of $n$ units. We denote this sequence by $X_n$. To increase its dimensionality, we add a unit to the network. This new unit takes a constant value, so that we obtain a $(n+1)$-dimensional sequence that explores *half* of the $(n+1)$-dimensional state space. Lemma 1 tells us that the second half should be the reflection of the first half in order to allow linear separability. The reflection step concludes the construction of an $(n+1)$-dimensional sequence $X_{n+1}$ of length $2^{n+1}$ starting from $X_n$. Algorithm 1 summarizes the sequence construction algorithm.

In the appendix we prove that the sequences devised according to Algorithm 1 are linearly separable and that the weights $w_{ij}$ for an implementation in a recurrent neural network can be constructed recursively. Here we provide the intuition of the proof and a simple algorithm for the construction of the weights.

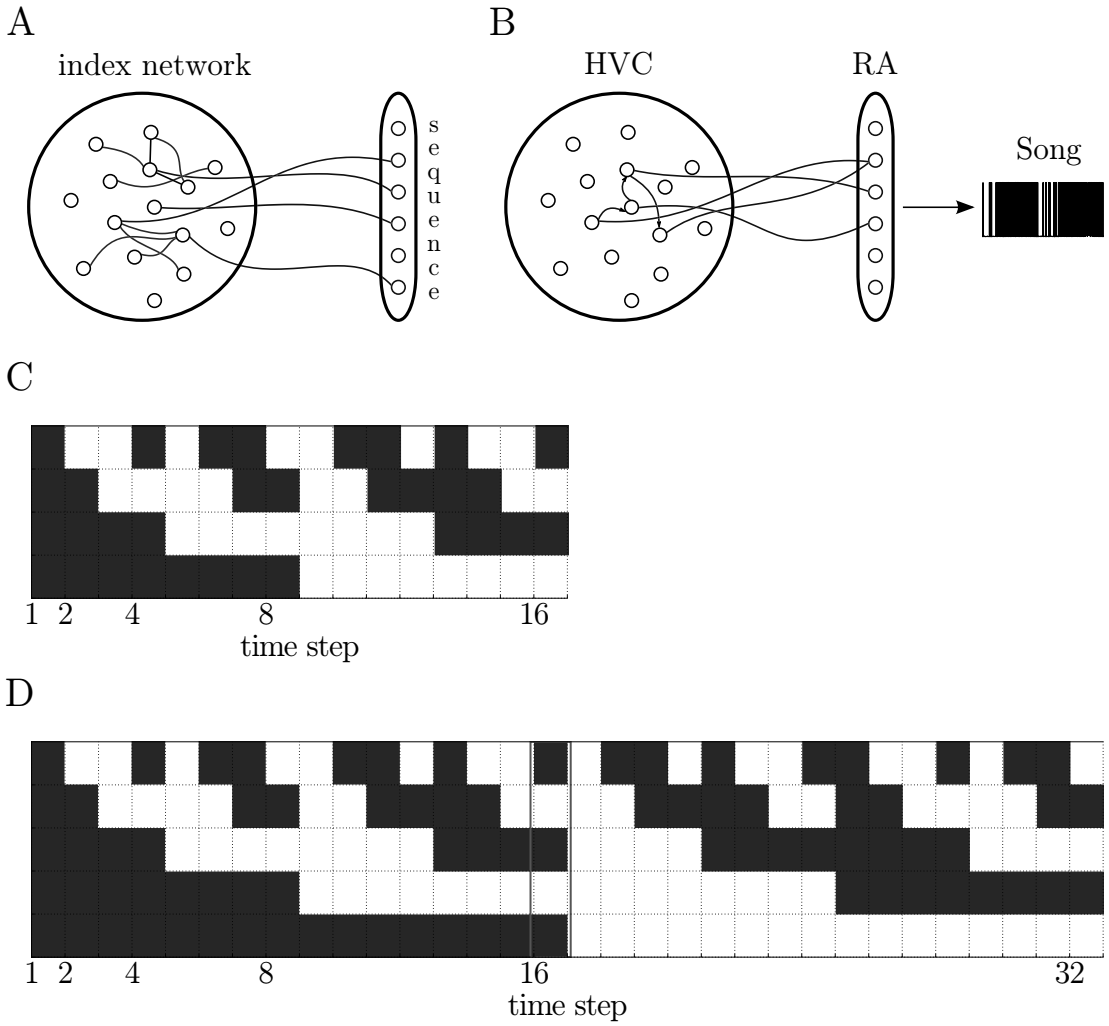



Figure 2.1 – **Network architectures and maximal-length sequences**. **A** Schematic representation of the index network connected to the sequence network. **B** Schematic representation of the hypothetical mechanism of song generation in the Zebra finch. Neurons in HVC are connected to form a chain and are only active once during a song. Neurons in RA read out their activity and can activate more than once. **C** Maximal length sequence for $N = 4$, constructed according to Eq.(1). Units are arranged from top to bottom according to their indices. A black rectangle indicates that the unit is active at that time step. **D** Maximal length sequence for $N = 5$, constructed according to Algorithm 1. Highlighted in red is the state at the critical time step $t = 2^{N-1}$.

**Algorithm 1** Construction of a maximal length sequence.

```
for n ∈ {1,...,N} do
    for t ∈ {1,...,2^{n-1}} do
        x_n(t) = 1
    end for
    for t ∈ {2^{n-1}+1,...,2^n} do
        for i ∈ {1,...,n} do
            x_i(t) = -x_i(t - 2^{n-1})
        end for
    end for
end for
```

The proof is done by induction, i.e. assuming that we have a linearly separable sequence $X_{n-1}$ for the $(n-1)$-dimensional case, we look for the existence of one in the $n$-dimensional case ($X_n$). We notice that the dynamics in Eq.(2.2) is symmetric under a simultaneous sign change of both $x_i(t-1)$ and $x_i(t)$, since this would correspond to a sign change of both sides of the equation. Given that $X_n$ is constructed according to Algorithm 1, i.e. the second half is the reflection of the first, we only have to show that the first half of the sequence, i.e. from $t = 1$ to $t = 2^{N-1}$, is linearly separable. Notice that this first half of $X_n$ is different from $X_{n-1}$, since it is its $n$-dimensional extension. We restrict to the case in which we do not modify the weights $w_{ij}$, for $i, j < n$. We introduce new weights to and from the added unit, $w_{in}, w_{ni}$, $i \leq n$. The proof consists in showing that the new weights can be chosen in a way that the $n$-dimensional sequence is linearly separable.

As we can see in Figure 2.1C, the $n$-th unit stays constant for the whole first half of the sequence. It flips its sign at $t = 2^{n-1}$ and stays then constant for the second half. Due to the special role of the switching point, we will refer to it as the *critical time point*. The activity of the first $n-1$ units evolves as in the $(n-1)$-dimensional case except for the critical time point. Indeed, while in the $(n-1)$-dimensional case all the $n-1$ units go from the state at $t = 2^{n-1}$ to the all-plus state (Figure 2.1C), in the $n$-dimensional case the first $n-1$ units should go to the all-minus state (Figure 2.1D). Since we do not change the weights between these units, this new transition should be caused by the interaction with the added unit.

These requirements can be translated into conditions on the new weights. We start by considering the input received by the $n$-th unit. A positive recurrent weight $w_{nn}$ ensures constant sign if it can overcome potentially negative input from the other units. However, since we want the $n$-th unit to flip sign at the critical time point, we need to have the input from the first $n-1$ units maximally negative at the critical time point. To obtain this, we set the weight from unit $i$ to the new unit $n$ equal to its activity $x_i$ at time $t = 2^{n-1}$.

$$w_{ni} = -x_i(2^{n-1}), \quad i \in \{1, \dots, n-1\}, \tag{2.10}$$

which yields

$$\sum_{j=1}^{n-1} w_{nj} x_j(2^{n-1}) = -\sum_{j=1}^{n-1} x_j(2^{n-1}) x_j(2^{n-1}) = -(n-1) \quad . \tag{2.11}$$

This choice ensures that at any time point different from the critical one the input from the first $n-1$ units is

$$\sum_{j=1}^{n-1} w_{nj} x_j(t) \geq \sum_{\substack{j=1 \\ j \neq j^\star}}^{n-1} w_{nj} x_j(2^{n-1}) - w_{nj^\star} x_{j^\star}(2^{n-1}) = -(n-3), \quad t \in \{1, \ldots, 2^{n-1}\} \quad , \tag{2.12}$$

since it exists a $t^\star$ for which $x_i(t^\star) = x_i(2^{n-1})$ for all $i \neq j^\star$, $i \in \{1, \ldots, n-1\}$. Therefore, by choosing

$$w_{nn} = n - 1 - \frac{1}{2} \quad , \tag{2.13}$$

we have a recurrent excitation which is always larger than the negative input from the first $n-1$ units except at the critical time point. The reason behind the choice of $w_{nn} = n - 1 - \frac{1}{2}$ and not, say, $w_{nn} = n-2$ is due to the presence of a stricter bound, as explained in the appendix and as can be seen in the next section. However, this stricter bound is necessary only if we want to be able to extend the system by another dimension, i.e. going to $n+1$ dimensions. If this is not the case, a larger range of weights gives rise to valid solutions.

We now consider the input received by each of the first $n-1$ units. The weights from the $n$-th unit to all the other ones should be negative to cause the transition to the all-minus state at the critical time point

$$w_{in} < 0, \quad i \in \{1, \ldots, n-1\} \quad . \tag{2.14}$$

The input from neuron $n$ to neuron $i$ should be bigger in magnitude than the one unit $i$ receives from the other $n-1$ units at the critical point

$$|w_{in}| > \sum_{j=1}^{n-1} w_{ij} x_j(2^{n-1}), \quad i \in \{1, \ldots, n-1\} \quad , \tag{2.15}$$

but this should be the only time point in which the $n$-th unit influences the others. This can be obtained if we set

$$w_{in} = -\left( \sum_{j=1}^{n-1} w_{ij} x_j(2^{n-1}) + \frac{1}{2^{n-1}} \right), \quad i \in \{1, \ldots, n-1\} \quad . \tag{2.16}$$

Intuitively, this corresponds to adding a "precision" bit to the lower bound of $|w_{in}|$. This choice is rigorously motived in the appendix, where we also provide exact bounds on the new weights. The recursive procedure for the weight construction is summarized in Algorithm 2, and an example

of a weight matrix built according to it is shown in Figure 2.2A.

**Algorithm 2** Construction of a maximal length orbit weight matrix.

**for** $n \in \{1,\dots,N\}$ **do**
  $w_{nn} = n - \frac{3}{2}$
  **if** $n > 1$ **then**
    **for** $i \in \{1,\dots,n-1\}$ **do**
      $w_{ni} = -x_i(2^{n-1})$
      $w_{in} = -\left(\sum_{j=1}^{n-1} w_{ij}x_j(2^{n-1}) + \frac{1}{2^{n-1}}\right)$
    **end for**
  **end if**
**end for**

## Exact bounds on the weights

Algorithm 2 is a special case within the more general conditions that the weights must satisfy. In the appendix, we derive the exact bounds that the new weight elements have to satisfy at *each recursive step*. Here we only report these bounds. In the following equations, $x_i(t)$ are the elements of the maximal length orbit constructed according to Algorithm 1.

- Elements of the added row:

$$w_{ni} = -x_i(2^{n-1})|w_{ni}|, \quad i \in \{1,\dots,n-1\} \quad , \tag{2.17}$$

  i.e. while their signs are constrained, their magnitudes are arbitrary.

- Diagonal element:

$$\sum_{j=1}^{n-1}|w_{nj}| - \min_{j\in\{1,\dots,n-1\}}|w_{nj}| < w_{nn} < \sum_{j=1}^{n-1}|w_{nj}| \quad . \tag{2.18}$$

- Column elements: $w_{in} = -|w_{in}|$ and

$$\begin{aligned} &|w_{in}| > \sum_{j=1}^{n-1} w_{ij}x_j(2^{n-1}) \\ &|w_{in}| < \frac{1}{2}\left\{\min_{t\in\mathcal{T}_i^+(2,2^{n-1})}\left[\sum_{j=1}^{n-1} w_{ij}x_j(t-1)\right] + \sum_{j=1}^{n-1} w_{ij}x_j(2^{n-1})\right\}, \\ &i \in \{1,\dots,n-1\} \quad , \end{aligned} \tag{2.19}$$

  where $\mathcal{T}_i^+(2,2^{n-1})$ is the set of all time points $t$ from $t=2$ to $t=2^{n-1}$ for which $x_i(t)=1$.

Eq.(2.19) represents the "tightest" bound to be satisfied. As we can see in Figure 2.2B, both the upper and lower bound on the new column elements go exponentially to zero with $n$, as well as

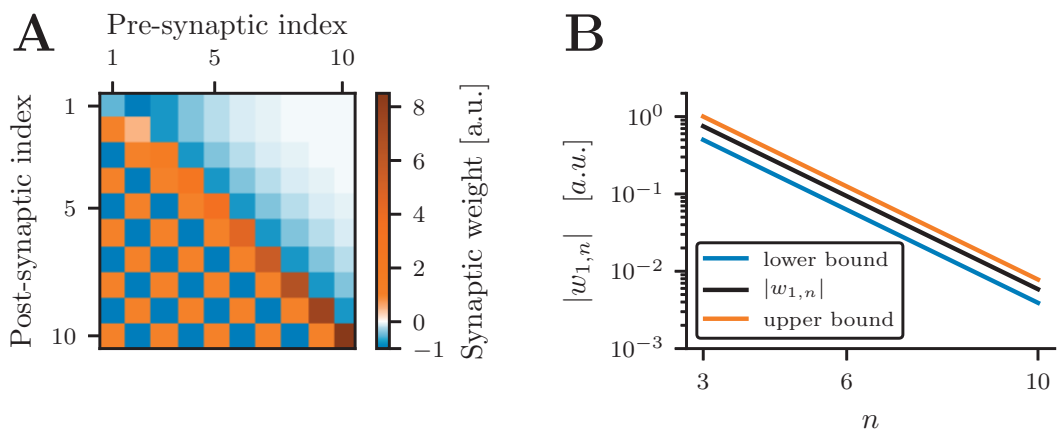



Figure 2.2 – **Weight matrix. A** Realization of the weight matrix according to Algorithm 2 for $N = 10$. Due to the exponential decrease of the super-diagonal weights, the color map is not able to capture its fine structure. **B** Exact bounds on the new column elements depending on the postsynaptic index. Due to the logarithmic scale both the bounds and the distance between them go to zero exponentially.

the distance between them. This means that new column elements need to be exponentially fine tuned.

## Comparison to co-prime chains

As mentioned in the introduction, it is straightforward to find weights such that a network of $N$ units produces a chain-like activity pattern, where $\xi_i(t) = 1$ if $t \bmod N = i - 1$ and $\xi_i(t) = -1$ otherwise (e.g. $w_{ij} = 1$ for $j \bmod N = i - 1$ and $w_{ij} = 0$ otherwise). If $K$ such networks with $N_1, \ldots, N_K$ units are combined into one network with $N = \Sigma(K) = \sum_{k=1}^{K} N_k$ units, and if $N_1, \ldots, N_k$ are co-prime, i.e. their greatest common divisor is 1, then the combined network will show a periodic orbit of length $T = \Pi(K) = \prod_{k=1}^{K} N_k$. Figure 2.3A shows an example with $N_1 = 2$ and $N_2 = 3$. Even though the sequence length grows asymptotically like $\Pi(K) \sim e^{(1+o(1))\,K \log K}$ (Sloane and Conway (2011)) and thus much faster than the number of units $\Sigma(K) \sim \frac{1}{2} K^2 \log K$ (Bach and Shallit (1996)), the orbit length of co-prime chains is considerably below the maximal sequence length, i.e. $\Pi(K) \ll 2^{\Sigma(K)}$ (see Fig. 2.3B).

In contrast to the network with chains of co-prime lengths, the maximal length orbit is produced by a network that cannot be split into distinct subnetworks; the weight matrix in figure 2.2A does not show block structure but reveals all-to-all connectivity of the network.

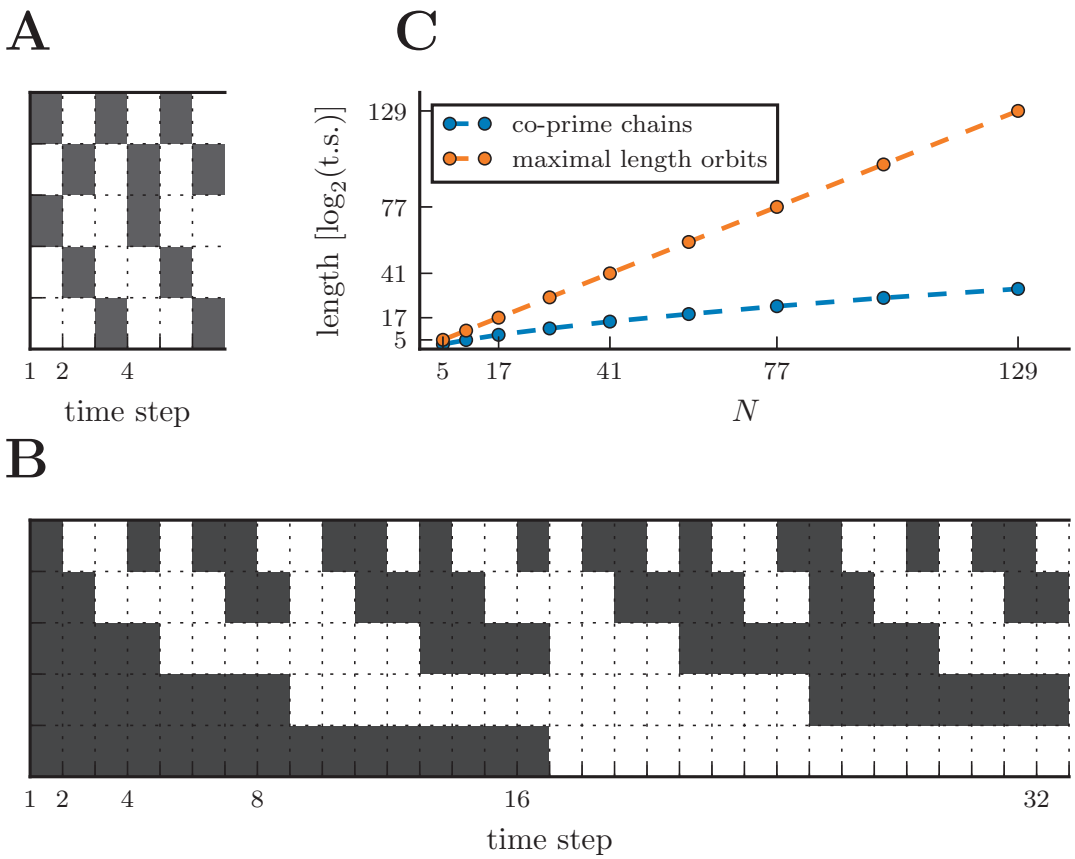



Figure 2.3 – **Maximal length sequences and co-prime chains**. **A** Co-prime chains of lengths 2 and 3 give rise to a periodic orbit of length 6. **B** The maximal length sequence with the same number of units, constructed according to algorithm 1 for comparison. **C** Increase of the sequence length with $N$. The scaling of the co-prime chains seems to be slightly sub-exponential.

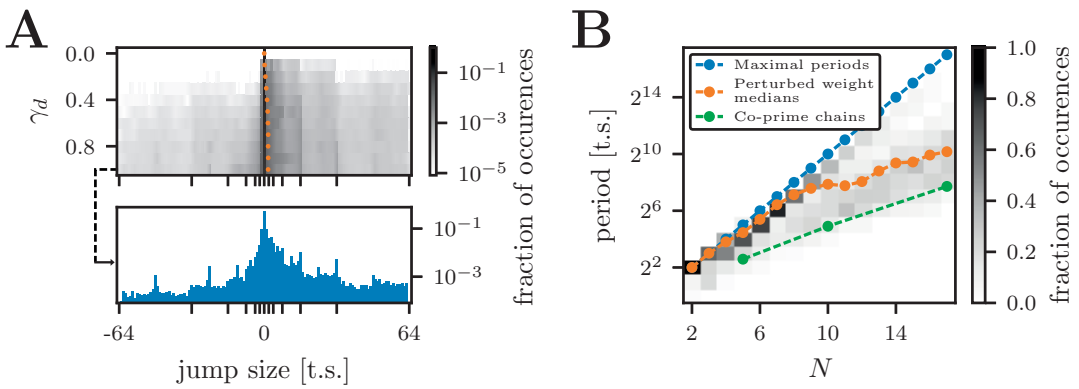



Figure 2.4 – **Effect of dynamical noise and weight noise**. **A** Top: Jump size distribution as a function of the dynamical noise level. Small jump sizes dominate (note the logarithmic grayscale). There is a slight asymmetry towards positive jumps, as revealed by the mean jump size (orange dots). Bottom: Jump distribution for $\gamma_d = 1.0$ **B** Distribution of longest orbits for perturbed weight matrices. For every $N$, the longest orbit was determined for 100 different weight matrices obtained according to Eq.(2.21) with $\gamma_w(N) = \frac{0.5}{N}$. We notice that at least in this range of $N$, the orbit lengths lie approximately between the orbit lengths of co-prime chains and the maximal lengths.

## Robustness to noise

Given the tightness of the bounds on the weight matrix, one may wonder whether the maximal length orbit is robust to perturbations. We considered two types of noise: Dynamical noise, i.e. perturbations of the total input onto each unit, and weight noise, i.e. perturbations of the weight matrix.

### Dynamical noise

In the presence of dynamical noise, the update rule becomes

$$\xi_i(t) = \operatorname{sign}\left(\sum_{j=1}^{N} w_{ij}\xi_j(t-1) + \gamma_d \epsilon_i(t)\right) \quad t \in \mathbb{N}, \tag{2.20}$$

where $\epsilon_i(t) \sim \mathcal{N}(0,1)$ and $\gamma_d$ is a parameter controlling the dynamical noise intensity.

The maximal length orbit covers the whole state space, therefore the orbit cannot be attractive. Indeed, for any "mistake" in the update, the network state jumps to a different point of the orbit. We define the size of a jump as the distance measured along the noiseless orbit and we estimate the distribution of jump sizes for different network sizes and noise intensities $\gamma_d$. The result for the case $N = 7$ can be seen in Figure 2.4A. The probability of having a jump of a certain size decreases rapidly with the size itself and increases with $\gamma_d$. This result is due to the fact that the average distance from threshold of the input onto a unit increases approximately linearly with the

unit index (not shown), and to the fact that large jumps require a large-index unit to flip sign. The distributions are slightly asymmetric towards positive jump sizes, as can be seen by looking at their means (orange dots). Nonetheless, since the probability of mistakes increases with $N$ and due to the asymmetry in the jump size distribution, errors accumulate more for larger $N$, causing an effective shortening of the orbit for high levels of noise.

### Weight noise

In the presence of weight noise, the weights $w_{ij}$ obtained with algorithm 2 are perturbed according to

$$w_{ij}^{noisy} = w_{ij} + \gamma_w(N)\epsilon_{ij} \quad , \tag{2.21}$$

where $\epsilon_{ij} \sim \mathcal{N}(0,1)$ and $\gamma_w(N)$ is a parameter regulating the weight noise intensity that can depend on $N$. The fact that the $w_{ij}$ span increasing orders of magnitude for increasing $N$, suggests that this type of noise could be detrimental for the length of the orbit for large $N$. For this reason, we decided to characterize how the period of the orbits scales with $N$ in the presence of weight noise, using three different functional forms of $\gamma_w(N)$. For all the forms of $\gamma_w(N)$ and for each $N = 2,\ldots,17$, we generated 100 independent weight matrices according to Eq.(2.21), and measured the longest orbit that is produced by each matrix. In the analysis of the effect of the weight noise, we removed the dynamical noise to assess the effects independently. If $\gamma_w(N) \sim \mathcal{O}\left(2^{-N}\right)$, we found (not shown), that the orbit length still scales exponentially with $N$. If $\gamma_w(N) \sim \mathcal{O}\left(\frac{1}{N}\right)$, the distribution of orbit length seems to slowly saturate, as shown in Figure 2.4B. However, it is interesting to notice that the distribution, for this range of $N$s and noise levels, lies almost entirely between the maximal lengths and the length of co-prime chains constructed with the same number of units. This is noteworthy because it shows the existence of other weight matrices that produce very long orbits. Finally, if the noise scales as $\mathcal{O}(1)$, we found the presence of a critical $N(\gamma_w)$, above which the distribution of orbit lengths becomes dominated by very short orbits.

## A substrate to read out slow sequences

We are interested in evaluating how the orbit we devised can be used for the readout of sequences. In this section we will refer to the recurrent network with the weight matrix constructed according to algorithm 2 as the reservoir network. We consider two types of readout units: Binary units and real-valued units. Binary readout units $1 \le i \le M$ are driven by the network activity according to

$$y_i(t) = \text{sign}\left(\sum_{j=1}^{N} v_{ij}\xi_j(t-1) + b_i\right) \quad , \tag{2.22}$$

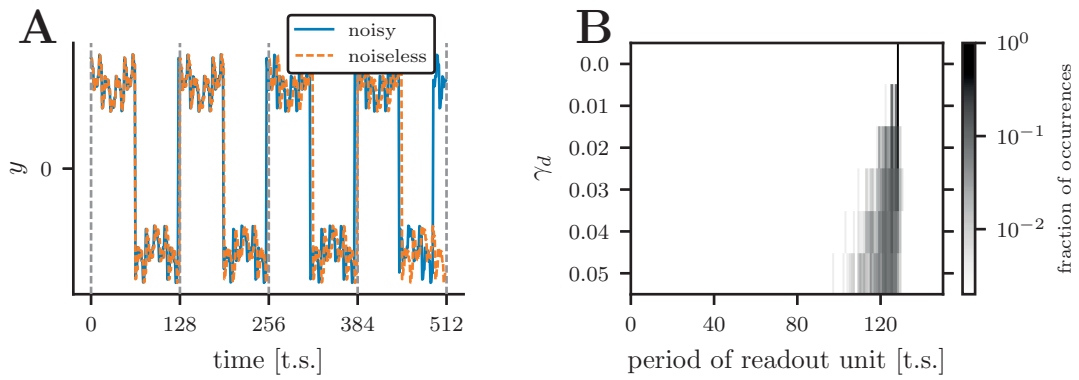



Figure 2.5 – **Examples of readout unit activities**. **A** Example of a real-valued readout unit in which the slow component of the oscillations is clearly visible. The reservoir network has $N = 7$ units ($T = 2^7 = 128$) and $\gamma_d = 0.05$. The addition of noise to the network dynamics does not disrupt the slow component, adding only small shifts, with a tendency for forward jumps, as also observed in Figure 2.4A. **B** Example of a binary readout unit set up to be a pattern detector. Its period, 128 time steps in the noiseless case, is perturbed when noise is added to the dynamics of the reservoir network. However, for small levels of noise, the distribution of periods remain centered around a value close to the noiseless case.

$v_{ij}$ are readout weights and $b_i$ is a bias parameter, $i \in \{1,\dots,M\}$ and $j \in \{1,\dots,N\}$. Similarly, real-valued readout units evolve according to

$$y_i(t) = \sum_{j=1}^{N} v_{ij}\xi_j(t-1) + b_i \quad . \tag{2.23}$$

Using these simple linear units, it is not possible to read out arbitrary sequences. This can be seen for example in the case of binary readout units. Suppose we want to generate a desired output sequence so that at each time point we fix an arbitrary target

$$y_i(t) = \pm 1 \quad \text{for} \quad t \in \{1,\dots,2^N\} \quad . \tag{2.24}$$

Finding the readout weights $\boldsymbol{v}_i = (v_{i1},\dots,v_{iN})$ for one binary readout unit is equivalent to finding a hyperplane that separates two sets defined on the vertices of an $N$-dimensional hypercube. The two sets are determined based on the desired activity $y_i(t)$, for $t \in \{1,\dots,2^N\}$. One set corresponds to the points in which $y_i(t) = +1$ and the other to the points in which $y_i(t) = -1$. Finding such a hyperplane is not possible for all arbitrary pairs of sets, therefore we cannot read out an arbitrary output sequence of length $2^N$ (Hertz et al. (1991)).

However, the orbit constructed according to Algorithm 1 is well suited to read out sequences with slow timescales. Indeed, if we measure the average number of time steps between two switches across the whole sequence for each unit (mean inter-switch interval), we see that it is exponentially increasing with the index (not shown). We can therefore say that higher index units

have *longer effective timescales*, because they change their state with an average interval much longer than the intrinsic timescale, which is equal to one time step. It is therefore possible to read out sequences that evolve on a slow timescale. A trivial example is a readout unit that copies the activity of one of the slow units. Combining the activity of several "slow" units, one could generate nontrivial sequences. Since the readout is not the main focus of this paper, we only provide two examples of how this can be done.

If a real-valued variable is read out from our maximal-length orbit, it will produce some form of oscillations on possibly multiple timescales. Figure 2.5A shows an example, generated with random readout weights, in which the slow timescales are well visible. As expected, if we add dynamical noise to the reservoir network, the slow timescales are maintained more than the fast ones. Noise has indeed the effect of producing small shifts either backwards or forward, but it will very rarely cause a jump to a very distant point.

A second possible application could be the readout of a "pattern detector", i.e. a binary readout unit that takes the value +1 only when the network is in a specific pattern. Since the reservoir network is in a specific pattern only once per cycle, the unit will be regularly active at intervals of $2^N$ time steps, in the noiseless case. For this reason, this type of readout unit could also be seen as a delay-counter. In order to set up this kind of readout, one could choose $v_{1j} = \bar{x}_j$, where $\bar{x}_j$ are the components of the pattern that we want to detect and $b_1 = -N + 1$. As before, we can study what happens in the presence of noise in the reservoir dynamics. In Figure 2.5B, we show the distribution of the activation periods of the readout unit, for $N = 7$. We see that for small amounts of noise, the performance of this type of readout unit degrades gracefully, with an asymmetric diffusion caused by the positive bias of jump sizes that was observed in Figure 2.4A.

## 2.3 Discussion

We have shown that a simple recurrent binary neural network with deterministic synchronous update dynamics can exhibit periodic orbits of maximal length $T = 2^N$. To prove this result we explicitly built a weight matrix that produces such an orbit. Although in principle it would have been possible to perform a search of long orbits or transients using random weights, the limit of learnability in the perceptron (Hertz et al. (1991); Gardner and Derrida (1988)) suggests that the expectation of finding a long orbit or transient would have been very low. However, the improvement on the length of the orbit comes at the cost of fine tuning the weights: the bounds in Eq.(2.19) become progressively tighter and the weights need to span multiple orders of magnitude. This requirement is rather unlikely to be exactly met by biological neural networks. But the simulations with weight noise showed that very long orbits are also possible with less fine-tuning. The bounds in Eq.(2.19) were found in a constructive proof that relies, in the inductive step ($N$ to $N+1$), on appending a row and a column to the $N \times N$-weight matrix while keeping the rest of the weight matrix fixed. It is possible that using a different procedure one would find a larger region of the weight space whose elements produce the desired orbit. However, the limit of learnability in the perceptron (Hertz et al. (1991); Gardner and Derrida (1988)) suggests that fine

tuning would be necessary anyway.

In a paper appeared after the publication of this work presented in this chapter, Hwang et al. (2018) studied the distribution of limit cycle for binary networks with the same dynamics as the one considered in this chapter, but in the presence of random connectivity. More precisely, they studied the number of periodic orbits of a certain length for variable symmetry degree of the connectivity matrix. Interestingly, they found that, while for almost-symmetric connectivity matrices the dynamics is dominated by short cycles, for larger degrees of asymmetry the number of longer limit cycles is large enough that the average cycle length diverges with $N$.

### Other maximal length orbits

The sequence presented above is not the unique maximal length orbit. Trivially, if we have *one* maximal length orbit, we can find other ones by relabeling unit indices, provided that one also permutes rows and columns of the weight matrix accordingly. Another allowed operation is to flip the sign of one unit along the entire orbit. Indeed, it is easy to show that changing the signs of all the weights in the row and column containing the flipped index, except for the diagonal element, one can produce the modified orbit.

On the other hand, lemma 1 provides a tool to exclude linear separability of other maximal length sequences. Two examples are binary count and Gray code (Gray (1953)), which do not have reflection symmetry and are therefore *not linearly separable*.

### Noise robustness and other approaches

In the results section we have shown that, in the presence of dynamical noise, the network state is unlikely to jump to an exponentially distant state on the orbit, but rather to the vicinity of the "correct" state. On the other hand, already small perturbations of the weights can significantly reduce the length of the longest orbit produced by the system, unless the noise level is also scaled down exponentially with $N$. This behavior is in contrast to what happens with co-prime chains that are robust to weight noise, since no fine-tuning of the weights is needed. However, dynamical noise is detrimental for co-prime chains. First, if individual chains are unstable, the activity in one subnetwork may vanish (all units inactive) or saturate at a maximal level (all units active). Second, even if we enforce only one unit per subnetwork to be active at each time step, such that jumps relative to the noiseless orbit can be measured as described in the paragraph after Eq.(2.20), the distribution of jumps is not peaked around small values (not shown). This is not surprising, since the subnetworks are uncoupled. For similar reasons, temporal versions of grid cell coding with different periods (Fiete et al. (2008); Sreenivasan and Fiete (2011); Mathis et al. (2012)) are likely to suffer from a high sensitivity to dynamical noise.

Models with continuous state space that rely on chaos to produce long transients, are by definition sensitive to noise. It has been shown that the time interval in which the activity of a noisy network

is reliable scales only linearly with the number of neurons (Ganguli et al. (2008)). Therefore, reading out from a chaotic or nearly-chaotic network also presents severe limitations in terms of noise robustness.

Although there is no obvious mapping between a binary network and a biological system, Hopfield networks have been shown to be useful conceptual tools. For example, the Hopfield model (Hopfield (1982)) had a strong conceptual influence on many associative memory models (Amit et al. (1985); Amit and Fusi (1994); Brunel (2000)). Moreover, a Hopfield network can be approximately mapped to a biological substrate, e.g. a multistable neural population (Zenke et al. (2015)). Seen from this perspective, the orbit discussed above could provide a method to produce long timescale sequences in a system that has only fast timescales, without exploiting any intrinsic slow time scale. Interestingly, this feature of the orbit would be largely robust to dynamical noise, because as we have already mentioned, the "slower" units are also more resistant to dynamical perturbations.

## 2.4 Methods: Proof of the theorem

For convenience, we rewrite here the theorem of the results section.

**Theorem** For all $N \in \mathbb{N}$ there are weights $w_{ij}$, $i \in \{1,\dots,N\}$ and $j \in \{1,\dots,N\}$ such that the dynamics in Eq.(2.2) admits a maximal length sequence $X^\star$ as orbit, i.e.

$$\exists\ t_0 : \quad \xi_i(t_0 + k\cdot 2^N + t) = x_i(t) \quad \forall k \in \mathbb{N} \quad . \tag{2.25}$$

The sequence covers the whole state space, i.e. it has the property $x_i(t) \neq x_i(t+\tau)$, $\forall\, t \in \{1,\dots,2^N\}$, $\forall\, i \in \{1,\dots,N\}$, $\forall \tau \in \{1,2,\dots,2^N-1\}$.

*Proof.* To prove the theorem, we need to show the existence of *at least one* sequence that covers the whole state space and that is linearly separable. Our approach is to *explicitly construct one particular maximal-length sequence* and to show that it is linearly separable. The theorem does not contain any restriction on the structure of the weights, therefore we are free to constrain them in any way as long that we show their existence.

We proceed by induction, building recursively both the sequence $X^\star$, according to Algorithm 1, and the weight matrix $w_{ij}$. For $X^\star$ to be a periodic orbit of the dynamics in Eq.(2.2), the weights have to satisfy linear separability constraints. We choose to perform the inductive step by *extending* the weight matrix, i.e. adding one column and one row without changing the other matrix elements. We stress that this does not restrict the statement of the theorem since it only requires the *existence of one set of weights*, regardless of how this is constructed.

Our inductive hypothesis contains the linear separability of the sequence for the $(N-1)$-dimensional case and an additional constraint on the weights that is necessary to be able to construct the weights by extension. This procedure not only shows the existence of a linearly separable sequence of maximal length but also provides a construction method for both $X^\star$ and $w_{ij}$.

### Inductive hypothesis and base case

The inductive hypothesis for a given $N \in \mathbb{N}$ contains the linear separability constraints

$$x_i(t) \cdot \sum_{j=1}^{N} w_{ij} x_j(t-1) > 0, \quad i \in \{1,\dots,N\},\, t \in \{1,\dots,2^N\} \quad . \tag{2.26}$$

Additionally, in order to prove the linear separability of $X^\star$ constructing $w_{ij}$ recursively, we assume that $w_{ij}$ satisfies

$$\sum_{j=1}^{N} w_{ij} x_j(2^N) < \min_{t \in \mathcal{T}_i^+(2,2^N)} \left[ \sum_{j=1}^{N} w_{ij} x_j(t-1) \right], \quad i \in \{1,\dots,N\}, \tag{2.27}$$

where $\mathcal{T}_i^+(2,2^N)$ is the set of all time points from $t=2$ to $t=2^N$ for which $x_i(t) = +1$.

We now prove the base case of the linear separability. For $N=1$ The maximal length sequence is $(1,-1)$. The sequence is linearly separable since for $w_{11} = -|w_{11}|$ we have

$$x_1(t=2)\, w_{11}\, x_1(t=1) = -1 \cdot (-|w_{11}|) \cdot 1 > 0 \tag{2.28}$$

$$x_1(t=1)\, w_{11}\, x_1(t=2) = 1 \cdot (-|w_{11}|) \cdot (-1) > 0 \quad . \tag{2.29}$$

The base case of the property in Eq.(2.27) is given by $N=2$, since for $N=1$ the $\min(\cdot)$ operator would be evaluated in an empty set. For $N=2$, Eq.(2.27) is satisfied by choosing

$$w_{11} < 0 \tag{2.30}$$

$$w_{22} > 0 \quad . \tag{2.31}$$

We notice that the first inequality is consistent with the one derived previously.

### The inductive step for linear separability requires bounds on the weights

We now assume that both Eq.(2.26) and Eq.(2.27) are true for $N-1$ and we prove that they also hold true for $N$.

We start with the linear separability condition. We split the sum in Eq.(2.26) into the contributions

that were already present in the case $N-1$ and into the new one

$$x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + x_i(t)w_{iN}x_N(t-1) > 0, \quad i \in \{1,\dots,N\},\, t \in \{1,\dots,2^N\}. \tag{2.32}$$

Then we divide the time range into four distinct sets

$$\begin{aligned}
&t = 1 && \Rightarrow && x_N(t) = 1, && x_N(t-1) = -1\\
&t \in \{1,\dots,2^{N-1}\} && \Rightarrow && x_N(t) = 1, && x_N(t-1) = 1\\
&t = 2^{N-1}+1 && \Rightarrow && x_N(t) = -1, && x_N(t-1) = 1\\
&t \in \{2^{N-1},\dots,2^N\} && \Rightarrow && x_N(t) = -1, && x_N(t-1) = -1
\end{aligned}$$

and for each of these sets we consider separately the case $i = N$ and $i \in \{1,\dots,N-1\}$. In the remainder of the proof the range of index $i$ is between 1 and $N-1$. We arrive at a system of eight inequalities:

$$\begin{aligned}
&\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) + w_{NN} > 0, && t \in \{1,\dots,2^{N-1}\}\\
&-\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) + w_{NN} > 0, && t \in \{2^{N-1}+1,\dots,2^N\}\\
&\sum_{j=1}^{N-1} w_{Nj}x_j(2^N) - w_{NN} > 0\\
&-\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) - w_{NN} > 0\\
&x_i(1)\sum_{j=1}^{N-1} w_{ij}x_j(2^N) - x_i(1)w_{iN} > 0\\
&x_i(2^{N-1}+1)\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) + x_i(2^{N-1}+1)w_{iN} > 0\\
&x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + x_i(t)w_{iN} > 0, && t \in \{1,\dots,2^{N-1}\}\\
&x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) - x_i(t)w_{iN} > 0, && t \in \{2^{N-1}+1,\dots,2^N\}.
\end{aligned} \tag{2.33}$$

Using the symmetry of $X^\star$ (line 7 in Algorithm 1), these equations can be reduced to four by

performing the substitution $x_i(t) \to -x_i(t-2^{N-1})$:

$$\begin{aligned}
&\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) + w_{NN} > 0, && t \in \{1,\dots,2^{N-1}\}\\
&-\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) - w_{NN} > 0\\
&-x_i(1)\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) - x_i(1)w_{iN} > 0\\
&x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + x_i(t)w_{iN} > 0, && t \in \{1,\dots,2^{N-1}\} \quad .
\end{aligned} \tag{2.34}$$

In the remainder we consider $t \in \{1,\dots,2^{N-1}\}$ unless explicitly stated. Intuitively, the first two inequalities represent the requirements on the influence of the first $N-1$ units on the $N$-th one and on the influence the $N$-th unit has on itself, while the last two inequalities represent the requirements on the influence of the $N$-th unit on the others.

From the first two inequalities in Eq.(2.34) we have, for the new diagonal element:

$$\begin{aligned}
&w_{NN} > -\sum_{j=1}^{N-1} w_{Nj}x_j(t-1)\\
&w_{NN} < -\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) \Rightarrow && (2.35)\\
&\Rightarrow -\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) < w_{NN} < -\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) \quad . && (2.36)
\end{aligned}$$

We now show that it is possible to construct $w_{ij}$ in such a way that the last inequality is satisfied.

We take $w_{Nj} = -x_j(2^{N-1})|w_{Nj}|$ with $|w_{Nj}| \neq 0,\ \forall\, j \in \{1,\dots,N-1\}$ and we find

$$\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) < w_{NN} < \sum_{j=1}^{N-1} |w_{Nj}| \quad . \tag{2.37}$$

The consistency condition $\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) < \sum_{j=1}^{N-1} |w_{Nj}|$ is always satisfied since to have an equality we would need that $\exists t \in \{1,\dots,2^{N-1}\}$ such that

$$x_j(2^{N-1})x_j(t-1) = 1, \qquad j \in \{1,\dots,N-1\} \quad , \tag{2.38}$$

but this is not possible due to the structure of $X^\star$. The case in which the lower bound in Eq.(2.37) is the closest to the upper one is when only one unit is flipped with respect to the state $x_j(2^{N-1})$, for which we obtain

$$\sum_{j=1}^{N-1} |w_{Nj}| - 2\min_j |w_{Nj}| < w_{NN} < \sum_{j=1}^{N-1} |w_{Nj}| \quad . \tag{2.39}$$

Eq.(2.39) gives upper and lower bounds on $w_{NN}$. We notice that $w_{11}$ is not constrained by Eq.(2.39) but only by $w_{11} < 0$.

We now perform a similar analysis on the last two inequalities in Eq.(2.34).

$$\begin{aligned} x_i(1)\, w_{iN} &< -x_i(1) \sum_{j=1}^{N-1} w_{ij} x_j(2^{N-1}) \\ x_i(t)\, w_{iN} &> -x_i(t) \sum_{j=1}^{N-1} w_{ij} x_j(t-1) \quad . \end{aligned} \tag{2.40}$$

Since the right hand side of the first equation is negative due to the inductive hypothesis and since $x_i(1) = 1$ due to the way the sequence is devised, we need

$$w_{iN} = -|w_{iN}|$$

$$|w_{iN}| > \sum_{j=1}^{N-1} w_{ij} x_j(2^{N-1}) > 0 \tag{2.41}$$

$$x_i(t)|w_{iN}| < x_i(t) \sum_{j=1}^{N-1} w_{ij} x_j(t-1) \quad . \tag{2.42}$$

The first inequality gives us a lower bound to the value of $|w_{iN}|$, while we can derive an upper bound from the second one.

For all $i$, we can divide the time interval into the time point in which $x_i(t) = 1$ from those in which $x_i(t) = -1$. If $x_i(t) = -1$ the inequality is satisfied since the left hand side is negative while the right hand side is positive due to the inductive hypothesis, Eq.(2.26). If $x_i(t) = 1$ we have

$$|w_{iN}| < \sum_{j=1}^{N-1} w_{ij} x_j(t-1), \qquad t \in \{2, \ldots, 2^{N-1}\}, \text{ where } x_i(t) = 1 \quad . \tag{2.43}$$

Therefore the upper bound is

$$|w_{iN}| < \min_{t \in \mathcal{T}_i^+(2, 2^{N-1})} \left[ \sum_{j=1}^{N-1} w_{ij} x_j(t-1) \right] \quad . \tag{2.44}$$

For the $w_{iN}$ to exist, we need the lower bound Eq.(2.41) and the upper bound Eq.(2.44) to be consistent, i.e.

$$\sum_{j=1}^{N-1} w_{ij} x_j(2^{N-1}) < \min_{t \in \mathcal{T}_i^+(2, 2^{N-1})} \left[ \sum_{j=1}^{N-1} w_{ij} x_j(t-1) \right] \quad , \tag{2.45}$$

which is ensured by the weight constrains that are part of the inductive hypothesis, Eq.(2.27).

**The inductive step on the weight constraints requires tighter bounds on the weights**

We now prove that Eq.(2.27) holds true given the inductive hypothesis.

We write the left hand side of Eq.(2.27) as

$$\sum_{j=1}^{N} w_{ij} x_j(2^N) = \sum_{j=1}^{N-1} w_{ij} x_j(2^N) + w_{iN} x_N(2^N) \quad . \tag{2.46}$$

As before, we treat the case $i = N$ and $i \in \{1, \dots, N-1\}$ separately.

For $i = N$ we have to ensure that

$$\sum_{j=1}^{N-1} w_{Nj} x_j(2^N) + w_{NN} x_N(2^N) < \min_{t \in \mathscr{T}_N^+(2,2^N)} \left[ \sum_{j=1}^{N-1} w_{Nj} x_j(t-1) + w_{NN} x_N(t-1) \right]. \tag{2.47}$$

Using the structure of $w_{ij}$ obtained previously and the properties of $X^\star$, we can rewrite this inequality as

$$\begin{aligned}
&\sum_{j=1}^{N-1} |w_{Nj}| - w_{NN} < \min_{t \in \{2,\dots,2^{N-1}\}} \left[ -\sum_{j=1}^{N-1} |w_{Nj}| x_j(2^{N-1}) x_j(t-1) \right] + w_{NN} \\
\Rightarrow\ & w_{NN} > \frac{1}{2} \left\{ \sum_{j=1}^{N} |w_{Nj}| - \min_{t \in \{2,\dots,2^{N-1}\}} \left[ -\sum_{j=1}^{N-1} |w_{Nj}| x_j(2^{N-1}) x_j(t-1) \right] \right\} \\
\Rightarrow\ & w_{NN} > \frac{1}{2} \left\{ \sum_{j=1}^{N-1} |w_{Nj}| + \max_{t \in \{2,\dots,2^{N-1}\}} \left[ \sum_{j=1}^{N-1} |w_{Nj}| x_j(2^{N-1}) x_j(t-1) \right] \right\}.
\end{aligned} \tag{2.48}$$

Following the same reasoning used in the previous section for the lower bound on the diagonal elements (after Eq.(2.37)), we rewrite the last bound as

$$w_{NN} > \sum_{j=1}^{N-1} |w_{Nj}| - \min_j |w_{Nj}| \quad . \tag{2.49}$$

This expression gives a stricter lower bound for the diagonal elements of the weight matrix. The bounds then read

$$\sum_{j=1}^{N-1} |w_{Nj}| - \min_j |w_{Nj}| < w_{NN} < \sum_{j=1}^{N-1} |w_{Nj}| \quad . \tag{2.50}$$

We now consider the case $i \in \{1, \dots, N-1\}$. We can rewrite the left hand side of Eq.(2.46) as

$$\sum_{j=1}^{N-1} w_{ij} x_j(2^N) + w_{iN} x_N(2^N) = \sum_{j=1}^{N-1} w_{ij} x_j(2^N) + |w_{iN}|, \quad . \tag{2.51}$$

Then we rewrite the right hand side of Eq.(2.27) as:

$$\begin{aligned}
&\min_{t\in\mathscr{T}_i^+(2,2^N)}\left[\sum_{j=1}^{N} w_{ij}x_j(t-1)\right] \\
&= \min\Bigg\{\min_{t\in\mathscr{T}_i^+(2,2^{N-1}+1)}\left[\sum_{j=1}^{N} w_{ij}x_j(t-1)\right], \\
&\qquad \min_{t\in\mathscr{T}_i^+(2^{N-1}+2,2^N)}\left[\sum_{j=1}^{N} w_{ij}x_j(t-1)\right]\Bigg\} \\
&= \min\Bigg\{\min_{t\in\mathscr{T}_i^+(2,2^{N-1}+1)}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) - |w_{iN}|\right], \\
&\qquad \min_{t\in\mathscr{T}_i^+(2^{N-1}+2,2^N)}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + |w_{iN}|\right]\Bigg\} \quad .
\end{aligned} \tag{2.52}$$

First, we suppose that the second term is the minimum. Therefore in order to prove Eq.(2.27) we need to show that the following inequality holds:

$$\begin{aligned}
&\sum_{j=1}^{N-1} w_{ij}x_j(2^N) + |w_{iN}| < \min_{t\in\mathscr{T}_i^+(2^{N-1}+2,2^N)}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1)\right] + |w_{iN}| \\
&\Rightarrow -\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) < \min_{t\in\mathscr{T}_i^+(2^{N-1}+2,2^N)}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1)\right] \quad .
\end{aligned} \tag{2.53}$$

The terms inside the minimum operator on the right hand side are all positive since we are considering only terms that lead to $x_i(t) = 1$ and because of the inductive hypothesis on linear separability, as can be seen by performing the substitution $t' = t - 2^{N-1}$. For the same inductive hypothesis the left hand side is negative. Therefore this inequality is always satisfied and it does not bring any additional requirements on the weights.

We now consider the case in which the first term in Eq.(2.52) is the minimum. We require that the following inequality holds:

$$\begin{aligned}
&\sum_{j=1}^{N-1} w_{ij}x_j(2^N) + |w_{iN}| < \min_{t\in\mathscr{T}_i^+(2,2^{N-1})}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1)\right] - |w_{iN}| \\
&|w_{iN}| < \frac{1}{2}\left\{\min_{t\in\mathscr{T}_i^+(2,2^{N-1})}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1)\right] + \sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1})\right\} \quad .
\end{aligned} \tag{2.54}$$

Note that in the time range of the minimum operator we could remove the time point $t = 2^{N-1}+1$ since we consider only $x_i(t) = 1$ and $x_i(2^{N-1}+1) = -1 \quad \forall i \in \{1,\dots,N-1\}$. We also exploited again the symmetry of $X^\star$ (line 7 of Algorithm 1), i.e. $x_j(2^N) = -x_j(2^{N-1})$ for $j < N$.

Eq.(2.54) gives us a new stricter upper bound on $|w_{iN}|$. Finally we need to show that this bound

is consistent with the lower one, i.e.

$$\sum_{j=1}^{N-1} w_{ij} x_j(2^{N-1}) < \frac{1}{2}\left\{ \min_{t\in\mathcal{T}_i^+(2,2^{N-1})} \left[ \sum_{j=1}^{N-1} w_{ij} x_j(t-1) \right] + \sum_{j=1}^{N-1} w_{ij} x_j(2^{N-1}) \right\}, \tag{2.55}$$

which can be rewritten as

$$\sum_{j=1}^{N-1} w_{ij} x_j(2^{N-1}) < \min_{t\in\mathcal{T}_i^+(2,2^{N-1})} \left[ \sum_{j=1}^{N-1} w_{ij} x_j(t-1) \right] \quad , \tag{2.56}$$

which is ensured by the inductive hypothesis on Eq.(2.27) □

## Acknowledgments

SPM was supported by the Swiss National Science Foundation, grant 200020_147200. JB was supported by the European Research Council, grant agreement 268 689.

## 2.5 Author contributions

SPM and JB conceived the study. SPM worked out the proof and wrote the simulation code, with the help of JB. SPM, WG and JB wrote the manuscript.

# 3 Dynamics of recurrent rate networks with adaptation

This chapter presents research carried out in collaboration with Tilo Schwalger and Wulfram Gerstner.

## 3.1 Introduction

History-dependent phenomena are ubiquitous in neuronal systems and are supported by multiple biophysical mechanisms that operate on different timescales. Among those, spike-frequency adaptation (SFA) has received great attention. Being present in neurons at all stages of sensory processing, SFA is believed to play a crucial role for efficient coding of external stimuli (Benda and Herz (2003)). Moreover, SFA over multiple timescales represents an efficient solution for information transmission of sensory signals whose statistics change dynamically (Fairhall et al. (2001); Lundstrom et al. (2008); Pozzorini et al. (2013)). Neurons exhibiting SFA are also widespread in highly recurrent networks, such as association cortex or motor cortex. Yet, the effect of SFA on the dynamics of recurrent networks is not well understood.

In the context of single neuron models, SFA shapes the $f-I$ curve (Ermentrout (1998); Richardson et al. (2003)) as well as the higher-order statistics of the output spike train (Schwalger and Lindner (2013)). When spiking neuron models are connected together in a recurrent network, the statistics of the stationary activity need to be computed self-consistently and this is known to be a hard analytical problem in the presence of SFA. One approach to tackle this problem is to use the Fokker-Planck formalism, which combined with linear response theory and with a slow-adaptation approximation, allows to obtain a good description of the statistics of the recurrent dynamics (Richardson (2009)). Adaptation does not only shape the statistics of the stationary state, but could also explain the emergence of population burst (Gigante et al. (2007)). Alternatively, using quasi-renewal theory (Naud and Gerstner (2012)), it is possible to obtain a mean-field description of a homogeneous population of spiking neurons, allowing to predict the role of adaptation in shaping the statistics of input noise (Deger (2014)). Moreover, this approach was extended to systematically include the effect of finite-size effects and their interaction with SFA (Schwalger

et al. (2017)).

The interest for adaptation in recurrent circuits is additionally motivated by neural network models that exploit adaptation to solve particular tasks. For example, adaptation has been proposed to play a role in sequential memory retrieval (Deco and Rolls (2005)), perceptual bistability (Shpiro et al. (2009)) and decision making (Theodoni et al. (2011)). Thanks to its slower dynamics, adaptation is also appealing for tasks that require slow sequential dynamics, for which one could exploit adaptation-induced slow activity propagation (Setareh et al. (2018)). Recently, SFA was shown to have beneficial consequences both for reservoir computing approaches (Nicola and Clopath (2017)) and for spiking neuron-based machine learning architectures (Bellec et al. (2018)). The possible role of adaptation in performing tasks that require slow timescales will be considered in more detail in chapter 5.

In this chapter, we use dynamic mean-field theory (DMFT) (Sompolinsky et al. (1988)) to describe the dynamics of randomly connected networks of rate units with adaptation. While being limited to rate models, DMFT is one of the few approaches that allows the self-consistent computation of the statistics of a recurrent network beyond the fixed-point regime, and it has been used to show the emergence of chaotic dynamics in large rate networks. The formalism that we use in this chapter is described in detail in chapter 4, where we study the general problem of DMFT for a network of $D$-dimensional rate units. Here we apply the results of chapter 4 to the specific case of rate units with adaptation, a two-dimensional instantiation of the $D$-dimensional model discussed in chapter 4. Using this theoretical framework, we are able to show how adaptation stabilizes the dynamics and how it shapes the statistics of the chaotic regime.

## 3.2 Results

### 3.2.1 Microscopic model and dynamical regimes

We are interested in studying the dynamics of a randomly connected recurrent network of rate units that undergo a simplified form of rate adaptation. The rate $\phi(x_i)$ should in fact be interpreted as the deviation of the actual rate $r_i$ from a reference rate $r_0$, i.e. $\phi(x_i) = r_i - r_0$, where $r_0$ could be the long-term average of the rate. Thus, $\phi(x_i)$ can take positive and negative value, but it must be bounded by $\phi(x_0) \geq -r_0$. We describe the system by the following set of differential equations

$$\tau_x \dot{x}_i(t) = -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) - a_i(t) + I_i(t) \tag{3.1}$$

$$\tau_a \dot{a}_i(t) = -a_i(t) + \beta x_i(t) \quad , \tag{3.2}$$

where $J_{ij} \sim \mathcal{N}\left(0, g^2/N\right)$, with $i, j = 1, \ldots, N$ are elements of a random matrix, sampled i.i.d. from a Gaussian distribution with mean zero and variance $g^2/N$; $\beta$ is a positive parameter that controls the strength of adaptation; $I_i(t)$ is an external stimulus whose statistics will be assumed

to be stationary in what follows; $\phi(\cdot)$ is the gain function of the rate model. In principle it could be left arbitrary, but for the simulation and for the mean-field theory results we use a piecewise linear gain function, given by

$$\phi_{PL}(x) = \begin{cases} -1 & \text{for} \quad x < -1 \\ x & \text{for} \quad -1 < x < 1 \\ 1 & \text{for} \quad x > 1 \end{cases} \quad . \tag{3.3}$$

To simplify our notation, we multiply Eq. (3.2) by $\gamma := \tau_x/\tau_a$ and we rescale time by $\tau_x$, so that the new time variable is unit-less

$$\begin{aligned} \dot{x}_i(t) &= -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) - a_i(t) + I_i(t) \\ \dot{a}_i(t) &= -\gamma a_i(t) + \gamma\beta x_i(t) \quad . \end{aligned} \tag{3.4}$$

We begin by studying the different dynamical regimes of the network. The point $(x_i, a_i) = 0\,, \forall\, i = 1, \ldots, N$ is a fixed point of the system if $\phi(0) = 0$. In what follows, we will consider this condition satisfied and for simplicity we will also assume that $\phi'(0) = 1$. Both conditions are fulfilled for the piecewise linear function of Eq. (3.3). We study the stability of this fixed point, assuming a state vector $(x_i, \ldots, x_N, a_1, \ldots, a_N)$. We need to compute the eigenvalues of the Jacobian of the system at the point $(x_i, a_i) = 0$, i.e.

$$\mathrm{B} := \mathscr{J}|_{(\boldsymbol{x},\boldsymbol{a})=(0,0)} = \begin{pmatrix} -\mathrm{I}_N + \mathrm{J} & -\mathrm{I}_N \\ \gamma\beta\mathrm{I}_N & -\gamma\mathrm{I}_N \end{pmatrix} \quad , \tag{3.5}$$

where each block is a $N$-by-$N$ matrix, J is the connectivity matrix whose elements are $J_{ij}$ and $\mathrm{I}_N$ is the $N$-dimensional identity matrix. The eigenvalues $\lambda_\mathrm{B}$ of B can be expressed as a function of the eigenvalues $\lambda_\mathrm{J}$ of J by solving the general formula in Eq. (4.4) (see also section 3.4.2)

$$\lambda_\mathrm{B}(\lambda_\mathrm{J}) = \frac{1}{2}\left(-1 - \gamma + \lambda_\mathrm{J} \pm \sqrt{(\lambda_\mathrm{J} - 1 + \gamma)^2 - 4\gamma\beta}\right) \quad . \tag{3.6}$$

In the large-$N$ limit, the eigenvalues $\lambda_\mathrm{J}$ are known to be uniformly distributed in a disk in the complex plane, centered at zero and with radius $g$ (Girko (1985)). The critical value of $g$ for which the stability of the fixed point is lost is given by

$$g_c(\gamma, \beta) = \begin{cases} \sqrt{1 - \gamma(\gamma + 2\beta) + 2\sqrt{\gamma^2\beta(2\gamma + 2\beta + 2)}} & \text{if} \quad \beta > \beta_H(\gamma) \\ 1 + \beta \quad \text{if} \quad \beta \leq \beta_H(\gamma) \end{cases} \quad , \tag{3.7}$$

where $\beta_H(\gamma) = -1 - \gamma + \sqrt{2\gamma^2 + 2\gamma + 1}$. Obtaining this result from the eigenvalue formula (Eq. (3.6)) is non obvious. However, one can more easily find the critical value of $g$ from the linear stability analysis of the mean-field theory (see section 3.4.4), and then use Eq. (3.6) to verify

that the expression for $g_c$ given by Eq. (3.7) gives the critical eigenvalue (in the $N \to \infty$ limit). Examples of the eigenvalue spectrum of B are shown in the insets of Fig. 3.1. If $g < g_c(\gamma, \beta)$, the network exhibit a transient dynamics before it settles at the zero fixed-point (Fig. 3.1A,B). From Eq. 3.7 we notice that $g_c(\gamma, \beta) \geq 1$, since both $\gamma$ and $\beta$ are positive. In the limit $\gamma \to 0$ or $\beta \to 0$ we retrieve the same dynamical regime as for the network without adaptation, for which $g_c = 1$ (Sompolinsky et al. (1988)). If the contribution of adaptation to the input of $x$ increases, due to either an increase in $\gamma$ or in $\beta$, the value of $g_c$ also increases (see Fig. 3.2C,D), as expected from the role of adaptation as a source of negative feedback. The introduction of adaptation therefore stabilizes the dynamics of the network.

The bifurcation that characterizes the loss of stability depends on two parameters, viz. the ratio of timescales $\gamma$ and the strength of the adaptation $\beta$. To further characterize the bifurcation at $g = g_c(\gamma, \beta)$, we can study the imaginary part of the critical eigenvalue, i.e. the one with real part equal to zero at $g = g_c(\gamma, \beta)$. It can be shown that, if the parameter $\beta$ that determines the strength of the adaptation has a value $\beta \leq \beta_H(\gamma)$, then the imaginary part of the critical eigenvalue is equal to zero and we have a saddle-node bifurcation at $g = g_c(\gamma, \beta)$. On the other hand, if $\beta > \beta_H(\gamma)$, then the critical eigenvalue is effectively a pair of complex-conjugate, purely imaginary eigenvalues, a signature of a Hopf bifurcation. Therefore, we introduce the curve $\beta = \beta_H(\gamma)$, which separates the positive quadrant of the $\gamma - \beta$ plane in two regions: in one region we have, at the critical value $g_c(\gamma, \beta)$, a saddle-node bifurcation, whereas in the other one we have a Hopf bifurcation (Fig. 3.2A). In the Hopf-bifurcation region, the imaginary part of the critical eigenvalues can be computed analytically and predicts the frequency $f_m$ of low-amplitude oscillations close to the bifurcation, if these are stable. In the finite-$N$ regime, we find numerically that low-amplitude oscillations are stable in the vicinity of the bifurcation. When $N \to \infty$ however, we find that chaotic dynamics onset right above the bifurcation (see 3.2.3). We find

$$\mathrm{Im}(\lambda_{\mathrm{B}}^c) = \sqrt{-\gamma^2 + \sqrt{\beta\gamma^2(\beta + 2\gamma + 2)}} =: 2\pi f_m \quad . \tag{3.8}$$

The frequency is monotonic in $\beta$ but non-monotonic in $\gamma$ (Fig. 3.2B), indicating that a slower adaptation variable (smaller $\gamma$) does not always correspond to slower oscillations. Finally, if $g > g_c(\gamma, \beta)$, the network exhibit self-sustained, irregular fluctuations (Fig. 3.1C,D) that will be characterized in the next sections.

### 3.2.2 Mean-field description in the frequency domain

The dynamics of the $2N$-dimensional dynamical system in Eqs. (3.1,3.2) for large $N$ is too high-dimensional to be studied at the microscopic level. In contrast, using dynamic mean-field theory (Sompolinsky et al. (1988)), we can find properties of the network dynamics that are independent of the specific connectivity realization. The mean-field approximation, valid in the large-$N$ limit of the randomly connected network (see Fig. 3.3D), can be obtained by replacing the recurrent input in Eq. (3.1) by a Gaussian process $\eta$, as described in detail in chapter 4 (see also Schücker et al. (2016a); Crisanti and Sompolinsky (2018)). The mean and the autocorrelation

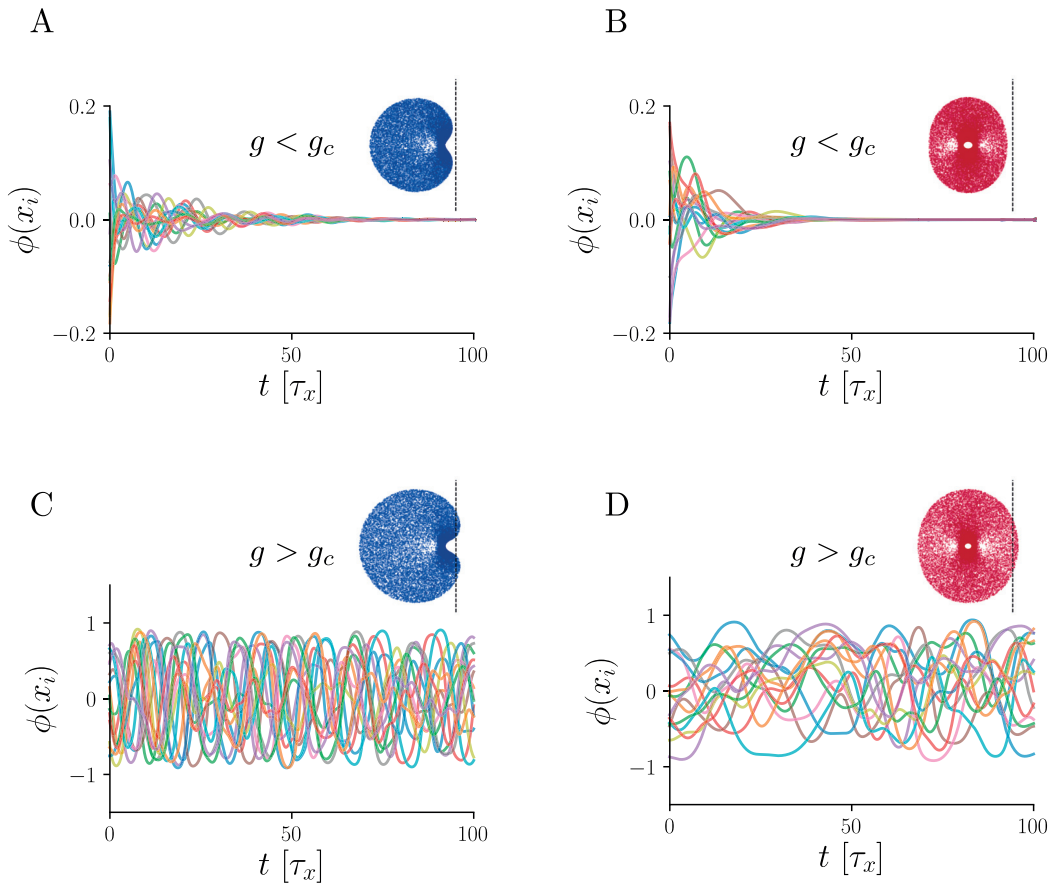



Figure 3.1 – **Dynamical regimes of the network with adaptation.** **A:** Evolution of the rate $\phi(x_i)$ in time. A randomly chosen subset of units is shown, out of $N = 1000$ units. Network parameters are $\gamma = 0.2$, $\beta = 0.5$ and $g = 0.96 g_c(\gamma, \beta)$. For these parameters the network is in the Hopf-bifurcation regime. Inset: Eigenvalue spectrum of the Jacobian at the fixed point, in the complex plane. The dashed line indicates $\mathrm{Re}(\lambda) = 0$. **B:** Same as **A**, but in the saddle-node bifurcation regime, with $\gamma = 1$, $\beta = 0.1$ and $g = 0.96 g_c(\gamma, \beta)$ **C:** Same as **A**, but above the instability, with $\gamma = 0.2$, $\beta = 0.5$ and $g = 1.3 g_c(\gamma, \beta)$. **D:** Same as **B**, but above the instability, with $\gamma = 1$, $\beta = 0.1$ and $g = 1.3 g_c(\gamma, \beta)$.

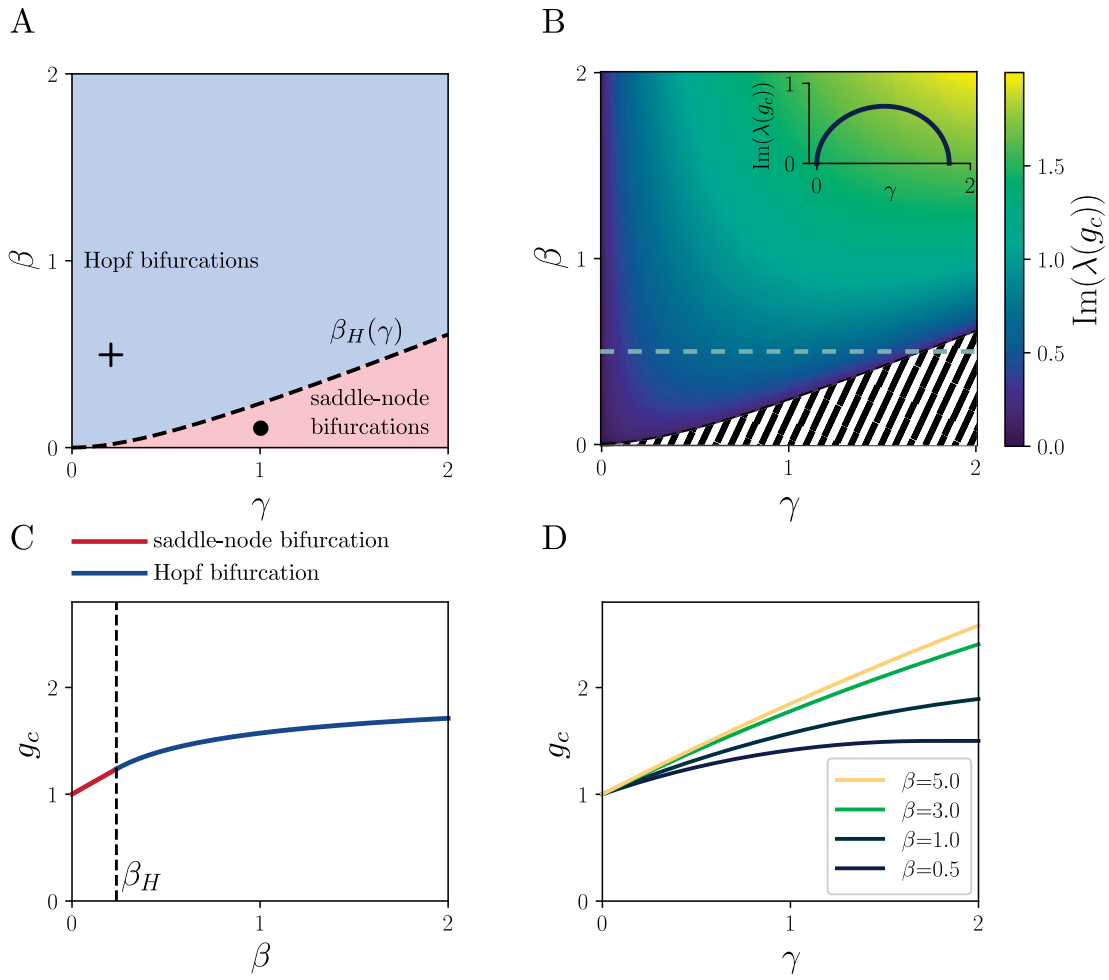



Figure 3.2 – **Stability of the fixed point and local properties. A:** Regions of the $\gamma - \beta$ plane in which for increasing $g$ we encounter a Hopf (blue region) or a saddle-node (red region) bifurcation. The two regions are separated by the curve $\beta_H(\gamma)$ (dashed). Cross and filled circle: parameters used in Fig. 3.1. **B:** Imaginary part of the eigenvalue with zero real part at the instability ($g = g_c$), in the $\gamma - \beta$ plane. The inset shows $\mathrm{Im}(\lambda(g_c))$ plotted against $\gamma$ for $\beta = 0.5$ (dashed gray line). **C:** Critical value of the parameter $g$ plotted against $\beta$, for $\gamma = 1$. The color-code indicates the type of bifurcation encountered when $g = g_c$. **D:** Critical value $g_c$ of the parameter $g$ plotted against $\gamma$, for different values of $\beta$. Larger values of $\beta$ or $\gamma$ stabilize the dynamics with respect to the network without adaptation, which is retrieved by setting $\gamma = 0$ or $\beta = 0$.

of $\eta$ should be matched to the statistics of the input that a neuron receives from the network. The mean-field equations read

$$\dot{x}(t) = -x(t) - a(t) + \eta(t) + I(t) \tag{3.9}$$

$$\dot{a}(t) = -\gamma a(t) + \gamma\beta x(t) \quad , \tag{3.10}$$

with $\langle\eta(t)\rangle = 0$ and $\langle\eta(t+\tau)\eta(t)\rangle = g^2\langle\phi(x(t+\tau))\phi(x(t))\rangle$, i.e. the second-order statistics of $\eta$ and those of $x$ depend on each other and therefore need to be matched self-consistently. We can express the self-consistency condition in the Fourier domain (see section 3.4.3)

$$S_x(f) = |\tilde{\chi}_0(f)|^2 \left(g^2 S_{\phi(x)}(f) + S_I(f)\right) \quad , \tag{3.11}$$

where $\delta(f-f')S_x(f) := \langle\tilde{x}^*(f)\tilde{x}(f')\rangle$ is the power spectral density of $x$, and analogously for $S_{\phi(x)}(f)$. The factor $|\tilde{\chi}_0(f)|^2$ coincides with the linear response function of a *single unit*, defined as $\tilde{\chi}_0(f) = \tilde{x}(f)/\tilde{I}(f)$. Notice that since for a single unit the relationship between $I$ and $x$ is linear, there is no need to consider small $I$ in the definition of $\tilde{\chi}_0(f)$. The factor $|\tilde{\chi}_0(f)|^2$ in Eq. (3.11) can be shown to be equal to (see section 3.4.3)

$$|\tilde{\chi}_0(f)|^2 = \frac{\gamma^2 + (2\pi)^2 f^2}{(2\pi)^4 f^4 + (1+\gamma^2-2\beta\gamma)(2\pi)^2 f^2 + \gamma^2(1+\beta)^2} \quad . \tag{3.12}$$

In order to solve Eq. (3.11), we need to compute $S_{\phi(x)}(f)$ as a functional of $S_x(f)$, which is known to be a hard problem and not possible in general. However, as discussed in section 4.6.2, the effect of the nonlinearity $\phi$ can be evaluated numerically or semi-analytically. Moreover, the transformation can be computed analytically in an integral form, which allows for a much faster computation of $S_{\phi(x)}(f)$ (see section 4.6.2). In the next section we show how the qualitative features of the dynamics of the network in the fluctuating regime can be predicted by the properties of the single unit linear response function $|\tilde{\chi}_0(f)|^2$.

### 3.2.3 Adaptation drives the network in a new chaotic regime

In the mean-field description, the dynamical state of the network is entirely described by the second-order statistics of the Gaussian process $x$, i.e. the power spectral density $S_x(f)$ in the Fourier domain or the autocorrelation $C_x(\tau)$ in time domain. Using the iterative method described in chapter 4, we find that if $g < g_c(\gamma,\beta)$, then $S_x(f) = 0, \forall f$, i.e. the mean-field variable $x$ is constantly equal to zero. This is consistent with the presence of a stable fixed-point at zero and it indicates that, in the large-$N$ limit, the fixed-point solution is the only possible one.

On the other hand, if $g > g_c(\gamma,\beta)$, the mean-field network is characterized by a nonzero, continuous power spectral density. This is an indication that, at the microscopic level, the network is in a chaotic state. However, we stress that a more rigorous proof of chaos would require the computation of the maximum Lyapunov exponent of the network. For the network without adaptation (Sompolinsky et al. (1988)) the chaotic state is always of the same type, i.e. characterized by a

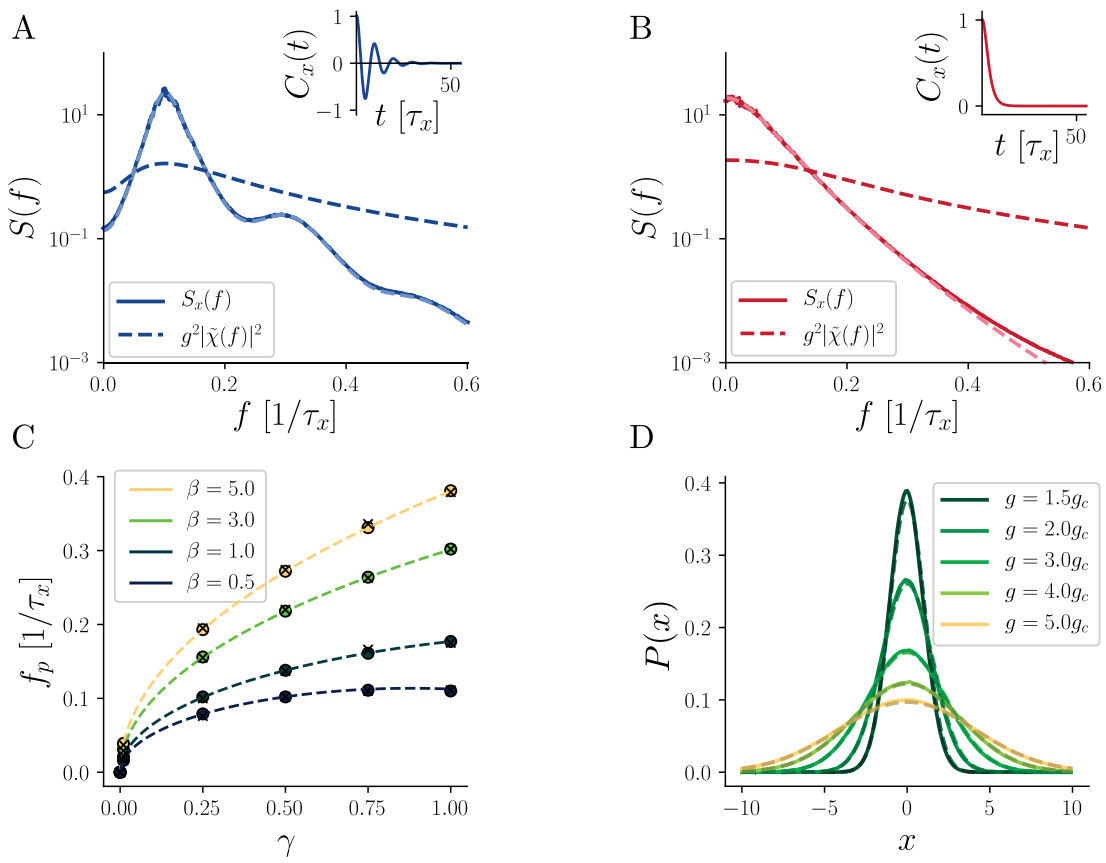



Figure 3.3 – **Dynamical regimes in the mean-field description A:** Power spectral density of the mean-field network (solid line) compared with microscopic simulations (light blue, dashed) for $\gamma = 0.25$, $\beta = 1$ and $g = 2g_c(\gamma, \beta)$. The dashed, dark blue line the power spectral density of a network of independent neurons, driven by white noise with variance $g^2$ and with the same adaptation parameters. Inset: Normalized mean-field autocorrelation $C_x(\tau)$ for the same parameters, plotted against the time lag in units of $\tau_x$. **B:** Same as A, but with $\gamma = 1$, $\beta = 0.1$ and $g = 2g_c(\gamma, \beta)$. The theory deviates from simulations in the tail of the power spectral density, due to numerical errors in the iterative method.
**C:** Maximum-power frequency $f_p$ of the recurrent network plotted against $\gamma$, for different $\beta$. Circles indicate the $f_p$ obtained for the mean-field network, crosses the one measured for microscopic simulations and the dashed lines indicate the prediction based on the single neuron response function $f_0$. Notice that for $\gamma = 0$ all curves are superimposed at $f_p = 0$. **D:** Distributions $P(x)$ of the activation $x$ from microscopic simulation ($N = 2000$, solid lines) and theoretical prediction (dashed lines). The adaptation parameter were $\gamma = 0.25$ and $\beta = 1$.

monotonically-decaying autocorrelation or, equivalently, by a power spectral density dominated by low frequencies. In contrast, we find that in the presence of adaptation the network can be in two qualitatively different chaotic regimes. For very weak and/or fast adaptation, the chaotic fluctuations are qualitatively the same as for the network without adaptation (Fig. 3.3B). We refer to this regime as to the non-resonant regime. On the other hand, for strong and/or slow adaptation, the mean-field network settles in a new regime, that we refer to as *resonant* regime, characterized by an autocorrelation that decays to zero via damped oscillations or, equivalently, by a power spectral density that exhibits a resonance band around a nonzero resonance frequency $f_p$ (Fig. 3.3A). The decaying autocorrelation function and the continuous power spectral density are an indication that also this regime corresponds to microscopic chaos. This new dynamical state, that we refer to as *resonant chaos*, is qualitatively different from the one of the non-resonant regime and from the one of the non-adaptive network, and it is unique to the network with adaptation.

Strikingly, whether the network settles in the resonant or in the non-resonant regime can be predicted purely based on the single-unit adaptation properties. More precisely, if $\beta < \beta_H(\gamma)$, then the linear response function $|\tilde{\chi}_0(f)|^2$ is monotonically decreasing with the frequency $f$ (see section 3.4.3), which is typical of a low-pass device (Fig. 3.3B). This behavior is reflected in the power spectral density of the spontaneous activity of the recurrent network, which turns out to be dominated by low frequencies. This corresponds to the non-resonant regime discussed above. In contrast, if $\beta > \beta_H(\gamma)$, then $|\tilde{\chi}_0(f)|^2$ has a maximum at a nonzero frequency $f_0 = \frac{1}{2\pi}\sqrt{-\gamma^2 + \sqrt{\beta\gamma^2(\beta+2\gamma+2)}}$, which is typical of the response of a band-pass filter (Fig. 3.3A). The frequency $f_0$ matches the imaginary part of the critical eigenvalue at the Hopf bifurcation, given in Eq. (3.8) (the reason of this match is detailed in section 3.4.4). The single neuron linear response characteristics are qualitatively preserved in the fluctuating activity of the recurrent network, which also exhibit a power spectral density dominated by a nonzero frequency $f_p$. Interestingly, the resonance frequency is not affected by the introduction of recurrent connections, since we find that $f_p = f_0$ (Fig. 3.3C). We notice that this result is consistent with the fixed point stability analysis, since the resonant and non-resonant regimes are matched to the regions in which we observe Hopf or saddle-node bifurcations, respectively.

### 3.2.4 Recurrent connections increase the coherence of the oscillations

While the resonance frequency in the resonant regime depends solely on the single-neuron properties, the introduction of recurrent connections does influence how coherent the resonant behavior is, i.e. the width of the resonance band. The narrower the resonance band, the more coherent the oscillatory behavior will be. To quantify the increase of coherence of the oscillations, we study the total area under the normalized autocorrelation, in absolute value (Fig. 3.4A), i.e.

$$s = \int_0^\infty \left| \frac{C_x(\tau)}{C_x(0)} \right| d\tau \quad . \tag{3.13}$$

As expected, the area $s$ diverges when approaching the criticality, i.e. when $g \to g_c(\gamma, \beta)$ (Fig. 3.4B), since the dynamics approach regular oscillations. Changes in the adaptation parameters

affect this behavior only by shifting the values of $s$. Interestingly, the total area $s$ does not depend strongly on the time constant of adaptation, since if we increase $\tau_a$ ($\gamma \to 0$), $s$ saturates around few tens of $\tau_x$ independently of how slow adaptation is (Fig. 3.4C). This behavior seems to be inherited from the single unit model, since also the network of independent neurons driven by white noise exhibits the same saturation of the total area.

To provide a more complete picture, we also study the correlation time, defined as

$$t_c = \frac{\int_0^\infty \tau\,|C_x(\tau)|\,d\tau}{\int_0^\infty |C_x(\tau)|\,d\tau} \quad . \tag{3.14}$$

While for relatively fast adaptation the introduction of recurrent connections also yields an increase of the correlation time $t_c$ (Fig. 3.4D), for slow adaptation we observe the opposite effect. This is due to the fact that in this regime the correlation time of the single unit driven by white noise is dominated by the long tail of the autocorrelation. The introduction of recurrent connections increases the oscillatory component, giving a larger "weight" to the short time lags and therefore decreasing $t_c$.

### 3.2.5 Response of the recurrent network to an external input

In this section, we go beyond the study of the spontaneous activity of the network by considering its response to an external drive. An interesting class of external drives are oscillatory signals, since we can study the locking properties of the network depending on the frequency and amplitude of the signal. Similarly to Rajan et al. (2010), we provide oscillatory input to each unit in the microscopic network, randomizing the phase

$$I_i(t) = A_I \cos\left(2\pi f_I t + \theta_i\right) \quad , \tag{3.15}$$

where $\theta_i \sim \mathrm{U}(0, 2\pi)$. The corresponding power spectral density of the input is given by $S_I(f) = A_I^2/4\left(\delta(f - f_I) + \delta(f + f_I)\right)$. Thanks to the phase randomization, the network still reaches a stationary state and the mean $\langle x(t)\rangle$ remains at zero. In Fig. 3.5A we see an example of how the presence of the input affects the dynamics of the mean-field network, quantified by the second-order statistics here summarized by the power spectral density. A sharp peak at the driving frequency $f_I$ and multiples thereof is elicited by the external input, while the nearby frequencies are suppressed (notice the log-scale in Fig 3.5A). For $f_I > f_p$, as in the example, the relative peaks of the spectrum are slightly shifted toward larger values. The opposite happens if $f_I < f_p$. Notice that both this shift and the suppression of spontaneous activity are nonlinear effects due the recurrent dynamics. As an additional nonlinear effect, the network activity also exhibits harmonics of the external input.

The formation of a sharp peak together with the suppression of other modes is an indication that at the microscopic level the network is driven towards a limit cycle while chaotic activity is suppressed. We quantify this effect by defining the chaos-suppression coefficient, similarly to

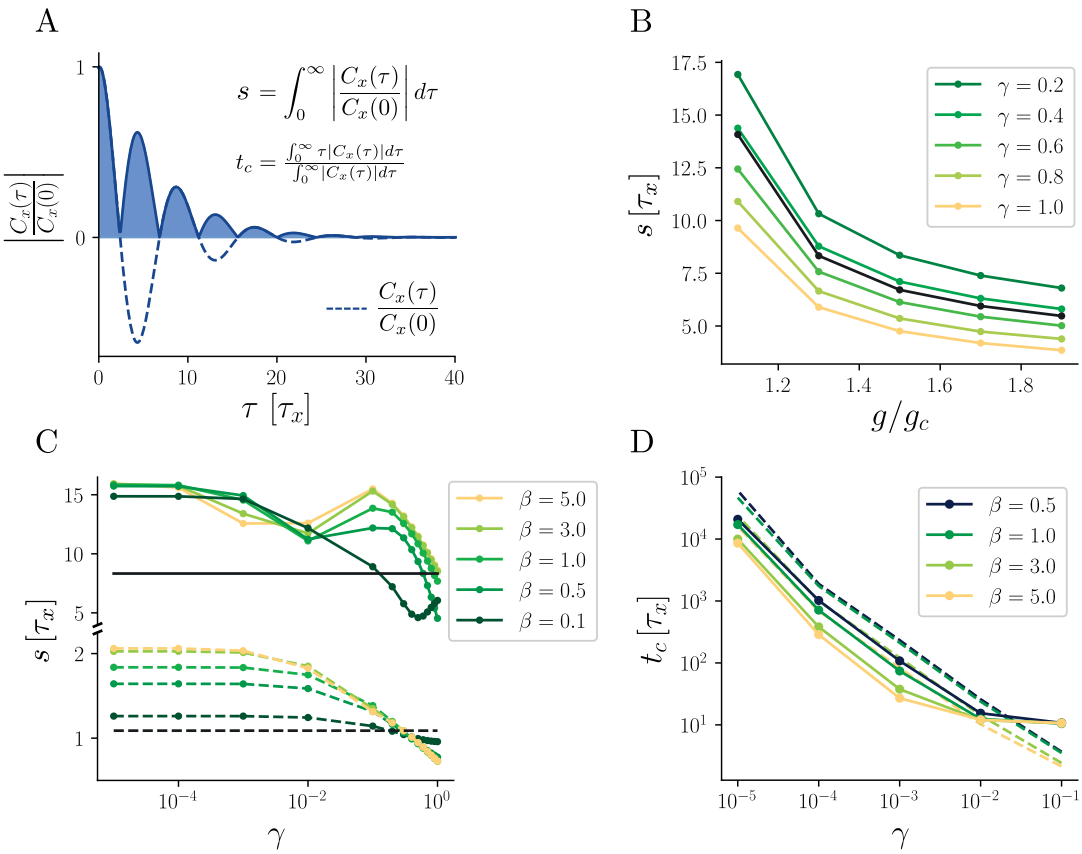



Figure 3.4 – **Correlation time and effect of recurrent connections. A:** Example autocorrelation and definition of relevant measures. **B:** Total area under the autocorrelation $s$ of the recurrent network as a function of $g/g_c$, for different values of $\gamma$ and fixed $\beta = 1$. As expected, $s$ increases when $g$ approaches the critical value. The black line indicates $s$ for the network without adaptation. **C:** Total area under the autocorrelation $s$ as a function of $\gamma$ for different values of $\beta$ and fixed $g = 1.3g_c$, both for the recurrent network (solid lines) and for the a single neuron driven by white noise (dashed lines). Black lines (solid and dashed) indicate the behavior of the network without adaptation (recurrent and independent neurons, respectively). Notice that the behavior of $s$ for $\beta = 0.1$ is qualitatively different from the other cases, since in this case for larger $\gamma$ the network is in the non-resonant regime. **D:** Correlation time $t_c$ as a function of $\gamma$, for different values of $\beta$, for both the recurrent network (solid lines) and for a single unit driven by white noise (dashed lines).

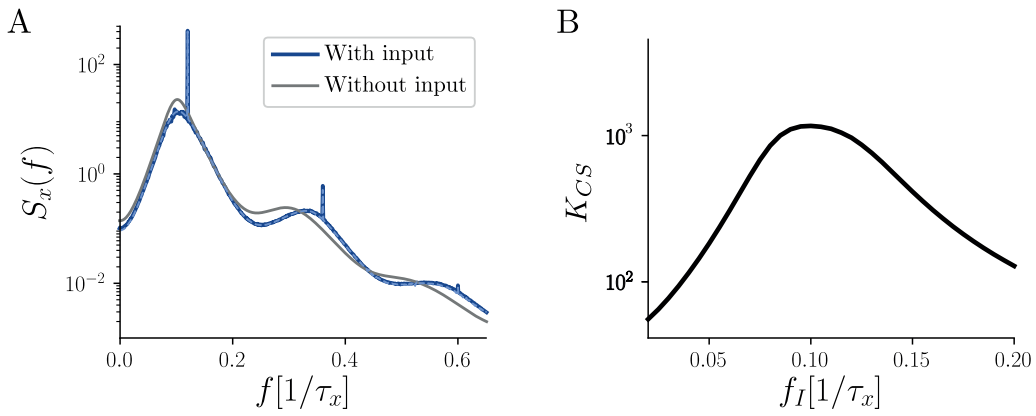



Figure 3.5 – **Response of the mean-field network to an oscillatory input. A:** Effect of the external input on the power spectral density $S_x(f)$. In the example, $\gamma = 0.25$, $\beta = 1$, $g = 2g_c(\gamma, \beta)$, $f_I = 0.12$ and $A_I = 5 \cdot 10^{-1}$. Simulation (solid blue) and theory (dashed blue) are superimposed. **B:** Chaos-suppression coefficient ($K_{CS}$, solid line), defined as in section 3.2.5, for different values of the input frequency $f_I$. Network parameters: $\gamma = 0.25$, $\beta = 1$, $g = 2g_c(\gamma, \beta)$ and $A_I = 5 \cdot 10^{-1}$.

(Rajan et al. (2010)), as the amplitude of the power spectral density at the peak divided by the mean power at the other frequencies, i.e.

$$K_{CS} = \frac{S_x(f_I)}{[S_x(f)]_{f \neq f_I}} \quad , \tag{3.16}$$

where the square brackets indicate the average over frequencies, excluding the input frequency $f_I$. We find that chaos suppression is more effective with input frequencies that are close to the resonance frequency of the network $f_p$ (Fig. 3.5B).

## 3.3 Discussion

We showed that adaptation stabilizes the dynamics of a recurrent network of rate units, since the transition from a fixed-point regime to a fluctuating regime happens at $g = g_c \geq 1$, i.e. for higher coupling strength than for the network without adaptation. Above the criticality and for slow adaptation, the dynamics settles in a state of resonant chaos that, unlike the chaotic activity of networks of rate units without adaptation, is dominated by a nonzero resonance frequency. Surprisingly, we observe empirically that the position of the resonance frequency can be computed purely based on the single unit properties and it is therefore independent of the connectivity strength $g$. Consistent with this result, the eigenvalue spectrum of the Jacobian at the fixed point indicates the appearance of a Hopf bifurcation. On the other hand, recurrent connections increase the coherence of the oscillations and therefore influence the correlation time. Indeed, as it is typical of critical behavior, the correlation time in the chaotic phase diverges when approaching the criticality. In the presence of adaptation, this happens because the oscillations

get more coherent and the system approaches a limit cycle.

It is interesting to observe that, while coherence of the oscillations is increased by recurrent connections, in the slow adaptation regime the correlation time is decreased by the introduction of recurrent connections, due to the increased correlation at short time lags. Despite this reduction, the correlation time increases with the adaptation timescale, and this is particularly interesting when considering the evidences for a beneficial role of adaptation in tasks requiring memory over long time lags (Nicola and Clopath (2017); Bellec et al. (2018)). The relation between correlation time and performance however is much more complex and requires further investigation.

The analysis was carried out using a linear model of adaptation. In spiking neurons, SFA is believed to have at least two sources (Pozzorini et al. (2013)): sub-threshold effects, e.g. voltage-dependent hyperpolarizing currents (Richardson et al. (2003)) and spike-triggered effects, such as sodium channel inactivation (Fuortes and Mantegazzini (1962); Geisler and Goldberg (1966)). The type of adaptation we considered is therefore more closely linked to sub-threshold adaptation in spiking neurons. Close to the criticality however, most of the units operate in the linear part of the transfer function $\phi$ and therefore we do not expect strong deviations if considering nonlinear adaptation.

## 3.4 Methods

### 3.4.1 Numerical methods

All numerical procedures were carried out using custom code written in Julia (Bezanson et al. (2017)). When using the sampling-based version of the iterative method, we considered the number of samples $M = 10000$. Network simulations were carried out using the fourth-order Runge-Kutta numerical integration method, with a time step $dt = 0.1\tau_x$. For Fig. 3.1 we used $N = 1000$, for Fig. 3.3A,B we used $N = 3000$, for Fig. 3.3C we used $N = 10000$, for Fig. 3.3D we used $N = 2000$ and for Fig. 3.5A we used $N = 2000$.

### 3.4.2 Calculation of the eigenvalue spectrum

In this section we provide the details of the derivation of the transformation from the eigenvalues of J to the eigenvalues of B. For clarity, we rewrite here B, i.e. the Jacobian at the fixed point

$$\mathrm{B} = \begin{pmatrix} -\mathrm{I}_N + \mathrm{J} & -\mathrm{I}_N \\ \gamma\beta\mathrm{I}_N & -\gamma\mathrm{I}_N \end{pmatrix} \quad , \tag{3.17}$$

where each block is a $N$-by-$N$ matrix, $J$ is the connectivity matrix whose elements are $J_{ij}$ and $I_N$ is the $N$-dimensional identity matrix. To find the eigenvalues of B, we need to compute the determinant of $\mathrm{B} - \lambda I_{2N}$, which is in the form $\begin{pmatrix} \mathrm{C}_{11} & \mathrm{C}_{12} \\ \mathrm{C}_{21} & \mathrm{C}_{22} \end{pmatrix}$. For block matrices of this form,

if $C_{21}$ and $C_{22}$ commute i.e. $C_{21}C_{22} = C_{22}C_{21}$, the following formula for the determinant holds (Silvester (2000))

$$\det\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \det\left(C_{11}C_{22} - C_{12}C_{21}\right) \quad . \tag{3.18}$$

Since in B both $C_{21}$ and $C_{22}$ are proportional to the identity matrix $I_N$, the commutation condition is satisfied, so that we find from Eq. (3.18)

$$\det\begin{pmatrix} (-1-\lambda_B)I_N + J & -I_N \\ \gamma\beta I_N & (-\gamma-\lambda_B)I_N \end{pmatrix} = \det\left(\left((-1-\lambda_B)(-\gamma-\lambda_B)+\gamma\beta\right)I_N + (-\gamma-\lambda_B)J\right) \quad . \tag{3.19}$$

We now rewrite the last expression in the form $\det\left(J - \lambda_J(\lambda_B) I_N\right) = 0$, that gives us $\lambda_J$ as a function of $\lambda_B$ by definition. Then, we invert $\lambda_J(\lambda_{_B})$ and get

$$\lambda_B(\lambda_J) = \frac{1}{2}\left(-1-\gamma+\lambda_J \pm \sqrt{(\lambda_J - 1 + \gamma)^2 - 4\gamma\beta}\right) \quad . \tag{3.20}$$

We notice that for every eigenvalue $\lambda_J$ we obtain two eigenvalues $\lambda_{_B}$, corresponding to the two choices of the sign in Eq. (3.20). This is consistent with the fact that the dimensionality of B is twice the dimensionality of J.

### 3.4.3 Self-consistent equation for the power spectral density and properties of $|\tilde{\chi}_0(f)|^2$

In this section, we show how we derived Eq. (3.11). First, we Fourier-transform the mean field equations (Eq. (3.9,3.10)), to get

$$\tilde{x}(f) = \frac{\gamma + 2\pi i f}{-(2\pi)^2 f^2 + 2\pi i(\gamma+1)f + \gamma(1+\beta)}\left(\tilde{\eta}(f) + \tilde{I}(f)\right) \tag{3.21}$$

$$\tilde{a}(f) = \frac{\gamma\beta}{\gamma + 2\pi i f}\tilde{x}(f) \quad . \tag{3.22}$$

From this expression we can find the power spectral density of $x$, by multiplying Eq. (3.21) by $\tilde{x}^*$ and averaging over the statistics of the input $\eta$. We obtain

$$S_x(f) = \frac{\gamma^2 + (2\pi)^2 f^2}{(2\pi)^4 f^4 + (1+\gamma^2-2\beta\gamma)(2\pi)^2 f^2 + \gamma^2(1+\beta)^2}\left(S_\eta(f) + S_I(f)\right) \quad , \tag{3.23}$$

where we can recognize the factor $|\tilde{\chi}_0(f)|^2$, defined in Eq. (3.12).

We stress that $|\tilde{\chi}_0(f)|^2$ is the linear response function of a single unit or, equivalently, of a network of uncoupled neurons. To see this, it is sufficient to set $\tilde{\eta}(f) = 0$, which gives us the case of

uncoupled neurons. From Eq. (3.21) we get

$$\tilde{x}(f) = \frac{\gamma + 2\pi i f}{-(2\pi)^2 f^2 + 2\pi i(\gamma+1)f + \gamma(1+\beta)} \tilde{I}(f) =: \tilde{\chi}_0(f)\tilde{I}(f) \quad . \tag{3.24}$$

It is trivial to verify that the definition of $\tilde{\chi}_0(f)$ is consistent with Eq. (3.12).

To study whether the single unit is in the resonant or non-resonant regime, we study the position of the maximum of $|\tilde{\chi}_0(f)|^2$. The derivative of $|\tilde{\chi}_0(f)|^2$ has five zeros, symmetric with respect to $f = 0$, given by

$$f \in \left\{ 0, \pm\frac{1}{2\pi}\sqrt{-\gamma^2 \pm \sqrt{\beta\gamma^2(\beta+2\gamma+2)}} \right\} \quad . \tag{3.25}$$

Among these, the only three that can take real values are $\left\{0, \pm\frac{1}{2\pi}\sqrt{-\gamma^2 + \sqrt{\beta\gamma^2(\beta+2\gamma+2)}}\right\}$. To have a non-monotonic $|\tilde{\chi}_0(f)|^2$, we require that

$$\frac{1}{2\pi}\sqrt{-\gamma^2 + \sqrt{\beta\gamma^2(\beta+2\gamma+2)}} =: f_0 \in \mathbb{R} \quad , \tag{3.26}$$

which is satisfied if

$$\gamma^2 < \sqrt{\beta\gamma^2(\beta+2\gamma+2)} \quad . \tag{3.27}$$

By solving this inequality with respect to $\beta$, we find the condition $\beta > \beta_H(\gamma)$.

### 3.4.4 Mean-field derivation of the full linear response function at the fixed point

In this section, we show how to compute the mean-field approximation of the full linear response function of the network at the fixed point in zero. For a similar derivation, see Kadmon and Sompolinsky (2015). Starting by the equations of the microscopic network (Eq. (3.1,3.2)), we find a set of differential equations for the linear response functions $\chi^x_{ik}(t,t') := \frac{\delta x_i(t)}{\delta h_k(t')}$ and $\chi^{ax}_{ik}(t,t') := \frac{\delta a_i(t)}{\delta h_k(t')}$, where $h_k(t')$ is a small perturbation given to the variable $x_k$ at time $t'$, i.e. $I_k(t) = h_k(t)\delta(t-t')$. We obtain

$$(\partial_t + 1)\chi^x_{ik}(t,t') = \phi'(0)\sum_{j=1}^{N} J_{kj}\chi^x_{kj}(t,t') - \chi^{ax}_{ik}(t,t') + \delta^{ik}\delta(t-t') \tag{3.28}$$

$$(\partial_t + \gamma)\chi^{ax}_{ik}(t,t') = \gamma\beta\chi^{ax}_{ik}(t,t') \quad . \tag{3.29}$$

Since we are at the fixed point, we consider only the time difference $\tau := t - t'$. Moreover, all the coefficients are time independent so that we can easily Fourier-transform them and solve for

$\tilde{\chi}^x_{ik}(t,t')$

$$\left(2\pi i f + 1 + \frac{\gamma\beta}{2\pi i f + \gamma}\right)\tilde{\chi}^x_{ik}(f) = \sum_{j=1}^{N} J_{kj}\tilde{\chi}^x_{kj}(f) + \delta^{ik} \quad , \tag{3.30}$$

where we set $\phi'(0) = 1$ for simplicity. The factor in parenthesis on the left-hand side is the inverse of the single-unit linear response defined in section 3.4.3. We now multiply this last equation by its complex conjugate to obtain

$$\begin{aligned} |\tilde{\chi}^x_{ik}(f)|^2 =& |\tilde{\chi}_0(f)|^2 \sum_{j,j'=1}^{N} J_{kj}J_{kj'}\tilde{\chi}^x_{kj}(f)\left(\tilde{\chi}^x_{kj'}(f)\right)^* \\ &+ \delta^{ik}|\tilde{\chi}_0(f)|^2\left(1 + \sum_{j=1}^{N} J_{kj}\tilde{\chi}^x_{kj}(f) + \sum_{j=1}^{N} J_{kj}\left(\tilde{\chi}^x_{kj}(f)\right)^*\right) \quad . \end{aligned} \tag{3.31}$$

Finally, we average over the quenched disorder and get

$$|\tilde{\chi}^x_{ik}(f)|^2 = g^2|\tilde{\chi}_0(f)|^2|\tilde{\chi}^x_{ik}(f)|^2 + |\tilde{\chi}_0(f)|^2 \quad , \tag{3.32}$$

from which we can solve for the mean-field linear response function

$$|\tilde{\chi}^x_{ik}(f)|^2 = \frac{|\tilde{\chi}_0(f)|^2}{1 - g^2|\tilde{\chi}_0(f)|^2} \quad . \tag{3.33}$$

From this equation we can conclude that in the $N \to \infty$ limit the fixed point is stable if $g^2|\tilde{\chi}_0(f)|^2 < 1 \ \forall f$. By imposing this condition for $f = f_0$ in the resonant regime (or $f = 0$ in the non-resonant one), we find the result for $g_c$ given by Eq. (3.7).

### 3.4.5 Spectral coherence of the mean-field network in the presence of an external input

We repeat here the definition of the spectral coherence given in section 3.2.5

$$\Gamma(f) := \frac{|S_{xI}(f)|^2}{S_x(f)S_I(f)} \quad . \tag{3.34}$$

The cross-spectrum $S_{xI}(f)$ is given by

$$S_{xI}(f) = \tilde{\chi}_0(f)S_I(f) \quad . \tag{3.35}$$

On the other hand, one can express $S_x(f)$ using Eq. (3.11) as $S_x = |\tilde{\chi}_0(f)|^2\left(g^2 S_{\phi(x)}(f) + S_I(f)\right)$. Using these two expressions we can write the spectral coherence as

$$\Gamma(f) = \frac{S_I(f)}{g^2 S_{\phi(x)}(f) + S_I(f)} \quad . \tag{3.36}$$

### 3.4.6 Time domain approach to the mean-field theory

For completeness, in this section we write down the explicit equation for the autocorrelation in time domain. Starting from Eq. (3.11), transforming back to time domain and defining $\Delta(\tau) = \langle x(\tau)x(0)\rangle$ the autocorrelation in the stationary regime, we obtain a fourth-order differential equation

$$\left[\partial_\tau^4 + (2\beta\gamma - \gamma^2 - 1)\partial_\tau^2 + \gamma^2(\beta+1)^2\right]\Delta(\tau) = (\gamma^2 - \partial_\tau^2)g^2 C_{\phi\phi}(\tau) \quad , \tag{3.37}$$

where $C_{\phi\phi}(\tau)$ is the autocorrelation of the rates, that results from imposing the self-consistent condition on the autocorrelation of $\eta$. Notice that we have to search for a self-consistent solution of Eq.(3.37), since its right-hand side depends ultimately on $x$ as the left hand side. To see this self-consistency explicitly, we notice that $C_{\phi\phi} = f_\phi(\Delta(\tau);\Delta_0)$, where

$$f_\phi(\Delta(\tau);\Delta_0) = \int\int \phi\left(\sqrt{\Delta_0 - \frac{\Delta^2(\tau)}{\Delta_0}}x + \frac{\Delta(\tau)}{\sqrt{\Delta_0}}z\right)\phi\left(\sqrt{\Delta_0}z\right)DxDz \quad , \tag{3.38}$$

where $Dx$ and $Dz$ are normalized Gaussian measures. We use the chain rule and Price's theorem (Price (1958)) to rewrite the derivative with respect to $\tau$ on the right-hand side of Eq. (3.37), resulting in

$$\begin{aligned}\left[\partial_\tau^4 + (2\beta\gamma - \gamma^2 - 1)\partial_\tau^2 + \gamma^2(\beta+1)^2\right]\Delta(\tau) =& g^2\gamma^2 f_\phi(\Delta(\tau),\Delta_0) \\ &- g^2(\partial_\tau\Delta(\tau))^2 f_{\phi''}(\Delta(\tau);\Delta_0) \\ &- g^2(\partial_\tau^2\Delta(\tau)) f_{\phi'}(\Delta(\tau);\Delta_0)\,.\end{aligned} \tag{3.39}$$

We can rewrite this fourth-order differential equation as a system of four first-order differential equations

$$\dot{\Delta}(\tau) = \Delta_1(\tau) \tag{3.40}$$

$$\dot{\Delta}_1(\tau) = \Delta_2(\tau) \tag{3.41}$$

$$\dot{\Delta}_2(\tau) = \Delta_3(\tau) \tag{3.42}$$

$$\begin{aligned}\dot{\Delta}_3(\tau) =& (1+\gamma^2 - 2\gamma\beta)\Delta_2(\tau) - \gamma^2(\beta+1)^2\Delta(\tau) + g^2\gamma^2 f_\phi(\Delta(\tau),\Delta_0) \\ &- g^2\Delta_1^2(\tau) f_{\phi''}(\Delta(\tau);\Delta_0) - g^2\Delta_2(\tau) f_{\phi'}(\Delta(\tau);\Delta_0) \quad .\end{aligned} \tag{3.43}$$

Using the symmetry properties of the autocorrelation and assuming that it is a smooth function, we have that $\Delta_1(0) = \Delta_3(0) = 0$. There are therefore two constants to be determined, $\Delta_0$ and $\Delta_2(0)$, the first of which also directly enters the system of differential equations.

In the case without adaptation, the resulting system of ODEs is two-dimensional, so only $\Delta_0$ needs to be determined. $\Delta_0$ can be found by imposing the boundary conditions $\Delta(\infty) = 0$, $\Delta_1(\infty) = 0$, corresponding to a chaotic solution (Sompolinsky et al. (1988)). Since in that case the system is conservative, these boundary conditions are then transformed in an initial condition by exploiting energy conservation (Sompolinsky et al. (1988)). In the case with adaptation however, the

system is non-conservative. Therefore, we cannot transform equivalent boundary conditions (all variables go to zero for $\tau \to \infty$) into initial conditions. In order to do so, we would need to find two conserved quantities of the system. Finding conserved quantities in a dynamical system is known to be a very hard problem, and we were not able to find any conserved quantity for our system. An alternative solution would be to use a shooting method to determine the two initial conditions. However, since we need to determine two initial conditions, such method is computationally very expensive.

## 3.5 Author contributions

SPM and TS designed the project. SPM performed the derivations, with the help of TS. Simulation and numerical integration code was written by SPM. SPM, WG and TS wrote the manuscript.

# 4 Dynamics of multi-dimensional rate units

This chapter presents research carried out in collaboration with Wulfram Gerstner and Tilo Schwalger.

## 4.1 Introduction

The brain exhibits very rich dynamics at multiple scales. Single neurons, beside integrating their input and emitting action potentials, also undergo a variety of spike-history-dependent effects, such as refractoriness and spike-frequency adaptation. Neurons in the cortex are usually embedded in highly recurrent networks, whose dynamics is shaped by both the connectivity and single neuron properties. How these two factors interact and contribute to the spontaneous and evoked dynamics of a recurrent neural network, is poorly understood.

Groups of spiking neurons are often described using firing rate models, i.e. by discarding the information about the exact spike-timing of single neurons. Despite being an approximation, such models have the advantage of being easier to study analytically, so that their dynamics can often be fully characterized. Firing rate models can also be combined to form networks, whose collective dynamics can be understood using mean-field techniques (Sompolinsky et al. (1988)), or that can be used to learn complex tasks (Sussillo and Abbott (2009); Mastrogiuseppe and Ostojic (2018)). However, commonly-used rate models (or *classic* rate models) are one-dimensional and as such cannot fully capture the dynamics of the mean activity of a population of spiking neurons. For example, classic rate models fail to account for rapid synchronization of neurons in response to a stimulus (Mainen and Sejnowski (1995); Bair and Koch (1996)), an effect that is readily observed even in simple spiking neuron models, such as the leaky integrate-and-fire model (Knight (1972); Konig et al. (1996); Gerstner (2000); Brette and Guignon (2003)). To capture rapid synchronization after stimulus onset, it is necessary to consider at least two-dimensional rate models (Mattia and Del Giudice (2002); Schaffer et al. (2013); Montbrió et al. (2015)). If in addition we want to account for history-dependent biophysical mechanisms present at the single neurons level such as refractoriness or spike-frequency adaptation, multiple auxiliary

variables should be added (Naud and Gerstner (2012); Deger (2014); Schwalger et al. (2017)), since the underlying spiking models are themselves multi-dimensional. Other examples of multi-dimensional spiking models have been developed to account for synaptic filtering (Fourcaud and Brunel (2002); Schwalger and Schimansky-Geier (2008)), subthreshold resonance (Richardson et al. (2003)) or for the effect of dendritic compartments (Ostojic (2015); Doose et al. (2016)).

In this chapter we study the dynamics of recurrent networks constructed by randomly connecting multi-dimensional rate models. We will see that a transition to chaos seems to occur consistently across different rate models, at a critical value of the connectivity strength that depends on the rate-model parameters. On the other hand, the characteristic of the chaotic regime are strongly dependent on the properties of the rate model under consideration, suggesting that the chaotic attractor in which the dynamic settles is model-dependent. This chapter also illustrates the general theoretical framework that we used in chapter 3 to analyze the network with adaptation.

## 4.2 Microscopic model and fixed-point stability

We consider a network of randomly connected, multi-dimensional firing-rate units where each unit is described by a set of $D$ variables. We assume that the first variable defines the output rate via a nonlinear gain function $\phi$, that we leave arbitrary for the moment, i.e. $y(t) = \phi(x_i^1(t))$. The remaining $D-1$ variables are auxiliary variables. In isolation, each unit obeys a system of $D$ first-order linear differential equations

$$\dot{x}_i^\alpha(t) = \sum_{\beta=1}^{D} \mathrm{A}^{\alpha\beta} x_i^\beta(t) \quad . \tag{4.1}$$

For the entire chapter, subscripts (in Latin letters) indicate the index of the unit in the network and run from 1 to $N$, while superscripts (in Greek letters) indicate the index of the variable in the rate model and run from 1 to $D$. The matrix $A$ is assumed to be non-singular and to have eigenvalues with negative real parts. We assume that the rate $\phi(x_i^1(t))$ is the only signal that unit $i$ uses to communicate with other units. Conversely, the signals coming from other units only influence the variable $x_i^1$, i.e. the rate of a unit is not directly coupled to the auxiliary variables of other units. Unit $i$ receives input from all the other units, via a set of random connections $J_{ij}$. When incorporating these assumptions, the network equations read

$$\dot{x}_i^\alpha(t) = \sum_{\beta=1}^{D} \mathrm{A}^{\alpha\beta} x_i^\beta(t) + \delta^{\alpha 1}\left(\sum_{j=1}^{N} J_{ij}\phi(x_j^1(t)) + I_i(t)\right) \quad , \quad J_{ij} \sim \mathcal{N}\left(0, g^2/N\right) \tag{4.2}$$

where $\delta^{\alpha\beta}$ is the Kronecker delta symbol and $J_{ij}$ are the randomly-chosen synaptic strengths. The external input $I_i(t)$ is assumed to have stationary statistics and zero mean.

Eq. (4.2) is a system of $N \cdot D$ coupled nonlinear differential equations that becomes clearly intractable for large $N$. However, if $\phi(0) = 0$ the system has a fixed point in $x_i^\alpha = 0, \forall i, \forall \alpha$, whose stability can be studied thanks to the clustered structure of the system. The Jacobian at the

fixed point is given by

$$\mathrm{B} := \mathscr{J}|_{x_i^a=0} = \begin{pmatrix} \mathrm{A}^{11}\mathrm{I}_N + \phi'(0)\mathrm{J} & \mathrm{A}^{12}\mathrm{I}_N & \dots & \mathrm{A}^{1D}\mathrm{I}_N \\ \mathrm{A}^{21}\mathrm{I}_N & \mathrm{A}^{22}\mathrm{I}_N & \dots & \mathrm{A}^{2D}\mathrm{I}_N \\ \dots & \dots & \dots & \dots \\ \mathrm{A}^{D1}\mathrm{I}_N & \mathrm{A}^{D2}\mathrm{I}_N & \dots & \mathrm{A}^{DD}\mathrm{I}_N \end{pmatrix} \quad , \tag{4.3}$$

where J is the random connectivity matrix and $\mathrm{I}_N$ is the $N$-dimensional identity matrix. The matrix B is of size $ND \times ND$ and it therefore admits $ND$ eigenvalues. Since all the blocks of B commute with each other, we can apply the result of Silvester (2000) to find a relation between the eigenvalues of J, A and B

$$\lambda_{\mathrm{J}} = \frac{\prod_{i=1}^{D}\left(\lambda_{\mathrm{B}} - \lambda_{\mathrm{A}}^{i}\right)}{\phi'(0)\prod_{j=1}^{D-1}\left(\lambda_{\mathrm{B}} - \lambda_{\mathrm{A}^-}^{j}\right)} \quad , \tag{4.4}$$

where $\mathrm{A}^-$ is the matrix obtained by removing the first column and the first row from the matrix A. This expression is valid for all the eigenvalues of B that are not coincident with those of $\mathrm{A}^-$. Eq. (4.4) is a degree-$D$ polynomial equation in $\lambda_{\mathrm{B}}$, so that for every value of $\lambda_{\mathrm{J}}$ we obtain $D$ eigenvalues of B, as expected. From now on we will assume that, for simplicity, $\phi'(0) = 1$.

In the $N \to \infty$ limit, the eigenvalues $\lambda_{\mathrm{J}}$ are known to be uniformly distributed on a disk in the complex plane, centered in zero and of radius $g$ (Girko (1985)). If one can invert Eq. (4.4) it becomes computationally fast to compute the eigenvalues of the Jacobian in the $N \to \infty$ limit without having to deal with finite-size effects. Whether one can obtain an explicit inverse formula depends on the dimensionality and on the entries of the matrix A.

Examples of the eigenvalue spectrum of the Jacobian of the network at the fixed point, obtained by solving Eq. (4.4) with respect to $\lambda_{\mathrm{B}}$, are shown in Fig. 4.1A,D. We show the examples of a three-dimensional model and a four-dimensional model, whose parameters are summarized in Table 4.1, for three different values of $g$ each.

## 4.3 Mean-field approximation

In order to study the dynamics of the system beyond the fixed point regime, we use a mean-field approach. The reason for this choice is twofold: first, the mean-field approximation allows us to describe the system dynamics for very large $N$, in which a study of the full microscopic system would be hopeless. Second, we are interested in the properties of a typical network and not of a specific realization of the connectivity matrix. This requires an average over the disorder, i.e. over the ensemble of matrices J. Following Sompolinsky et al. (1988), we approximate the network input to a representative unit $i$ with a Gaussian process $\eta$ and substitute the average over time, initial conditions and network realizations with the average over realizations of $\eta$. This approximation is valid in the large-$N$ limit, in which neurons become independent (Schücker et al. (2016a); Crisanti and Sompolinsky (2018)). Notice that we are interested in the regime in

which the number of auxiliary variables per unit $D$ remains finite. In the mean-field description, the activity of each individual unit in the network follows a realization of the following system of $D$ stochastic differential equations, to which we refer to as mean-field equations (see section 4.6.3 for more details)

$$\dot{x}_i^\alpha(t) = \sum_{\beta=1}^{D} A^{\alpha\beta} x_i^\beta(t) + \delta^{\alpha 1}\left(\eta(t) + I(t)\right) \quad , \tag{4.5}$$

where $\eta(t)$ is a Gaussian process with mean zero and whose autocorrelation needs to be determined self-consistently by imposing

$$\langle \eta(t)\eta(s)\rangle = g^2 \langle \phi(x^1(t))\phi(x^1(s))\rangle \quad . \tag{4.6}$$

Thanks to the mean-field approximations, we reduced a ND-dimensional, deterministic, nonlinear system to a D-dimensional, stochastic, linear system. The effect of the nonlinearity however, needs to be considered when performing the self-consistent moment matching.

One advantageous aspect of the mean-field equations is that all nonlinear effects are summarized in the self-consistency condition (Eq. (4.6)). The mathematical structure of Eq. (4.5) enables a straightforward Fourier transform. After inverting the matrix equation,we solve for $\tilde{x}^\alpha(f)$ and find

$$\tilde{x}^\alpha(f) = \left[\left(2\pi i f - \mathrm{A}\right)^{-1}\right]^{\alpha 1}\left(\tilde{\eta}(f) + \tilde{I}(f)\right) \quad , \tag{4.7}$$

where $\left[\left(2\pi i f - \mathrm{A}\right)^{-1}\right]^{\alpha 1}$ indicates the $\alpha^{\text{th}}$ element of the first column of the matrix $\left(2\pi i f - \mathrm{A}\right)^{-1}$. In analogy with the theory of linear time-invariant (LTI) systems (see for example Hespanha (2009)), we define the linear response function of a network of uncoupled neurons (i.e. obtained by setting $\eta = 0$) in the frequency domain as

$$\tilde{\chi}_0^{\alpha 1}(f) = \left[\left(2\pi i f - \mathrm{A}\right)^{-1}\right]^{\alpha 1} \quad . \tag{4.8}$$

Since we assume a stationary input with zero mean, the mean of all variables is equal to zero. The second-order statistics are summarized, in the frequency domain, by the power spectral density matrix, whose elements are defined by $\delta(f - f')S_x^{\alpha\beta}(f) = \langle(\tilde{x}^\alpha)^*(f)\tilde{x}^\beta(f')\rangle$, where the average is over the Gaussian process $\eta$. $S_x^{\alpha\beta}(f)$ is the Fourier transform of the cross-correlation between the variable $x^\alpha$ and the variable $x^\beta$. The elements of the matrix of power spectral densities obeys the following set of algebraic equations

$$S_x^{\alpha\beta}(f) = \left(\tilde{\chi}_0^{\alpha 1}(f)\right)^* \tilde{\chi}_0^{\beta 1}(f)\left(S_\eta(f) + S_I(f)\right) \quad . \tag{4.9}$$

Moreover, to the self-consistency condition reads, in frequency domain

$$S_\eta(f) = g^2 S_{\phi(x^1)}(f) \quad . \tag{4.10}$$

Among the set of equations in Eq.(4.9), the only difficult one equation for $\alpha = \beta = 1$, which we need to solve self-consistently; the other equations can be trivially solved if the solution for $S_x^{11}(f)$ is known. We rewrite Eq. (4.9) for the case of $S_x^{11}(f)$ and insert the linear response function

$$S_x^{11}(f) = |\tilde{\chi}_0^{11}(f)|^2 \left(S_\eta(f) + S_I(f)\right) = \frac{\left|\left[\mathrm{adj}(2\pi i f - \mathrm{A})\right]^{11}\right|^2}{\prod_{i=1}^{D} |2\pi i f - \lambda_{\mathrm{A}}^i|^2} \left(S_\eta(f) + S_I(f)\right) \quad , \tag{4.11}$$

where we used the expression for the inverse of the matrix $\left(2\pi i f - \mathrm{A}\right)^{-1}$ in terms of its adjoint $\mathrm{adj}\left(\left(2\pi i f - \mathrm{A}\right)^{-1}\right)$. We notice that in the numerator we have the squared absolute value of a polynomial of degree $D-1$, i.e. a polynomial of degree $2D-2$. On the other hand, in the denominator we find a polynomial of degree $2D$. We can see two examples of the behavior of $|\tilde{\chi}_0^{11}(f)|^2$ in Fig. 4.1B,E (solid lines). The parameters of the corresponding rate model are summarized in Table 4.1. Notice that the linear response function can be non-monotonic and even multi-modal, while for a one-dimensional model it is always monotonic.

In section 4.6.1, we show that $|\tilde{\chi}_0^{11}(f)|^2$ allows to compute the critical value of $g$ more easily than Eq. (4.4). We find that $g_c$ is implicitly defined by

$$g_c^2 \max_f |\tilde{\chi}_0^{11}(f)|^2 = 1 \quad . \tag{4.12}$$

In the next section we show that $|\tilde{\chi}_0^{11}(f)|^2$ has a crucial role in shaping the statistics of the dynamics in the fluctuating regime.

## 4.4 Qualitative study of the mean-field solution

The traditional approach in the DMFT literature is to consider the time-domain version of Eq. (4.9). This can be obtained by applying an inverse Fourier transform, which leads to a differential equation of order $2D$. However, the problem of determining the initial conditions becomes harder for increasing $D$. For this reason, we remain in the frequency domain and apply an iterative approach to solve Eq. (4.9) self-consistently (i.e. introducing Eq.(4.10)). Iterative methods to solve self-consistent problems have already been proposed both in the context of spiking networks (Dummer et al. (2014); Wieland (2015)) and of DMFT (Stern et al. (2014)). Here we discuss how such a method in the frequency domain allows to qualitatively understand several features of the network dynamics.

The difficulty in solving the self-consistent problem in the frequency domain lies in calculating the nonlinear effect that $\phi$ has on the statistics of $x$. Concretely, we need to express $S_{\phi(x^1)}$ as a functional of $S_x^{11}(f)$. While this calculation cannot be carried out analytically for a general nonlinearity, it is possible to compute it numerically or semi-analytically. The idea of using an iterative method is to start with an arbitrary initial power spectral density $S_{\phi(x^1)}^{(0)}(f)$, which could be, for example, the one of white noise. We then apply multiple iterations each consisting of a linear step followed by a nonlinear step. At each iteration, the linear step is simply a

multiplication by $g^2|\tilde{\chi}_0^{11}(f)|^2$ and it allows us to compute $(S_x^{11})^{(n+1)}(f)$. The nonlinear step afterwards transforms $(S_x^{11})^{(n+1)}(f)$ into $S_{\phi(x^1)}^{(n+1)}(f)$. Depending on the nonlinearity $\phi$ that we consider, this step can introduce numerical errors of variable magnitude. Due to the importance of understanding the effect of the nonlinearity on the second order statistics, the discussion on how to numerically or semi-analytically implement the nonlinear step is treated separately in section 4.6.2.

For a qualitative understanding of the effect of the iterations on the power spectral density, we exploit the fact that $x^1$ is a Gaussian process, since $\eta$ was assumed to be a Gaussian process. For Gaussian processes, the following formula holds (Stratonovich (1967))

$$C_{\phi(x^1)}(\tau) = \sum_{n=0}^{\infty} \frac{1}{n!} \left\langle \frac{d^n \phi}{d(x^1)^n} \right\rangle^2 C_{x^1}^n(\tau) \quad , \tag{4.13}$$

where the angular brackets indicate the mean over the statistics of $x^1$. Eq. (4.13) gives the effect of a nonlinearity $\phi$ on a the autocorrelation of a Gaussian process $x^1$. By truncating the series after the first term, we get

$$C_{\phi(x^1)}(\tau) \simeq \left(\langle \phi'(x^1) \rangle\right)^2 C_{x^1}(\tau) \quad . \tag{4.14}$$

Fourier transforming this equation we get an approximation of the power spectral density of $\phi(x^1)$

$$S_{\phi(x^1)}(f) \simeq s_1 \left( \int_{-\infty}^{\infty} S_x^{11}(f') df' \right) S_x^{11}(f) \quad , \tag{4.15}$$

where we rewrote $\left(\langle \phi'(x^1) \rangle\right)^2 =: s_1 \left( \int_{-\infty}^{\infty} S_x^{11}(f') df' \right)$ to highlight the fact that the coefficient that multiplies $S_x^{11}(f)$ depends on the area under the power spectral density, i.e. on the variance of $x^1$. We stress that retaining only the first term in Eq. (4.13) is different than considering a linear approximation of $\phi$, since the dependence of the coefficient on the variance would not appear in that case.

Using this approximation, we can express the power spectral density at the $n^{\text{th}}$ iteration of the iterative method, as a function of the initial power spectral density $S_{\phi(x^1)}^{(0)}(f)$ from which we started to iterate. We obtain

$$(S_x^{11})^{(n)}(f) = \left( \prod_{k=1}^{n-1} s_1^{(k)} \right) \left( g^2 |\tilde{\chi}_0^{11}(f)|^2 \right)^n S_{\phi(x^1)}^{(0)}(f) \quad , \tag{4.16}$$

where $s_1^{(k)} := s_1 \left( \int_{-\infty}^{\infty} (S_x^{11})^{(n)}(f') df' \right)$. If we take $S_{\phi(x^1)}^{(0)}(f)$ to be constant and we define $a_n = \left( \prod_{k=1}^{n-1} s_1^{(k)} \right)$, we can rewrite the above expression as

$$(S_x^{11})^{(n)}(f) = a_n \left( g^2 |\tilde{\chi}_0^{11}(f)|^2 \right)^n \quad . \tag{4.17}$$

If $g > g_c$, there will be a range of frequencies for which $g^2|\tilde{\chi}_0^{11}(f)|^2 > 1$, which implies that its

$n^{\text{th}}$ power diverges when $n$ grows. In a purely linear network, this phenomenon would lead to a blow-up of the power spectral density, in agreement with the fact that activity in a linear network is unbounded for $g > g_c$. If $\phi$ is a compressive nonlinearity however, the coefficient $a_n$ will tend to zero for growing $n$, counterbalancing the unbounded growth of $\left(g^2|\tilde{\chi}_0^{11}(f)|^2\right)^n$. Notice that this constraint on the coefficient $a_n$ is necessary independently of the truncation of the series in Eq. (4.13), since all the neglected terms are positive and would not provide a different mechanism for contrasting the growth of the first term. Based on Eq. (4.17), we would predict that all the modes for which $|\tilde{\chi}_0^{11}(f)|^2 > 1/g^2$ will get amplified over multiple iterations, while all the other modes will get suppressed. While this is a highly simplified description, the suppression and the amplification of modes is clearly visible when comparing the dynamics of the self-consistent solution (Fig. 4.1C,F) to the corresponding linear response function (Fig. 4.1B,E). When truncating the series after the first order however, the mean-field network does not admit a self-consistent solution, for which we need to retain also higher order terms. The presence of those terms will be reflected, among others, in the interference among amplified modes.

We now consider also higher order terms of the sum, which allow the existence of a self-consistent solution and that are responsible for the formation of harmonics. For example, the second order term in Eq. (4.13) is given by

$$\frac{1}{2}\left(\langle\phi''(x^1)\rangle\right)^2\left(C_{x^1}(\tau)\right)^2 \xrightarrow{FT} \frac{1}{2}s_2\left(\int_{-\infty}^{\infty} S_x^{11}(f')df'\right)\left(S_x^{11} * S_x^{11}\right)(f) \tag{4.18}$$

where $s_2$ is defined analogously to $s_1$. In general, higher-order terms will contain convolutions of the power spectral density with itself, which are responsible for the creation of higher harmonics. Indeed, if a function has a bump-shaped profile, then its $n$-times self-convolution shifts the center of the bump to the $n^{\text{th}}$ multiple of the bump center. This implies that if the power spectral density is resonant, i.e. if it has a peak at a nonzero frequency, then to be self-consistent it should also exhibit harmonics.

In Fig. 4.1A,B,C we consider a three-dimensional rate model that has the global maximum at zero frequency and a local maximum at a higher frequency. For the weakest connectivity strength ($g = g_1$), only the modes near the global maximum are amplified (Fig. 4.1C), in agreement with our qualitative analysis. At the other extreme, i.e. for very strong connectivity ($g = g_3$), also the modes near the second local maximum survive in the self-consistent solution. Finally, for intermediate connectivity strength ($g = g_2$), the interference among modes is strong enough to prevent the amplification of the modes close to the local maximum, so that only those next to the global one survive. Similar observations can be made for the four-dimensional model shown in Fig. 4.1D,E,F.

## 4.5 Robustness of the iterative method

The formalism we presented is independent of the choice of the nonlinearity and the only assumptions that we made in deriving the existence of a microscopic fixed-point in zero is that

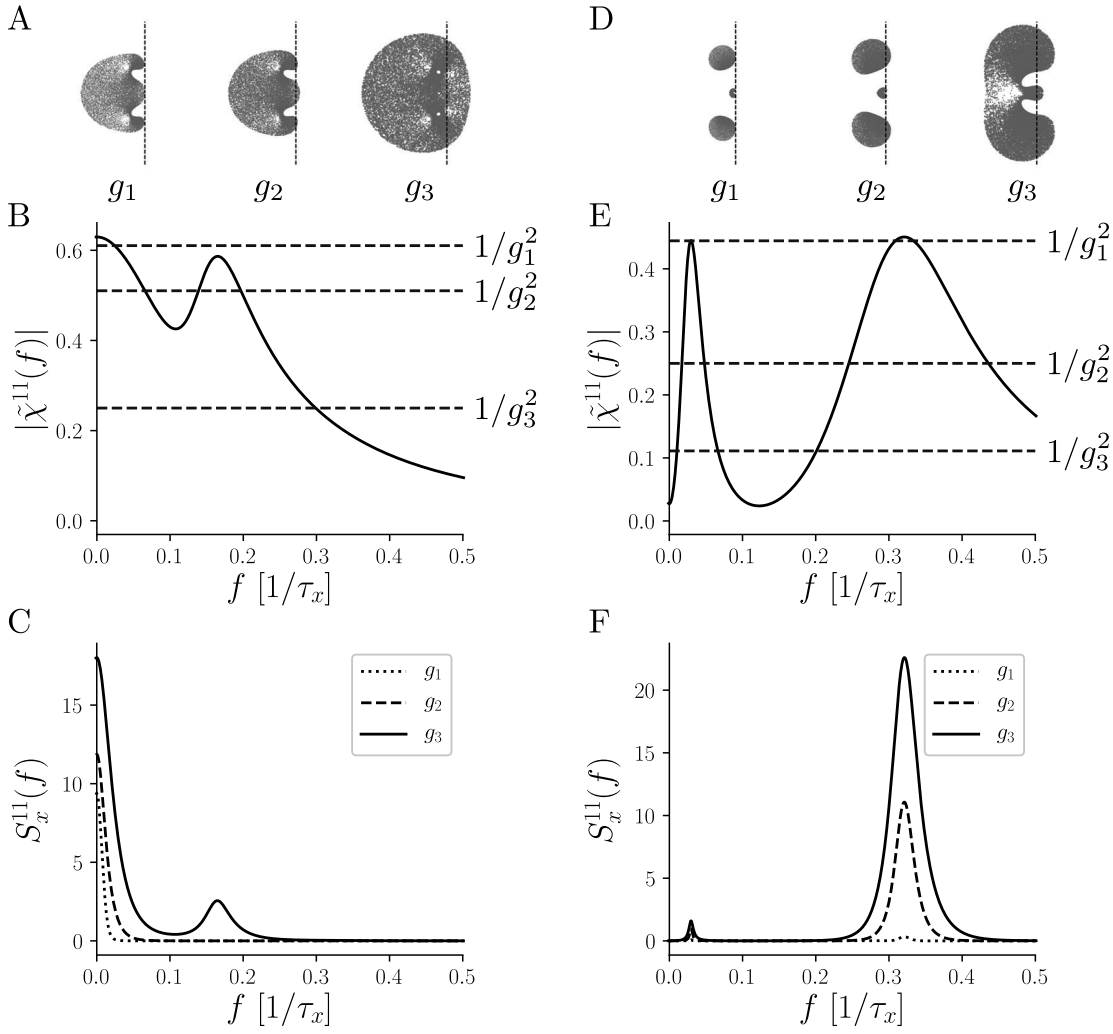



Figure 4.1 – **Two examples of multi-dimensional rate models. A-B-C:** Analysis of a three-dimensional rate model. Eigenvalue spectra (A) corresponding to the coupling values $g_1 = 1.28$, $g_2 = 1.4$ and $g_3 = 2$. The dashed line indicates the imaginary axis. In B we plot the linear response function of the single unit $|\tilde{\chi}_0^{11}(f)|^2$ (solid line), and the instability threshold corresponding to the three coupling values $g_1$, $g_2$ and $g_3$ (dashed lines). In C we plot the solution of the mean field theory obtained with the iterative method for the three values of $g$, $g_1 = 1.5$, $g_2 = 2$ and $g_3 = 3$. **D-E-F:** Same as A-B-C, but for a four-dimensional rate model.

| Fig. 4.1A,B,C | Fig. 4.1D,E,F | Fig. 4.2 |
|---|---|---|
| $\begin{pmatrix} -1 & -1 & -1 \\ 0.1 & -0.1 & 1.7 \\ 0.1 & -0.4 & -0.5 \end{pmatrix}$ | $\begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & -0.5 & -0.65 & -0.6 \\ 1 & 0.35 & -0.05 & -0.57 \\ 1 & 0.35 & 0.28 & -0.005 \end{pmatrix}$ | $\begin{pmatrix} -1 & -1 \\ 0.25 & -0.25 \end{pmatrix}$ |

Table 4.1 – **Parameters of the models in the examples.** Matrix A defining the rate model for the different example in Fig. 4.1 and 4.2.

$\phi(0) = 0$. In solving the mean-field theory for the examples presented in Fig. 4.1, we used a piecewise-linear approximation of the hyperbolic tangent, given by

$$\phi_{PL}(x) = \begin{cases} -1 & \text{for} \quad x < -1 \\ x & \text{for} \quad -1 \leq x \leq 1 \\ 1 & \text{for} \quad x > 1 \end{cases} . \tag{4.19}$$

The self-consistent solution that we obtain does not vary qualitatively when considering similar gain functions, such as the hyperbolic tangent or a cubic approximation of it (shown in Fig. 4.2A, for the two-dimensional model studied in chapter 3). However, since the cubic gain function is unbounded, the dynamics will be unstable above a certain $g_c^*$, which is different from $g_c$ in general.

The evolution of the power spectral density $S_x^{11}(f)$ over iterations is shown in Fig. 4.2C, where the formation of harmonics over iterations is clearly visible. While we have no guarantee that the iterative method converges, empirical tests of the method indicate convergence properties that match the one of stability of the network itself, i.e. if the network converges, so does the iterative method. The mean-squared distance between the power spectral density at consecutive realizations decreases approximately exponentially after an initial transients (Fig. 4.2B) and saturates at a value dependent on the numerical error made in performing the nonlinear pass. The iterative method is also robust over different initializations of $S_x^{11}(f)$. To follow the evolution of $S_x^{11}(f)$ over iterations, we measure total area under it, i.e. $\int_{-\infty}^{+\infty} S_x^{11}(f)df = \text{Var}(x^1)$ and the maximum height of the power spectral density $\max_f(S_x^{11}(f))$. Despite different initializations, the trajectories in the subspace of these two measures converge quite rapidly to a one-dimensional sub-manifold (Fig. 4.2D).

## Discussion

We analyzed how the dynamics of a random network of multi-dimensional rate units is influenced by the properties of the rate model itself. We used DMFT, a well-established theoretical tool,

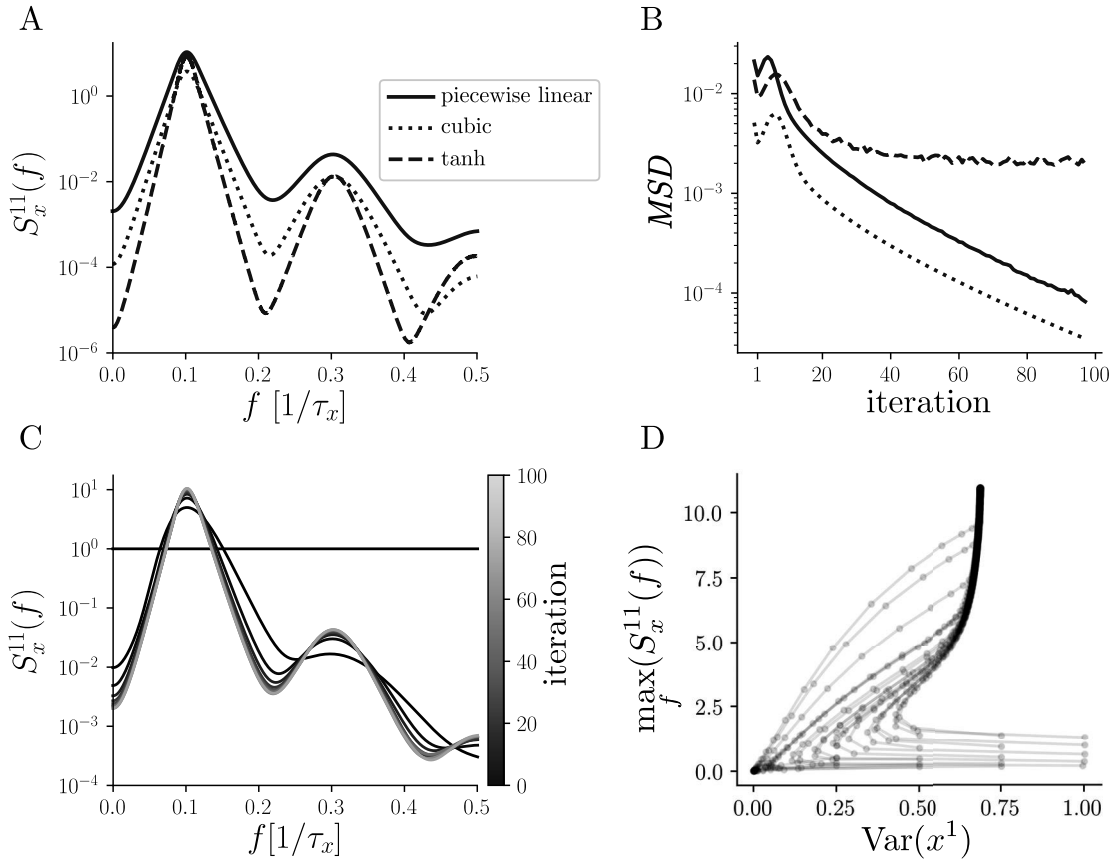



Figure 4.2 – **Stability of the iterative method for a two-dimensional rate model. A:** Power spectral density $S_x^{11}(f)$ for three different nonlinearities, as indicated in the legend. **B:** Mean-squared distance (*MSD*) between two consecutive iterations of $S_x^{11}(f)$. Conventions are the same as in A. Notice that the curve corresponding to the hyperbolic tangent is saturating at a much higher value than for the other nonlinearities, which is due to the sampling method used to evaluate the nonlinear pass. **C:** Evolution of the power-spectral density over iterations, shown for piecewise linear nonlinearity. **D:** Evolution of the total area under the power spectral density $\mathrm{Var}(x)$ and of the maximum amplitude of $S_x(f)$, over iterations. Shown for piecewise linear nonlinearity.

to reduce the high-dimensional, deterministic network model to a low dimensional system of stochastic differential equations. Standard approaches to solve the mean-field equations were not fruitful in the multi-dimensional setting. However, a qualitative study of the semi-numerical solution lead us to conclude that the linear response function of the rate model plays a crucial role in shaping the statistics of the recurrent dynamics. More precisely, bands of preferred frequencies in the linear response function become narrower due to recurrent connections. Moreover, if multiple resonance bands are present, they seem to interfere so that in the self-consistent dynamics exhibit only one dominant frequency.

One interesting application of multi-dimensional rate models is to spike-frequency adaptation (SFA). A phenomenological 2-dimensional rate model that include a form of SFA has been studied in detail in chapter 3. However, SFA is known to have multiple timescales that are power-law distributed (Lundstrom et al. (2008); Pozzorini et al. (2013)). The example models shown in Fig. 4.1 are instantiations of this type of adaptation. Interestingly, if no coupling is present between different adaptation variables, the linear response function exhibits only one band of preferred frequencies. On the other hand, if there is coupling, the linear response function can be multimodal and this can be reflected also in the recurrent dynamics (as shown in Fig. 4.1).

We carried out our analysis for the case of balanced network, i.e. in which the mean of the input is zero. There are no conceptual obstacle in extending the analysis to include more recent developments. For example, the case of non-balanced input was considered in (Kadmon and Sompolinsky (2015); Mastrogiuseppe and Ostojic (2017)), in which it was shown that a transition to chaos can be observed in many different network architectures and in some cases also for unbounded nonlinearities. We expect this observation to hold when considering multi-dimensional rate models. As another example, Mastrogiuseppe and Ostojic (2018) extended the formalism to include the effect of low-rank perturbation of the random connectivity, showing that the resulting dynamics becomes effectively low dimensional and that it allows the network to perform complex tasks. It is an open question how the properties of the multi-dimensional rate model would shape this effective low-dimensional dynamics.

## 4.6 Additional details

### 4.6.1 Mean-field linear stability analysis

In this section, we generalize the calculation performed in section 3.4.4 to find the linear response function, at the fixed point, in the mean-field approximation. The steps are conceptually the same as in section 3.4.4. However, here we consider the full matrix of linear response functions (see below), to conclude that the only quantity that matters for the stability at the fixed point is $|\tilde{\chi}^{11}(f)|^2$.

Starting from the microscopic network equations (Eq. (4.2)), we derive a set of differential

equations, that we write in matrix form

$$(\mathrm{I}_D\partial_\tau - \mathrm{A})\,\chi_{ik}(\tau) = \sum_{j=1}^{N} J_{ij}\Delta_1\chi_{jk}(\tau) + \delta_{ik}\mathrm{I}_D\delta(\tau) \quad , \tag{4.20}$$

where $\Delta_1 = \delta^{\alpha 1}\delta^{\beta 1}$ is a matrix whose only nonzero element is $[\Delta_1]^{11} = 1$. $\chi_{ik}(\tau)$ is a $D$ by $D$ matrix, whose component are defined as $\chi_{ik}^{\alpha\beta}(\tau) = \frac{\delta x_i^\alpha(\tau)}{\delta h_k^\beta(0)}$, where $h_k^\beta$ is a small perturbation given to the variable $x_k^\beta$ at time $\tau = 0$. Notice that in deriving Eq. (4.20), we have assumed stationarity and that $\phi'(0) = 1$. We now Fourier transform Eq. (4.20) and get

$$\left(2\pi i f\,\mathrm{I}_D - \mathrm{A}\right)\tilde{\chi}_{ik}(f) = \sum_{j=1}^{N} J_{ij}\Delta_1\tilde{\chi}_{jk}(f) + \delta_{ik}\mathrm{I}_D \quad . \tag{4.21}$$

Inverting the matrix $\left(2\pi i f\,\mathrm{I}_D - \mathrm{A}\right)$ and recognizing the linear response function of the single unit $\tilde{\chi}_0(f)$, we obtain

$$\tilde{\chi}_{ik}(f) = \sum_{j=1}^{N} J_{ij}\tilde{\chi}_0(f)\Delta_1\tilde{\chi}_{jk}(f) + \delta_{ik}\tilde{\chi}_0(f) \quad , \tag{4.22}$$

where $\tilde{\chi}_0(f)$ is a $D$ by $D$ matrix whose elements are $\tilde{\chi}_0^{\alpha\beta}(f)$, defined in section 4.3.

Since in the mean-field approximation the mean of the linear response function is zero, we look for the second moments (Kadmon and Sompolinsky (2015)). We multiply every element of the matrix equation (Eq. (4.22)) by its complex conjugate and average over the quenched disorder. We obtain

$$|\tilde{\chi}(f)|^2 = g^2|\tilde{\chi}_0(f)\Delta_1\tilde{\chi}(f)|^2 + |\tilde{\chi}_0(f)|^2 \quad , \tag{4.23}$$

where the absolute value is intended element-wise. Due to the structure of the matrix $\Delta_1$, we have that $|\tilde{\chi}_0(f)\Delta_1\tilde{\chi}(f)|^2 = |\tilde{\chi}_0(f)|^2\Delta_1|\tilde{\chi}(f)|^2$, as it can be verified simply by using the definition of $\Delta_1$. Finally, we can solve for $|\tilde{\chi}(f)|^2$

$$|\tilde{\chi}(f)|^2 = \left(\mathrm{I}_D - g^2|\tilde{\chi}_0(f)|^2\Delta_1\right)^{-1}\left(|\tilde{\chi}_0(f)|^2\right) \quad . \tag{4.24}$$

Since the only nonzero eigenvalue of the matrix $|\tilde{\chi}_0(f)|^2\Delta_1$ is $|\tilde{\chi}_0^{11}(f)|^2$, the stability condition for the fixed point is given by

$$g^2 \max_f |\tilde{\chi}_0^{11}(f)|^2 < 1 \quad . \tag{4.25}$$

### 4.6.2 Effect of nonlinearities on second order statistics

In this section, we provide some additional details on how to compute the effect of nonlinearities on the second order statistics (autocorrelation or power spectral density) of a Gaussian process. We

consider three cases of interest: polynomials, piecewise linear functions and arbitrary nonlinear functions. To simplify our notation, we drop the superscript of and consider a generic Gaussian process $x^1$.

The effect of polynomial nonlinearities can be expressed in closed form in time domain. This can be seen by considering again the infinite series expression (Eq. 4.13), valid for stationary Gaussian processes $x$

$$C_{\phi(x)}(\tau) = \sum_{n=0}^{\infty} \left( \left\langle \frac{d^n \phi}{dx^n} \right\rangle \right)^2 C_x^n(\tau) \quad , \tag{4.26}$$

where the angular brackets indicate the average over the statistics of $x$. In the case in which $\phi$ is a polynomial of degree $p$, only the terms in the sum up to $p$ are nonzero. As an example, we can compute the effect of a cubic approximation of the hyperbolic tangent, i.e. $\phi(x) \simeq \phi_3(x) := x - \frac{x^3}{3}$

$$C_{\phi_3(x)}(\tau) = \left(1 + C_x^2(0) - 2C_x(0)\right) C_x(\tau) + \frac{2}{3} C_x^3(\tau) \quad . \tag{4.27}$$

As expected, the effect of the nonlinearity depends on $C_x(0)$ i.e. on the variance of $x$ itself. Notice that the coefficient of the first term is compressive (i.e. smaller than one) only if $C_x(0)$ is smaller than one itself. This type of behavior is expected since $\phi_3$ is unbounded.

Another interesting case are piecewise linear nonlinearities. In this case, we use Price's theorem twice to get

$$\frac{\partial^2 C_{\phi(x)}(t)}{\partial (C_x(t))^2} = C_{\phi''(x)}(t) \quad . \tag{4.28}$$

For a piecewise linear $\phi$, the second derivative $\phi''$ is a sum of Dirac's delta functions with variable coefficients. More precisely, we consider

$$\phi_{PL}(x) = \Theta(x_1 - x) c_0 x + \sum_{p=1}^{P-1} \Theta(x - x_p)\Theta(x_{p+1} - x) c_p x_p + \Theta(x - x_P) c_P x \quad , \tag{4.29}$$

where $x_p$ are the points in which the first derivative is discontinuous, $c_p$ are some arbitrary coefficients and $\Theta(\cdot)$ is the Heaviside function. The second derivative of $\phi_{PL}$ is given by

$$\phi''_{PL}(x) = \sum_{p=1}^{P} (c_p - c_{p-1}) \delta(x - x_p) \quad . \tag{4.30}$$

The delta functions allow us to compute the correlation function $C_{\phi''_{PL}}(t)$ explicitly

$$C_{\phi''_{PL}}(t) = \sum_{p,p'=1}^{P} \frac{(c_p - c_{p-1})(c_{p'} - c_{p'-1})}{2\pi C_x(0)\sqrt{1-\rho^2(t)}} \exp\left( -\frac{x_p^2 + x_{p'}^2 - 2\rho(t) x_p x_{p'}}{2C_x(0)(1-\rho^2(t))} \right) \quad , \tag{4.31}$$

where we defined $\rho(t) := \frac{C_x(t)}{C_x(0)}$. Inserting Eq. (4.31) in Eq. (4.28) and integrating twice with

respect to $C_x(t)$ we get

$$C_{\phi_{PL}(x)}(t) = f_\phi(0; C_x(0)) + f_{\phi'}(0; C_x(0))\, C_x(t) + \sum_{p,p'=1}^{P} \int_0^{C_x(t)} \int_0^{\sigma'} \frac{(c_p - c_{p-1})(c_{p'} - c_{p'-1})}{2\pi C_x(0)\sqrt{1 - \frac{\sigma^2}{C_x^2(0)}}} \times \\ \times \exp\left( -\frac{x_p^2 + x_{p'}^2 - 2\frac{\sigma}{C_x(0)} x_p x_{p'}}{2C_x(0)\left(1 - \frac{\sigma^2}{C_x^2(0)}\right)} \right) d\sigma d\sigma' . \tag{4.32}$$

In the case in which $\phi$ is an odd function, the term $f_\phi(0; C_x(0))$ is equal to zero. For the specific case of the piecewise linear approximation of the hyperbolic tangent considered in this chapter, i.e.

$$\phi_{PL}(x) = \begin{cases} -1 & \text{for} \quad x < -1 \\ x & \text{for} \quad -1 < x < 1 \\ 1 & \text{for} \quad x > 1 \end{cases} , \tag{4.33}$$

the expression in Eq. (4.32) reduces to

$$C_{\phi_{PL}(x)}(t) = \mathrm{Erf}^2\left( \frac{1}{\sqrt{2C_x(0)}} \right) C_x(t) + \frac{2}{\pi C_x(0)} \int_0^{C_x(t)} \int_0^{\sigma'} \frac{1}{\sqrt{1 - \frac{\sigma^2}{C_x^2(0)}}} \quad \times \\ \times \quad \exp\left( -\frac{1}{C_x(0)\left(1 - \frac{\sigma^2}{C_x^2(0)}\right)} \right) \sinh\left( \frac{\sigma}{C_x^2(0)\left(1 - \frac{\sigma^2}{C_x^2(0)}\right)} \right) d\sigma d\sigma' \quad . \tag{4.34}$$

For the piecewise linear function, an alternative approach based on the infinite series in Eq. (4.13) and on Hermite polynomials was proposed by Kruscha and Lindner (2016).

For an arbitrary nonlinear function, we can use two methods. The first method is a semi-analytical approach that relies on the integral form of the autocorrelation of the rate $C_{\phi(x)}(\tau)$ as a functional of the autocorrelation $C_x(\tau)$ of $x$ (Schücker et al. (2016a))

$$C_{\phi(x)}(\tau) = \int\int \phi\left( \sqrt{C_x(0) - \frac{C_x^2(\tau)}{C_x(0)}}\, x + \frac{C_x(\tau)}{\sqrt{C_x(0)}} z \right) \phi\left( \sqrt{C_x(0)} z \right) Dx Dz \quad , \tag{4.35}$$

where $Dx = e^{-x^2/2} dx$. Notice that a slightly different version of this formula was already proposed in Sompolinsky et al. (1988). Therefore, to obtain the effect of $\phi$ on the power spectral density, one should 1) inverse Fourier transform $S_x(f)$ to get $C_x(\tau)$ 2) apply Eq.(4.35), by computing the two integrals numerically 3) Fourier transform $C_{\phi(x)}(\tau)$ to get $S_{\phi(x)}(f)$. Practically, this procedure requires the application of the fast Fourier transform algorithm and the numerical evaluation of two integrals.

The second method is fully numerical and it can be useful in cases in which the integrals in the

first method are expensive to evaluate numerically. This method consists in approximating the power spectral density $S_{\phi(x)}$ via Monte Carlo sampling. More precisely, we sample multiple realizations in frequency domain of the Gaussian process with zero mean and power spectral density $S_x(f)$. We then transform each sample to time domain and apply the nonlinearity $\phi(x)$ to each sample $x(t)$ individually. Finally, we transform back to Fourier domain and get $S_{\phi(x)}$ by averaging. This method, whose steps are summarized in Alg. 3, introduces additional errors due to the finite amount of samples that one considers. Despite being computationally more expensive than the closed form expressions, this sampling method provides a solution of the mean-field theory for an arbitrary nonlinearity and it is computationally much cheaper that running the full microscopic simulation.

**Algorithm 3** Computation of the effect of an arbitrary nonlinearity $\phi$ on the power spectral density.

$M$ = number of samples
**for** $m \in \{1, \dots, M\}$ **do**
    $\tilde{x}_m(f) = \text{samplefrom}(S_x(f))$
    $x_m(t) = FT^{-1}[\tilde{x}_m(f)]$
    $\tilde{\phi}_m(f) = FT[\phi(x_m(t))]$
**end for**
$S_{\phi(x)}(f) = \frac{1}{TM} \sum_{m=1}^{M} \tilde{\phi}_m^*(f)\tilde{\phi}_m(f)$

### 4.6.3 Derivation of mean-field theory

In this section, we will extend the derivation of the dynamic mean-field theory (DMFT) for the case of the network of multi-dimensional rate units. Since there are no additional complication with respect to the standard case, we report here only the main steps. For a review of the path-integral approach to DMFT, see e.g. (Schücker et al. (2016a); Crisanti and Sompolinsky (2018)). The moment-generating functional corresponding to our differential equations is

$$Z[\boldsymbol{j}^x, \tilde{\boldsymbol{j}}^x](J) = \int \mathscr{D}\boldsymbol{x}\mathscr{D}\tilde{\boldsymbol{x}} \exp\left[S_0[\boldsymbol{x}, \tilde{\boldsymbol{x}}] - (\tilde{\boldsymbol{x}}^1)^T \mathrm{J}\phi(\boldsymbol{x}^1(\boldsymbol{t})) + \boldsymbol{j}^T\boldsymbol{x} + \tilde{\boldsymbol{j}}^T\tilde{\boldsymbol{x}}\right] \quad , \tag{4.36}$$

where

$$S_0[\boldsymbol{x}, \tilde{\boldsymbol{x}}] := \tilde{\boldsymbol{x}}^T(I_D\partial_t - A)\boldsymbol{x} \tag{4.37}$$

and we introduced the notation $\tilde{\boldsymbol{x}}^T\boldsymbol{x} = \sum_\alpha \sum_i \int \tilde{x}_i^\alpha(t) x_i^\alpha(t) dt$. The integral is over paths and bold symbols indicate vectors, over both the network space and the rate model space, so that $\mathscr{D}\boldsymbol{x} := \prod_\alpha \prod_i \mathscr{D}x_i^\alpha$.

We are interested in properties that are independent of the particular realization of the coupling matrix J. In order to extract those properties, we average over the quenched disorder by defining

the averaged generating function

$$\bar{Z}[\boldsymbol{j}^x, \tilde{\boldsymbol{j}}^x] := \int \prod_{ij} dJ_{ij} \mathcal{N}\left(0, \frac{g^2}{N}, J_{ij}\right) Z[\boldsymbol{j}^x, \tilde{\boldsymbol{j}}^x](\mathrm{J}) \quad . \tag{4.38}$$

The average over each $J_{ij}$ can be computed by recognizing that the terms corresponding to different $J_{ij}$ factorize and the integral can be solved using the square-completion method. Since the details of this calculation are analogous to the one-dimensional case, we directly report the result

$$\begin{aligned}\bar{Z}[\boldsymbol{j}^x, \tilde{\boldsymbol{j}}^x] = \int \mathscr{D}\boldsymbol{x}\mathscr{D}\tilde{\boldsymbol{x}} \exp\left[S_0[\boldsymbol{x}, \tilde{\boldsymbol{x}}] + \boldsymbol{j}^T\boldsymbol{x} + \tilde{\boldsymbol{j}}^T\tilde{\boldsymbol{x}}\right] \times \\ \times \exp\left[\frac{1}{2}\int_{-\infty}^{\infty}\left(\sum_i \tilde{x}_i^1(t)\tilde{x}_i^1(t')\right)\left(\frac{g^2}{N}\sum_j \phi(x_j^1(t))\phi(x_j^1(t'))\right)\right] \quad .\end{aligned} \tag{4.39}$$

We now aim to decouple the interaction term in the last line by introducing the auxiliary field

$$Q_1(t,s) := \frac{g^2}{N}\sum_j \phi(x_j^1(t))\phi(x_j^1(s)) \quad . \tag{4.40}$$

We rewrite the averaged generating functional as a field theory for two auxiliary fields $Q_1, Q_2$. The result is, following the same steps for the one-dimensional case,

$$\begin{aligned}\bar{Z}[j, \tilde{j}] &= \int \mathscr{D}Q_1\mathscr{D}Q_2 \exp\left(-\frac{N}{g^2}Q_1^T Q_2 + N\ln Z[Q_1, Q_2] + j^T Q_1 + \tilde{j}^T Q_2\right) \\ Z[Q_1, Q_2] &:= \int \mathscr{D}\boldsymbol{x}\mathscr{D}\tilde{\boldsymbol{x}} \exp\left(S_0[\boldsymbol{x}, \tilde{\boldsymbol{x}}] + \frac{1}{2}(\tilde{x}^1)^T Q_1 \tilde{x}^1 + \phi(x^1)^T Q_2 \phi(x^1)\right) \quad ,\end{aligned} \tag{4.41}$$

where we extended our notation to $Q_1^T Q_2 := \int\int Q_1(s,t)Q_2(s,t)dsdt$. The crucial observation to make is that essentially all factors associated to different units factorized yielding the factor $N$. For this reason, the integration is now only over all rate model indices but over only one unit index. The remainder is the problem of one unit, characterized by $D$ variables, interacting with two external fields $Q_1, Q_2$.

The final step is to perform a saddle-point approximation, i.e. replace $Q_1, Q_2$ by their values that make the action stationary. After this step, the averaged generating functional reduces to

$$\bar{Z}^* \propto \int \mathscr{D}\boldsymbol{x}\mathscr{D}\tilde{\boldsymbol{x}} \exp\left(S_0[\boldsymbol{x}, \tilde{\boldsymbol{x}}] + \frac{g^2}{2}(\tilde{x}^1)^T C_{\phi(x^1)}\tilde{x}^1\right) \quad . \tag{4.42}$$

This is the statistical field theory corresponding to $D$ linearly interacting variables, with $x^1$ that receives a Gaussian noise whose autocorrelation is given by $C_{\phi(x^1)}$. Writing the corresponding differential equations results in our mean-field description (Eq. 4.5).

## 4.7 Author contributions

SPM and TS designed the project. SPM performed the derivations, with the help of TS. Simulation and numerical integration code was written by SPM. SPM, WG and TS wrote the manuscript.

# 5 Reservoir computing using networks of neurons with adaptation

This chapter presents preliminary work that I carried out during the last year of my PhD, in parallel to the projects presented in the previous two chapters. Some interesting and novel results have been obtained, but a more careful analysis should still be conducted and the mechanisms should be investigated in depth.

## 5.1 Introduction

The ability to integrate information over long timescales in a variety of contexts is fundamental in several complex tasks that humans and some animals are able to solve. For example, when listening to somebody speaking, the meaning of a word that appears at the end of a sentence might be correctly understood only if we are able to relate it to what was said at the beginning of the sentence.

In chapter 3 we analyzed the dynamics of recurrent networks with a rate adaptation mechanism, and we found that this additional feature increases correlation time in the recurrent network. On the other hand, the interaction between recurrent connections and adaptation sharpens the linear response function of a single rate unit, resulting in an increased amount of correlation at short time lags.

Recurrent networks have proven to be very successful in machine learning tasks that require integration of information over time. However, these networks are notoriously hard to train. State of the art performance is obtained with so-called long-short term memory (LSTM) networks (Hochreiter and Schmidhuber (1997)), in which additional multiplicative mechanisms are added to the units to facilitate memory retention and training. In a recent publication, Bellec et al. (2018) used a recurrent network of spiking neurons with spike-frequency adaptation, trained with back-propagation-through-time, to obtain performance close to LSTM networks. Reservoir learning approaches differ from the aforementioned techniques in that only readout weights, or a subset of the recurrent weights, are trained. This facilitates learning, but it requires a rich enough dynamics of the recurrent network (or *reservoir*), to achieve a good performance. In

this chapter, we explore the possibility of using random recurrent networks with adaptation in reservoir computing.

From the theoretical perspective, one of the advantages of reservoir computing is that the dynamics of the network prior to learning is relatively well understood using mean-field theory approaches. In the more recent reservoir learning techniques (Sussillo and Abbott (2009)) the recurrent connections are partially modified in that only a rank-one perturbation of the recurrent weights is learned. The dynamics of recurrent networks in which the random connectivity is perturbed via a rank-one matrix can also be studied analytically (Mastrogiuseppe and Ostojic (2018)), which allows to implement complex computation with such perturbations.

One of the limitations of reservoir computing is the relatively poor performance in tasks that require bridging long time intervals, either during the generation of very long temporal patterns or during sequence recognition, which might require integration of information over long timescales. The results of chapter 3 regarding the correlation time motivate the study of the performance of the network with adaptation on tasks requiring slow timescales. Moreover, the evidence from the literature discussed above indicates a beneficial role of adaptation for learning, at least in networks of spiking neurons. In this chapter, we begin a systematic study of the effect that adaptation has in the context of reservoir computing with rate units. From our preliminary analysis, it appears that the benefits of adaptation for learning are dependent on the task under consideration.

## 5.2 Results

Our aim is to compare the performance of a network with adaptation (also called *adaptive* network in what follows) to the performance obtained without adaptation, on tasks that require to maintain information over long time lags. Both networks are constructed by introducing a linear readout unit $z$ in a random recurrent network, of the type described in section 1.4 (standard) and in chapter 3 (adaptive). The value of this linear readout unit is fed back to the network via some fixed feedback connections (Sussillo and Abbott (2009), see section 5.4.1). We train only the readout weights, i.e. those that connect the recurrent network to $z$, using the supervised learning algorithm proposed by Sussillo and Abbott (2009). The tasks we devise all require the network to retain information over long timescales. It is important to stress that we neither aim at achieving the best performance for machine-learning type of applications, nor to propose biologically plausible mechanisms that real neural networks might use to solve such tasks. In contrast, we consider these tasks as additional probing tools for the dynamics of the network and in particular for its capability of generating slow dynamics.

For all tasks described below, we used $g = 1.5 g_c$ for both networks, where $g_c = 1$ for the standard network while it depends on the adaptation parameters for the adaptive network (see Eq. (3.7)). The adaptation parameters are $\tau_a = 10\tau_x$ and $\beta = 0.5$.

### 5.2.1 Network traces during different trials decorrelate over time

The first task we consider is a delayed-recognition (DR) task (Fig. 5.1A), which is designed to test for how long the trace of an input stimulus can be read out. At time $t = 0$ (beginning of the trial), the network receives an input pattern which is fixed over trials but randomized over sessions. The network target output is zero at all times, except at a fixed delay $d$ after the stimulus, at which the network has to emit a brief pulse. We consider a trial successful if a pulse at the required time point can be isolated via a thresholding procedure on the output $z$ (see section 5.4.4).

If the stimulus is strong or long enough to drive the network to exactly the same state at every trial, then the task is perfectly solvable as long as 1) the dynamics remains chaotic for at least a time $d$ after the stimulus is released and 2) the state-space is sufficiently high-dimensional to linearly separate the state of the network at the pulse time from its state at all the previous time points. In realistic conditions however, the stimulus has a finite strength and duration, which implies that the state of the network when the stimulus is released varies from trial to trial. Since the dynamics is chaotic, trajectories associated to different trials decorrelate with a rate related to the maximum Lyapunov exponent, which implies that the performance decreases when we increase the delay $d$ (Fig. 5.1B,D).

First, we check for the dependence of the performance on the number of units $N$. We find that for $N \geq 200$, the performance consistently drops at $d \sim 300\tau_x$ (Fig 5.1B). The reason for the significantly worse performance for $N = 100$ is that due to finite-size effects the network settles in either a fixed point or in a limit cycle before the expected response time.

We then compare the performance of the adaptive network to that of the standard one. We did not observe any significant difference between the two networks (Fig. 5.1D), but we should stress that we did not conduct a careful exploration of the adaptation parameter space. When comparing the evolution of the readout over time for the two networks (Fig. 5.1B), we qualitatively observe that the adaptive network exhibits a higher trial-to-trial variability. This is most likely due to the longer correlation time of the adaptation variable, which implies that the stimulus is less effective in driving the adaptive network into a consistent state. Based on this argument, we would expect adaptation to even worsen the performance. However, this effect does not seem to be significantly strong.

### 5.2.2 Adaptation improves sequence discrimination over long timescales

The network performance on the DR task seems to be limited by the speed of decorrelation of the dynamics across different trials. To test the ability of the network to integrate information across long time lags, we test its performance on a delayed matching-to-sample (DMTS) task (Fig. 5.2A). In this task, the network is exposed to two stimuli $S_1$ and $S_2$, with $S_i \in \{A, B\}$, where $A$ and $B$ are patterns chosen randomly for every session and kept fixed within each session (see section 5.4.1). $S_1$ and $S_2$ are separated by a delay $d_1$ and a short go-cue is delivered after a second

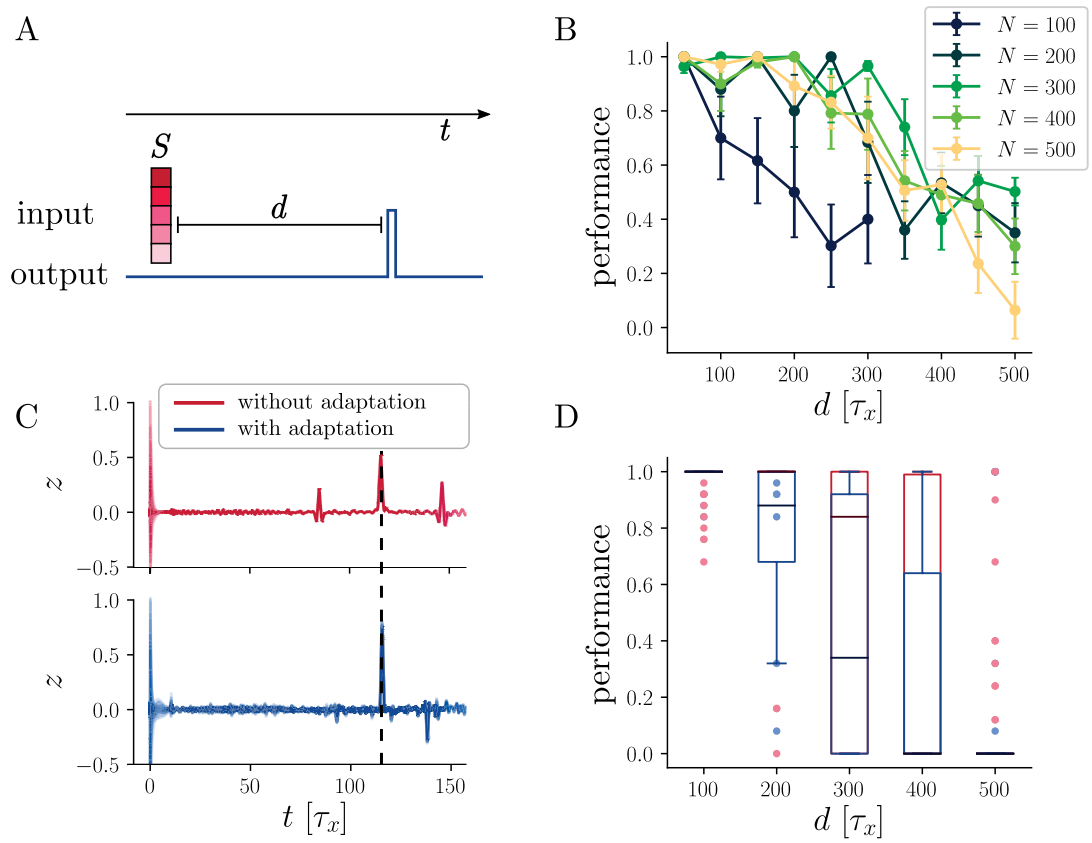



Figure 5.1 – **The network without adaptation has limited integration time. A:** Schematic of the delayed-response (DR) task. A fixed pattern $S$ is presented to the network for a time $t_S = 10\tau_x$ and the network has to output a pulse after a fixed delay of length $d$. **B:** Performance of the network without adaptation on the DR task, plotted against $d$ for different $N$. The performance is defined as the fraction of trials in which the network produce a response pulse with correct timing, averaged over initial conditions and over network realizations. The performance in each single trial is either one, if the pulse at correct timing can be isolated, or zero otherwise. **C:** Linear readout $z$ during multiple trials for the same session, for the network without adaptation (top) and with adaptation (bottom). In this plot, $d = 100\tau_x$. The dashed line indicates the target response time. The task and network parameters are summarized in Table 5.1. **D** Comparison of the performance for the network with (blue) and without (red) adaptation, for $N = 500$ and variable $d$.

delay $d_2$ from $S_2$. Similarly to the DR task, the target output of the network is zero for most of the trial duration. If $S_1 = S_2$ however, the network has to respond with a brief pulse after the go-cue.

For this task, adaptation yields a much higher performance across the whole range of delays $d$ (Fig. 5.2B). In fact, the standard network seems to be unable to solve the task even for rather short delays and it simply learns to respond with a pulse independently of the stimulus identities (Fig. 5.2C). The reason for this failure is that the second stimulus quenches the variability across trials, effectively making the network "forget" about the identity of the first stimulus. Thanks to the longer correlation time of the adaptation variable, however, the adaptive network preserves a memory of the first stimulus even after the second one, which allows to correctly solve the task. We can visualize this mechanism in a low-dimensional projection of the network activity (see Fig. 5.2D). For the standard network, the second stimulus drives the network into the same state independently of the identity of the first, and from that point the network continues along a stereotyped trajectory. In the case of the adaptive network on the other hand, different trajectories are brought closer to each other by the second stimulus but not to the point to be indistinguishable. It is interesting to notice the role of the go-cue, which causes an additional quenching of the variability, allowing a more reliable readout.

We were interested in seeing whether the network is able to retain information for long time lags, while at the same time being exposed to continuous additional stimuli. To this end, we tested the network on a sequence discrimination task, in which the network has to discriminate between a correct (or *Go*) sequence and a wrong (or *No-Go*) one (Fig. 5.3A). Both sequences are formed by a concatenation of random patterns. Crucially, the second half of the sequence is the same for both sequences, while the first half is different. This task is conceptually similar to the DMTS task. However, in the previous case the network underwent brief but strong stimulation, while in this case the stimulation is continuous but weaker.

We find that the adaptive network performs this task moderately better than the standard one (Fig. 5.3B). The explanation for this observation seems to be similar to that of the DMTS task: the coherent input during the second half erases the trace of the first half more easily for the standard network (Fig. 5.3C,D). In contrast, the trajectories of the adaptive network dynamics remain clearly well-separated throughout the trial (Fig. 5.3D).

### 5.2.3 Adaptation increases robustness to noise

Finally, we tested the performance of the two networks in the presence of input noise, both during training and during testing, for the three tasks described above. As expected, the performance is decreased with respect to the noiseless case for all tasks (Fig. 5.4). The DR and the DMTS tasks seemed to be the most sensitive to noise, while the performance of the networks on the sequence discrimination task seems to be more robust. Interestingly, the presence of adaptation makes the performance more robust to noise. Indeed, a difference in performance between the

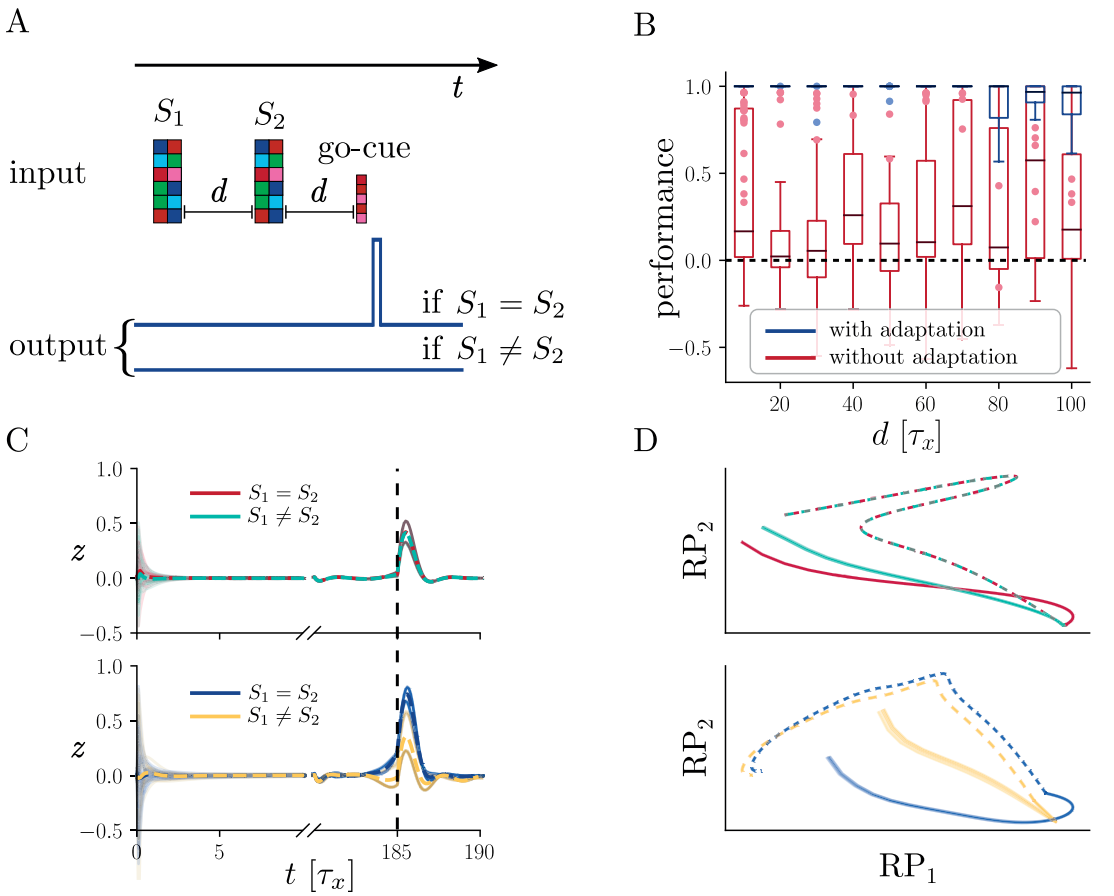



Figure 5.2 – **Performance on a delayed matching-to-sample task (DMTS). A:** Schematics of the task. Colored rectangles indicate random patterns. **B:** Comparison of the performance on the DMTS task of the network with or without adaptation. The dashed line indicates the chance level, which is at performance = 0 (see section 5.4.4 for the definition of the performance). **C:** Output traces for the network without adaptation (top) and with adaptation (bottom), for $d = 80\tau_x$. Dashed colored lines indicate the mean of the readout, across trials of the same type. The dashed black line indicates the target response time. **D:** Evolution of the network activity in a two-dimensional random projection space, for the network without adaptation (top) and with adaptation (bottom). The evolution of the activity is plotted during the presentation of the second input pattern (solid lines) and during the following delay period (dashed lines). Color conventions and task parameters are the same as in panel C. Notice that for the network without adaptation the input drives the network into the same state, independently of the initial condition.

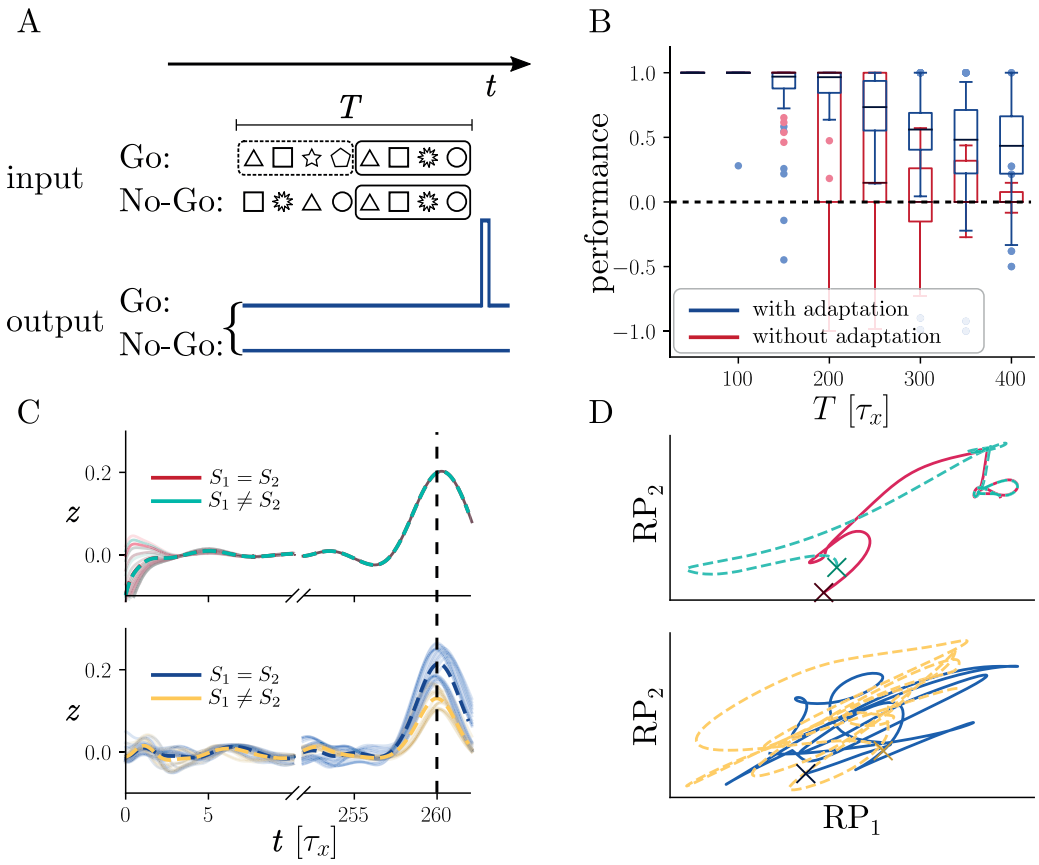



Figure 5.3 – **Performance on a two-sequence discrimination task. A:** Schematics of the task. Geometrical shapes are associated to specific random patterns. **B:** Comparison of the performance on the two-sequence discrimination task of the network with or without adaptation. The dashed line indicates the chance level, which is at performance = 0 (see section 5.4.4 for the definition of the performance) **C:** Output traces for the network without adaptation (top) and with adaptation (bottom). Dashed colored lines indicate the mean of the readout, across trials of the same type. The dashed black line indicates the target response time. **D:** Evolution of the network activity in a two random projection space, for the network without adaptation (top) and with adaptation (bottom), for $T = 250\tau_x$. The evolution of the activity is plotted from $t_0 = 120\,[\tau_x]$ to $t_1 = 250\,[\tau_x]$, for $T = 250$ i.e. shortly before and during the time when the two input sequences coincide. The crosses indicate the respective network states at $t = t_0$. Notice that for the network without adaptation the input drives the network into the same state, independently of the initial condition.

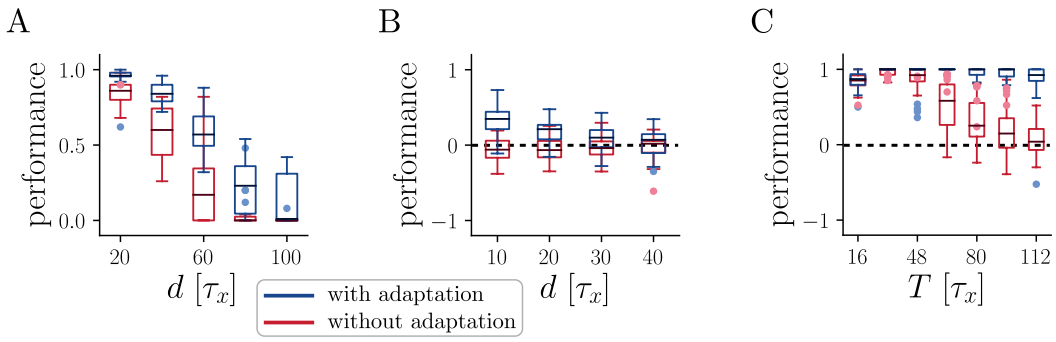



Figure 5.4 – **Effect of adaptation on noise robustness. A:** Performance of the two network with and without adaptation in the delayed-response task, in the presence of noise, for different values of the delay $d$. **B:** Same as A, but for the delayed matching-to-sample task. **C:** Same as A, but for the sequence-discrimination task. In this case the performance is plotted against the sequence length $T$. In panels B and C, the dashed line indicates the chance level.

| Task | Task parameters |
|---|---|
| Delayed reaction | $s = 10,\ t_s = 10\tau_x,\ R_{\min} = 10\tau_x,\ R_{\max} = 50\tau_x$ |
| Delayed matching-to-sample | $s = 10,\ t_s = 10\tau_x,\ R_{\min} = 10\tau_x,\ R_{\max} = 50\tau_x,\ s_{GC} = 1,\ t_{GC} = 5\tau_x$ |
| Sequence discrimination | $s = 1,\ R_{\min} = 20\tau_x,\ R_{\max} = 40\tau_x$ |

Table 5.1 – **Task parameters.**

two networks can be seen also for the DR task and in the sequence discrimination task, for which we did not observe a strong difference in the noiseless case. This effect is particularly visible for the sequence discrimination task, in which the adaptive network maintains an almost perfect performance until $d \sim 100\tau_x$, while the standard one drops to chance level.

## 5.3 Discussion

We have shown that recurrent networks with adaptation might have advantageous properties for some tasks that require integrating information over long time intervals. More precisely, the adaptation variables can maintain a trace of input stimuli for a longer time than the activation variable. This makes the network more robust to the memory erasing effect of incoming stimuli, and allows the network to generate specific output depending on the characteristics of the input even if relevant inputs are separated by long time lags. For the very same reasons, it is harder to set a desired initial condition on the network with adaptation, since this requires a stronger stimulus than for the standard one.

Adaptation seems to play an interesting role also in increasing the robustness of the network to noise, but the mechanisms underlying this effect need to be investigated more. It is important to notice that the feedback connections do not play an important role for the tasks we considered.

This is due to the fact that the output is required to be zero for most of the time at each trial, which prevents the network from learning fixed points or limit cycles. To verify this, we ran preliminary simulations with no feedback present (not shown), that resulted in a slightly worse performance. This could be because the feedback can still have a beneficial role in stabilizing the dynamics.

The idea that adaptive mechanisms could be helpful in solving temporal tasks was also proposed in a recent paper (Bellec et al. (2018)), in which the authors managed to obtain a performance comparable with state-of-the-art LSTM approaches using spiking neural networks. It is tempting to hypothesize that the effect that adaptation has in reducing the sensitivity to noise could be related to the results of Bellec et al. (2018). Due to the complexity of the neural model however, such claims are highly speculative.

Adaptation mechanisms are ubiquitous in real brain circuits, and have multiple timescales. Spike-frequency adaptation (SFA) in particular is present in multiple neuron types and in some cases it exhibits a scale-free profile (Lundstrom et al. (2008)), which is thought to have beneficial consequences for sensory processing (Fairhall et al. (2001)). For this reason, it would be interesting to study the effect of scale-free SFA in rate- or spiking-based learning tasks.

## 5.4 Methods

### 5.4.1 Network setup and simulation

In this chapter we considered two types of networks, one without adaptation (standard network) and one with adaptation (adaptive network). The neurons in the standard network follow the system of $N$ differential equations

$$\tau_x \dot{x}_i(t) = -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) + w_i^{FB} z(t) + I_i(t) \quad , \tag{5.1}$$

where $z(t) = \sum_j w_j^{RO}\phi(x_j(t))$ and $\phi$ is a nonlinear transfer function, which will be chosen to be the hyperbolic tangent in what follows. On the other hand, the adaptive network follows the modified equations

$$\begin{aligned} \tau_x \dot{x}_i(t) &= -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) - a_i(t) + w_i^{FB} z(t) + I_i(t) \\ \tau_a \dot{a}_i(t) &= -a_i(t) + \beta x_i(t) \quad , \end{aligned} \tag{5.2}$$

where $z$ and $\phi$ are defined in the same way as in the standard case. Notice that in both cases, due to the linear read-out $z$, the output-feedback loop represents a rank-one perturbation of the weight matrix J, given by $w^{RO} \wedge w^{FB}$. All time intervals and time constants were measured in units of $\tau_x$. For the adaptive network, we used in all tasks $\tau_a = 10\tau_x$ and $\beta = 0.5$.

The recurrent connections were initialized by sampling from a Gaussian distribution with mean

zero and variance $g^2/N$ and kept fixed during one session. $g$ is a parameter that varied depending on the task and on the analysis under consideration. The readout weights $w^{RO}$ were initialized at zero and trained using the FORCE procedure (Sussillo and Abbott (2009)), which consists in performing a recursive least-squares regression on the output, choosing P(0) = I for the initialization of the running estimate of the inverse of the correlation matrix P (see Sussillo and Abbott (2009), Eqs. 5,6). The feedback weights $w^{FB}$ were initialized by sampling from a uniform distribution between -1 and 1 and kept fixed during one session.

The equations were integrated using the Euler method with time step equal to $0.1\tau_x$, while the FORCE update rule was applied every two time steps, as in the original publication (Sussillo and Abbott (2009)). We also tested a fourth order Runge-Kutta integration method, which did not yield significantly different results.

### 5.4.2 Task implementation

**Delayed-Response task** In the Delayed-Response (DR) task, in each trial the network receives an input pattern of duration $s = 10\tau_x$ after which it has to wait for a certain delay $d$ and then respond with a short pulse of length $t_R = \tau_x$. For every learning session, an $N$-dimensional input pattern was chosen randomly by sampling each input component from a uniform distribution defined in the interval $(-s, s)$.

**Delayed Matching-To-Sample task** In the Delayed Matching-To-Sample (DMTS) task, in each trial the network receives two input patterns ($S_1$ and $S_2$) separated by a delay of duration $d$, after which it has to wait for a second delay of the same duration and eventually respond with a pulse if $S_1 = S_2$. At every session, two patterns were generated independently in the same way as for the DR task. In this task, the moment at which the network has to respond was signaled by a short go-cue, a very brief ($t_{\text{gc}} = \tau_x$) input pattern that was the same throughout a session and that was sampled independently from the other patterns.

**Sequence discrimination task** In the sequence discrimination task, in each trial the network receives one out of two sequences. The two sequences consist of eight patterns each, concatenated in time and of equal duration $\tau = T/8$. The total duration of each sequence is indicated by $T$ and is a variable task parameter. The first sequence was constructed by randomly choosing eight patterns from a set of ten random patterns (generated as in the other tasks), that were the same for all the sessions. The second sequence was constructed to have the same second half as the first one, but again choosing randomly the first half. In this way, the two sequences share the same input during the second half. The first half, in contrast, can have different degrees of similarity across sessions. The network has to respond with a pulse directly after the end of the correct sequence.

### 5.4.3 Learning procedure

For all tasks, the learning session was organized in multiple epochs, themselves being divided into multiple trials. Each trial was followed by a relaxation period of a duration randomly chosen between $R_{\min}$ and $R_{\max}$, after which the next trial was started. The randomness of the duration of the relaxation period has the purpose of randomizing the initial condition at the beginning of each trial. The distinction between sessions and epochs is merely due to the fact that at the end of each epoch the performance of the network was tested in order to monitor the advancement of learning. A learning session ended if either the network achieved perfect performance in the inter-epoch testing phase, or if the maximum number of epochs was reached. After learning, a testing session of $n = 50$ trials was used to assess performance.

We should stress that the distinction between learning and testing in this context is simply due to the presence or absence of weight change respectively. The input patterns were the same for both learning and testing since we were not interested in the network generalization capabilities but rather in the ability to integrate information over long timescales.

### 5.4.4 Performance measure

In order to compute the performance, we need to identify response pulses in the network output. This was done by first low-pass filtering the network output $z$ with a timescale $\tau_f = 10\tau_x$ in order to avoid detection of multiple pulses due to fast fluctuations. The filtered output was then passed through a Heaviside function $z_{\text{th}}(t) = \Theta(z(t) - \theta)$. For each testing session, the threshold $\theta$ was optimized to match the number of pulses in the network output to the number of pulses in the target.

For the DR task, the performance was simply the number of trials in which the output pulse could be isolated at the correct time point. For the DMTS and sequence-discrimination task, the performance was calculated as the number of correct pulses minus the number of wrong pulses, i.e. pulses not produced at the right moment, normalized by the total number of pulses. In formulas,

$$P = \frac{n_{cp} - n_{wp}}{n_t} \quad , \tag{5.3}$$

where $n_{cp}$ is the number of correctly predicted pulses in the network output, $n_{wp}$ is the number of wrongly predicted pulses and $n_t$ is the total number of pulses in the target. In this way the performance will lie between -1 and 1, and even if all the pulses are correctly predicted the performance will still be penalized if the network emits extra pulses.

## 5.5 Author contributions

SPM designed the project and wrote the simulation code. SPM and WG wrote the text.

# 6 Towards hierarchical sequence generation

This chapters summarizes unfinished work that I carried out in the first part of my PhD, before starting to work on the projects presented in chapters 2 to 5. A toy model for hierarchical sequence generation is motivated and discussed. Despite the fact that this work was chronologically the first, it represents a logical outlook for this thesis.

The research presented in this chapter was carried out in collaboration with Johanni Brea and Wulfram Gerstner.

## 6.1 Introduction

In chapter 2 and in chapter 3 we discussed the possibility of producing slow dynamics by connectivity tuning or by adding rate adaptation, respectively. We have seen that obtaining slow dynamics by acting on the recurrent connections leads to a trade-off between precision on the weights and slowness of the dynamics. On the other hand, slow biophysical mechanisms like adaptation seem to allow the network to act on both fast and slow timescales (see chapter 5). This can be advantageous in order to learn to generate patterns that feature slow timescales or to recognize signals that require integration of information over long time intervals.

Besides containing long timescales, sequences that humans face in behaviorally relevant situations also exhibit complex structures that are often organized in a hierarchical fashion. Manifest examples of such organization are language and music. More generally, motor sequences are usually formed by binding together motor primitives to form complex movements, which themselves can be chained together to achieve a certain goal. Hierarchical organization seems therefore widespread in behaviorally relevant sequences, but is this property also reflected in the sequence generation mechanisms? Ever since Lashley's seminal work (Lashley (1951)) there is general consensus that the answer to this question is yes, supported by an increasing amount of evidences coming from behavioral studies in humans (see Rosenbaum et al. (2007) for a review). For example in sequential tapping tasks, the existence of a hierarchical motor generation system is supported both by the timing patterns of different actions and by the distribution of mistakes

that subjects make (Rosenbaum et al. (1983)).

From the behavioral modeling and artificial intelligence perspective, the way in which humans manage to perform complex sequences of actions by subdividing them in simpler tasks has inspired hierarchical modeling approaches. In particular hierarchical reinforcement learning (Sutton et al. (1999); Barto and Mahadevan (2003)) might provide a way out from the curse of dimensionality from which classic reinforcement learning approaches suffer in large state spaces. More generally hierarchical Bayesian models, beside accounting for how sequences of actions are learned or performed, might also be a powerful way to abstract knowledge (Tenenbaum et al. (2011)). Recent approaches to language and relational reasoning combined traditional hierarchical models with connectionist architectures to obtain relevant insights of how humans might be able to infer relations and to understand sentences (Doumas et al. (2008); Martin and Doumas (2017)).

On the other hand, how hierarchical sequence generation could emerge from biologically plausible networks of neurons is an open question. Kiebel et al. (2009), proposed a combination of a dynamical system approach to neural dynamics and Bayesian inference as a model for hierarchical sequence recognition. The same dynamical system framework was used as a model for hierarchical sequence generation (Rabinovich et al. (2014); Fonollosa et al. (2015)). While these models bear interesting similarities with experimental findings of hierarchical timescales in both human and primate cortices (Hasson et al. (2008); Murray et al. (2014)), a description at the neural population level that also explains the learning process with biologically plausible learning rule, is still missing.

In this chapter, we propose a toy model that illustrates how the slow dynamics generated by the combination of adaptation and recurrent connections can be combined with a hierarchical network architecture to obtain a model that generates sequences that exhibit a hierarchical structure. Despite its simplicity, the model allows to generate a large class of sequences while requiring a relatively small number of units. We will not explicitly address the problem of learning the parameters of the model, but we discuss how Hebbian plasticity could allow learning of some parts of the hierarchical structure. We first describe the simple mechanism that allows to produce transitions between network states with flexible timescales, an approach that is closely related to the idea of stable heteroclinic channels (Rabinovich et al. (2008)). Then, we present the network architecture and show how it can generate sequences with hierarchical structure. Finally, we discuss the possibility of introducing plasticity in the system to partially learn the connectivity.

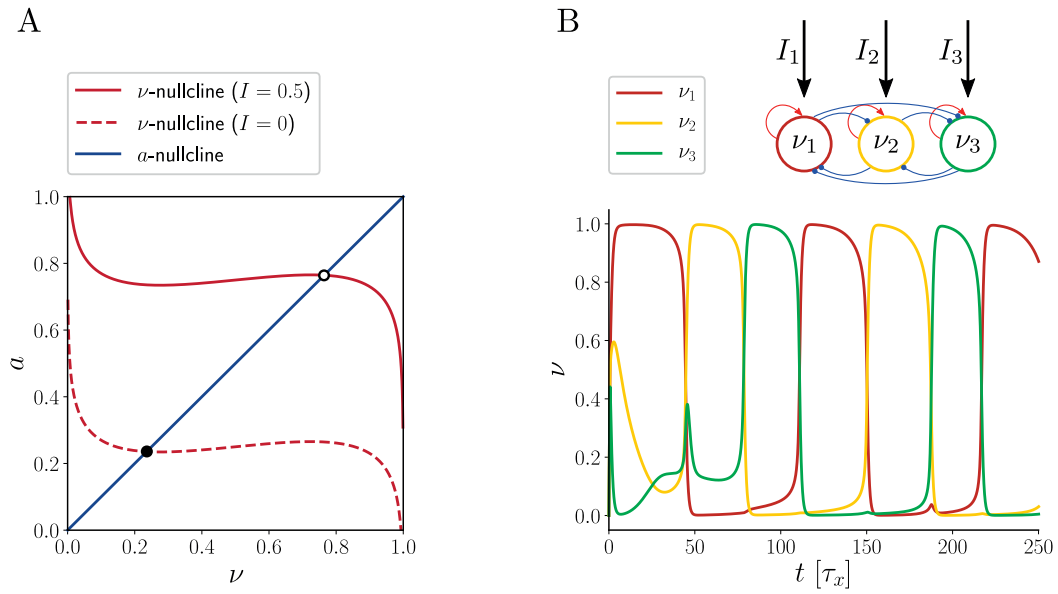



Figure 6.1 – **Implementation of elementary sequence generation. A:** Phase-plane diagram for a single unit with adaptation. In most cases there is only one fixed point, which can be stable (filled circle) or unstable (empty circle). The external bias $I$ changes both the position of the $\nu$-nullcline and the stability of the fixed point. **B:** Example of generation of an elementary sequence with three elements, using the small network depicted in the inset. In this example, $\tau_a = 100\tau_x$ and $I_1 > I_2 > I_3$.

## 6.2 Results

### 6.2.1 Generation of elementary sequences using bistable units with adaptation

We consider a network of $N$ rate units, each described by two differential equations:

$$\tau_\nu \dot{\nu}_i(t) = -\nu_i(t) + \sigma\left(w_i^R \nu_i(t) - \sum_{j\neq i} w^I \nu_j(t) - a_i(t) + I_i^{\text{ext}}(t)\right) \tag{6.1}$$

$$\tau_a \dot{a}_i(t) = -a_i(t) + \beta\nu_i(t) \quad , \tag{6.2}$$

where all parameters are positive and $\sigma(\cdot)$ is a sigmoid nonlinearity. Self-excitation in combination with lateral inhibition introduces competition among units, such that at any given time there is only one unit that has a high rate while all the other units have firing rates close to zero, a behavior known as winner-take-all (WTA). Due to adaptation however, the temporary "winner" will eventually transition to a low-rate level while a new high-rate unit takes over. In this way, the network produces a sequence of activations that can be seen as an elementary sequence and that can be used as a building block to generate more complex sequences.

The order in which different units activate is mostly determined by $I_i^{\text{ext}}(t)$. On the other hand, the time that each unit is in a state of high firing rate ($t_H$) is mostly influenced by the adaptation

parameters $\tau_a$ and $\beta$ as well as by the self-excitation $w_i^R$. We can qualitatively understand this last point by assuming that because of the WTA behavior, the temporary winner unit is momentarily uncoupled to the others, since their rates are approximately zero. The behavior of the winner unit can then be studied in a phase-plane (Fig. 6.1A). It is crucial that the parameters of each unit, namely the self-excitation $w_i^R$ and the adaptation parameters, are set up in such a way that the fixed point is stable in the absence of input while it is unstable once the input is present. Since the system is bounded, the theorem of Poincaré-Bendixson theorem implies that the system admits one limit cycle, which means the winner unit will tend to oscillate. If the sigmoid is very sharp, we can approximate it with a Heaviside function. In this approximation and in a regime of slow adaptation ($\tau_v \ll \tau_a$), we can find a relation between the value of the synaptic connections and the resulting $t_H$ (see also Additional details)

$$w_i^R = \beta\left(1 - e^{-\frac{t_H}{\tau_a}}\right) - I_i^{\text{ext}} \quad . \tag{6.3}$$

From Eq. (6.3) we see that an exponential fine tuning on $w_i^R$ is required to have a linear precision on $t_H$. Despite this limitation, the above setup represents a simple way to obtain an effective slow timescale that can be regulated by acting purely on the synaptic connections. We can see an example of a length-three elementary sequence generated in this way in Fig. 6.1B. We notice that during the first activation different units compete before the first element of a sequence is determined, a process similar to the one proposed for some models of decision making (Wong and Wang (2006)).

Since the excitatory units are not directly interacting, the different sequence elements maintain their "identities", in the sense that they are not bound to be in a particular order with respect to other elements. Their order on the other hand, is determined by the external bias provided by $I_i^{\text{ext}}$. Using this approach, one can generate only Markovian sequences, i.e. each element should be uniquely determined by the previous one. Moreover, while the length of an elementary sequence is formally limited only by the size of the dictionary (i.e. the number of different sequence elements), the longer the sequence to be generated the more fine tuning of the weights is necessary. In the next section, we describe how to combine multiple of these elementary sequences hierarchically so that each unit at a certain level of the hierarchy is biased by the units in the level above and provides bias to the units in the level below.

### 6.2.2 Generation of hierarchical sequences by combining elementary blocks

The key idea in combining multiple elementary blocks in a hierarchical fashion is to assume a hierarchical network with several layers where the external bias $I_i^{\text{ext}}$ to unit $i$ in layer $n$ is generated by units in layer $n+1$. As a simple example, let us consider an elementary block that consists of only two sequence elements, that we indicate as A-B. To produce this block, it is sufficient to have a single biasing unit in the level above in the hierarchy that has a stronger bias towards A than towards B. In this way, the unit corresponding to A is the first to become active, i.e. to transition to a state of high firing rate, after a "decision" mechanism similar to the one used

in models of decision-making processes (Wong and Wang (2006)). After a time $\sim t_H$, the unit corresponding to B will become the active one, provided that the biasing unit is still active. To have this last condition verified, we require that units in the higher layers have a longer activation time $t_H$ than those in the lower layers. We also introduce bottom-up weights, i.e. connections from a lower layer of the hierarchy to the one above it, in order to help synchronizing different layers. By repeating this procedure across units and across layers, we set up a multi-layer network in which units in the lowest layer are associated directly with sequence elements. In the second layer from the bottom, there are units that code for elementary sequences (or *chunks*) of two or three units. Finally, in the higher layers there are units that code for combinations of chunks. We setup an independent WTA circuit in every layer, so that there can only be one unit per layer active at any given time point. For simplicity, we introduce inter-layer connections only between layers belonging to neighboring levels of the hierarchy. However, direct connections between units belonging to more "distant" layers can also be introduced without harming the approach.

We provide some examples of these architecture in Fig 6.2, each one with eight sequence elements (indicated by letters A to H, on colored disks), combined to form a length-eight sequence. If the desired sequence is Markovian (see Fig. 6.2A, i.e. given one element the next one is uniquely determined, then it could be produced simply by chaining elements together. By doing this however, sequence elements would lose their "identities" since they would always appear as part of the same sequence. By constructing the hidden structure as shown in Fig. 6.2A (left), one would avoid this problem since no direct interaction is present among sequence elements. The same result could be obtained with a delay-line of the same length as the desired sequence instead of a hierarchical structure, in a single hidden layer. While this last solution has the advantage of being applicable to any sequence, it has two drawbacks. First, it is expensive in terms of required number of units, as it will be explained later; second, it does not allow to capture a possible hidden structure in the sequence. Using our approach we can use a separate hierarchical structure to generate a different sequence using the same elements, since we did non introduce any direct relation among sequence elements. Non-Markovian sequences can also be generated by this type of hierarchical structure (see Fig. 6.2B). A sequence is non-Markovian if to determine the next sequence element we need to know a certain number of previous elements and not just the last one. The hierarchical structure solves this problem by encoding the sequence history in a hierarchical representation.

The advantage of the approach that we propose is that it allows to reuse chunks in different parts of a sequence. For example, the network in Fig. 6.2C generates a sequence in which the chunk A-B appears twice, and the network is connected in such a way that the same chunk-related unit is activated twice during the sequence generation. We notice that in this case the sequence of chunks is itself non-Markovian. This however does not pose additional problems since the same strategy applied for the bottom layer can be used at any level of the hierarchy. This procedure for sequence generation is not restricted to chunks of length two, but can also be used with longer chunks as demonstrated in Fig. 6.2D, for length-3 chunks. However, as discussed in 6.2.1 longer chunks require more fine-tuning and are therefore less robust, which makes the use of shorter chunks advantageous.

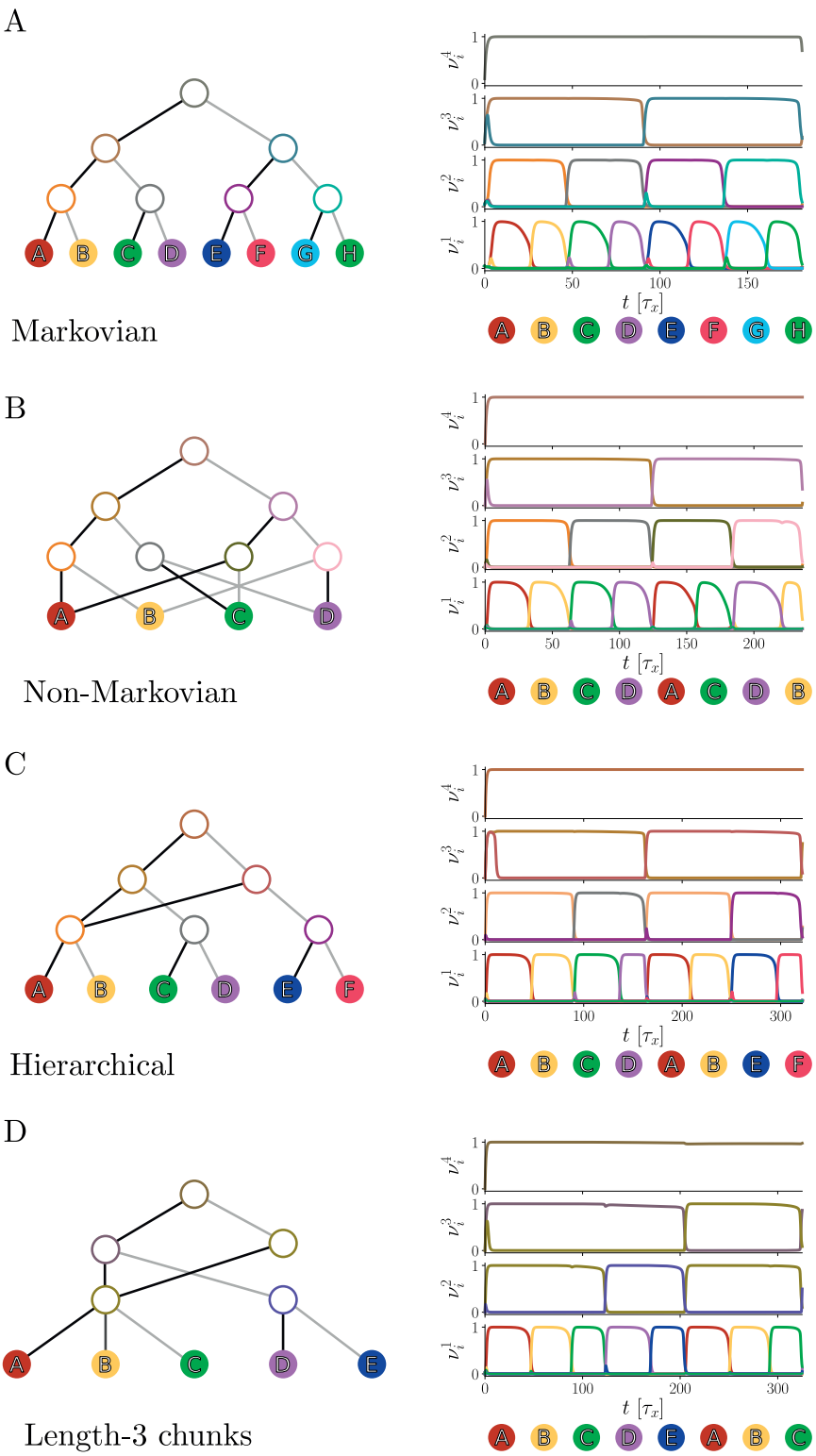



Figure 6.2 – *Caption next page.*

Figure 6.2 – *(Previous page)* **Generation of sequences with hierarchical structure.** For all panels, the desired sequence is on the bottom right whereas on the left we represent diagrammatically the structure of the multi-layer network that allows to produce it. In the network structure diagram, dark gray lines indicate stronger bidirectional connections, while light gray indicate weaker bidirectional connections. Self-excitation and lateral inhibition within each layer is not shown but always present. When the sequence-related unit in the topmost layer is activated by an external input, the network produces the desired sequence of activations (right). **A:** Example of Markovian sequence. **B:** Example of non-Markovian sequence with no hierarchical structure. **C:** Example of non-Markovian sequence in which length-2 chunks are combined in a non-Markovian fashion. **D:** Example of non-Markovian sequence in which length-3 chunks are combined in a non-Markovian fashion.

We notice that chunks can be reused not only in different parts of a single sequence, but also across sequences. For example, the chunk A-B and others appear multiple times in different sequences in the examples of Fig. 6.2. Whenever this happens, the same chunk-related unit can be reused, leading to a big advantage in terms of used resources. The same is not possible when using the delay-line approach, which instead would require an entirely separate hidden structure for each sequence. As a practical example, to generate all the examples of Fig. 6.2 we used 18 hidden units, while by using the delay-line approach one would need 32 units.

### 6.2.3 Learning asymmetric biases

In the previous section we showed that a hierarchical hidden structure provides a flexible way to generate complex sequences in a compact form thanks to the possibility of reusing different chunks of the sequence multiple times, within or across sequences. However, all the networks in Fig. 6.2 were hand-wired, so it is natural to ask whether they could also be learned in a biologically plausible way. While it seems unlikely that such a structure could emerge from a purely random initialization of the hidden connections without using non-local learning rules such as back-propagation, some assumptions on initialization of the weights might introduce the right inductive bias to converge to the desired hierarchical structure. Even more problematically, supervisory signals are available only at the terminal level, i.e. for the sequence element-related units, while for all the higher layers in the hierarchy learning should happen in a fully unsupervised way.

One possible approach could be to introduce some spatial structure in the connectivity, so that the network would spontaneously exhibit a richer dynamics (Litwin-Kumar and Doiron (2012); Setareh et al. (2017)). In particular, one could use either a bimodal or a power-law distribution of the out-degree, depending on whether one considers also connections between more distant layers or not, to obtain a hidden structure similar to the ones in Fig. 6.2. Moreover, since we are interested in obtaining largely symmetric structures, i.e. networks in which most of the connections are bi-directional, the in- and out-degree should be highly correlated. Together with variability in the level of self-excitation, this might lead to a natural hierarchy of activation

timescales.

Thank to the built-in hierarchy of timescales, units with slower effective dynamics will co-activate with multiple "faster" units, which through Hebbian learning will lead to potentiation of the synaptic connections. As a consequence, when "presenting" a sequence by activating the units corresponding to different elements, "slower" units will naturally tend to get associated to multiple "faster" units and therefore become chunk-related.

One last feature of the hand-wired networks of Fig. 6.2 is that the biases are asymmetrical, meaning that chunk-related units are more strongly connected to the lower-layer units that appear earlier in the chunk than to those that appear in a later position. In other words, the strength of the top-down connections decreases with the position of the downstream unit in the chunk. Developing such a position-dependent connectivity using biologically plausible rules might be challenging, but we can take advantage of the adaptation of the chunk-related unit. Indeed, the value of the adaptation variable of the pre-synaptic unit naturally encodes the time since that unit is active, and therefore it also encodes the position of the downstream units in the chunk. To exploit this property, we propose two approaches: one would be that plasticity directly depends on the pre-synaptic adaptation variable, which does not violate locality. More precisely, one would assume that it is harder to induce LTP when the pre-synaptic neuron is in an adapted state. The second option is to exploit the effect of adaptation only indirectly by its effect on the pre-synaptic rate, which influences plasticity in any Hebbian-type plasticity rule. More precisely, the higher the value of the adaptation variable, the lower the rate (if all the other inputs are the same), which in turn implies less potentiation in commonly used plasticity rules.

## 6.3 Discussion

We propose an approach to sequence generation that relies on a hierarchical hidden structure from which complex sequences can be read out. The main advantage is the possibility of reusing chunk-related hidden units both within and across sequences, which is particularly convenient when sequences share several subparts, as it often happens in behavior. The hierarchical hidden structure is built by combining multiple elementary sequence generators, constructed using the same computational mechanism at different scales.

The elementary sequence generator that we proposed relies on adaptation and competition between units to obtain a form of "winnerless competition" (Rabinovich et al. (2008)). Thanks to the large adaptation timescale, elementary sequences can be very slow with respect to the synaptic timescales, but their speed can still be tuned by modifying the amount of self-excitation. This is a useful feature in the context of behavioral sequences, which often need to be generated on the timescale of seconds. A limitation of the implementation that we proposed in this chapter is that it is unclear how to implement two consecutive appearances of the same element/chunk. Indeed the winnerless competition mechanism is by design producing transitions from an element/chunk to a second one and not to itself. While the duration of each element could be extended or even

doubled by acting on the parameters, inactivation and subsequent reactivation of the same element seems to be hard to achieve in such a setting.

There could be alternative implementations of the elementary sequence generator, that might have different properties with respect to the one discussed in this chapter. For example, one could use the same connectivity structure discussed in chapter 2 to efficiently generate units that evolve on multiple timescales without the need of any intrinsic slow mechanism. Moreover, while the winnerless competition mechanism proposed in this chapter requires an exponential fine tuning of the weights to obtain a linear precision on the timescale, using the approach of chapter 2 allows to have exponential control on the timescale by requiring exponential precision. Alternatively, different elementary sequences could be produced as trajectories of a recurrent network, as discussed in chapter 5. Consistently with the approach presented in this chapter, which trajectory gets executed could depend on a bias coming from an upstream network with slower dynamics. Using a reservoir instead of a WTA circuit would require an entirely different approach to learning with respect to the one discussed in section 6.2.3. However, promising recent results on unsupervised chunking using reservoir computing (Asabuki et al. (2017)) make this approach worth additional investigation. Finally, despite the limitations of using a delay-line to read out a full sequence (see section 6.2.2), it might still be advantageous to use short delay-lines to implement the elementary sequence generator. Indeed, it is impossible to construct networks in which different populations activate in a consecutive way with very different timescales, from very fast as in a synfire chain (Diesmann et al. (1999)) to much slower, by exploiting intrinsic slow mechanisms as adaptation (Setareh et al. (2018))

A similar approach to hierarchical sequence generation and chunking was proposed by Rabinovich et al. (2014); Fonollosa et al. (2015), in which a hierarchy of three layers is constructed by having a system of generalized Lotka-Volterra equations in each layer that gives rise to a winnerless competition among units. From the mathematical point of view the two implementations share strong similarities, in that both result in a series of transitions between quasi-stable states. One main conceptual difference is that in our approach any interaction between elements/chunks at the same level of the hierarchy is absent, while in the approach proposed by Rabinovich et al. (2014); Fonollosa et al. (2015) the sequential outcome is a consequence of both direct connections between elements and top-down interaction.

One interesting feature of our implementation of a hierarchical structure is that to become active each unit has to win after a "decision" process. If a new chunk needs to be started, then both the element-related unit and the chunk-related unit need to become the winner in their respective layers, and this will happen in a serial fashion from top to bottom. Therefore, chunks that are higher in the hierarchy will require more time to get initialized. This property resembles what is observed in motor sequence generation studies (Rosenbaum et al. (1983); Koch and Hoffmann (2000)), in which subjects are asked to repeatedly perform the same finger tapping sequence as fast as possible, while the latencies between consecutive taps and the possible mistakes are recorded. If the sequence being performed has a clear chunk-separable structure, then subjects will exhibit a distinctive pattern of latencies, in which taps that correspond to the beginning of a

chunk require longer time before getting initiated than those in the middle. Moreover, subjects are more likely to make mistakes on the first element of a chunk, which is compatible with our model since, in the presence of noise, each decision process would be a possible source of mistake.

Another interesting consequence of our approach is that it might enforce chunking and hierarchical sequence production even when the sequence under consideration does not have any hierarchical structure, as in the Markovian example in Fig. 6.2A. This seems to be a strategy also used by humans (Sakai et al. (2003)) when learning sequences. This strategy might not only be a bi-product of a hidden structure optimized for recognition of hierarchical structures, but also a way to overcome the limitations of working memory. The capability of humans in retaining in memory multiple items belonging to the same category is limited to a few items (Miller (1956)). However, we are clearly able to overcome these limitation in some situations, e.g. when retaining a phone number in memory. Chunking might be a strategy to solve this problem, i.e. grouping together items might reduce the interference of different working memory representation.

Similarly, chunking a sequence to construct a hierarchical structure is a way to compress it. One possible approach to quantify the compressibility of a sequence is in terms of its Kolmogorov complexity, which is defined as the length of the shortest computer program that produces the sequence as output (in a predefined programming language). The smaller the Kolmogorov complexity, the more a certain sequence can be compressed by storing the program that generates it. Our approach to sequence generation carries some similarities with this idea, in that the hierarchical hidden structure can be seen as a simple program that generates a sequence after having compressed some features such as chunk repetitions.

## 6.4 Additional details

### 6.4.1 Relation between synaptic connections and time in the high-rate state

If adaptation is very slow compared to the rate timescale ($\tau_\nu \ll \tau_a$), we can use the separation of timescales approach, i.e. we assume that for all values of $a_i(t)$, $\nu_i(t)$ immediately reaches the temporary fixed point, given by the solution of the system of equations

$$\bar{\nu}_i = \sigma\left( w_i^R \bar{\nu}_i + \sum_{j \neq i} w_{ij}^I \bar{\nu}_j - a_i(t) + I^{\text{ext}}(t) \right) \quad . \tag{6.4}$$

We are interested in the case in which the system has only one unit with high rate, while all the other units have firing rate close to zero, therefore we will set $\bar{\nu}_j \simeq 0 \, \forall \, j \neq i$. If we approximate $\sigma(\cdot)$ with the Heaviside function $\Theta(\cdot)$, we can can find the condition for the existence of the high-rate state for unit $i$

$$a_i(t) - I_i^{\text{ext}}(t) - w_i^R > 0 \quad . \tag{6.5}$$

Therefore, the high-rate fixed point is lost when $a_i(t) = a_i^* := I_i^{\text{ext}}(t) + w_i^R$ By solving the linear differential equation for the adaptation with $v_i(t) = \bar{v}_i$ and looking for the time point at which $a_i(t) = a_i^*$, we can find an approximation for the time in the high-rate state, that we indicate as $\bar{t}_H$, for the case of constant $I^{\text{ext}}$

$$\bar{t}_H := \tau_a \ln\left(\frac{\beta}{\beta - w_i^R - I_i^{\text{ext}}}\right) \quad . \tag{6.6}$$

Notice that in general, $\bar{t}_H$ is an approximation of $t_H$, since it does not take into account nonzero rates of the other units, different values of the adaptation variables when the unit is activated, and the real shape of the nonlinearity.

## 6.5 Author contributions

SPM, JB and WG designed the project. SPM performed the work, wrote the simulation code, and wrote the text, with the help of WG.

# A Additional publications

## A.1 Algorithmic Composition of Melodies with Deep Recurrent Neural Networks

Florian Colombo, **Samuel P. Muscinelli**, Alexander Seeholzer, Johanni Brea and Wulfram Gerstner

Published in: Proceeding of the 1st Conference on Computer Simulation of Musical Creativity, Huddersfield University (DOI: 10.13140/RG.2.1.2436.5683; Colombo et al. (2016))

### Summary

Algorithmic music composition exemplifies very well the challenges discussed in this thesis. Music has a hierarchical structure on multiple scales, which in a machine learning approach should be extracted from the examples in the dataset. For these reasons, a model for algorithmic composition that is both easily trainable and able to reproduce the long-range temporal dependencies typical of music is still lacking.

Here we investigate how artificial neural networks can be trained on a large corpus of melodies and turned into automated music composers able to generate new melodies coherent with the style they have been trained on. To capture the long timescales present in the dataset, we employ gated recurrent unit networks which are related to long-short term memory (LSTM) units. Thanks to multiplicatives gates, these type of units have been shown to be particularly efficient in learning complex sequential activations with arbitrary long time lags. Our model processes duration and pitch in parallel while modeling the relation between these two properties.

Once the network was trained on a dataset of Irish folk songs, we could verify the acquisition of the distinctive song feature by running the network in a generative mode. We performed a musical analysis of the generated songs, and observed that they have coherent metrical structure, present some recurrent distinctive features.

### Author contributions

The model was conceived by FC, SPM, JB and AS. The model implementation was done by FC. SPM performed the musical analysis. All authors designed the project and wrote the publication.

## A.2 Optimal stimulation protocol in a bistable synaptic consolidation model

Chiara Gastaldi, **Samuel P. Muscinelli** and Wulfram Gerstner

Preprint published at ArXiv:1805.10116 (2018) (Gastaldi et al. (2018))

### Summary

In the introduction of this thesis, we have seen how synaptic plasticity includes a large variety of mechanisms and timescales. Among these, synaptic consolidation is responsible for late-LTP and it allows to maintain the synaptic changes induced by neural activity over a timescale of hours. In experiments, synaptic consolidation can be induced by repeating a stimulation protocol several times. However, the effectiveness of consolidation depends crucially on the repetition frequency of the stimulations.

Here we propose a simple mathematical model that allows to systematically study the interaction between the stimulation protocol and synaptic consolidation. The model consists of a two-dimensional, nonlinear dynamical system in which both variables are intrinsically bistable and feature different timescales. We show the existence of optimal stimulation protocols in our model which, similarly to LTP experiments, depend on the repetition frequency of the stimulation. Interestingly, this sensitivity to a particular frequency is more pronounced when the difference in timescale between the two variables is strong.

Our results show that the complex dependence of LTP on the stimulation frequency emerges naturally from a minimal model with only two bistable variables. In the context of this thesis, this result highlights a different aspect of systems with multiple intrinsic timescales, namely the emergence of a non-trivial response to external stimulation.

### Author contributions

CG performed most of the derivations, with the help of SPM. SPM performed the bifurcation analysis. CG wrote the code and ran the experiments. All authors conceived the study and wrote the manuscript.

# Bibliography


S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.*, 27:77–87, 1977.

D. J. Amit. *Modeling brain function.* Cambridge University Press, 1989.

D. J. Amit and S. Fusi. Learning in neural networks with material synapses. *Neural Comput.*, 6: 957–982, 1994.

D. J. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite number of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.*, 55:1530–1533, 1985.

T. Asabuki, N. Hiratani, and T. Fukai. Chunking sequence information by mutually predicting recurrent neural networks. *bioRxiv*, 2017.

Y. Aviel and W. Gerstner. From spiking neurons to rate models: a cascade model as an approximation to spiking neuron models with refractoriness. *Phys. Rev. E*, 73:51908, 2006.

E. Bach and J. Shallit. *Algorithmic Number Theory*. MIT Press, 1996.

W. Bair and C. Koch. Temporal precision of spike trains in extrastriate cortex of the behaving macaque monekey. *Neural Comput.*, 8:1185–1202, 1996.

A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *DEDS*, 13(1):41–77, 2003.

G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *arXiv*, 2018.

R. Ben-Yishai, R.L. Bar-Or, and H. Sompolinsky. Theory of orientation tuning in visual cortex. *Proc. Natl. Acad. Sci. USA*, 92:3844–3848, 1995.

J. Benda and A. V. M. Herz. A universal model for spike-frequency adaptation. *Neural Comput.*, 15(11):2523–2564, 2003.

M. K. Benna and S. Fusi. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19(12):1697–1706, oct 2016. doi: 10.1038/nn.4401.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, jan 2017. doi: 10.1137/141000671.

T. V. P. Bliss and G. L. Collingridge. A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361:31–39, 1993.

N. G. Bowery, B. Bettler, W. Froestl, J. P. Gallagher, F. Marshall, M. Raiteri, T. I. Bonner, and S. J. Enna. International union of pharmacology. xxxiii. mammalian $\gamma$-aminobutyric acidb receptors: Structure and function. *Pharmacological Reviews*, 54(2):247–264, 2002. ISSN 0031-6997.

J. Brea, W. Senn, and J.-P. Pfister. Matching recall and storage in sequence learning with spiking neural networks. *J. Neuroscience*, 33:9565–9575, 2013.

R. Brette and E. Guignon. Reliability of spike timing is a general property of spiking neurons. *Neural Comput.*, 12:279–308, 2003.

N. Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J of Comput Neurosci*, 8(3):183–208, 2000.

N. Brunel and X.-J. Wang. Effects of neuromodulation in a cortical network model of object workingmemory dominated by recurrent inhibition. *J. Comput. Neurosci.*, 11:63–85, 2001.

J Buhmann and K Schulten. Noise-driven temporal association in neural networks. *Europhys. Lett.*, 4:1205–1209, 1987.

C. Clopath, Ziegler L., E. Vasilaki, L. Busing, and W. Gerstner. Tag-trigger-consolidation: A model of early and late long-term-potentiation and depression. *PLOS Comput. Biol.*, 4: e1000248, 2008.

C. Clopath, L. Busing, E. Vasilaki, and W. Gerstner. Connectivity reflects coding: A model of voltage-based spike-timing-dependent-plasticity with homeostasis. *Nature Neuroscience*, 13: 344–352, 2010.

F. Colombo, S. P. Muscinelli, A. Seeholzer, J. Brea, and W. Gerstner. Algorithmic Composition of Melodies with Deep Recurrent Neural Networks. *ArXiv e-prints*, 2016.

A. Compte. Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, 10(9):910–923, 2000.

A. Crisanti and H. Sompolinsky. Path integral approach to random neural networks. *arXiv e-prints*, 2018.

G. Deco and E. T. Rolls. Neurodynamics of biased competition and cooperation for attention:a model with spiking neurons. *J. Neurophysiol.*, 94:295–313, 2005.

T.and Naud R.and Gerstner W. Deger, M.and Schwalger. Fluctuations and information filtering in coupled populations of spiking neurons with adaptation. *Phys. Rev. E*, 90(6-1):062704, 2014.

B. DePasquale, C. J. Cueva, K. Rajan, G. S. Escola, and L. F. Abbott. full-force: A target-based method for training recurrent networks. *PLOS ONE*, 13(2):1–18, 02 2018. doi: 10.1371/journal.pone.0191527.

M. Diesmann, M.-O. Gewaltig, and A. Aertsen. Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402:529–533, 1999.

J. Doose, G. Doron, M. Brecht, and B. Lindner. Noisy juxtacellular stimulation in vivo leads to reliable spiking and reveals high-frequency coding in single neurons. *Journal of Neuroscience*, 36(43):11120–11132, 2016. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0787-16.2016.

Leonidas A. A. Doumas, John E. Hummel, and Catherine M. Sandhofer. A theory of the discovery and predication of relational concepts. *Psychological Review*, 115(1):1–43, 2008.

B. Dummer, S. Wieland, and B. Lindner. Self-consistent determination of the spike-train power spectrum in a neural network with sparse connectivity. *Front. Comp. Neurosci.*, 8:104, 2014.

A. Edman, S. Gestrelius, and W. Grampp. Analysis of gated membrane currents and mechanisms of firing control in the rapidly adapting lobster stretch receptor neurone. *The Journal of Physiology*, 384(1):649–669. doi: 10.1113/jphysiol.1987.sp016475.

H. Eichenbaum. Time cells in the hippocampus: a new dimension for mapping memories. *Nature. Rev. Neurosci.*, 15(11):732–744, 2014.

D. Elizondo. The linear separability problem: Some testing methods. *IEEE Transactions on Neural Networks*, 17(2):330–344, 2006.

B. Ermentrout. Neural networks as spatio-temporal pattern-forming systems. *Reports on Progress in Physics*, 61(4):353–430, 1998.

A. L. Fairhall, G.D Lewen, W. Bialek, and R.R.D. van Steveninck. Efficiency and ambiguity in an adaptive neural code. *Nature*, 412:787–792, 2001.

M. S. Fee, A. A. Kozhevnikov, and R. H. R. Hahnloser. Neural mechanisms of vocal sequence generation in the songbird. *Annals of the New York Academy of Sciences*, 1016(1):153–170, 2004.

I. R. Fiete, Y. Burak, and T. Brookings. What grid cells convey about rat location. *Journal of Neuroscience*, 28(27):6858–6871, 2008.

J. Fonollosa, E. Neftci, and M. Rabinovich. Learning of chunking sequences in cognition and behavior. *PLoS Comput. Biol.*, 11(11):1–24, 2015.

N. Fourcaud and N. Brunel. Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural Computation*, 14:2057–2110, 2002.

U. Frey and R.G.M. Morris. Synaptic tagging and long-term potentiation. *Nature*, 385:533 – 536, 1997.

U. Frey and R.G.M. Morris. Weak before strong: dissociating synaptic tagging and plasticity-factor accounts of late-ltp. *Neuropharmacology*, 37:545–552, 1998.

M.G.F. Fuortes and F. Mantegazzini. Interpretation of the repetitive firing of nerve cells. *J. General Physiology*, 45:1163–1179, 1962.

S. Fusi, P.J. Drew, and L.F. Abbott. Cascade models of synaptically stored memories. *Neuron*, 45:599–611, 2005.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proc. Natl. Acad. Sci. USA*, 105(48):18970–18975, 2008.

E. Gardner and B. Derrida. Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, 21(1):271–284, 1988.

C. Gastaldi, S. P. Muscinelli, and W. Gerstner. Optimal stimulation protocol in a bistable synaptic consolidation model. *ArXiv e-prints*, 2018.

C.D. Geisler and J.M. Goldberg. A stochastic model of repetitive activity of neurons. *Biophys. J.*, 6:53–69, 1966.

W. Gerstner. Population dynamics for spiking neurons: fast transients, asynchronous states and locking. *Neural Computation*, 12:43–89, 2000.

W. Gerstner, R. Kempter, J.L. van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–78, 1996.

W. Gerstner, W.M. Kistler, R. Naud, and L. Paninski. *Neuronal Dynamics. From single neurons to networks and cognition*. Cambridge Univ. Press, 2014.

G. Gigante, M. Mattia, and P. Del Giudice. Diverse population-bursting modes of adapting spiking neurons. *Phys. Rev. Lett.*, 98:148101, 2007.

A. Gilra and W. Gerstner. Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *eLife*, 6:e28295, nov 2017. ISSN 2050-084X. doi: 10.7554/eLife.28295.

V. Girko. Circular law. *Theory of Probability & Its Applications*, 29(4):694–706, 1985.

A. Gorchetchnikov and S. Grossberg. Space, time and learning in the hippocampus: How fine spatial and temporal scales are expanded into population codes for behavioral control. *Neural Networks*, 20(2):182–193, 2007.

F. Gray. Pulse code communication, 1953. US Patent 2632058.

R. Gütig and H. Sompolinsky. The tempotron: a neuron that learns spike timing-based decisions. *Nat Neurosci*, 9(3):420–428, 2006.

R. H. R. Hahnloser, A. A. Kozhevnikov, and M. S. Fee. An ultra-sparse code underliesthe generation of neural sequences in a songbird. *Nature*, 419(6902):65–70, 2002.

U. Hasson, E. Yang, I. Vallines, D. J. Heeger, and N. Rubin. A hierarchy of temporal receptive windows in human cortex. *J. Neurosci.*, 28(10):2539–2550, 2008.

A. Hayashi-Takagi, S. Yagishita, M. Nakamura, F. Shirai, Y. I. Wu, A. L. Loshbaugh, B. Kuhlman, K. M. Hahn, and H. Kasai. Labelling and optical erasure of synaptic memory traces in the motor cortex. *Nature*, 525(7569):333–338, sep 2015. doi: 10.1038/nature15257.

D. O. Hebb. *The Organization of Behavior*. Wiley, 1949.

G. Hennequin. Stability and amplification in plastic cortical circuits. 2013.

J Hertz, A Krogh, and R G Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.

J. P Hespanha. *Linear systems theory*. Princeton Univ. Press, Princeton, NJ, 2009.

S. Hestrin, R. A. Nicoll, D. J. Perkel, and P. Sah. Analysis of excitatory synaptic action in pyramidal cells using whole-cell recording from rat hippocampal slices. *The Journal of Physiology*, 422(1):203–225, 1990. doi: 10.1113/jphysiol.1990.sp017980.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol*, 117(4):500–544, 1952.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554–2558, 1982.

S. Hwang, V. Folli, E. Lanza, G. Parisi, G. Ruocco, and F. Zamponi. On the number of limit cycles in asymmetric neural networks. *ArXiv e-prints*, 2018.

E.M. Izhikevich. Resonate-and-fire neurons. *Neural Networks*, 14:883–894, 2001.

H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304:78–80, 2004.

J. Kadmon and H. Sompolinsky. Transition to chaos in random neuronal networks. *Phys. Rev. X*, 5:041030, 2015.

S. J. Kiebel, K. von Kriegstein, J. Daunizeau, and K. J. Friston. Recognizing sequences of sequences. *PLoS Computat. Biol.*, 5(8):1–13, 2009.

W. M. Kistler and W. Gerstner. Stable propagation of activity pulses in populations of spiking neurons. *Neural Computation*, 14:987–997, 2002.

D. Kleinfeld. Sequential state generation by model neural networks. *Proc. Natl. Acad. Sci. USA*, 83:9469–9473, 1986.

B. W. Knight. Dynamics of encoding in a population of neurons. *J. Gen. Physiology*, 59:734–766, 1972.

I. Koch and J. Hoffmann. Patterns, chunks, and hierarchies in serial reaction-time tasks. *Psychological Research*, 63(1):22–35, 2000.

P Konig, A K Engel, and W Singer. Integrator or coincidence detector? the role of the cortical neuron revisited. *Trends Neurosci*, 19(4):130–137, 1996.

A. Kruscha and B. Lindner. Partial synchronous output of a neuronal population under weak common noise: Analytical approaches to the correlation statistics. *Phys. Rev. E*, 94:022422, Aug 2016. doi: 10.1103/PhysRevE.94.022422.

G. La Camera, A. Rauch, H. Lüscher, W. Senn, and S. Fusi. Minimal models of adapted neuronal response to in Vivo–Like input currents. *Neural Comp.*, 16(10):2101–2124, 2004.

R. Laje and D.V. Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.*, 16:925–933, 2013.

L. Lapicque. Recherches quantitatives sur l'excitation electrique des nerfs traitée comme une polarization. *J. Physiol. Pathol. Gen.*, 9:620–635, 1907.

K. S. Lashley. The problem of serial order in behaviour. In *Cerebral Mechanisms in Behaviour; the Hixon Symposium*. HAFNER PUBLISHING COMPANY New York and London 1967, 1951.

S. Lim and M. S Goldman. Balanced cortical microcircuitry for maintaining information in working memory. *Nature Neuroscience*, 16(9):1306–1314, 2013.

A. Litwin-Kumar and B. Doiron. Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nat. Neurosci.*, 15(11):1498–1505, 2012.

B.N. Lundstrom, M.H. Higgs, W.J. Spain, and A.L. Fairhall. Fractional differentiation by neocortical pyramidal neurons. *Nature Neuroscience*, 11:1335–1342, 2008.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, pages 2531–2560, 2002.

Z. F. Mainen and T. J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268:1503–1506, 1995.

A. H. Marblestone, G. Wayne, and K. P. Kording. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience*, 10, 2016.

A. E. Martin and L. A. A. Doumas. A mechanism for the cortical computation of hierarchical linguistic structure. *PLoS Biol.*, 15(3):1–23, 2017.

S.J. Martin, P.D. Grimwood, and R.G.M. Morris. Synaptic plasticity and memory: an evaluation of the hypothesis. *Ann. Rev. Neurosci.*, 23:649–711, 2000.

F. Mastrogiuseppe and S. Ostojic. Intrinsically-generated fluctuating activity in excitatory-inhibitory networks. *PLoS Comput. Biol.*, 13(4):1–40, 2017.

F. Mastrogiuseppe and S. Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609 – 623.e29, 2018.

A. Mathis, A. V. M. Herz, and M. B. Stemmler. Resolution of nested neuronal representations can be exponential in the number of neurons. *Phys. Rev. Lett.*, 109(1), 2012.

M. Mattia and P. Del Giudice. On the population dynamics of interacting spiking neurons. *Phys. Rev. E*, xx:xx, 2002.

W. S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of mathematical Biophys.*, 5:115–133, 1943.

G. Miller. The magical number seven plus minus two. *Psych. Rev.*, 63:81–97, 1956.

G. Mongillo, D.J. Amit, and N. Brunel. Retrospective and prospective persistent activity induced by hebbian learning in a recurrent cortical network. *Europ. J. Neurosci.*, 18:2011–2024, 2003.

G. Mongillo, E. Curti, S. Romani, and D.J. Amit. Learning in realistic networks of spiking neuronsand spike-driven plastic synapses. *Europ. J. Neurosci.*, 21:3143–3160, 2005.

G. Mongillo, O. Barak, and M. Tsodyks. Synaptic theory of working memory. *Science*, 319: 1543–1546, 2008.

E. Montbrió, D. Pazó, and A. Roxin. Macroscopic description for networks of spiking neurons. *Phys. Rev. X*, 5(2):021028, 2015.

J. D. Murray, A. Bernacchia, D. J Freedman, R. Romo, J. D Wallis, X. Cai, C. Padoa-Schioppa, T. Pasternak, H. Seo, D. Lee, and X.-J. Wang. A hierarchy of intrinsic timescales across primate cortex. *Nat. Neurosci.*, 17(12):1661–1663, 2014.

S. P. Muscinelli, Wulfram Gerstner, and Johanni Brea. Exponentially long orbits in hopfield neural networks. *Neural Comput.*, 29(2):458–484, 2017.

R. Naud and W. Gerstner. Coding and decoding with adapting neurons: A population approach to the peri-stimulus time histogram. *PLOS Comput. Biol.*, 8:e1002711, 2012.

W. Nicola and C. Clopath. Supervised learning in spiking neural networks with force training. *Nat. Commun.*, 8(1):2208, 2017. doi: 10.1038/s41467-017-01827-3.

G.and Schwartz E.and Barbour B.and Brunel N.and Hakim V. Ostojic, S.and Szapiro. Neuronal morphology generates high-frequency firing resonance. *J. Neurosci.*, 35(18):7056–7068, 2015.

S. Ostojic. Interspike interval distributions of spiking neurons driven by fluctuating inputs. *J Neurophysiol*, 106(1):361–373, 2011.

J.-P. Pfister and W. Gerstner. Beyond pair-based stdp: a phenomenological rule for spike triplet and frequency effects. In *Advances in Neural Information Processing Systems 18*. MIT Press Cambridge, 2006.

J.-P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner. Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning. *Neural Computation*, 18:1318–1348, 2006.

C. Pozzorini, R. Naud, S. Mensi, and W. Gerstner. Temporal whitening by power-law adaptation in neocortical neurons. *Nat. Neurosci.*, 16:942–948, 2013.

R. Price. A useful theorem for nonlinear devices having gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72, June 1958. ISSN 0096-1000. doi: 10.1109/TIT.1958.1057444.

M. Rabinovich, P. Varona, I. Tristan, and V. Afraimovich. Chunking dynamics: heteroclinics in mind. *Front. Comput. Neurosci.*, 8:22, 2014.

M. I. Rabinovich, R. Huerta, P. Varona, and V. S. Afraimovich. Transient cognitive dynamics, metastability, and decision making. *PLOS Computational Biology*, 4(5):1–9, 2008.

K. Rajan, L. F. Abbott, and H. Sompolinsky. Stimulus-dependent suppression of chaos in recurrent neural networks. *Phys. Rev. E*, 82:011903, 2010.

M.J.E. Richardson. Dynamics of populations and networks of neurons with voltage-activated and calcium-activated currents. *Physical Review E*, 80:021928, 2009.

MJE Richardson, N. Brunel, and V. Hakim. from subthreshold to firing-rate resonance. *J. Neurophysiology*, 89:2538–2554, 2003.

D. A. Rosenbaum, S. B. Kenny, and M. A. Derr. Hierarchical control of rapid movement sequences. *J Exp Psychol Hum Percept Perform*, 9(1):86–102, 1983.

D. A. Rosenbaum, R. G. Cohen, S. A. Jax, D. J. Weiss, and R. van der Wel. The problem of serial order in behavior: Lashley's legacy. *Human Movement Science*, 26(4):525–554, 2007.

D. E. Rumelhart, J.L. McClelland, and the PDP research group. *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations*. MIT Press, 1986.

K. Sakai, K. Kitaguchi, and O. Hikosaka. Chunking during human visuomotor sequence learning. *Experimental Brain Research*, 152(2):229–242, 2003.

E. S. Schaffer, S. Ostojic, and L. F. Abbott. A complex-valued firing-rate model that approximates the dynamics of spiking networks. *PLoS Comput. Biol.*, 9(10):e1003301, 2013.

J. Schücker, S. Goedeke, D. Dahmen, and M. Helias. Functional methods for disordered neural networks. *arXiv*, 2016a.

J. Schücker, S. Goedeke, and M. Helias. Optimal sequence memory in driven random networks. *arXiv*, 2016b.

T. Schwalger and B. Lindner. Patterns of interval correlations in neural oscillators with adaptation. *Front. Comput. Neurosci.*, 7(164):164, 2013.

T. Schwalger and L. Schimansky-Geier. Interspike interval statistics of a leaky integrate-and-fire neuron driven by Gaussian noise with large correlation times. *Phys. Rev. E*, 77:031914–9, 2008.

T. Schwalger, M. Deger, and W. Gerstner. Towards a theory of cortical columns: From spiking neurons to interacting neural populations of finite size. *PLoS Comput. Biol.*, 13(4):e1005507, 2017.

A. Seeholzer, M. Deger, and W. Gerstner. Stability of working memory in continuous attractor networks under the control of short-term plasticity. *bioRxiv*, 2018. doi: 10.1101/424515.

H. Setareh, M. Deger, C. C. H. Petersen, and W. Gerstner. Cortical dynamics in presence of assemblies of densely connected weight-hub neurons. *Front. Comput. Neurosci.*, 11:52, 2017.

H. Setareh, M. Deger, and W. Gerstner. Excitable neuronal assemblies with adaptation as a building block of brain circuits for velocity-controlled signal propagation. *PLoS Comput. Biol.*, 14(7):1–30, 2018.

A. Shpiro, R. Moreno-Bote, N. Rubin, and J. Rinzel. Balance between noise and adaptation in competition models of perceptual bistability. *J. Comput. Neurosci.*, 27(1):37–54, 2009.

J. R. Silvester. Determinants of block matrices. *The Mathematical Gazette*, 84(501):460–467, 2000.

N.J.A. Sloane and J.H. Conway. The on-line encyclopedia of integer sequences. http://oeis.org/A002110, 2011.

H. Sompolinsky and I. Kanter. Temporal association in asymmetric neural networks. *Phys. Rev. Lett.*, 57:2861–2864, 1986.

H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in random neural networks. *Physical Review Letters*, 61(3):259–262, 1988.

S. Sreenivasan and I. Fiete. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nat. Neurosci.*, 14(10):1330–1337, 2011.

M. Stern, H. Sompolinsky, and L. F. Abbott. Dynamics of random neural networks with bistable units. *Phys. Rev. E*, 90:062710, Dec 2014. doi: 10.1103/PhysRevE.90.062710.

R. L. Stratonovich. *Topics in the Theory of Random Noise*, volume 1. Gordon and Breach, 1967.

D. Sussillo and L. F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

T. Sussillo, D.and Toyoizumi and W. Maass. Self-tuning of neural circuits through short-term synaptic plasticity. *Journal of Neurophysiology*, 97(6):4079–4095, 2007. doi: 10.1152/jn.01357.2006.

R. S. Sutton, D Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1):181 – 211, 1999.

J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011.

P. Theodoni, G. Kovács, M. W. Greenlee, and G. Deco. Neuronal adaptation effects in decision making. *J. Neurosci.*, 31(1):234–246, 2011.

T. Toyoizumi and L. F. Abbott. Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime. *Phys. Rev. E*, 84(5), 2011.

L. N Trefethen and M. Embree. *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005.

C. van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274:1724–1726, 1996.

D.and Schwalger T.and Lindner B. Wieland, S.and Bernardi. Slow fluctuations in recurrent networks of spiking neurons. *Phys. Rev. E*, 92(4):040901, 2015.

H. R. Wilson and J. D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.*, 12:1–24, 1972.

K.F. Wong and X.J. Wang. A recurrent network mechanism of time integration in perceptual decisions. *J. Neurosci.*, 26:1314–1328, 2006.

F. Zenke, E.J. Agnes, and W. Gerstner. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature Commun.*, 6:6922, 2015.

L. Ziegler, F. Zenke, D.B. Kastner, and W. Gerstner. Synaptic consolidation: from synapses to behavioral modeling. *J. Neuroscience*, 35:1319–1334, 2015.

R. Zillmer, N. Brunel, and D. Hansel. Very long transients, irregular firing, and chaotic dynamics in networks of randomly connected inhibitory integrate-and-fire neurons. *Physical Review E*, 79(3), 2009.

R S Zucker and W G Regehr. Short-term synaptic plasticity. *Annu Rev Physiol*, 64:355–405, 2002.

A. Zumdieck, M. Timme, T. Geisel, and F. Wolf. Long chaotic transients in complex networks. *Physical Review Letters*, 93(24), 2004.

# Samuel Pavio Muscinelli

## Education

2013–present **PhD Candidate – Neuroscience**, *EPFL*, Lausanne.
Prof. Wulfram Gerstner's lab

2010–2012 **Master of Science – Physics**, *Università La Sapienza*, Rome, *110/110*.
Average exam grade: 29.0/30

2007–2010 **Bachelor of Science – Physics**, *Università di Perugia*, Perugia, Italy, *110/110*.
Average exam grade 28.6/30

2002–2008 **Bachelor of Arts – Trumpet**, *Conservatorio di musica di Perugia*.
Final grade: 9.0/10

## Journal Papers

2018 **C. Gastaldi, S. P. Muscinelli, W. Gerstner**, *Optimal stimulation protocol in a bistable synaptic consolidation model.*, arXiv preprint arXiv:1805.10116.

2017 **S. P. Muscinelli, W. Gerstner and J. Brea**, *Exponentally long orbits in Hopfield neural networks*, Neural Computation 29(2), 458–484.

2013 **M. Bochicchio, S. P. Muscinelli**, *Ultraviolet asymptotics of glueball propagators*, Journal of high energy physics 2013 (8), 1–51.

## Conference Abstracts/Papers (selected)

2018 **S. P. Muscinelli, W. Gerstner and T. Schwalger**, *Oscillations and chaos in random adaptive neural networks*, CNS poster.

2017 **S.P. Muscinelli, W. Gerstner**, *Long timescale sequence recognition using adaptive neural networks.*, Conference on Cognitive Computational Neuroscience 2017.

2016 **F. Colombo, S. P. Muscinelli, A. Seeholzer, J. Brea and W. Gerstner.**, *Algorithmic Composition of Melodies with Deep Recurrent Neural Networks*, 1st Conference on Computer Simulation of Musical Creativity, .

## Ongoing Research

Dynamics of random recurrent rate networks in the presence of rate adaptation - in preparation

Learning of slow tasks with recurrent neural networks with adaptation - ongoing project

*Rue du Bugnon 4 – 1005 Lausanne – Switzerland*
*+41 78 695 90 39* • *+41 21 69 35265*
*samuel.muscinelli@epfl.ch*

## Teaching Experience

2013–2016 **Teaching Assistant**, *EPFL*.

List of taught courses:

- Biological modeling of neural networks, prof. Wulfram Gerstner
- Linear Algebra, prof. Michel Cibils
- Vector Analysis, prof. Michel Cibils
- Complex analysis, prof. Robert Dalang

## Languages

| | |
|---|---|
| Italian | Mother Tongue |
| English | Fluent |
| French | Fluent |

## Computer skills

| | |
|---|---|
| Programming languages | MATLAB, Python, Julia |
| Graphics | Inkscape, Illustrator, GIMP, Photoshop, Adobe Premiere |
| OS | Linux, Microsoft Windows |
| Miscellaneous | Git, Microsoft Office Suite |

## Other Interests

| | |
|---|---|
| Music | Currently playing in the student chamber orchestra of EPFL |
| Movie Making | Filming and directing multiple amateur short films |

*Rue du Bugnon 4 – 1005 Lausanne – Switzerland*
*+41 78 695 90 39* • *+41 21 69 35265*
*samuel.muscinelli@epfl.ch*