Network Dynamics as an Inverse Problem:

Reconstruction, Design and Optimality

Dissertation

for the award of the degree
"Doctor rerum naturalium"
Division Mathematics and Natural Sciences
of the Georg-August-Universität Göttingen
within the doctoral program
Physics of Biological and Complex Systems
of the Georg-August-Universität Göttingen

submitted by
José Luis Casadiego Bastidas
from
Valencia, Venezuela

Göttingen 2015

Thesis committee

Prof. Dr. Marc Timme, Network Dynamics, Max Planck Institute for Dynamics and Self-Organization
Prof. Dr. Ulrich Parlitz, Biomedical Physics, Max Planck Institute for Dynamics and Self-Organization
Prof. Dr. Reiner Kree, Institute for Theoretical Physics, Georg August University Göttingen

Members of examination board

1st Referee: Prof. Dr. Marc Timme, Network Dynamics, Max Planck Institute for Dynamics and Self-Organization
2nd Referee: Prof. Dr. Reiner Kree, Institute for Theoretical Physics, Georg August University Göttingen
Prof. Dr. Ulrich Parlitz, Biomedical Physics, Max Planck Institute for Dynamics and Self-Organization
Prof. Dr. Stephan Herminghaus, Dynamics of Complex Fluids, Max Planck Institute for Dynamics and Self-Organization
Prof. Dr. Theo Geisel, Nonlinear Dynamics, Max Planck Institute for Dynamics and Self-Organization
Prof. Dr. Patrick Cramer, Molecular Biology, Max Planck Institute for Biophysical Chemestry

Date of oral examination

January 13th 2016

A mis padres y hermanos,
Ustedes son la luz que ilumina mi camino...

Statement of originality

I hereby declare that this dissertation entitled **Network Dynamics as an Inverse Problem: Reconstruction, Design and Optimality** has been written independently and with no other sources and aids than those cited.

José Luis Casadiego Bastidas
Göttingen, October 30th 2015

Contents

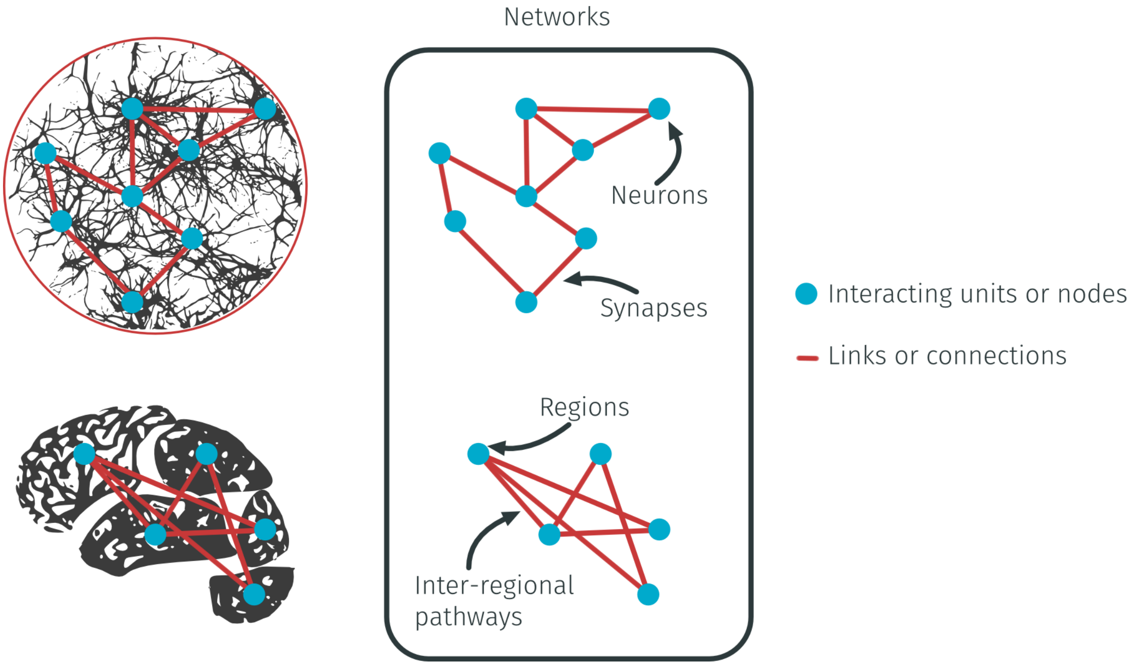
Introduction

Networks are everywhere and dominate practically all phenomena surrounding us; and phenomena happening within us. From the Internet to social networks, from metabolic networks to neural circuits, from power grids to transportation systems, networks can be found at every single stage of our daily life. Because of this, the study of networks has been at the forefront of interdisciplinary research for the past two decades and it promises to remain at so for many more.

Historically, we can trace back the earliest study on networks to the famous problem of the *Seven Bridges of Königsberg* [1]. Given the map of Königsberg (the present Kaliningrad), where the Pregel river divides the landscape into four regions connected by seven bridges, Leonhard Euler resolved the question of whether it is possible to take a walk starting from any region, crossing all bridges at once and returning to the starting point. Spoiler alert! it is not possible. Nonetheless and most importantly, Euler here introduced a new manner to look at problems, a new abstraction in terms of parts connected via links, setting the groundwork to what in later centuries and currently is known as graph theory and network science.

Despite the simplicity of Euler's ideas, this network concept bridged its way through to a broad variety of scientific fields in the course of history while helping in understanding phenomena of diverse nature. Such fields include (but are not restricted to) biology, chemistry, physics and social sciences. For instance, an interesting example is that of metabolic networks. The metabolism of a living cell is the collection of processes by which the cell sustains itself [2]. Such processes involve groups of consecutive chemical reactions that transform raw inputs into beneficial products for the cell. The diagrams depicting the relations between these inputs and products are known as metabolic networks [3]. Therefore, metabolic networks may help in understanding how fundamental processes within cells are related. Likewise, gene regulatory networks, which are networks that describe how genes and proteins interact in biological cells [4–10], have helped in understanding how the cell differentiation is carried out in cells of living organisms [3]. Furthermore, neural networks, which are structures that describe how neurons process and transmit information in our brains, have helped in explaining in simple terms how diseases such as schizophrenia and multiple sclerosis affect our natural function by changing the intrinsic wiring schemes of our brains [11, 12].

But, what exactly is a network? A network, in its simplest form, is just a collection



Networks are abstract representations of interconnected systems. (Left) The human brain at an (top) anatomical scale describing the synaptic connections among neurons, and at a (bottom) large scale describing the relations between brain regions through inter-regional pathways. (Right) Both descriptions may be reduced to simpler structures known as networks that represent how the interacting systems are connected to each other.

of points called *nodes* or *units* connected with each other through *links* or *edges* [3]. In spirit, this definition indicates that the behavior of networks is determined by the interactions of each single unit with the rest of the network. Thus, in general, the nodes represent interacting systems, and the links, the means by which these systems are interacting. For instance, in gene regulatory networks, the nodes represent the expression levels and concentrations of genes and proteins, respectively; and the links the regulatory features (who regulates whom) [4–10]. Moreover, in neural networks, the neurons are the interacting systems and the links are given by the synapses among them [11–13].

Traditionally, when studying networks theoretically, one tends to break them down into their fundamental parts and interaction forces (i.e. units and connections). Next, one characterizes the behavior of each part as if they were completely isolated from the rest of the network, hoping to capture the intrinsic dynamics of each unit in mathematical models involving specific parameters (e.g. dynamical systems). Later, one puts all pieces back together by coupling all mathematical models for units dynamics in a larger model, trusting that such model reproduces the key features of the entire network [14]. Finally, one characterizes this larger mathematical model (i) by exploring it as a function of its parameters, and (ii) by tracking the onset of non-trivial collective behavior, such as steady states, periodic orbits or synchronization, among others. Such approaches are known as *forward approaches* or *forward* *problems* on networks and they have helped in understanding diverse non-trivial phenomena in many different fields. For instance, they have been used to comprehend how different kinds of biological oscillators, such as cardiac pacemaker cells and flashing fireflies, synchronize their dynamics in time [14–16]. Also, they have helped in explaining how collections of pulse coupled neurons may generate spike trains in different dynamical regimes (e.g. synchronous-asynchronous and regular-irregular) [17], and moreover, how fixed anatomical connections in neural networks may yield distinct functional relationships [11].

However, modeling a network is a rather complicated task due to a myriad of reasons [14]. For instance, the connectivity scheme of a network may be too entangled. It has been estimated that in the human brain there are approximately 20 billion neocortical neurons, where each averages around 7,000 incoming connections [18]. Another reason is that the network structure may be evolving at a high rate in time. In the World Wide Web, for instance, websites, web pages and links are constantly being created and deleted [19]. Furthermore, there may also be different types of nodes and interactions simultaneously present in the network. For example, in gene regulatory networks, there may be activators and repressors (promoters and inhibitors of gene expression), and moreover, gene expression levels may be interacting with proteins and vice-versa [20, 21].

Alternatively, in recent years, there has been an increasing interest in using *inverse approaches* to design, control and understand the function of networks. Such approaches are particularly important when one needs to indirectly estimate specific parameters and functions from available data. As we will explain in forthcoming paragraphs, inferring the connectivity of networks from their nodal dynamics constitutes an important and challenging inverse problem [22].

In general, inverse approaches rely on measurements to infer the parameters and functions that determine a specific system [23]. Yet, whereas in forward approaches there is a unique relation from parameters and functions to measurements, inverse approaches generally produce families of solutions for parameters and functions given a specific set of measurements [23]. Thereby, extending the inference problem to selecting the actual parameters and functions that generated the given measurements. A simple example is that of a moving particle. If we know all the forces acting on this particle, we can easily predict the trajectory that the particle will describe in space. Conversely, if we only know the trajectory, (at most) we can infer the set of all possible forces that may generate that specific trajectory, unless more information is provided. Interestingly, this aspect may be advantageous for designing networks. Designing a network often means to engineer network static structures for particular function. And thanks to the intricate relation existing between the function and structure in networks, very disparate networks may yield identical dynamics [24]. This phenomenon has been reported in networks of spiking neurons [25–27] and gene regulatory networks [28, 29]. Moreover, there is a rapidly growing trend in designing synthetic gene circuits for specific biological function, cf. Refs. [30, 31].

Nonetheless, often just knowing the family of networks that generate a specific dynamics is not sufficient. Generally, knowing the actual network structure behind a certain dynamics may provide valuable information. In gene regulatory networks, for instance, the effect of specific genetic anomalies may spread across the network and modify the expression of other genes, thereby, leading to disease phenotypes [32]. Thus, inferring the actual genes involved in the progression of specific diseases may provide means to identify better targets for drug development [33].

Frequently, the connectivity of networks is not directly accessible. Whether one studies gene regulatory networks, metabolic networks or neural networks, sometimes we may have no means of directly measuring the network connectivity [20, 21, 33–39]. Instead, one is forced to resort to indirect approaches to estimate the network connections from available data. For instance, spike trains of neural networks may be used ti reveal network connections between spiking neurons [34, 35, 40]. Moreover, gene expression levels may also reveal the regulatory features of gene regulatory networks [4,7,9,20,21,33,37]. However, when inferring the connectivity of networks, one may stumble upon two basic types of connectivities, functional and structural connectivities.

A functional connectivity is constructed by computing correlation-based measures on the dynamics of pairs of units [22]. For example, if correlations between two units exceed a certain value, one assumes that there is a link between these two units. Conversely, if the measure is below, then one assumes there is no connection. Some examples of these measures are mutual information [41], transfer entropy [42], Granger causality [43, 44] and extensions thereof. Although approaches of this nature are widely abundant in the literature [37, 45–49], they generally show a series of conceptual challenges when used to infer network connectivities. For instance, the functional connectivity may depend on the current network state, therefore, a given physical network may span different functional connectivities for different collective states [11]. Thus, functional connectivities may not reflect the actual network structure. Also, one may measure high correlations between two units even though they are not directly connected [22]. Such high correlations may be induced by a third unit controlling the other two, or by the entire network or both. Furthermore, even the correlations of two physically connected units may be decorrelated by external noise or common external drivings [22].

On the other hand, the structural connectivity (as its name indicates) captures the actual structure of the network. In other words, the structural connectivity is determined by the physical links existing among network units [22]. For instance, the structure of networks of spiking neurons is given by the synaptic connections existing among neurons [11, 12], or in power grids, the structural connectivity is defined by the physical power lines connecting the elements in the grid [50–53].

The network structure is of particular importance because it may provide crucial features about the network that cannot be inferred from the individual description of any of its units. For instance, the network connectivity of social networks may provide means for measuring how information or diseases would spread in a certain population [14, 54–57].

So, how can one infer the structural connectivity of a network from its units dynamics? Recent approaches have tackled this problem from two basic mechanistic angles. It must be noted that here we employ the word mechanistic to indicate that these approaches assume that the network dynamics is defined by deterministic rules, such as dynamical systems.

The first angle, a driving-response strategy, relies on perturbing or driving simple and stable collective states (e.g. steady states or limit cycles) through external signals applied to the network. For instance, networks operating close to a steady state relax back to such state after transient perturbations, only if these perturbations are sufficiently small such as to not kick the network out of the basin of attraction of the stable state [58–62]. Specifically, the driving effectively changes the initial state of the network while preserving the network characteristic features (e.g. structural connections, coupling functions and intrinsic dynamics) intact. Perturbing the steady state a sufficient number of times provides a linear mapping from network responses (caused by perturbations) to the actual structural connectivity of the network [4, 6, 7]. Similarly, steady states and limit cycles may be translated in the state space with sufficiently weak and constant external driving signals. Given that these translations are partly determined by the network connections [63], generating many different translations may reveal the network structural connections [64–67].

The second angle, observing the network dynamics, relies on prior knowledge about the parameters and functions involved in the network dynamics. In particular, knowing the actual parameters and functions generating the network dynamics reduces the problem of inferring network structural connections to just fitting a collection of unknown parameters (in this case the connections) to measured data [34, 35, 40, 68–73]. Such approaches are especially reliable when estimating connections from time series [68–74], even in the presence of challenging dynamics (e.g. chaotic and noisy dynamics) [69]. Furthermore, such approaches may also reveal the structural connectivity of hybrid systems [75] where the coupling dynamics occurs at discrete events in times, such as in networks of spiking neurons [34, 35, 40].

However, despite the considerable advances in the state of the art of network inference approaches, there are numerous fundamental questions yet to be answered. For instance, how can one find classes of networks capable of generating a common dynamics? How are these networks related? Also, is a stable state really necessary for revealing the network connectivity from driving-response approaches? Can similar strategies perhaps reveal connections in networks describing more complex dynamics? Furthermore, is a prior knowledge of network features, such as the intrinsic dynamics, coupling functions and other parameters, necessary to understand how units interact with each other? Or can the recorded dynamics of a network alone uniquely determine its underlying structure?

In this thesis, we take a general view on the study of networks from inverse perspectives. By proposing a general representation for the dynamics of networks in terms of explicit dependencies among units, we develop new concepts for understanding, among other things, (i) how disparate networks achieve identical dynamics, and (ii) how we may reconstruct the structural connectivity of networks regardless of the type of network under study (e.g. gene regulatory networks, metabolic networks or neural networks). Specifically, here we heavily rely on concepts and tools coming from nonlinear dynamics and linear algebra to introduce physics-inspired inverse approaches for explaining the fundamental mechanisms for revealing connections in networks. Furthermore, the content and results of this thesis are self-sufficient, such that the reader may not need to refer to additional scientific sources. Still, we advise the reader to check Ref. [22] for an extensive review on the state of the art of the field previous to this contribution.

The thesis is organized as follows. This introduction is followed by chapter 1, where we briefly analyze the role of dynamical systems for representing the dynamics of networks. We also highlight the limitation of mainstream representations of network dynamics in terms of coupled dynamical systems having pairwise interactions. We end the chapter with an alternate representation of network dynamics which captures network connections in coarser manners. Such representation is of particular importance given that it constitutes the starting point of all other chapters. In chapter 2, we explain how disparate networks of generic dynamical systems may generate identical collective dynamics in time. Also, we derive a theoretical framework to parametrize the family of networks reproducing a specific dynamics, given a time series of its units. In chapter 3, we develop a model-independent approach for inferring network structural connections solely from time series. Specifically, we propose a functional decomposition of network dynamics in terms of network interactions of several orders. We demonstrate that this decomposition combined with the Algorithm for Revealing Network Interactions (ARNI) (also introduced in that chapter) reveals the structural connection of networks simply from recorded time series. In chapter 4, we show another model-independent approach for revealing the connectivity of networks based on local samplings of the network dynamics. In general, these local samplings provide linear mappings from collections of network states to structural connections. In chapter 5, we further study the application of local samplings by developing a model-independent approach for reconstructing structural connections from average responses of networks to driving signals. Interestingly, averaging the dynamics filters chaotic and noisy effects out from the measurements. This enhances the capabilities of local samplings to reveal structural connections of networks in the presence of strongly stochastic dynamics. In chapter 6, employing local samplings, we demonstrate that representing the network dynamics in different variables than the original variables where interactions occur may provide better dynamical features for inferring connections. In chapter 7, we show that under proper representations of the dynamics of networks, also discrete events in time may reveal structural connections of networks. Specifically, we demonstrate that by reconstructing the synaptic connections of pulse-coupled networks of spiking neurons (under a variety of dynamical regimes) simply from spike trains. Finally, we summarize our results and give a brief outlook to future research.

Network dynamics as an inverse problem

Chapter 1

Network dynamics

In this thesis, we analyze (i) how different networks generate a specific dynamics, and (ii) how one can infer network parameters, specifically the network structure, from the dynamics of networks through inverse approaches. As a starting point, we introduce in this chapter a general definition for the dynamics of complex networks in terms of coupled dynamical systems. However, contrary to mainstream descriptions in terms of pairwise interactions (cf. Ref. [22]), here we describe the dynamics of networks in terms of a more general class of interactions. In addition, we replace the standard notion of adjacent nodes by a general concept in terms of explicit dependencies among units' dynamics. In particular, a single explicit dependency parameter groups different orders of interactions coming from a specific unit into a single value. Thus, it may be considered as a coarser definition of connection between two units. Interestingly, this general definition of network dynamics eases all calculations along this thesis.

The first section of this chapter introduces the concept of dynamical systems and their role describing how systems evolve in time. In the second section, we describe the temporal evolution of complex networks in terms of coupled dynamical systems having pairwise interactions and give a set of applicactions of such systems in the real-world. Next, using a simple example inspired in gene regulatory networks, we show the limitations of networks of dynamical systems having pairwise interactions for capturing the structure of networks presenting higher-order interactions. Finally, in the third section, we present (i) our general model for describing the dynamics of networks, and (ii) our definition of explicit dependencies among units.

1.1. Dynamical systems

Loosely speaking, a *dynamical system* is a rule that determines the temporal evolution of a set of variables. Despite the simple it seems, this concept has been broadly used to understand phenomena happening around us. For instance, the swing of a pendulum, a population's growth or even a predator-prey phenomenon may be easily explained in terms of dynamical systems [76].

Formally, a dynamical system is defined as [77]

$$\dot{\boldsymbol{x}} = h(\boldsymbol{x}), \tag{1.1}$$

where $\dot{\boldsymbol{x}} = d\boldsymbol{x}/dt$ and $\boldsymbol{x} = \left[x^1(t), x^2(t), \dots, x^D(t)\right] \in \mathbb{R}^D$, is the triple $(\mathbb{R}^+, \mathcal{S}, \Phi)$ where $\mathcal{S} \subset \mathbb{R}^D$ is a *state space* and Φ is a flow

$$\Phi : \mathbb{R}^+ \times \mathcal{S} \to \mathcal{S}, \tag{1.2}$$

that is determined by the solution $\boldsymbol{x}(t) = \Phi(t, \boldsymbol{x}(0))$ of (1.1) with $\boldsymbol{x}(0)$ denoting an initial condition and $t \in \mathbb{R}^+$ denoting time.

Concretely, the state space $\mathcal{S}$ is an abstract space that depicts all possible states of the system. Thus, each possible state $\boldsymbol{x}(t)$ of (1.1) represent a unique point $s(t) \in \mathcal{S}$. Moreover, the mapping $\Phi(\cdot, \boldsymbol{x})$ is known as the *evolution rule* and defines the temporal evolution of the system. Hence, $\Phi(t, \boldsymbol{x})$ determines the subsequent state following from a current state in $\mathcal{S}$.

Combining both $\mathcal{S}$ and Φ, one may define a *trajectory* $\boldsymbol{s}$ *in the state space* $\mathcal{S}$ as a sequential collection of states $\boldsymbol{s} := [s(1), s(2), \dots, s(T)] \in \mathbb{R}^T$, where T is the number of states in the trajectory. Thus, a trajectory $\boldsymbol{s}$ characterizes the behavior of the system for a given initial condition $\boldsymbol{x}(0)$, cf. Fig. 1.1.

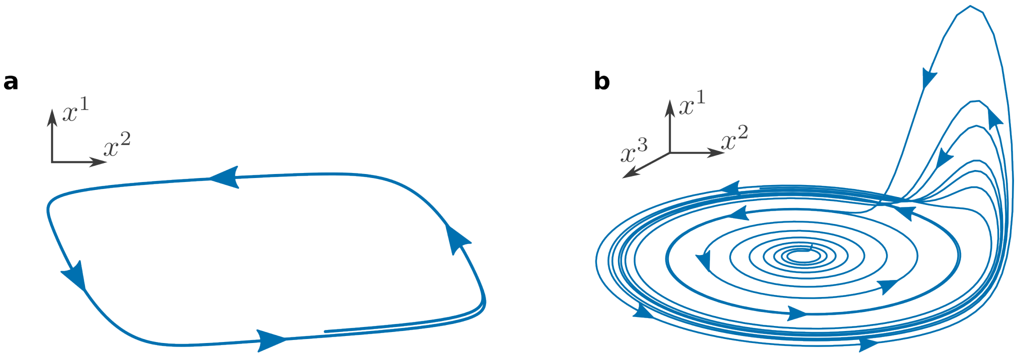


Figure 1.1. – Trajectories in the state space determine the temporal evolution of systems. Single trajectories in the state space for **a**, 2-dimensional FitzHugh-Nagumo [78] and **b**, 3-dimensional Roessler oscillators [79]. Trajectories in the state space may reveal, for instance, the existence of periodic orbits or chaotic attractors [76].

1.2. Networks of dynamical systems

As mentioned in sec. 1.1, dynamical systems are useful tools to describe the temporal behavior of systems. However, many aspects of our daily life result from the interaction of several dynamic parts. For instance, the biochemical reactions that regulate the metabolism of cells in living organisms; power grids and other distribution grids of commodities which make our everyday life more manageable; and most importantly, the neural circuitry in our brains that makes us functional beings; are just some of many fundamental examples.

In particular, one may represent such interconnected and dynamic systems in terms of networks of dynamical systems. These networks are basically structures depicting how collection of *units or nodes* — representing the dynamical systems — are interacting with each other. Hence, in this representation, the evolution of single dynamical systems in the network will be partly determined by (i) their individual or intrinsic dynamics, and by (ii) the interactions with the rest of the network.

1.2.1. Networks with pairwise interactions

It is customary to write networks of dynamical systems as [22]

$$\dot{\boldsymbol{x}}_i = \boldsymbol{h}_i(\boldsymbol{x}_i) + \sum_{j=1}^{N} J_{ij}\boldsymbol{g}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \boldsymbol{I}_i(t) + \boldsymbol{\xi}_i(t), \tag{1.3}$$

where $i, j \in \{1, 2, \ldots, N\}$, $\boldsymbol{x}_i(t) = \left[x_i^1(t), x_i^2(t), \ldots, x_i^D(t)\right] \in \mathbb{R}^D$ describes the state of the i-th unit at time $t \in \mathbb{R}$ and the functions $\boldsymbol{h}_i : \mathbb{R}^D \to \mathbb{R}^D$ and $\boldsymbol{g}_{ij} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^D$ represent the *intrinsic* and *pairwise coupling dynamics* of the D-dimensional dynamical units, respectively. The terms $\boldsymbol{I}_i(t) \in \mathbb{R}^D$ and $\boldsymbol{\xi}_i(t) \in \mathbb{R}^D$ are a vector of *external driving signals* and *external noise* acting on the i-th unit, respectively. The $J_{ij} \in \mathbb{R}^{D\times D}$ define the *topology* or *structural connectivity of the network* and set the strength of interactions among x_i^d for all $i \in \{1, 2, \ldots, N\}$ and $d \in \{1, 2, \ldots, D\}$.

Yet, for simplicity and throughout this thesis, we will mostly focus on systems determined by just one state variable, $x_i \in \mathbb{R}$. However the concepts introduced in this also apply to more general models (1.3).

So, for scalar state variables, model (1.3) reduces to

$$\dot{x}_i = h_i(x_i) + \sum_{j=1}^{N} J_{ij}g_{ij}(x_i, x_j) + I_i(t) + \xi_i(t), \tag{1.4}$$

where $h_i : \mathbb{R} \to \mathbb{R}$, $g_{ij} : \mathbb{R}^2 \to \mathbb{R}$, $J_{ij} \in \mathbb{R}$, $I_i(t) \in \mathbb{R}$ and $\xi_i(t) \in \mathbb{R}$. In its simplest setup, model (1.4) may be written in terms of its *adjacency matrix* $A \in \{0, 1\}^{N\times N}$, where $J_{ij} = A_{ij} = 1$ if j directly acts on i, and $J_{ij} = A_{ij} = 0$ otherwise, cf. Fig. 1.2a.

Models (1.3) and (1.4) may effectively describe the behavior of a broad variety of systems. For instance, large-scale brain networks are often modeled as [11]

$$\dot{\boldsymbol{x}}_i = \boldsymbol{h}_i(\boldsymbol{x}_i) + \sum_{j=1}^{N} J_{ij}\boldsymbol{S}\left(\boldsymbol{x}_j(t - \tau_{ij})\right), \tag{1.5}$$

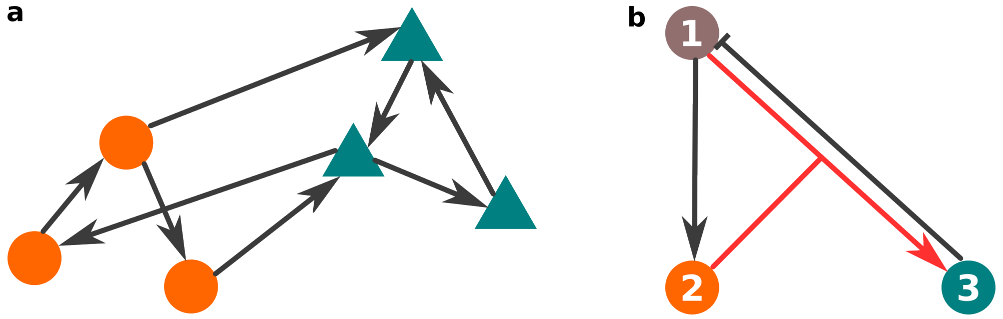


Figure 1.2. – Purpose and limitations of adjacency matrices. **a**, Scheme of a neural network having (△) excitatory and (○) inhibitory neurons, cf. [11]. The adjacency matrix describes the pairwise interactions among coupled units by defining links A_{ij} — represented by the arrows — starting from a unit j which is influencing the dynamics of a unit i. **b**, Scheme of a gene regulatory network of three genes, cf. [80]. We can distinguish three different types of interactions, where A_{13} ($\vdash$) indicates that gene 3 represses the expresion of gene 1; A_{21} ($\rightarrow$) indicates that gene 1 activates the expression of gene 2, but the third interaction ($\rightarrow$) represents a three-point interaction affecting the dynamics of gene 3. Three-point and higher-order interactions cannot be expressed in terms of adjecency matrices.

where $\boldsymbol{x}_i \in \mathbb{R}^D$ denotes the activity of the i-th neural population in the network, $\boldsymbol{S} : \mathbb{R}^D \to \mathbb{R}^D$ is a sigmoidal function representing the firing rates of populations and $\tau_{ij} \in \mathbb{R}$ is the time delay between populations j and i.

Also, models for networks of chemical reactions — systems that describe the concentrations of a given set of chemicals — are commonly posed as [81]

$$\dot{x}_i = \sum_{j=1}^{N} J_{ij} g_j(x_j), \tag{1.6}$$

where the entries of J_{ij} indicate the number of molecules of chemical species involved in each reaction and $g_j(x_j)$ is a component of the reaction rate vector $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^N$.

1.2.2. Networks with higher-order interactions

However, models (1.3) and (1.4) fail to represent systems with higher-order interactions. For instance, the structure of simple networks such as

$$\begin{aligned} \dot{x}_1 &= h_1(x_1) + g_{13}(x_1, x_3), \\ \dot{x}_2 &= h_2(x_2) + g_{21}(x_2, x_1), \\ \dot{x}_3 &= h_3(x_3) + g_{12}(x_1, x_2), \end{aligned} \tag{1.7}$$

may not be representable in terms of adjacency matrices due to the *three-point interaction* affecting the dynamics of unit 3, cf. Fig. 1.2b. Furthermore, one would need to resort to more elaborate concepts such as the *adjacency tensor* [82] *of order 3* to represent the structure of (1.7). Moreover, networks with higher-order interactions may even require adjecency tensors of higher orders to accurately represent the network structure.

Alternatively, we may represent the structure of a network having higher-order interactions in terms of its *incidence matrix* $H \in \{0,1\}^{M \times N}$, where N is the number of units in the network and M is the number of interactions [83]. Specifically, H defines how units are organized or involved in the network interactions. For instance, the incidence matrix H for (1.7) is given by

$$H = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \tag{1.8}$$

where the rows of H represent the existing interactions and its columns the network units. Nonetheless, despite H reliably captures the network structure, it may not be simply included in the functional forms of networks, as for the adjecency matrices in (1.4).

Do other network representations exist that are comprehensive but simple enough to capture the network structure in the presence of higher-oder interactions while explicitly appearing in the functional form of networks? In the next section we introduce an orginal and novel description for networks of dynamical system which solves this particular challenge.

1.3. Explicit dependency matrix

For networks of units whose state is defined by scalar variables $x_i(t) \in \mathbb{R}$, we propose that

$$\dot{x}_i = f_i(\Lambda^i \boldsymbol{x}) + I_i(t) + \xi_i(t), \tag{1.9}$$

where $\boldsymbol{x} = [x_1(t), x_2(t), \ldots, x_N(t)]^\top \in \mathbb{R}^N$ is a vector containing the network state at t and $f_i : \mathbb{R}^N \to \mathbb{R}$ is a function that specifies the dynamics of unit i. The terms $I_i(t) \in \mathbb{R}$ and $\xi_i(t) \in \mathbb{R}$ represent external driving signals and external noise acting on the i-th unit, respectively. Especially, we introduce the diagonal matrix $\Lambda^i \in \{0,1\}^{N \times N}$, that establishes which units are affecting the dynamics of unit i. We call Λ^i *the explicit dependency matrix of unit* i.

We define the diagonal entries of Λ^i as

$$\Lambda^i_{jj} = \begin{cases} 0 & \text{if } \frac{\partial f_i}{\partial x_j} \equiv 0 \\ 1 & \text{if } \frac{\partial f_i}{\partial x_j} \not\equiv 0 \end{cases}, \tag{1.10}$$

where the partial derivatives are identically zero, $\partial f_i / \partial x_j \equiv 0$, if and only if $(\partial f_i / \partial x_j)(\Lambda^i \boldsymbol{x}) = 0$ for all $\boldsymbol{x}$. Hence, if a unit j is directly affecting the dynamics of unit i, it follows that $\Lambda^i_{jj} = 1$ and $\Lambda^i_{jj} = 0$ otherwise.

In particular, the set of explicit dependency matrices Λ^i for all $i \in \{1, 2, \ldots, N\}$ replaces the concept of adjacency matrices in (1.4). Yet, differently from adjacency matrices, the explicit dependency matrices Λ^i comprise all types of network interactions within its structure, cf. (1.10). Thus, independently of whether we deal with networks having pairwise, three-point or higher-order interactions, the Λ^i matrices reliably capture the network structure.

For instance, let us construct the explicit dependency matrices for the system described in (1.7). Within our framework, we may represent (1.7) as

$$\begin{aligned} \dot{x}_1 &= f_1(\Lambda^1 \boldsymbol{x}) = & h_1(x_1) + g_{13}(x_1, x_3), \\ \dot{x}_2 &= f_2(\Lambda^2 \boldsymbol{x}) = & h_2(x_2) + g_{21}(x_2, x_1), \\ \dot{x}_3 &= f_3(\Lambda^3 \boldsymbol{x}) = & h_3(x_3) + g_{12}(x_1, x_2). \end{aligned} \tag{1.11}$$

The gradients $\boldsymbol{\nabla} f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_N}\right] \in \mathbb{R}^N$ of (1.11) are given by

$$\begin{aligned} \boldsymbol{\nabla} f_1 &= \left[\frac{\partial h_1}{\partial x_1} + \frac{\partial g_{13}}{\partial x_1}, 0, \frac{\partial g_{13}}{\partial x_3}\right], \\ \boldsymbol{\nabla} f_2 &= \left[\frac{\partial g_{21}}{\partial x_1}, \frac{\partial h_2}{\partial x_2} + \frac{\partial g_{21}}{\partial x_2}, 0\right], \\ \boldsymbol{\nabla} f_3 &= \left[\frac{\partial g_{12}}{\partial x_1}, \frac{\partial g_{12}}{\partial x_2}, \frac{\partial h_3}{\partial x_3}\right]. \end{aligned} \tag{1.12}$$

Therefore, according to (1.10), we may describe the structure of (1.7) as

$$\Lambda^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Lambda^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Lambda^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1.13}$$

where Λ^1, Λ^2 and Λ^3 are the dependency matrices for units 1,2 and 3, respectively.

Particularly, the advantage of explicit dependency matrices over other representations (e.g. adjacency and incidence matrices) is their coarser definition of connections. Grouping all orders of interactions into individual quantities eases the description of a network structure, as shown in the preceding example. Moreover, explicit dependencies are intrinsic to the definition of networks of dynamical systems, and thereby, one can always derive such matrices from models (1.4) and (1.9) through (1.10)

For arbitrary D-dimensional units $\boldsymbol{x}_i \in \mathbb{R}^D$, we may expand (1.9) as

$$\dot{x}_i^d = f_i(\Lambda^{id} \boldsymbol{x}) + I_i^d(t) + \xi_i^d(t), \tag{1.14}$$

where $i \in \{1, 2, \dots, N\}$ and $d \in \{1, 2, \dots, D\}$, $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \dots, \boldsymbol{x}_N(t)]^{\mathsf{T}} \in \mathbb{R}^{ND}$ is a vector containing the state of every unit (and their components) at time t,

$f_i^d : \mathbb{R}^{ND} \to \mathbb{R}$ is a function that specifies the dynamics of the d-th component of unit i and $\boldsymbol{I}_i(t) = \left[I_i^1(t), I_i^2(t), \ldots, I_i^D(t)\right]^\mathsf{T} \in \mathbb{R}^D$ and $\xi_i(t) = \left[\xi_i^1(t), \xi_i^2(t), \ldots, \xi_i^D(t)\right]^\mathsf{T} \in \mathbb{R}^D$ are a vector of external driving signals and external noise. The explicit dependency matrix $\Lambda^{id} \in \{0,1\}^{ND \times ND}$ is a diagonal matrix that establishes the structural dependency of the d-th component of unit i with respect to other units and their components. Thus, in general, the entries of $\Lambda^{i,d}$ are given as

$$\Lambda^{id}_{k_{js}k_{js}} = \begin{cases} 0 & \text{if } \frac{\partial f_i^d}{\partial x_j^s} \equiv 0 \\ 1 & \text{if } \frac{\partial f_i^d}{\partial x_j^s} \not\equiv 0 \end{cases}, \tag{1.15}$$

where $k_{js} := (j-1)D + s$ is the index of the s-th component of unit j. Thus, if the s-th component of unit j, x_j^s, affects the dynamics of the d-th component of unit i, it follows that $\Lambda^{id}_{k_{js}k_{js}} = 1$ and $\Lambda^{i}_{k_{js}k_{js}} = 0$ otherwise.

For later use, we further define the *dynamics spaces* $\mathcal{D}_i \subset \mathbb{R}^{(N+1)}$ for all $i \in \{1, 2, \ldots, N\}$ with $x_i \in \mathbb{R}$. Such spaces $\mathcal{D}_i$ are defined for each unit i as extensions of the common state space $\mathcal{S}$ explained in sec. 1.1 including the rate of change $\dot{x}_i$ of the unit. Thus, each possible system state defines a point $\delta_i(t) := (\boldsymbol{x}(t), \dot{x}_i(t)) \in \mathcal{D}_i$, cf. Fig. 1.3 for a simple example. Furthermore, states $\delta_i(t) \in \mathcal{D}$ may be mapped to $s(t) \in \mathcal{S}$ by a simple orthogonal projection,

$$s(t) = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \end{bmatrix} \delta_i(t). \tag{1.16}$$

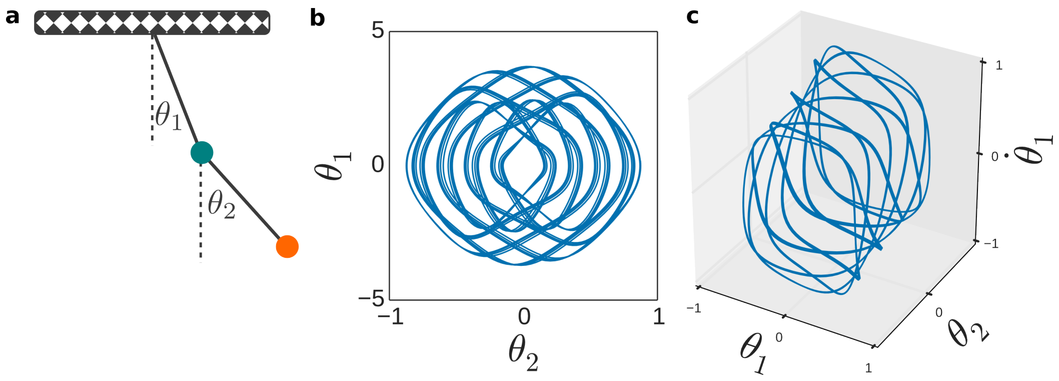


Figure 1.3. – A dynamics space better explains dynamics than a state space. a, Scheme of a chaotic double pendulum, cf. [84]. **b**, State space of the system for initial conditions $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = (0.87, -0.17, 2.00, -2.00)$. **c**, Dynamics space for pendulum 1. While a trajectory in the state space defines the evolution of the system from a initial condition (**b**), the dynamics space provides more information on the evolution of system by including the rate of change of units. Thus, it explains in more detail how a system yields a specific trajectory (**c**).

Moreover, likewise to sec. 1.1, we define a *trajectory* $\boldsymbol{\delta}_i$ *in the state space* $\mathcal{D}_i$ as a sequential collection of states $\boldsymbol{\delta}_i := [\delta_i(1), \delta_i(2), \ldots, \delta_i(T)] \in \mathbb{R}^T$, where T is the number of states and $\boldsymbol{\delta}_i$ determines the evolution of unit i following a initial condition $(\boldsymbol{x}(0), \dot{x}_i(0))$.

These representation in dynamics spaces will prove useful in chapter 2 for understanding how very disparate networks may perform identical dynamics, in chapter 3 for explaining how time series alone may uniquely reveal structural connection, in chapter 4, chapter 5, chapter 6 and chapter 7 for illustrating how local samplings of network dynamics may be used to infer the structure of networks.

Chapter 2

Parametrization of network dynamics

Recent studies coming from different fields suggest that, apart from structure, there may be other important factors defining and shaping network dynamics [28, 29, 57, 85]. For example, it has been shown that models of spiking neural networks [25–27, 34,35,40] may exhibit specific spike patterns which may be generated by any network from a high-dimensional family of networks. Modeling and experimental results in gene regulatory networks also indicate that disparate structures may yield identical dynamics under different mechanisms involving distinct network parameters and rate laws [28, 29]. These results suggest the existence of a sort of *give-and-take mechanism* in networks, where reducing one parameter may be compensated by increasing (or reducing) other parameters.

Such evidences raise some important questions: is this give-and-take mechanism unique to specific systems or is it a general property of networks? How are distinct mechanisms yielding an identical dynamics related? Moreover, how can one find classes of networks capable of generating a common dynamics?

In this chapter, we propose an inverse approach to answer these questions from a general perspective. First, we determine how different networks of generic units may generate a common collective dynamics by separating the contributions of structure and function to network dynamics. Second, we derive a theoretical framework to simultaneously restrict connectivities and local and coupling functions given predefined time series of the units. This proves helpful in understanding how distinct dynamical mechanisms generating a common network dynamics are related. In particular, we show (i) how the collective dynamics of networks may be parametrized in the dynamics space, (ii) how different network connectivities may achieve identical dynamics, and (iii) how these networks energetically differ from each other. As an example, we rewire networks of Kuramoto [86] and Kuramoto-like [87] oscillators with random network topologies into different networks that display the same collective dynamics. Parts of the results of this chapter were published in [24].

2.1. Parametrization of network dynamics

To understand how disparate networks yield identical dynamics, we firstly need to understand how explicit dependencies Λ^i influence the unit's dynamics. Therefore, let us differentiate (1.9) with respect to time,

$$\ddot{x}_i = \sum_{j=1}^{N} \Lambda^i_{jj} \frac{\partial f_i}{\partial x_j} \dot{x}_j, \tag{2.1}$$

or in vector form,

$$\ddot{x}_i = \boldsymbol{\nabla} f_i \Lambda^i \dot{\boldsymbol{x}}, \tag{2.2}$$

where $\dot{\boldsymbol{x}} = [\dot{x}_1(t), \dot{x}_2(t), \ldots, \dot{x}_N(t)]^\mathsf{T} \in \mathbb{R}^N$ and

$$\boldsymbol{\nabla} f_i = \left[\frac{\partial f_i}{\partial x_1}, \frac{\partial f_i}{\partial x_2}, \ldots, \frac{\partial f_i}{\partial x_N}\right] \in \mathbb{R}^N \tag{2.3}$$

is a row vector containing the gradient of function f_i. Thus, each entry in $\boldsymbol{\nabla} f_i$ comprises a unit-specific dependency of unit i on unit j, $\partial f_i / \partial x_j \not\equiv 0$.

The implications of equation (2.2) may be better understood in terms of the unit's dynamics space $\mathcal{D}_i$, $i \in \{1, 2, \ldots, N\}$. The unit's rate of change $\dot{x}_i(t)$ may be thought of as the position of a moving particle in the dynamics space $\mathcal{D}_i$, where (2.2) represents the rate of change of the particle with respect to time when it is moving at a velocity $\dot{\boldsymbol{x}}$ and the gradient sets the direction in which the directional derivative is greatest, cf. Fig. 2.1. So, if the elements in $\boldsymbol{\nabla} f_i$ were considered as *tunable parameters* of the unit's dynamics (e.g., f_i may be freely designed), one would be able to tune the gradient entries $\partial f_i / \partial x_j$ such that (2.2) is always satisfied for given $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and $\ddot{x}_i$. Specifically, equation (2.2) provides $N - 1$ degrees of freedom for selecting the entries of $\boldsymbol{\nabla} f_i$. Mostly important, equation (2.2) separates the contributions of structure Λ^i and function f_i to network dynamics, thus, demonstrating the existence of an *interplay mechanism between structure and function in networks*.

Also, equation (2.2) explains how networks having very different structures and interactions may yield identical dynamics in time. In general, by modifying units' interactions while preserving (2.2), one may explore the space of networks yielding identical dynamics. Designing $\boldsymbol{\nabla} f_i$ at will implies to establish how the unit (and consequently, the network) would evolve in the dynamics space while yielding a pre-specified dynamics, as shown in sec. 2.2.

2.1.1. Parametrization of specific dynamics

How can we parametrize the network dynamics in the dynamics space? Specifically, how can we find the set of explicit dependencies Λ^i and gradients $\boldsymbol{\nabla} f_i$ that yield identical network dynamics? Here, we introduce a rather simple approach

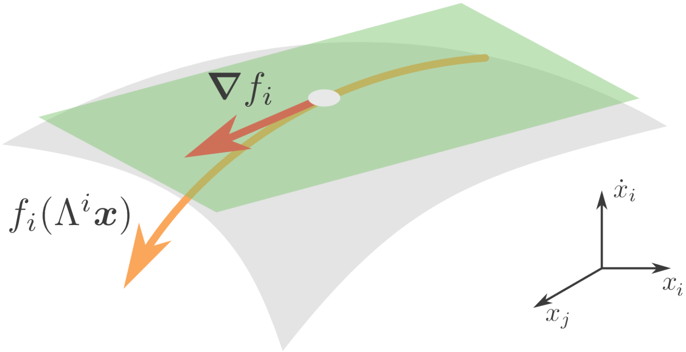


Figure 2.1. – The gradient ∇f_i indicates where the unit evolves in the dynamics space in time. The gradient ∇f_i (green) points in the direction in which the unit's dynamics (orange) evolves in time. The green surface depicts the tangent space that contains ∇f_i and the gray surface is the manifold on which the dynamics is embedded. The gradient ∇f_i points in the direction in which the unit evolves in time, thus, by controlling the entries in ∇f_i one would be able to control how the unit evolves in the dynamics space.

for parametrizing network dynamics in terms of its family of generating network structures and interactions.

First, let us assume that a high-dimensional time series of a network is given by

$$x_{i,m} = x_i(t_m), \tag{2.4}$$

for all $i \in \{1, 2, \dots, N\}$, $t_m = m\Delta t + t_0$ and Δt is sufficiently small such as to reliably estimate the first two time derivatives $\dot{x}_i$ and $\ddot{x}_i$ from (2.4) by

$$\dot{x}_{i,m} = \frac{x_{i,m+1} - x_{i,m-1}}{2\Delta t}, \tag{2.5}$$

$$\ddot{x}_{i,m} = \frac{x_{i,m+2} - 2x_{i,m} + x_{i,m-2}}{4\Delta t^2}. \tag{2.6}$$

Additionally, let us define matrices $\Gamma^i \in \mathbb{R}^{NM \times NM}$ and $Y \in \mathbb{R}^{NM \times M}$ as

$$\Gamma^i := \begin{bmatrix} \Lambda^i & 0 & \dots & 0 \\ 0 & \Lambda^i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Lambda^i \end{bmatrix}, \tag{2.7}$$

where 0 is a matrix full of zeros of size $N \times N$ and

$$Y := \begin{bmatrix} \dot{\boldsymbol{x}}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dot{\boldsymbol{x}}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \dot{\boldsymbol{x}}_M \end{bmatrix}, \tag{2.8}$$

with $\dot{\boldsymbol{x}}_m = [\dot{x}_{1,m}, \dot{x}_{2,m}, \ldots, \dot{x}_{N,m}]^\mathsf{T} \in \mathbb{R}^N$ and $\mathbf{0}$ is a vector full of zeros of size N. Equation (2.2) may be rewritten for M different time steps as

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{g}_i \Gamma^i Y, \tag{2.9}$$

where $\ddot{\boldsymbol{x}}_i = [\ddot{x}_{i,1}, \ddot{x}_{i,2}, \ldots, \ddot{x}_{i,M}] \in \mathbb{R}^M$ is a vector that contains the rate of change of unit dynamics $\dot{x}_i$ for M different time steps, $\boldsymbol{g}_i = [\boldsymbol{\nabla} f_{i,1}, \boldsymbol{\nabla} f_{i,2}, \ldots, \boldsymbol{\nabla} f_{i,M}] \in \mathbb{R}^{NM}$ is a vector composed by the gradients $\boldsymbol{\nabla} f_{i,m}$ evaluated at M time steps. The M-dimensional time series (2.4) for each i imposes M linear constraints on the non-linear system of equations (2.9) with $(NM + N)$ unknowns coming from the entries of $\boldsymbol{g}_i$ and Λ^i. However, entries in Λ^i are either ones or zeros (whether a unit interacts with another or not), thereby we may combine unknowns of $\boldsymbol{g}_i$ and Γ^i into a single vector of unknowns

$$\Gamma^i \boldsymbol{g}_i^\mathsf{T} = \left[\boldsymbol{\nabla} f_{i,1}\Lambda^i, \boldsymbol{\nabla} f_{i,2}\Lambda^i, \ldots, \boldsymbol{\nabla} f_{i,M}\Lambda^i\right]^\mathsf{T} \in \mathbb{R}^{NM}, \tag{2.10}$$

where the magnitude of its entries are strictly related to the gradients' entries. Thus, one may solve the under-determined system (2.9) instead as

$$\Gamma^i \boldsymbol{g}_i^\mathsf{T} = V\Sigma^\dagger U^\mathsf{T} \ddot{\boldsymbol{x}}_i^\mathsf{T} + W\boldsymbol{\zeta}_i, \tag{2.11}$$

where the operator † stands for pseudo-inverse [88] and $Y^\mathsf{T} = U\Sigma V^\mathsf{T}$ (singular value decomposition), $W \in \mathbb{R}^{NM \times M(N-1)}$ is a orthonormal basis for the nullspace[1] of Y^T and $\boldsymbol{\zeta}_i \in \mathbb{R}^{M(N-1)}$ is a vector of *tunable parameters*, cf. Appendix B.

Equation (2.11) parametrizes the space of solutions for (2.9). So, different choices of $\boldsymbol{\zeta}_i$ produce different explicit dependencies and interactions $\Gamma^i \boldsymbol{g}_i^\mathsf{T}$ that are consistent with (2.9). Furthermore, equation (2.11) spans all possible configurations for structures and interactions, and we will refer to them from now on as *network configurations*.

The main advantage of parametrization (2.11) is that by imposing specific constraints on units' dynamics (e.g., specific network connectivities), the gradient entries may be optimized for particular purposes. This is done by solving $\boldsymbol{\zeta}_i$ given a specific set of constraints, cf. sec. 2.1.2.

An approach using equation (2.11) is of practical importance when replacing or shutting down units, as shown in sec. 2.2. We may want to perform changes on the network structure while affecting its dynamics the least. So, by parametrizing the network dynamics through (2.11), we may alter the network structure and still achieve a pre-specified dynamics given problem-specific constraints (e.g. wiring costs), cf. Fig. 2.3 and Fig. 2.4. Furthermore, we may move across equivalent networks producing identical dynamics through (2.11) with little effort. This feature makes this approach a promising tool for designing networks for specific function, and moreover, it poses an alternative viewpoint to understand the interplay between the structure and function of complex networks.

In the next section, we propose a method to arrange the gradient entries with a set of allowed dependency matrices Λ^i.

[1] W may be easily extracted from the singular value decomposition of Y^T, cf. [4, 34, 35, 64, 69, 88]

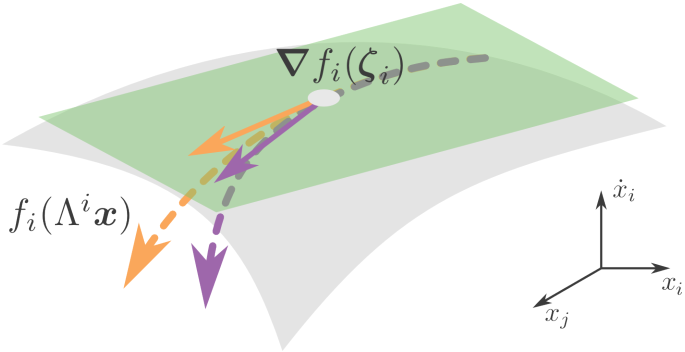


Figure 2.2. – Two equivalent trajectories in the dynamics space produce identical dynamics in time. The gradient ∇f_i (solid) points in the direction in which a specific unit's dynamics (dashed) evolves in time. The green surface depicts the tangent space that contains ∇f_i and the gray surface is the manifold on which the dynamics is embedded. Both dynamics yield different trajectories in $\mathcal{D}_i$, but both yield equal $\dot{x}_i(t)$ (according to (2.11)), thus, producing identical dynamics in time.

2.1.2. Designing networks

We consider *designing a network* as selecting a network structure and arranging the gradients such that the selected network structure satisfies a specific pre-defined dynamics. To design a network, we propose to arrange the block entries $\nabla f_{i,m}\Lambda^i$ of $\boldsymbol{g}_i\Gamma^i$ as the rows of a matrix $G^i \in \mathbb{R}^{M\times N}$ defined as

$$G^i := \begin{bmatrix} \frac{\partial f_{i,1}}{\partial x_1}\Lambda^i_{11} & \frac{\partial f_{i,1}}{\partial x_2}\Lambda^i_{22} & \dots & \frac{\partial f_{i,1}}{\partial x_N}\Lambda^i_{NN} \\ \frac{\partial f_{i,2}}{\partial x_1}\Lambda^i_{11} & \frac{\partial f_{i,2}}{\partial x_2}\Lambda^i_{22} & \dots & \frac{\partial f_{i,2}}{\partial x_N}\Lambda^i_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{i,M}}{\partial x_1}\Lambda^i_{11} & \frac{\partial f_{i,M}}{\partial x_2}\Lambda^i_{22} & \dots & \frac{\partial f_{i,M}}{\partial x_N}\Lambda^i_{NN} \end{bmatrix}, \tag{2.12}$$

where $\partial f_{i,m}/\partial x_j = \partial f_i(\Lambda^i \boldsymbol{x}_m)/\partial x_j$ is the partial derivative of f_i with respect to the activity of unit j evaluated at time step m. Unless the network is fully connected, the diagonal of Λ^i for all $i \in \{1, 2, \dots, N\}$ are composed of zeros and ones depending on the network connectivity. If $\Lambda^i_{jj} = 0$, it follows from equation (2.12) that

$$\forall\, m \in \{1, 2, \dots, M\} \quad : \quad G^i_{mj} = 0, \tag{2.13}$$

where G^i_{mj} is the (j, m) entry of G^i. The conditions (2.13) may be used as constraints to choose a specific $\hat{\boldsymbol{\zeta}}_i$ from the family of solutions (2.11) such that (2.13) holds[2].

[2] The dimension of the space of solutions for $\boldsymbol{\zeta}_i$ depends on the number of conditions (2.13). By imposing more conditions, one may reduce the dimensionality of the solution space.

After a specific $\hat{\boldsymbol{\zeta}}_i$ has been chosen, one may compute the remaining entries in G^i by computing (2.11) using $\hat{\boldsymbol{\zeta}}_i$.

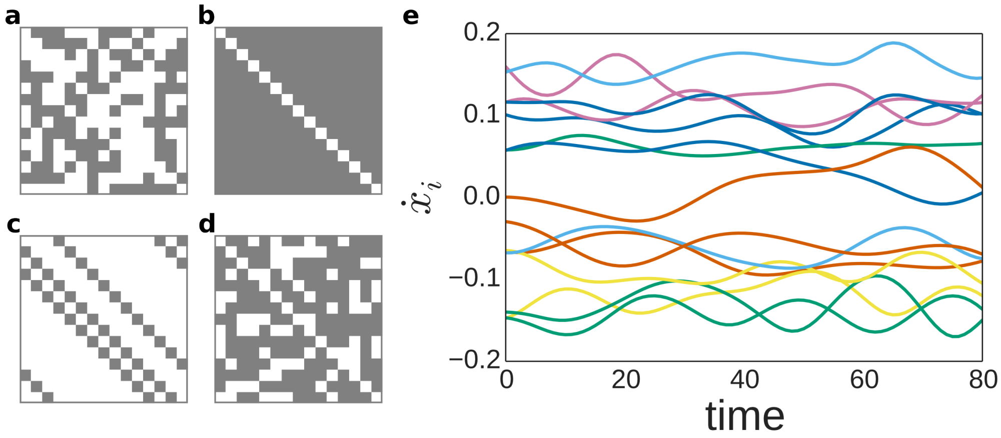


Figure 2.3. – **Equivalent networks show identical dynamics**. (**a**-**d**), Network connectivity for four different networks of $N = 15$ coded as gray for connections and white otherwise. **a**, Network of directionally coupled Kuramoto oscillators (2.14). (**b**,**c**,**d**), Fully-connected, regular and randomly-connected networks with optimized gradients. **e**, Dynamics generated by the network shown in **a**. The other three networks (**b**,**c**,**d**) display identical dynamics to that of **a** if the interactions among their units are properly arranged, cf. Fig.2.4.

2.2. Example: re-engineering networks

Similarly to network design, re-engineering networks often implies to rewire networks into feasible structures while preserving the dynamics of the original network. This, in turn, may require that network interactions change in the rewired network for preserving the desired network dynamics.

Here, to test the framework described in sec. 2.1.1 and sec. 2.1.2, we simulated networks of directionally coupled Kuramoto [86, 89],

$$\dot{x}_i = \omega_i + \frac{1}{k}\sum_{j=1}^{N} J_{ij}\sin\left(x_j - x_i\right), \tag{2.14}$$

and Kuramoto-like oscillators [87],

$$\dot{x}_i = \omega_i + \frac{1}{k}\sum_{j=1}^{N} J_{ij}\left[\sin\left(x_j - x_i - 1.05\right) + 0.33\sin\left(2\left(x_j - x_i\right)\right)\right], \tag{2.15}$$

having k incoming connections per node, cf. Fig. 2.3 a realization. Here, the entries of $J \in \mathbb{R}^{N\times N}$ are given by $J = R \odot A$, where $\odot$ stands for entry-wise-matrix product and $A \in \{0,1\}^{N\times N}$ is a randomly generated adjacency matrix. Both the ω_i and the R_{ij} are drawn from a uniform distribution on the respective intervals $\omega_i \in [-2, 2]$ and $R_{ij} \in [0.5, 1]$.

The time derivatives $\dot{x}_{i,m}$ and $\ddot{x}_{i,m}$ are estimated from sampled time series as in (2.5) and (2.6). Then, we parametrize the dynamics of each unit in an interval of M time steps utilizing (2.11) and rewired original networks (2.14) and (2.15) for different sets of dependency matrices Λ^i for all $i \in \{1, 2, \ldots, N\}$. The resulting G^ipredict the dynamics $\dot{y}_i$ of unit i in the rewired networks via

$$\dot{y}_{i,m+1} \approx \dot{x}_{i,m} + \boldsymbol{G}^i_m \left[\boldsymbol{x}_{m+1} - \boldsymbol{x}_m\right], \tag{2.16}$$

where $\dot{y}_{i,m}$ is the predicted dynamics for i at time step m, $\boldsymbol{G}^i_m$ is the m-th row of a G_i calculated for a specific Λ^i and $\boldsymbol{x}_m = \left[x_{1,m}, x_{2,m}, \ldots, x_{N,m}\right]^{\mathsf{T}} \in \mathbb{R}^N$.

2.2.1. Moving across network structures

Exploiting the parametrization (2.11) permits to explore many different network structures with a desired dynamics at will, regardless the topological features of such structures, cf. Fig. 2.3 and Fig. 2.4. The interplay between function and structure described above determines the rearrangement of interactions among units such that the system as a whole describes a pre-defined dynamics. Furthermore, equation (2.11) permits to actively modify network structure and rearrange interactions in time while preserving a desired dynamics, feature that may be of interest for designing time-dependent structures, cf. Fig. 2.5.

2.2.2. Network variance and time resolution

To measure the reliability of predictions, we quantified the differences between predicted and original dynamics at t by means of the network variance Nv defined as

$$Nv(t) := \frac{1}{N}\sum_{j=1}^{N}\left(x_{i,t} - y_{i,t}\right)^2, \tag{2.17}$$

for several rewired networks, cf. Fig. 2.6. Additionally, to understand the source of variances between networks, we computed the mean network variance $\langle Nv\rangle_t$. The numerics suggest that the mean variance $\langle Nv\rangle_t$ roughly scales as a power-law $\langle Nv\rangle_t \sim (\Delta t)^\alpha$ where $\alpha \approx 2.2$, cf. Fig. 2.6. This implies that differences between network dynamics may be due to numerical aspects (e.g, approximations (2.16) and numerical differentiations (2.5) and (2.6)), and not due to the parametrization itself.

2.3. Energy of network configurations

Can we quantify differences among distinct network configurations? As shown above, distinct configurations achieve a specific dynamics by rearranging the gradient entries $\partial f_{i,m}/\partial x_j$, cf. Fig. 2.4. Furthermore, the collection of entries $\partial f_{i,m}/\partial x_j$

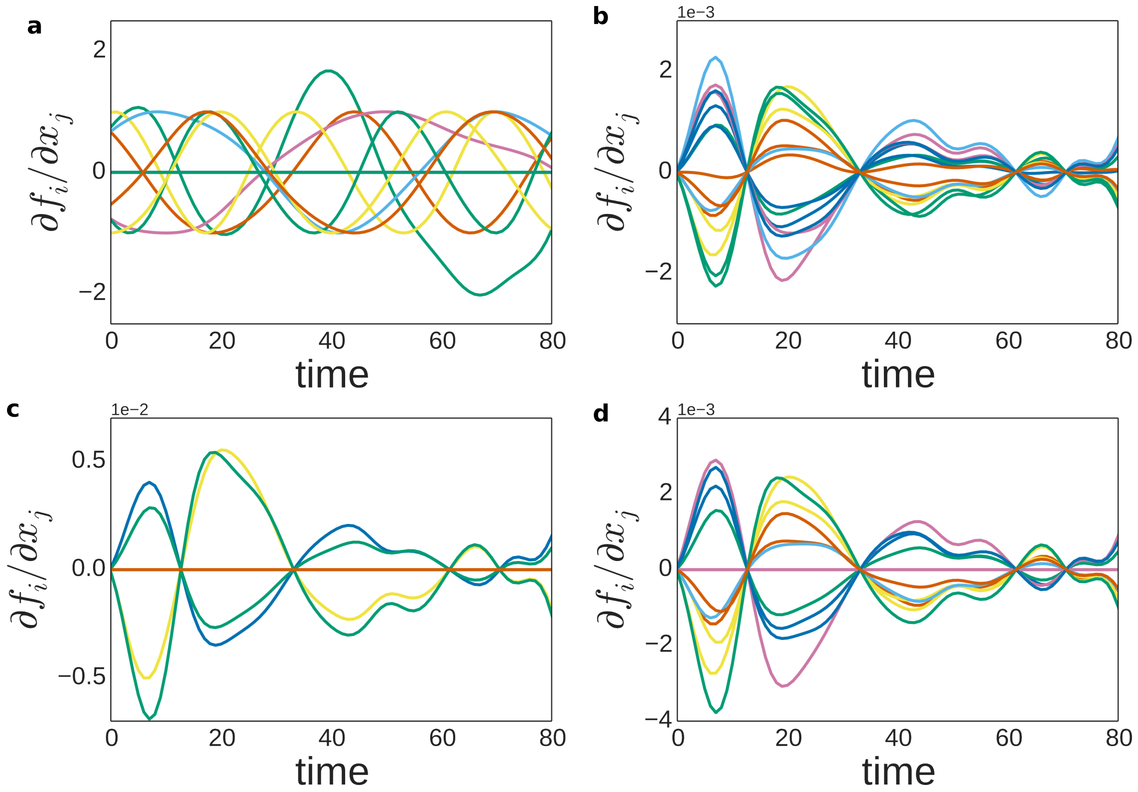


Figure 2.4. – Different network configurations yield identical dynamics by rearranging of network interactions. (**a**-**d**), $\partial f_i/\partial x_j$ for unit $i = 2$ and $j \in \{1, 2, \ldots, N\}$ in the four networks shown in Fig.2.3. **a**, Network of directionally coupled Kuramoto oscillators (2.14). (**b**,**c**,**d**), Fully-connected, regular and randomly-connected networks. Different networks manage to yield identical dynamics by reconstituting network interactions. Thus, different structures force units to interact in different manners for achieving a common network dynamics.

for all $i \in \{1, 2, \ldots, N\}$ and $m \in \{1, 2, \ldots, M\}$ are specific to each configuration. Thus, we propose to measure the effort put on by a network configuration to achieve a specific dynamics in terms of the *energies* $E_i(\boldsymbol{\zeta}_i) \in \mathbb{R}$ defined as

$$E_i = E(G^i(\boldsymbol{\zeta}_i)), \tag{2.18}$$

for all $i \in \{1, 2, \ldots, N\}$. However, what is a good choice for the energy function E? A candidate function must satisfy $E(G^i(\boldsymbol{\zeta}_i)) > 0$ and $E(0_{N\times M}) = 0$ where $0_{N\times M} = \{0\}^{N\times M}$. Interestingly, the squared Frobenius norm $\|\cdot\|_F^2$ is a natural choice satisfying the aforementioned conditions. So, we may define the energy function E as

$$E(G^i(\boldsymbol{\zeta}_i)) := \left\|G^i(\boldsymbol{\zeta}_i)\right\|_F^2 = \mathrm{Trace}\left[\left(G^i(\boldsymbol{\zeta}_i)\right)^\mathsf{T} G^i(\boldsymbol{\zeta}_i)\right] = \sum_{j=1}^{N} \sigma_{ij}^2, \tag{2.19}$$

where $\sigma_{ij\in\{1,2,\ldots,N\}}$ are the singular values of G^i.

Equation (2.19) directly relates the parametrization (2.11) with energy $E_i(\boldsymbol{\zeta}_i)$, which eases a general analysis as we will see in short. Equation (2.19) may also be written

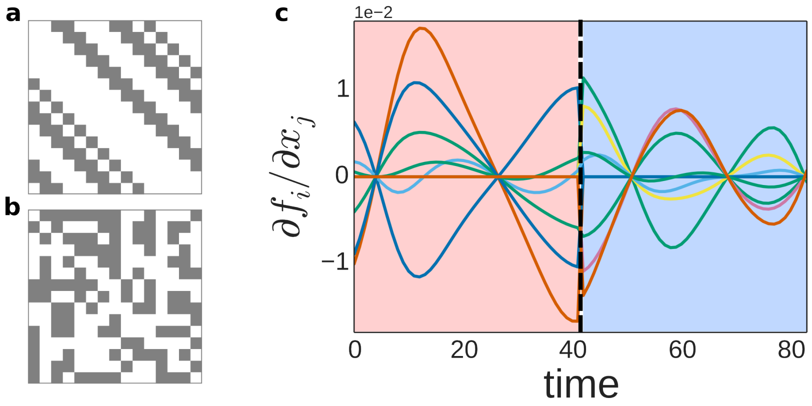


Figure 2.5. – Time dependent network connectivity and active rearrangement of network interactions. (**a**,**b**), Network connectivity for two different networks of $N = 15$ coded as gray for connections and white otherwise. (**a**,**b**), Regular and randomly-connected networks with optimized gradients. **c**, $\partial f_i/\partial x_j$ for unit $i = 5$ and $j \in \{1, 2, \ldots, N\}$ in a network actively changing from the structure depicted in **a** (pink area) to **b** (blue area). Even if the network structure changes, the gradient entries may be rearranged to satisfy the pre-defined dynamics. Specifically, at $t = 40$ the network changes its structure, thus, shuts down some interactions and activate others (transition from pink to blue).

as

$$\sum_{j=1}^{N} \sigma_{ij}^2 = \sum_{j=1}^{N} \sum_{m=1}^{M} \left(G_{mj}^i\right)^2. \tag{2.20}$$

By defining $\boldsymbol{r}_i = V\Sigma^\dagger U^\intercal \ddot{\boldsymbol{x}}_i^\intercal$, we may rewrite (2.20) as

$$\sum_{j=1}^{N} \sigma_{ij}^2 = \sum_{j=1}^{N} \sum_{m=1}^{M} \left(r_{i,(j+(m-1)N)} + \boldsymbol{w}_{j+(m-1)N}\boldsymbol{\zeta}_i\right)^2, \tag{2.21}$$

where $\boldsymbol{w}_k \in \mathbb{R}^{1\times M(N-1)}$ is the k-th row of W. Expanding (2.21) yields

$$\sum_{j=1}^{N} \sigma_{ij}^2 = \sum_{j=1}^{N} \sum_{m=1}^{M} \left[r_{i,(j+(m-1)N)}^2 + \left(\boldsymbol{w}_{j+(m-1)N}\boldsymbol{\zeta}_i\right)^2 + 2r_{i,(j+(m-1)N)}\boldsymbol{w}_{j+(m-1)N}\boldsymbol{\zeta}_i\right]. \tag{2.22}$$

Replacing both $j \in \{1, 2, \ldots, N\}$ and $m \in \{1, 2, \ldots, M\}$ indices on the right hand of (2.22) by a single index $k \in \{1, 2, \ldots, NM\}$ yields

$$\sum_{j=1}^{N} \sigma_{ij}^2 = \sum_{k=1}^{NM} r_{i,k}^2 + \sum_{k=1}^{NM} (\boldsymbol{w}_k\boldsymbol{\zeta}_i)^2 + 2\sum_{k=1}^{NM} r_{i,k}\boldsymbol{w}_k\boldsymbol{\zeta}_i. \tag{2.23}$$

Further inspection on (2.23) reveals that

$$\sum_{j=1}^{N} \sigma_{ij}^2 = \boldsymbol{r}_i^\intercal \boldsymbol{r}_i + \boldsymbol{\zeta}_i^\intercal W^\intercal W \boldsymbol{\zeta}_i + 2\boldsymbol{r}_i^\intercal W \boldsymbol{\zeta}_i. \tag{2.24}$$

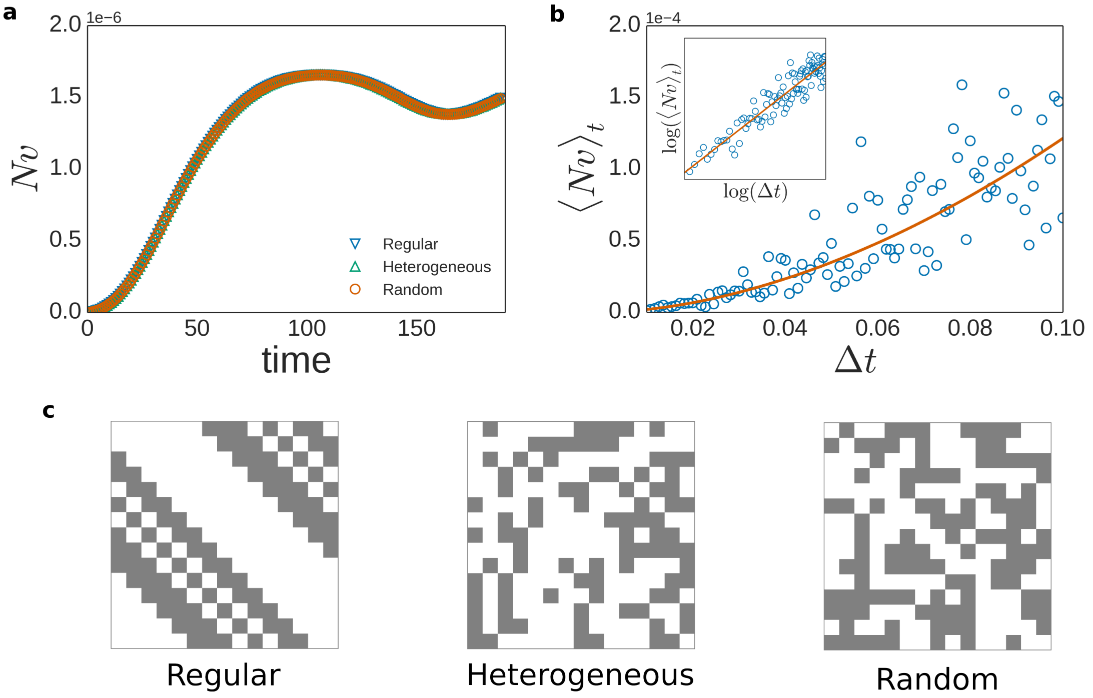


Figure 2.6. – Network variance scales with the sampling rate. Network variance for networks of $N = 15$ Kuramoto-like oscillators (2.15) and sampling rate dependence. **a**, Network variance in time for three different proposed network connectivities. **b**, Mean network variance $\langle Nv \rangle_t$ versus Δt with the best power-law fit. The inset shows same data on a logarithmic scale. **c**, Topology A (regular network), Topology B (heterogeneous network) and Topology C (random network).

However, given that (i) W is orthonormal basis for the nullspace of Y^T, thus $W^\mathsf{T} W = I_{M(N-1)}$, and (ii) $\forall \boldsymbol{\zeta}_i : \boldsymbol{r}_i \perp W\boldsymbol{\zeta}_i$, equation (2.24) turns into

$$E(\boldsymbol{\zeta}_i) = \sum_{j=1}^{N} \sigma_{ij}^2 = \boldsymbol{r}_i^\mathsf{T} \boldsymbol{r}_i + \boldsymbol{\zeta}_i^\mathsf{T} \boldsymbol{\zeta}_i. \tag{2.25}$$

Equation (2.25) provides a simple relation between the energies $E_i(\boldsymbol{\zeta}_i)$ and the network configurations defined by vectors $\boldsymbol{\zeta}_i$. Also, it provides an unique set of lower bounds

$$E_{i0} := E_i \left(\mathbf{0}_{M(N-1)} \right), \tag{2.26}$$

where $\mathbf{0}_{M(N-1)} \in \{0\}^{M(N-1)}$, for the minimal energy required to yield a specific network dynamics. Here we will refer to this network configuration yielding energies $\forall i : E_i = E_{i0}$ as the *minimal network configuration.*

Furthermore, equation (2.25) predicts that different network configurations may achieve identical energies. Specifically, as long as two different vectors $\boldsymbol{\zeta}_i$ and $\boldsymbol{\zeta}_i'$ satisfy the condition $\zeta = \|\boldsymbol{\zeta}_i\|_2^2 = \|\boldsymbol{\zeta}_i'\|_2^2$, both configurations will yield identical energies E_i, cf. Fig. 2.7.

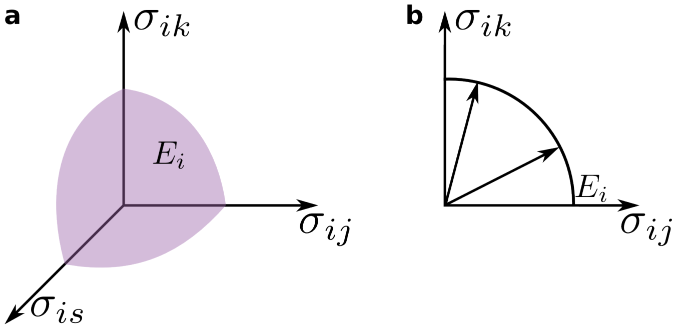


Figure 2.7. – Different network configurations may yield identical energies. Equation (2.25) defines isoenergetic hyper-surfaces E_i. (**a**,**b**), Examples of isoenergetic surfaces in 3-D and 2-D. Two different network configurations may achieve identical energies E_i by means of distinct sets of σ_{ij} satisfying (2.25). Thus, one may rewire network structures while satisfying energy constraints.

2.3.1. Optimality of network configurations

Can we say something about how optimal a given network configuration is? Given that now we have a concept of minimal network configurations, by defining the set of energy ratios

$$R_i := \frac{E_{i0}}{E_i(\boldsymbol{\zeta}_i)} \in [0,1]$$

to the minimal configuration, we may quantify the energy-wise optimality of a given configuration producing a specific network dynamics. Specifically, higher values of R_i indicate more optimal configurations with respect to the minimal configuration.

As an example, we compare the optimality of networks of Kuramoto (2.14) and Kuramoto-like oscillators (2.15) with their minimal counterparts, cf. Fig. 2.8. We observed that both types of networks are by no means particularly optimal. Meaning that the energy invested by such is too high when compared to their minimal configurations. Yet, are other network configurations more energetically optimal for producing the same network dynamics?

To briefly answer the latter, we tested the optimality of re-engineered networks using randomly generated network structures with a fixed number of incoming connections per node. Opposite to our example, these networks yielded high optimality values, thus largely outperforming original networks, cf. Fig. 2.9. Such results suggest that interactions between units may be optimized, energetically speaking, into different coupling functions than those used in (2.14) and (2.15). Therefore, our results affirmatively answer that original networks may be (considerably) optimized employing different configurations.

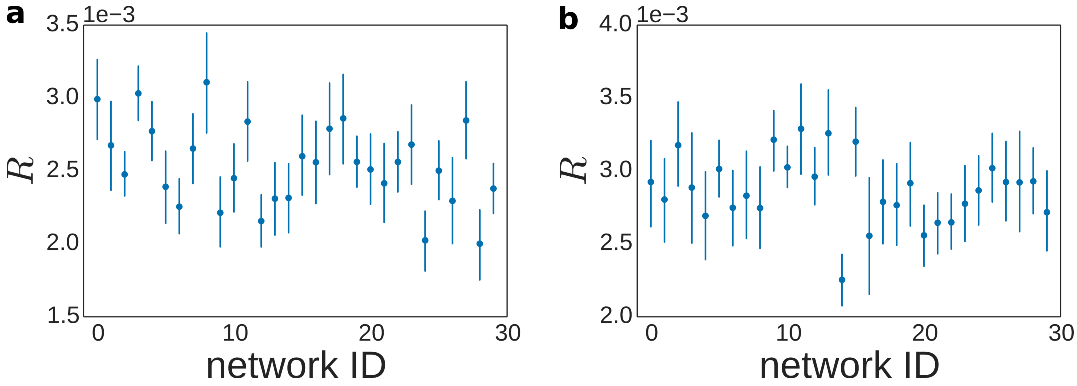


Figure 2.8. – Networks of Kuramoto and Kuramoto-like oscillators are not energy-wise optimal. (**a**,**b**), Mean energy-wise optimality $R = N^{-1}\sum_{n=1}^{N} R_i$ for ID $\in \{1, 2, \ldots, 30\}$ different networks of $N = 15$ and $k = 7$ of Kuramoto (2.14) and Kuramoto-like (2.15). ID represents labeling of networks. When compared to their minimal configurations, the simulated networks look rather non-optimal. Values indicate that these networks invest large energies to yield their specific dynamics.

2.4. Discussion

In this chapter, we proposed to view network dynamics as an inverse problem. Starting from our definition of networks of dynamical systems, we developed a method to find the set of all networks yielding a common collective dynamics. Specifically, by separating contributions to network dynamics coming from structure and function, we analytically parametrized the set of all ways in which a collection of interconnected units have to interact to reproduce a specific dynamics. Additionally, we demonstrated that interactions may be optimized for specific network structures and still achieve pre-defined dynamics. Interestingly, also time-dependent network structures satisfying specific structural constraints may be designed using our approach. Furthermore, here we introduced a set of physical quantities for measuring an *energy-wise* optimality of networks.

Previous works on inverse problems on networks of continuously coupled units have mainly focused on finding the actual network structure generating a specific dynamics [7, 9, 10, 22, 34, 35, 63–65, 67, 69, 90]. However, works on networks of spiking neurons suggest the existence of an interplay between the structure and function of networks that can be exploited for designing networks for specific dynamics [26, 27].

Here we extended the ideas presented in [26, 27] to demonstrate that such interplay also exists in networks of continuously coupled units. In particular, this mechanism explains that disparate networks yield identical dynamics by rearranging interactions while satisfying specific constraints. For instance, rewiring connections within a network will require that functional interactions (coupling functions) among units also change if one wants the network dynamics to remain intact. Specifically, this *give-and-take* mechanism exploits the fact that a given dynamics may be achieved in the dynamics space in several different manners spanned by different connections,

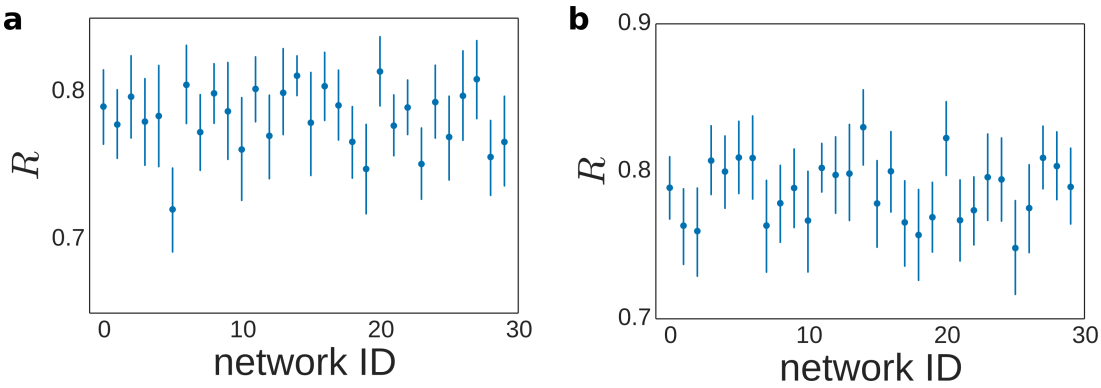


Figure 2.9. – Energy-wise optimality of re-engineered networks. (**a**,**b**), Mean energy-wise optimality $R = N^{-1}\sum_{n=1}^{N} R_i$ for ID $\in \{1, 2, \ldots, 30\}$ different networks of $N = 15$ and $k = 7$ of Kuramoto (2.14) and Kuramoto-like (2.15). ID represents labeling of networks. Re-engineered networks achieve specific dynamics in more optimal ways than original Kuramoto and Kuramoto-like oscillators, cf. Fig.2.8.

local dynamics and coupling functions. Thus, confirming that *networks are more than just its structure*. Instead, these are intricate combinations of several factors (e.g., connections, local dynamics, coupling functions) occurring simultaneously and shaping network dynamics. Furthermore, here we demonstrated the existence of *minimal network configurations*, networks that reproduce a specific dynamics while investing the minimal energy. This provides a natural way to measure and compare the optimality of networks designed for specific function. Yet, it is still unclear how this optimality scheme compares to other schemes observed in nature, such as the minimization of wiring-costs.

To our knowledge, the present chapter might be one of the few first studies (cf. Refs. [26,27]) tackling the interplay between network's structure and function from a general and analytical perspective. Future work still needs to completely understand this *give-and-take* mechanism and its relation to functional form of coupling functions. Although, here we limited ourselves to numerically find the family of network configurations for a specific dynamics, one may also want to construct analytical functions for such configurations through basis functions for network modeling purposes. Furthermore, an immediate next step in this area is to devise optimization schemes for particular purposes such as robustness to dynamical or adaptability to structural perturbations. Such schemes may bridge the results of this chapter with applications in real-world networks such as power grids or other type of distribution networks of commodities.

Chapter 3

Network connectivity from time series

In recent years, there has been an increasing interest in developing methods to infer how interacting units are connected. For instance, understanding interactions at cellular levels may help in identifying genes and pathways involved in the progression of specific human diseases [33]. Also, understanding how genes regulate themselves in gene regulatory networks may explain how cell differentiation is carried out in living organisms [80, 91–93].

Yet, the network connectivity is often not directly accessible in experimental setups. Instead, one must resort to indirect approaches to retrieve connections from available data. Generally, given a set of observations of a network under different conditions (e.g., perturbations of a steady-state), heuristic similarity measures (e.g., mutual information [41], transfer entropy [42], Granger causality [43, 44] and extensions thereof) are computed between pairs of units and thresholded to discern between actual and absent links. Such methods are susceptible to recover spurious connections, leading to faulty results.

Alternatively, methods stemming from the theory of networks of dynamical systems have proven to safely recover structural connections (physical links) by exploiting certain features of network dynamics [22]. For instance, it was demonstrated in systems defined as [22, 69]

$$\dot{x}_i = b_i\left(x_i\right) + \sum_{j=1}^{N} J_{ij} g_{ij}\left(x_i, x_j\right), \tag{3.1}$$

where $b_i : \mathbb{R} \to \mathbb{R}$ sets the local dynamics, the $g_{ij} : \mathbb{R}^2 \to \mathbb{R}$ set how units i and j interact and the elements $J_{ij} \in \mathbb{R}$ set the connectivity; that if functions b_i and g_{ij} are known, the elements J_{ij} may be recovered from simply observing the network dynamics. Nonetheless, such functions are usually not at hand when dealing with real-life networks.

Thus, is it possible to retrieve connections purely from observing the network dynamics? Particularly, is the pre-knowledge of functions b_i and g_{ij} really necessary to understand how units are interacting with each other? Furthermore, can time series of a network uniquely determine its structure?

In this chapter, we propose a model-independent framework to reveal network structural connections from time series alone. We formulate the problem in terms of a *functional decomposition of dynamical systems* by decomposing units' dynamics in terms of network interactions of several orders. This decomposition (when properly generated) entirely map structural connections between units. We also introduce the *Algorithm for Revealing Network Interactions* (ARNI) that reverse-engineers the set of interactions that best captures units' dynamics. Interestingly, sufficiently long time series and compositions of several different short time series safely reveal network structural connections. This suggests that the sampling of units' dynamics space is the responsible for revealing structural links. We demonstrate the efficiency of our method by successfully revealing structural connections in networks of coupled oscillators exhibiting complex dynamics.

3.1. Interaction decomposition of network dynamics

To describe our framework for inferring connections, here we will describe the dynamics of networks in terms of generic dynamical systems (1.9), cf. chapter 1.

In real-world problems, functions b_i are generally not at hand. Instead one is forced to estimate them while relying on units' dynamics $x_i(t)$. This renders the problem of inferring explicit connections Λ^i from $\dot{x}_i$ and $\boldsymbol{x}$ into a difficult (and possibly combinatorial) task to solve.

Still, researchers often select particular models for h_i, involving known functions and specific types of interactions to explain the network dynamics [22,69,70,73,74]. Subsequently, based on these models, one may reconstruct the network connectivity from recorded measurements. Nevertheless, proposing models for the underlying mechanisms defining the network dynamics requires a strong *a priori* knowledge about the network intrinsic features, such as the local dynamics coupling functions.

We circumvent these limitations by functionally decomposing units' dynamics in interaction terms with the rest of the network as

$$\begin{aligned}\dot{x}_i = f_i(\Lambda^i \boldsymbol{x}) = \sum_{j=1}^{N} \Lambda^i_{jj} g^i_j(x_j) + \sum_{j=1}^{N}\sum_{s=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss} g^i_{js}(x_j, x_s) + \\ \sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss}\Lambda^i_{ww} g^i_{jsw}(x_j, x_s, x_w) + \dots \qquad (3.2)\end{aligned}$$

where $g^i_j : \mathbb{R} \to \mathbb{R}$, $g^i_{js} : \mathbb{R}^2 \to \mathbb{R}$, $g^i_{jsw} : \mathbb{R}^3 \to \mathbb{R}$ and, in general $g^i_{j_1 j_2 \cdots j_K} : \mathbb{R}^K \to \mathbb{R}$, represent the K-th order interactions between units j_k for all $k \in \{1, 2, \dots, K\}$ and unit i.

Specifically, decomposition (3.2) separates contributions to units' dynamics arising from different types of network interactions. Thus, it may be employed to infer network connectivity.

3.2. Reconstruction problem

Here reconstructing the network connectivity implies identifying which interactions in (3.2) capture the network dynamics the best. So, given that f_i and Λ^i are unknown, functions $g^i_{j_1 j_2 \cdots j_K}$ are not directly accessible. Still, one may expand these functions in terms of basis functions as

$$\dot{x}_i = \sum_{j=1}^{N} \Lambda^i_{jj} \sum_{p=1}^{P_1} c^i_{j,p} h_{j,p}(x_j) + \sum_{j=1}^{N}\sum_{s=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss} \sum_{p=1}^{P_2} c^i_{js,p} h_{js,p}(x_j, x_s) + \\ \sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss}\Lambda^i_{ww} \sum_{p=1}^{P_3} c^i_{jsw,p} h_{jsw,p}(x_j, x_s, x_w) + \ldots, \tag{3.3}$$

where P_k indicates the number of basis functions employed in the expansion. The rationale of (3.3) is to express the unknown functions $g^i_{j_1 j_2 \cdots j_K}$ in terms of linear combinations of simpler and known functions [94].

Hence, defining

$$\boldsymbol{h}_j(x_j) := [h_{j,1}(x_j), h_{j,2}(x_j), \ldots, h_{j,P_1}(x_j)]^{\mathsf{T}} \in \mathbb{R}^{P_1} \tag{3.4}$$

and $\boldsymbol{c}^i_j := \left[c^i_{j,1}, c^i_{j,2}, \ldots, c^i_{j,P_1}\right] \in \mathbb{R}^{P_1}$, and analogously repeating the same process for all expansions in equation (3.3), we may rewrite it as

$$\dot{x}_i = \sum_{j=1}^{N} \Lambda^i_{jj} \boldsymbol{c}^i_j \boldsymbol{h}_j(x_j) + \sum_{j=1}^{N}\sum_{s=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss} \boldsymbol{c}^i_{js} \boldsymbol{h}_{js}(x_j, x_s) \\ + \sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss}\Lambda^i_{ww} \boldsymbol{c}^i_{jsw} \boldsymbol{h}_{jsw}(x_j, x_s, x_w) + \ldots. \tag{3.5}$$

Now, let us assume that we are given a high dimensional time series of a network $x_{i,m} = x_i(t_m)$ for all $i \in \{1, 2, \ldots, N\}$, where $t_m = m\Delta t + t_0$ is the sampling time for $m \in \{1, 2, \ldots, M\}$. Additionally, let us assume that Δt is sufficiently small such as to reliably estimate time derivatives $\dot{x}_i$ from measured time series, thereby, $\dot{x}_{i,m}$ is the time derivative of i computed at t_m. We may extend equation (3.5) as

$$\dot{\boldsymbol{x}}_i = \sum_{j=1}^{N} \Lambda^i_{jj} \boldsymbol{c}^i_j H_j + \sum_{j=1}^{N}\sum_{s=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss} \boldsymbol{c}^i_{js} H_{js} \\ + \sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N} \Lambda^i_{jj}\Lambda^i_{ss}\Lambda^i_{ww} \boldsymbol{c}^i_{jsw} H_{jsw} + \ldots, \tag{3.6}$$

where $\dot{\boldsymbol{x}}_i := [\dot{x}_{i,1}, \dot{x}_{i,2}, \ldots, \dot{x}_{i,M}] \in \mathbb{R}^M$,

$$H_j := [\boldsymbol{h}_j(x_{j,1}), \boldsymbol{h}_j(x_{j,2}), \ldots, \boldsymbol{h}_j(x_{j,M})] \in \mathbb{R}^{P_1 \times M}, \tag{3.7}$$

and matrices H_{js} and H_{jsw} are constructed in a similar manner to (3.7).

Finally, the reconstruction problem becomes identifying the non-zero interaction terms in equation (3.6) that best capture the unit's dynamics. Yet, the vector of coefficients $\boldsymbol{c}^i_j, \boldsymbol{c}^i_{js}, \boldsymbol{c}^i_{jsw}$ (and so on) are unknown, thereby, hindering the retrieval of Λ^i from (3.6).

3.3. Algorithm for revealing network interactions ARNI

Solving linear systems (3.6) consist in finding *block structures for* $\boldsymbol{c}^i$. These structured solutions are composed by blocks $\boldsymbol{c}^i_s$ of non-zero entries (representing the non-zero interactions acting on unit i) distributed along $\boldsymbol{c}^i$, cf. Fig. 3.1. However, such solutions cannot be retrieved in standard manners (e.g. Moore-Penrose pseudo-inversion or minimization of L_1-norm, cf. Appendix A and Appendix B). Thus, solving this kind of systems requires more sophisticated methods.

Specifically, given a linear system

$$\boldsymbol{y} = \boldsymbol{c}D, \tag{3.8}$$

where $\boldsymbol{y} \in \mathbb{R}^{1\times M}$ is a vector of M measurements and $D \in \mathbb{R}^{Nd\times M}$ is a matrix divided into N matrix blocks $D_j \in \mathbb{R}^{d\times M}$ for all $j \in \{1, 2, \ldots, N\}$, we may define a structured solution in blocks for $\boldsymbol{c}$ as [95–99]

$$\boldsymbol{c} = \Big[\underbrace{c_1, c_2, \ldots, c_d}_{\boldsymbol{c}_1}, \underbrace{c_{d+1}, c_{d+2}, \ldots, c_{2d}}_{\boldsymbol{c}_2}, \ldots, \underbrace{c_{(N-1)d+1}, c_{(N-1)d+2}, \ldots, c_{Nd}}_{\boldsymbol{c}_N}\Big], \tag{3.9}$$

where $\boldsymbol{c}_j \in \mathbb{R}^d$ for all $j \in \{1, 2, \ldots, N\}$ represent the blocks.

Recent efforts on solving systems (3.8) subjected to solutions (3.9) in blocks have been mainly focused on *block sparse*[1] *solutions on under-determined systems* ($Nd \gg M$) [95–99] , cf. Fig. 3.1. Such approaches (e.g. group lasso [95], sparse regression using mixed norms [97], block orthogonal matching pursuit [98]) take advantage of the indeterminacy of systems (3.8) to iteratively construct a block sparse solution from a family of solutions for (3.8) employing specific heuristics (e.g. minimizing $\|\boldsymbol{c}_j\|_2$ for $j \in \{1, 2, \ldots, N\}$ while maximizing the number of zero blocks [96]).

Nonetheless, despite returning block solutions, these approaches do not retrieve non-sparse solutions. Thus, limiting their applicability only to systems showing sparseness. Furthermore, preliminary tests revealed that these approaches fail to

[1] A sparse vector consist in an array containing only few entries different from zero [95–100].

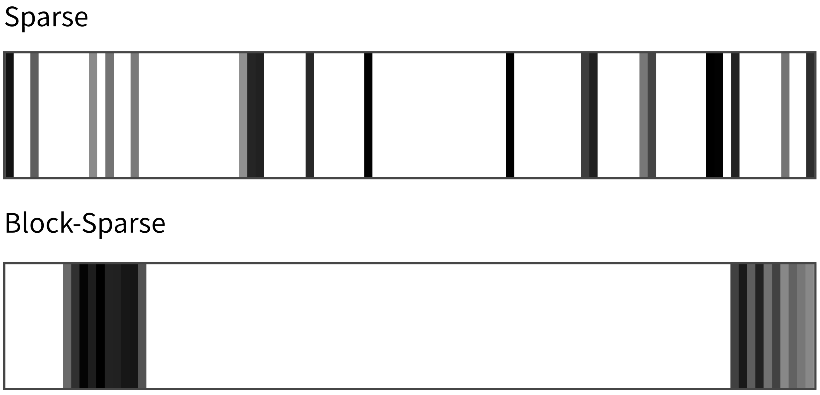


Figure 3.1. – Sparsity differs from block sparsity. A sparse vector contains only few entries different that zero distributed along the vector. Instead, a block sparse vector is composed by few blocks of non-zero entries.

recover the network connectivity in our reconstruction problems (3.6), even when the networks were considerably sparse.

Thus, in view of this, we developed the *Algorithm for Revealing Network Interactions* (ARNI) that reverse-engineers the non-zero interactions in equation (3.6) while solely relying on measured time series of $\boldsymbol{x}$. Specifically, ARNI: (i) constructs functional spaces for each interaction using the known basis functions and measured time series, (ii) finds the best suited set of interactions that reproduces the measured $\dot{x}_i$ under a squared error-minimization scheme and (iii) returns the explicit connections Λ^i of the unit.

In particular, ARNI recovers connections by minimizing projections on *sequentially-enlarged* functional spaces generated by the basis functions, cf. Fig. 3.2 and Fig. 3.3.

Conceptually, the algorithm consist of the following steps:

1. Select a generic model for interactions by choosing orders K, P_1, P_2, P_3 and so on, in (3.6).
2. Select models for basis functions $h_{j,p}(x_j)$, $h_{js,p}(x_j, x_s)$, $h_{jsw,p}(x_j, x_s, x_w)$ and so on, cf. sec. 3.4.3.3.
3. Compute vectors $\dot{\boldsymbol{x}}_i = [\dot{x}_{i,1}, \dot{x}_{i,2}, \ldots, \dot{x}_{i,M}] \in \mathbb{R}^M$ for all i and construct the set of matrices $\mathcal{H}$ up to chosen K, where $\mathcal{H}$ is defined as

$$\mathcal{H} := \bigcup_{q=1}^{Q} H_q \quad , \tag{3.10}$$

   where $H_{N+j} := H_{jj}$, $H_{N+N^2+j} := H_{jjj}$ and so on, and $Q = \sum_{k=1}^{K} N^k$.
4. Define the set $\mathcal{N} := \{1, 2, \ldots, Q\}$ that contains all s indices of $\mathcal{H}$.
5. Define the set of incoming interactions of unit i at the l-th iteration as $\hat{\mathcal{L}}_{i,l}$, and initialize $\hat{\mathcal{L}}_{i0} = \varnothing$.

6. Compute $\forall\, n \in \mathcal{N} \setminus \hat{\mathcal{L}}_{i,(l-1)}$ the error function

$$\epsilon\left(\hat{\mathcal{L}}_{i,l}^{n}\right) := \sigma\left(\dot{\boldsymbol{x}}_i - H^{\dagger}_{\hat{\mathcal{L}}_{i,l}^{n}} H_{\hat{\mathcal{L}}_{i,l}^{n}} \dot{\boldsymbol{x}}_i\right), \tag{3.11}$$

   where $\sigma\left(\boldsymbol{y}\right)$ stands for the standard deviation of entries of $\boldsymbol{y}$, † indicates the Moore-Penrose pseudo-inverse, $\hat{\mathcal{L}}_{i,l}^{n} = \hat{\mathcal{L}}_{i,(l-1)} \cup \{n\}$ and

$$H_{\hat{\mathcal{L}}_{i,l}^{n}} := \left[H_{j_1}, H_{j_2}, \ldots, H_{j_{l-1}}, H_n\right]^{\mathsf{T}} \in \mathbb{R}^{l(K+1)\times M}, \tag{3.12}$$

   where $j_p \in \hat{\mathcal{L}}_{i,(l-1)}$ and $p = \{1, 2, \ldots, l-1\}$.

7. Store results in $\boldsymbol{\epsilon}_i(l) \in \mathbb{R}^L$ where $\left[\boldsymbol{\epsilon}_i(l)\right]_r = \epsilon\left(\hat{\mathcal{L}}_{i,l}^{n_r}\right)$, $n_r \in \mathcal{N} \setminus \hat{\mathcal{L}}_{i,(l-1)}$, $r \in \{1, 2, \ldots, L\}$ and $L = \left|\mathcal{N} \setminus \hat{\mathcal{L}}_{i,(l-1)}\right|$.

8. If the standard deviation $\sigma_l := \sigma(\boldsymbol{\epsilon}_i(l)) \leq \theta$, STOP algorithm and RETURN $\hat{\mathcal{L}}_{i,(l-1)}$ as set of incoming interactions.

9. Else, UPDATE $\hat{\mathcal{L}}_{i,l} = \hat{\mathcal{L}}_{i,(l-1)} \cup \{n^*\}$ where

$$n^* = \arg\min_{n\in\mathcal{N}} \left(\epsilon\left(\hat{\mathcal{L}}_{i,l}^{n}\right)\right) \tag{3.13}$$

   and REPEAT from step 6 until step 7 is satisfied or $\mathcal{N} \setminus \hat{\mathcal{L}}_{i,(l-1)} \neq \varnothing$.

For the condition $\mathcal{N} \setminus \hat{\mathcal{L}}_{i,(l-1)} \neq \varnothing$, the algorithm has a complexity of $\mathcal{O}\left(Q^2\right)$.

3.4. Reconstructing networks from dynamics

To test the framework described in this chapter, we generated high dimensional time series simulating directionally coupled networks of dynamical systems. The resulting time series were used as inputs for reconstructing network structural connections by means of ARNI.

So, to introduce a functional complexity in the coupling functions, we selected a model of Kuramoto-like oscillators with coupling functions having two Fourier modes [87]

$$\dot{x}_i = \omega_i + \frac{1}{n_i}\sum_{j=1}^{N} J_{ij}\left[\sin\left(x_j - x_i - 1.05\right) + 0.33\sin\left(2\left(x_j - x_i\right)\right)\right] + \xi_i, \tag{3.14}$$

having n_i randomly-selected incoming connections per node. Here, the entries of $J \in \mathbb{R}^{N\times N}$ are given by $J = R \odot A$, where $\odot$ stands for entry-wise-matrix product, $A \in \{0,1\}^{N\times N}$ is a randomly generated adjacency matrix with a fixed number n_i of non-zero entries per row and $R \in \mathbb{R}^{N\times N}$ with R_{ij} randomly drawn from a uniform distribution $R_{ij} \in [0.5, 1]$. The natural frequencies ω_i are randomly selected from a

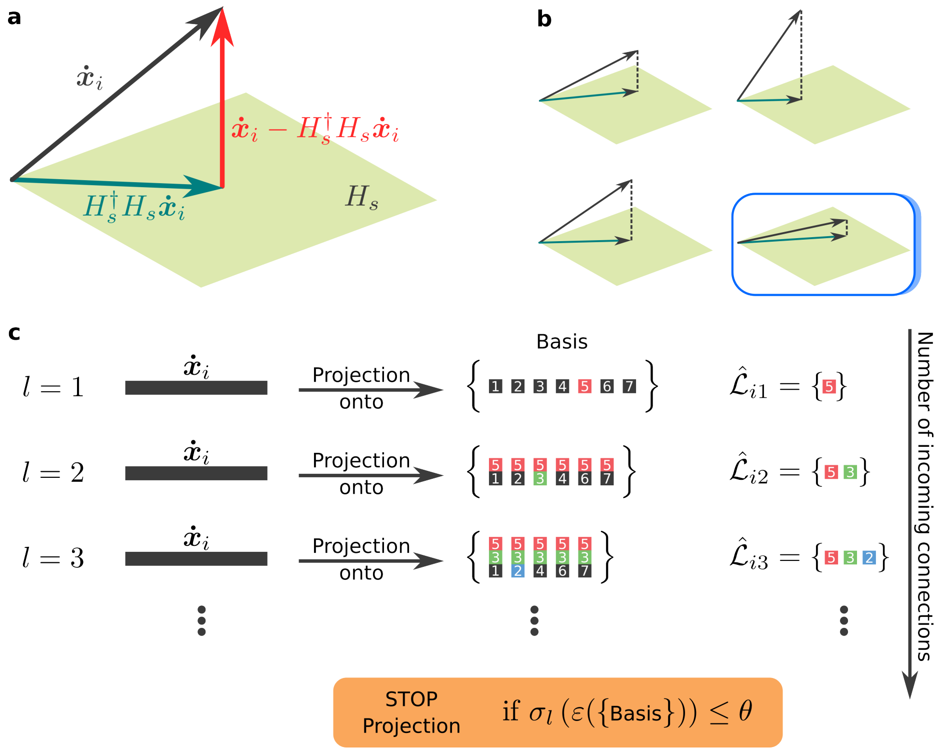

a
$\dot{\boldsymbol{x}}_i$
$\dot{\boldsymbol{x}}_i - H_s^{\dagger}H_s\dot{\boldsymbol{x}}_i$
$H_s^{\dagger}H_s\dot{\boldsymbol{x}}_i$
H_s
b
c
Basis
$l = 1$
$l = 2$
$l = 3$
Projection onto
$\hat{\mathcal{L}}_{i1} = \{5\}$
$\hat{\mathcal{L}}_{i2} = \{5\ 3\}$
$\hat{\mathcal{L}}_{i3} = \{5\ 3\ 2\}$
Number of incoming connections
STOP Projection
if $\sigma_l\left(\varepsilon(\{\text{Basis}\})\right) \leq \theta$

Figure 3.2. – ARNI reveals connections by projecting dynamics onto spaces spanned by basis functions. a, ARNI projects dynamical vectors $\dot{\boldsymbol{x}}_i$ onto spaces spanned by H_s, where $H_s^{\dagger}H_s\dot{\boldsymbol{x}}_i$ is the projection and $\dot{\boldsymbol{x}}_i - H_s^{\dagger}H_s\dot{\boldsymbol{x}}_i$ the error of the projection. **b**, ARNI selects the H_s yielding the largest projection, or conversely, the smallest error. **c**, ARNI iteratively constructs $\hat{\mathcal{L}}_i$ by means of $\hat{\mathcal{L}}_{i,l}$. At $l = 1$, $\dot{\boldsymbol{x}}_i$ is projected on all spaces and one incoming interaction n^* (red) is selected according to (3.13). At $l = 2$, $\dot{\boldsymbol{x}}_i$ is now projected on the composite spaces formed by the inferred incoming connection in $\hat{\mathcal{L}}_{i,1}$ and the rest of possible incoming interaction in $\mathcal{C}_1$. Then, one incoming interaction n^* (green) is selected according to (3.13). At $l = 3$, $\dot{\boldsymbol{x}}_i$ is projected in the composite spaces formed by the inferred incoming interactions in $\hat{\mathcal{L}}_{i,2}$ and the rest of possible incoming interactions in $\mathcal{C}_2$. Then, an incoming interaction n^* (blue) is selected according to (3.13). This process is repeated until the stopping criterion $\sigma_l \leq \theta$ is reached.

uniform distribution $\omega_i \in [-2, 2]$ and ξ_i represents external white noise affecting the dynamics of unit i.

To test the robustness of our approach against chaotic dynamics, we simulated networks of coupled Rössler oscillators [79]. The dynamics of each oscillator $\boldsymbol{x}_i = [x_i^1, x_i^2, x_i^3] \in \mathbb{R}^3$ is set by three differential equations

$$\begin{aligned}
\dot{x}_i^1 &= -x_i^2 - x_i^3 + \frac{1}{n_i}\sum_{j=1}^{N} J_{ij}\sin(x_j^1) + \xi_i^1, \\
\dot{x}_i^2 &= x_i^1 + 0.1x_i^2 + \xi_i^2, \\
\dot{x}_i^3 &= 0.1 + x_i^3\left(x_i^1 - 18.0\right) + \xi_i^3,
\end{aligned} \tag{3.15}$$

where n_i is the number of incoming connections to unit i. As before, the entries of J are given by $J = R \odot A$, where $R_{ij} \in [0.5, 1]$. Similarly, ξ_i^k with $k \in \{1, 2, 3\}$

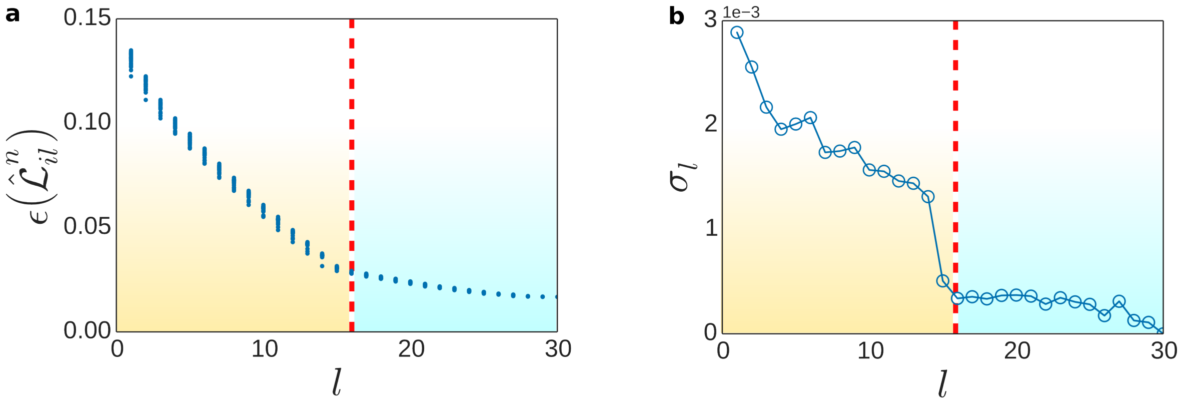


Figure 3.3. – ARNI iteratively arrives at the right solution. Reconstruction of a selected unit of a Kuramoto-like oscillators (3.14) of $N = 30$ and $n_i = 15$. **a,** Error function $\epsilon(\hat{\mathcal{L}}^{n}_{i,l})$ versus iteration index l. Each column represents the distribution of $\epsilon(\hat{\mathcal{L}}^{n}_{i,l}) \ : \ \forall n \in \mathcal{N} \setminus \hat{\mathcal{L}}_{i,(l-1)}$. **b,** Standard deviation σ_l of error vectors $\boldsymbol{\epsilon}_i(l)$ versus l. Red lines indicate when the connectivity is completely revealed.

represent external noisy signals acting on the unit's components.

Additionally, for mimicking real-world recordings, we sampled time series with sampling rates of $\Delta t = 1$ for all simulations. Nonetheless, it is important to note that simulations were computed with time steps $\delta t \ll \Delta t$.

3.4.1. Quantifying quality of reconstruction

Our reconstruction problem may be seen as a binary classification problem where one has to identify whether network interactions are active or not. Thus, if an *interaction is active* and it is *classified as active*, it is counted as a *true positive*, yet if it is *classified as inactive*, it is counted as a *false negative*. Conversely, if the *interaction is inactive* and it is *classified as inactive*, it is counted as a *true negative*, but if it is *classified as active*, it is counted as a *false positive*.

3.4.1.1. True positives and false positives rates

We measure the predictive power of our approach using the *true positives* (TPR) and *false positives* (FPR) *rates*. Basically, the TPR defines how many active interactions were *correctly identified as active* from all active interactions (actual interactions) present in the dynamics. Analogously, the FPR determines how many inactive interactions were *incorrectly identified as active* from all inactive interactions available. Both quantities are defined as

$$\text{TPR} = \frac{\text{True positives}}{\text{Total number of positives}}, \tag{3.16}$$

$$\text{FPR} = \frac{\text{False positives}}{\text{Total number of negatives}}. \tag{3.17}$$

3.4.1.2. Receiver Operating Characteristics (ROC) curve

A ROC curve is a two-dimensional graph in which TPR is plotted against FPR. Its purpose is to depict the trade-off between *true positives* and *false positives* of a given classifier [101].

Depending on its internal parameters, the classifier may yield better or worse results. Thus, ROC curves provide a graphical way to assess the predictive power of such classifiers with respect to their internal parameters, cf. Fig. 3.4a. Ideally, one seeks to have points in the ROC space as close to the upper-left corner as possible. Points there represent classifiers with high TPR and low FPR. Thus, classifiers in this area are more likely to correctly predict *true positives* than those close to the diagonal line.

The diagonal line TPR $=$ FPR represents a random-guessing classifier, cf. Fig. 3.4a. Thus, any classifier that appears below this line performs worse than random-guessing. Conversely, any classifier that appears above performs better than random-guessing.

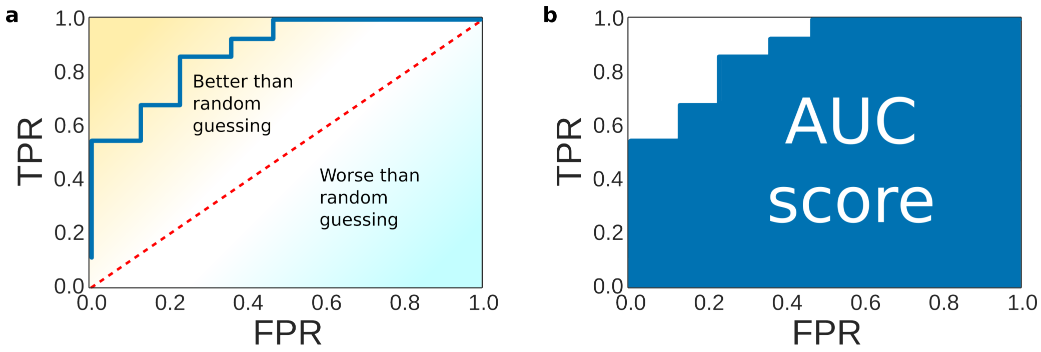


Figure 3.4. – Schematics of ROC curves and AUC scores. a, Generic example of a ROC curve. The ROC space may be divided in two elemental regions separated by the line TPR $=$ FPR. Classifiers above the diagonal line behave better than random-guessing whereas classifiers below the line behave worse than random-guessing. This curve indicates that the classifier correctly predicts true positives while having few false positives. **b**, AUC score for the classifier shown in **a**. The AUC score measures the area under the ROC curve, thus, it reaches a maximum at AUC score $= 1$ for a perfect classifier.

3.4.1.3. Area Under the Curve (AUC) score

Still, one may want to characterize the predictive power of a classifier in terms of a single scalar variable rather than interpreting a two-dimensional graph. A broadly used method is to measure the Area Under the ROC Curve, also known as the AUC score [101]. As its name indicates, the AUC score quantifies the area below the ROC curve of a given classifier, and thereby, AUC score $\in [0, 1]$, cf. Fig. 3.4b.

The AUC score provides an easy way to compare the predictive power of different classifiers. For instance, the higher the AUC score of a classifier is, the higher its predictive power. Also, the AUC score is equivalent to the probability that the classifier will rank (in a list of probable interactions) a randomly-selected *active interaction* higher than a randomly-selected *inactive interaction* [101]

3.4.2. Sampling of dynamics space reveals network connectivity

It was demonstrated in network models of spiking neurons [26, 27] and in network models with generic interactions [24] (cf. chapter 3) that a specific high-dimensional time series may be generated by a family of disparate networks having different connectivities and interactions. However, if one assumes that functional forms f_i and connections Λ^i do not change with time, then a specific network state $\boldsymbol{x}$ is always mapped in the same manner, as set by $\dot{x}_i = f_i\left(\Lambda^i \boldsymbol{x}\right)$. Hence, one may expect (in an extreme case) that long time series that do not converge to any stable dynamics (e.g., steady-states, limit cycles or periodic orbits) reveal distinctive features of the actual network under study by *extensively sampling its dynamics space*, cf. Fig. 3.5a.

Reconstructing networks of Kuramoto-like oscillators (3.14) employing ARNI shows that AUC scores for predictions of connections improve with longer time series, cf. Fig. 3.5d. Thus, indicating that the quality of reconstruction improves with the length M of time series.

This suggests that an extensive sampling of the dynamics space through time series may indeed reveal network interactions, as guessed initially. Yet, if this extensive sampling is the responsible of revealing interactions, then sampling different regions of the dynamics space through shorter time series should also reveal network connections, cf. Fig. 3.5b.

Reconstructions from compositions of short but distributed time series reveal that AUC scores also improve with longer time series, cf. Fig. 3.5d. Thus, confirming that an extensive sampling of the dynamics space reveals network interactions.

These results also hold for networks of systems determined by more than one state variable $x_i(t)$. Intrinsically, higher-dimensional units are low-dimensional networks. For instance, Rössler oscillators (3.15) may be seen as networks of three units interacting with each other, cf. Fig. Fig. 3.7a. Thus, for networks of higher-dimensional dynamical systems, connections may be recovered through ARNI by extending equation (3.2) to include all components $x_i^d(t)$ of $i \in \{1, 2, \ldots, N\}$, where $d \in \{1, 2, \ldots, D_i\}$ and D_i is the number of components of unit i, cf. Fig. 3.7c.

3.4.3. Technical issues

In the following, we show certain aspects to be considered when reconstructing a network using this approach.

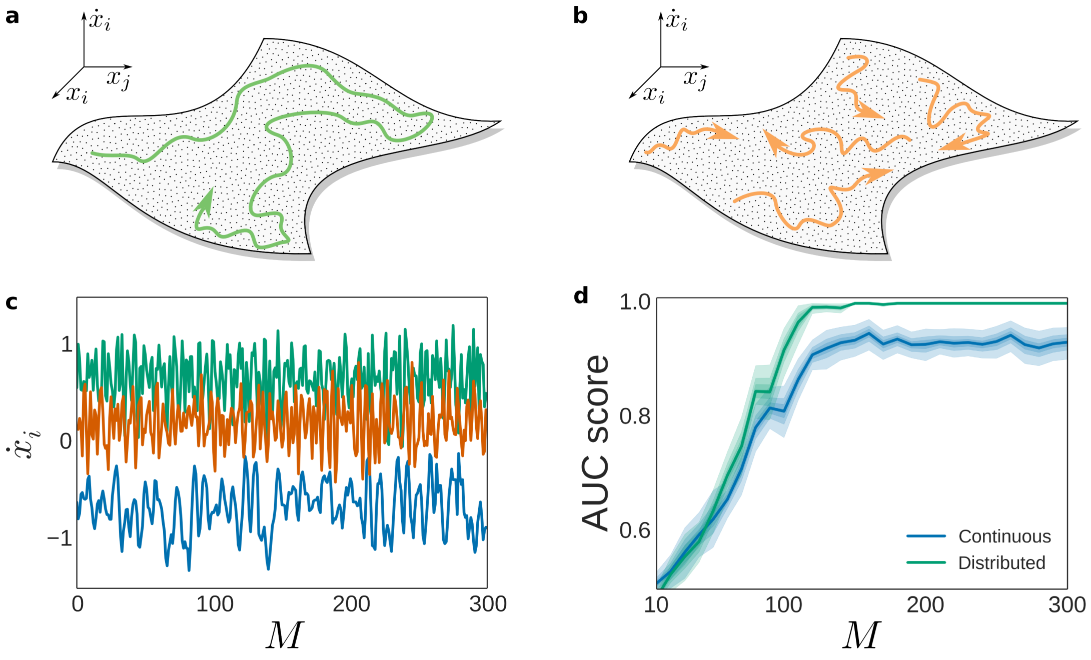


Figure 3.5. – Extensive sampling of the dynamics space reveals network interactions. (**a**,**b**), Sampling strategies. **a**, A long sampling of the dynamics space of a network (and units) by means of time series may reveal distinctive features that are unique to the actual network under study. **b**, Shorter time series may also reveal particular features of the actual network when distributed across the dynamics space, thus sampling different regions of it. (**c**,**d**), Reconstruction of networks of Kuramoto-like oscillators (3.14) of $N = 20$ and $n_i = 10$ for $i \in \{1, 2, \ldots, N\}$. **c**, Derivatives $\dot{x}_i$ for three selected units. The data does not show any kind of regular dynamics. **d**, Area under the curve (AUC) score versus length of time series M for both sampling strategies. For the distributed sampling, we simulated short time series of $m = 10$ starting from different initial conditions, and created composite longer time series from the shorter ones. The reconstruction quality improves with M for both sampling strategies. However, a distributed sampling determines the connectivity the best.

3.4.3.1. Noisy measurements

Random fluctuations are ubiquitous in nature. For instance, in gene networks, noisy dynamics is simultaneously present at several different levels (e.g. gene-intrinsic, network intrinsic and cell-intrinsic) [102, 103]. Thus, how feasible is it to reconstruct networks from noisy dynamics through ARNI?

Fundamentally, noise hinders the identification problem by corrupting measurements of units' dynamics, thereby masking network interactions, cf. Fig. 3.6a and Fig. 3.7b. However, sampling over longer periods of time mitigates this effect, cf. Fig. 3.6b and Fig. 3.7c.

Furthermore, our numerics indicate that the necessary length $M_{0.95}$ of time series for achieving reconstructions of AUC scores larger than 0.95 scales super-linearly (presumably as $M_{0.95} \sim e^{\lambda}$) with the noise strength λ, cf. Fig. 3.6c.

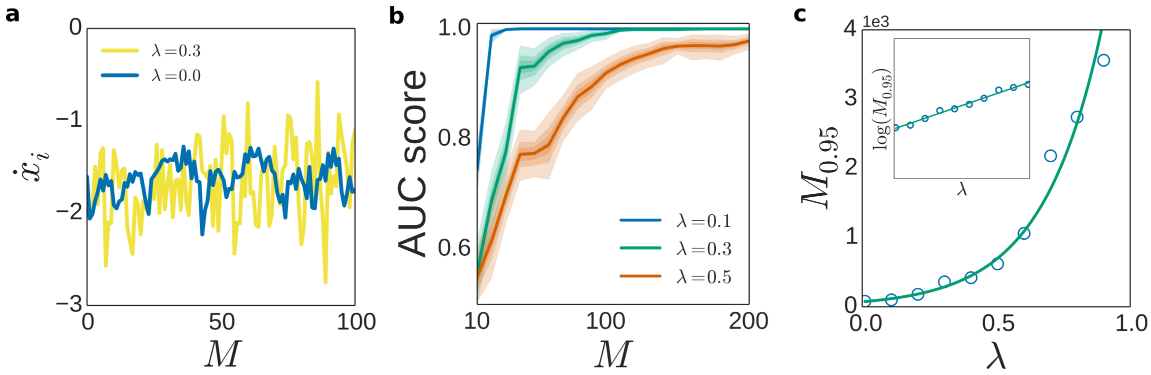


Figure 3.6. – ARNI reveals structural connections from noisy time series. (**a**,**b**,**c**) Reconstruction of networks of Kuramoto-like oscillators (3.14) of $N = 20$ and $n_i = 5$. **a**, Derivative $\dot{x}_i$ for a noise-free ($\lambda = 0$) and noisy dynamics ($\lambda = 0.3$). **b**, Area under the curve (AUC) score versus M. Reconstruction improves with M. This indicates that longer samplings mitigate errors induced by noisy measurements. **c**, Minimum length $M_{0.95}$ of time series required for achieving AUC scores larger than 0.95 versus noise strength λ with the best exponential fit. The inset shows the same data with $M_{0.95}$ on logarithmic scale.

3.4.3.2. Network size and number of incoming connections

Is reconstruction still feasible with increasing size? How does the number of incoming connections affect the identification problem? Our numerical results suggest that reconstruction scales with both network size and number of incoming connections.

For networks of different sizes with a fixed number of incoming connections per node, we found that reconstruction scales sub-linearly with network size, cf. Fig. 3.8a. Specifically, we found that the length of time series $M_{0.95}$ for achieving successful reconstructions increases sub-linearly (presumably as $M_{0.95} \sim \log(N)$) with the size N of the network.

Reconstructing networks of fixed size shows that reconstruction scales linearly with the number of incoming connections, cf. Fig. 3.8b. Thus, higher M is necessary for achieving optimal results with increasing number of incoming connections n_i.

Furthermore, these results highlight the applicability of our approach for reconstructing real-world networks, which are often sparse (reduced n_i per node) and large in size.

3.4.3.3. Choosing proper basis functions

Selecting an appropriate class of basis functions to represent the network interactions is a vital step of our approach. Choosing basis functions that capture the intrinsic nature of interactions (e.g. single $h(x_i)$, pairwise $h(x_i, x_j)$, triplet $h(x_i, x_j, x_w)$, and so on) will lead to optimal results. However, finding appropriate class of basis functions should not be confused with finding a specific set of basis functions. While the former implies to find basis functions of correct order $k \in \{1, 2, \ldots, K\}$, cf. Fig. 3.9,

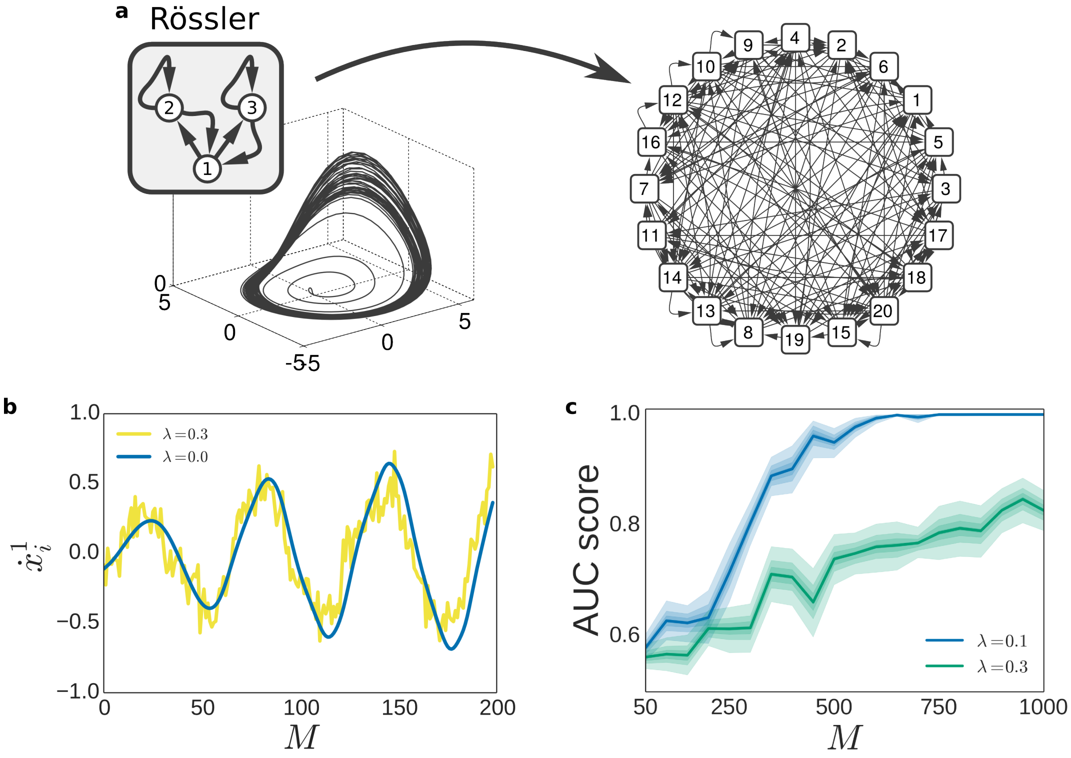


Figure 3.7. – Reconstructing networks of higher-dimensional chaotic systems from noisy time series. **a**, Rössler oscillators, when assembled together, may be seen as networks composed of smaller and simpler sub-networks represented by the individual systems. (**b**,**c**) Reconstruction of networks of Rössler oscillators (3.15) of $N = 20$ and $n_i = 5$. **b**, Derivative $\dot{x}_i$ for a noise-free ($\lambda = 0$) and noisy dynamics ($\lambda = 0.3$). **c**, Area under the curve (AUC) score versus M. Reconstruction improves with M. Thus, indicating that reconstruction of chaotic systems is feasible even from noisy measurements. However, the necessary M for achieving successful reconstruction increases with larger noise strengths λ, cf. Fig.3.6.

the latter implies to find a unique set of basis functions capable of representing the network dynamics.

Reconstruction of networks of Kuramoto-like oscillators (3.14) using different sets of basis functions reveal that employing bases resembling the intrinsic nature of interactions (e.g. pairwise) lead to optimal results, cf. Fig. 3.9. The rationale is that expansions in basis functions seek to span functional spaces, thus, functions involving x_i and x_j together, $h(x_i, x_j)$, are more suitable to represent the dependencies induced by pairwise interactions in (3.14) than functions just involving single interactions, $h(x_i)$ and $h(x_j)$.

3.5. Discussion

In this chapter, we introduced a novel and general approach for reconstructing structural connections of networks of dynamical systems from time series alone. While

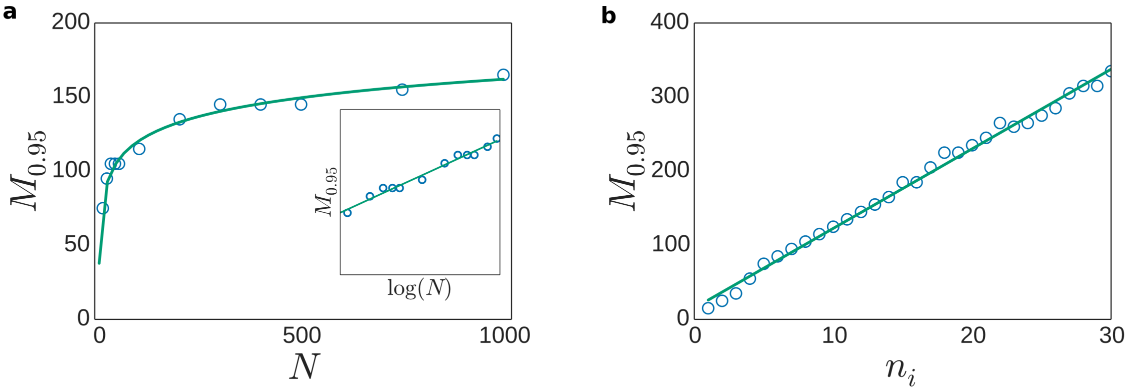
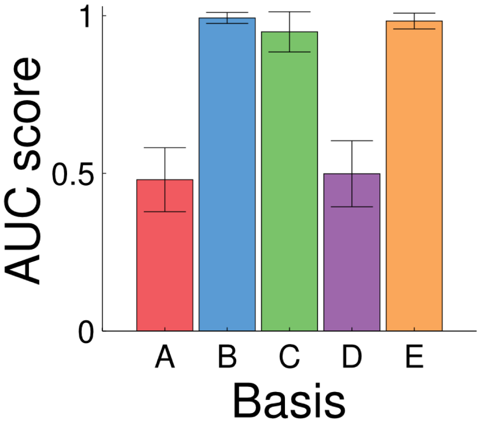


Figure 3.8. – **Reconstruction scales with network size and with number of incoming connections.** Reconstruction of Kuramoto-like oscillators (3.14). **a**, Minimum length $M_{0.95}$ of time series required for achieving AUC score larger than 0.95 versus network size N with the best logarithmic fit. The inset shows same data with N on logarithmic scales. The number of incoming connections $n_i = 8$ was fixed for all networks. **b**, Minimum length $M_{0.95}$ of time series required for achieving AUC score larger than 0.95 versus number of incoming connections n_i with best linear fit. All networks were of size $N = 50$.

only relying on the assumption that units are interacting with each other, we proposed a decomposition of units' dynamics in terms of network interactions. These decompositions proved useful for revealing physical links between units from recorded time series. Furthermore, we developed an efficient algorithm for retrieving the network structure from such decompositions. Interestingly, we found that sufficiently long observations and composition of several short observations of units' dynamics reveal network structural connections, suggesting that the sampling of the dynamics space through time series is the responsible of revealing structural links of networks.

Previous studies on inferring structural connections from time series have focused on the reconstruction of networks with known local and coupling functions [22,68–70,73,74]. Such prior knowledge reduces the problem of inferring connections to a standard linear algebra problem, where one has to solve a linear system of equations to reveal the network connections, cf. Ref. [22] for a comprehensive review. Here, however, we extended the up-to-date theory and developed an approach that does not require any information about local and coupling functions. Instead, our approach solely relies on recorded time series, thereby making it especially suitable when measuring techniques are restricted to simply observe the network dynamics. Particularly, we achieved this model-independency by expressing coupling functions in terms of expansions in basis functions.

However, it is important to mention that we are not the first suggesting the use of basis expansions for network reconstruction. These were previously used in [70,71,74] for inferring physical links utilizing the compressed sensing framework, cf. Ref. [104] for an introductory review. Specifically, the authors expand the coupling functions in terms of linear combinations of basis functions, and then proceed to determine which linear coefficients are different than zero. Such coefficients determine the



$$A : h_{j,p}(x_j) = x_j^p$$

$$B : h_{ij,p}(x_i, x_j) = (x_j - x_i)^p$$

$$C : h_{ij,p_1p_2}(x_i, x_j) = x_j^{p_1} x_i^{p_2}$$

$$D : h_{j,p}(x_j) = \sin(p(x_j)) \\ \cos(p(x_j))$$

$$E : h_{ij,p}(x_i, x_j) = \sin(p(x_j - x_i)) \\ \cos(p(x_j - x_i))$$

Figure 3.9. – Appropriate basis functions reveal network interactions. AUC score for reconstruction of networks of $N = 60$ and $n = 20$ Kuramoto-like oscillators (3.14) employing distinct basis functions, Basis $\in [A, B, C, D, E]$. Choosing a set of basis functions that do not capture the nature of interactions leads to erroneous results.

presence or the absence of connections. Nevertheless, in their examples, the basis expansions contain the actual coupling functions involved in the network dynamics, cf. Ref. [71] for a clear example. Thus, instead of representing the unknown coupling functions as linear combinations of basis functions, the authors simply propose a collection of proxy functions (including the actual coupling functions) for representing the network dynamics. Such strategy is similar to that proposed in [69], where the authors provide the coupling functions defining the network dynamics and solve a linear system of equations that yields the network connectivity.

Here, in contrast to [70, 71, 74], we take a more general perspective. Specifically, rather than assuming that expansions may contain the actual coupling functions, we instead assume that coupling functions are only formed by linear combinations of basis functions. Such assumption enhances the capability of expansions to span broader functional spaces while reducing the necessary pre-knowledge about the network features. Moreover, it increases the expansions' efficiency for representing more complex coupling functions.

Finally, the present study affirmatively answers the question of whether time series may uniquely define network structures. Yet, further efforts should be focused on devising ways to implement different types of experimental recordings jointly with time series. Such additional recordings may reduce the number of time points necessary to achieve successful reconstructions. In addition, further efforts should also be oriented to develop methods for reconstructing networks where only a restricted subset of the network is measured, *networks with hidden units*. Furthermore, based on the findings of this chapter, a promising direction for future research is to extend our framework to predictability of network dynamics based on recorded time series.

Chapter 4

Network connectivity from local samplings

In chapter 3, we demonstrated that time series alone may uniquely determine the connectivity structure underlying the dynamics of networks. Specifically, we showed that a sufficiently extensive sampling of the dynamics space reliably reveals physical links among units. However, producing such long samplings may be experimentally costly (or even infeasible), especially in cases where networks are densely connected and large in size.

So, is there an alternate strategy for recovering network connections that does not require an extensive sampling of network dynamics? Interestingly, there is. It has been shown in networks exhibiting stable collective dynamics (e.g. steady states or periodic orbit) that applying external driving signals on such dynamics may reveal network connections [4,6,7,64–66,105] . For instance, applying small driving signals (such as to not kick the system out of the basin of attraction of the stable state) effectively shifts the stable dynamics in the state space while qualitatively keeping the stable dynamics the same [22], cf. Fig. 4.1. Thus, applying different signals on a stable dynamics generates qualitatively similar responses. Such are determined partly by (i) the driving signals, and (ii) the network's particular features, such as its underlying interactions and connections. In particular, these collective responses yield linear constraints on the network topology such that sufficiently many driving-response experiments may reveal the entire network connectivity [22].

However, most of these driving-response approaches require that the local dynamics and the approximate forms of coupling functions to be explicitly known. Moreover, these approaches also need the collective dynamics (i) to be simple enough (e.g. stable steady state or periodic orbit), or (ii) to admit to be driven to stable states. Furthermore, networks exhibiting more complex dynamics, such asynchronous chaotic activity or other common features of higher-dimensional and nonlinear systems are certainly not suited to be reconstructed by this kind of approaches [4, 7, 22, 64–66]. Therefore, the scope of these approaches is limited to only well-accessible systems.

Yet, is a stable dynamics necessary at all for revealing the network connectivity from driving-response experiments? Also, do we need to know network features, such as

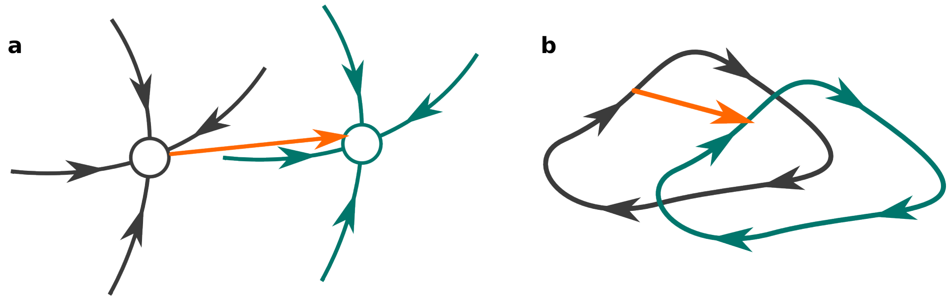


Figure 4.1. – Driving-response approaches on stable dynamics for reconstructing networks. a, Constant external driving signal shifts a stable steady state (black) in the state space to another position (teal). **b**, Constant external driving signal shifts a stable periodic orbit (black) in the state space to another position (teal). In both panels, the displacement (orange) depends on (i) the driving signal and (ii) the network intrinsic features (e.g. local dynamics, coupling functions and connections). Thus, many displacements under different driving conditions yields information on the network connectivity. Figure modified from [22].

coupling functions, in advance to understand how units interact with each other? Moreover, can we explore similar strategies for inferring connections in networks describing more complex dynamics than simple stable dynamics?

In this chapter, we propose a model-independent framework to reveal the network connectivity from local samplings of network dynamics. Assuming *linearity around a reference dynamical trajectory* in the dynamics space, we infer the connectivity of complex networks by comparing reference trajectories to nearby dynamical trajectories. In contrast to the approaches described in [4, 7, 22, 64–66], our approach is not constrained to networks displaying only stable dynamics nor need to be driven to steady-states. Instead, here we exploit the linearity of local samplings to demonstrate the direct relation between nearby trajectories and Jacobian matrices evaluated at the reference trajectory. Furthermore, these Jacobian matrices fully map structural connections of networks. Interestingly, the length of used trajectories is much smaller than the time series employed in chapter 3 by several orders of magnitude ($\sim$ 10 measurements). To demonstrate the capabilities of our approach, we retrieve full network connectivity of networks of coupled Kuramoto-like oscillators exhibiting complex dynamics [87] by recording nearby trajectories under K independent driving signals. Moreover, assuming that networks arising in real-life situations are essentially sparse, we enhanced the approach for recovering the network connectivity even if the number of different nearby trajectories is much smaller than the number of units in the network.

4.1. Mapping nearby dynamics to network connectivity

As in the previous chapter, here we will describe the dynamics of networks in terms of generic dynamical systems (1.9), cf. chapter 1.

To understand how sets of nearby dynamics reveal network structures, we will focus in this section on the special case of having many *perturbed replicas* of the same system starting from identical initial conditions. Specifically, we will drive the replicas through external constant signals to generate different nearby trajectories. Then, these trajectories will be used to map the network connections. Nonetheless, the concepts, arguments and results developed in this section also apply to cases where a local sampling is possible but interfering with the network dynamics is infeasible.

So, how can we map transient response dynamics resulting from perturbations to the network connectivity? Specifically, how can we relate the evolution of a network following a perturbation of its units' dynamics to network structural connections? To answer these, let us assume we are given K perturbed replicas

$$\dot{y}_{ik} = f_i\left(\Lambda^i \boldsymbol{y}_k\right) + I_{ik}, \tag{4.1}$$

of a system of the form (1.9) where I_{ik} is a driving signal acting on the dynamics of i-th unit of the k-th replica and $\boldsymbol{y}_k = [y_{1k}, y_{2k}, \ldots, y_{Nk}]^{\mathsf{T}} \in \mathbb{R}^N$. Let us denote the state of the original network as $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^{\mathsf{T}} \in \mathbb{R}^N$ and the time derivative of the i-th unit as $\dot{x}_i \in \mathbb{R}$. If the difference between systems states[1] $\|(\boldsymbol{x}, \dot{x}_i) - (\boldsymbol{y}_k, \dot{y}_{ik})\|_2$ is sufficiently small, we may calculate a first order approximation of (4.1) around $\Lambda^i \boldsymbol{x}$ as

$$\dot{y}_{ik} = \dot{x}_i + \boldsymbol{\nabla} f_i \Lambda_i \left[\boldsymbol{y}_k - \boldsymbol{x}\right] + I_{ik}, \tag{4.2}$$

where $\boldsymbol{\nabla} f_i = \left[\dfrac{\partial f_i}{\partial x_1}, \dfrac{\partial f_i}{\partial x_2}, \ldots, \dfrac{\partial f_i}{\partial x_N}\right] \in \mathbb{R}^N$ is the gradient of f_i, cf. Fig. 4.2. The entries of the product $\boldsymbol{\nabla} f_i \Lambda^i = \left[\dfrac{\partial f_i}{\partial x_1}\Lambda^i_{11}, \dfrac{\partial f_i}{\partial x_2}\Lambda^i_{22}, \ldots, \dfrac{\partial f_i}{\partial x_N}\Lambda^i_{NN}\right] \in \mathbb{R}^N$ serve as predictors for the network connectivity[2]. Thus, by producing sufficiently many $K \gg N$ replicas satisfying approximation (4.2) one may solve for $\boldsymbol{\nabla} f_i \Lambda^i$ in a similar manner as it is done for the Laplacian matrix in [64]. However, given that $\boldsymbol{\nabla} f_i \Lambda^i$ evolves in time, we may infer some entries $\partial f_i \Lambda^i_{jj} / \partial x_j$ as zero due to the time evolution of the network encoded in $\partial f_i / \partial x_j$ and not due to the explicit dependencies in Λ^i.

One may reduce the chances of inferring false absent connections by analyzing the evolution of gradients in a short time window of size M following the perturbation. Thus, defining $\boldsymbol{g}_i := [\boldsymbol{\nabla} f_{i,1}, \boldsymbol{\nabla} f_{i,2}, \ldots, \boldsymbol{\nabla} f_{i,M}] \in \mathbb{R}^{NM}$ with $\boldsymbol{\nabla} f_{i,m}$ being the gradient of the function evaluated at the m-th time step, we may extend equation (4.2) for M time steps as

[1] Time derivatives $\dot{x}_i$ and $\dot{y}_{ik}$ may be estimated from recorded time series as shown in equation (4.16).

[2] When all gradients are considered they form the Jacobian Matrix Jf_i at a specific point in the state space.

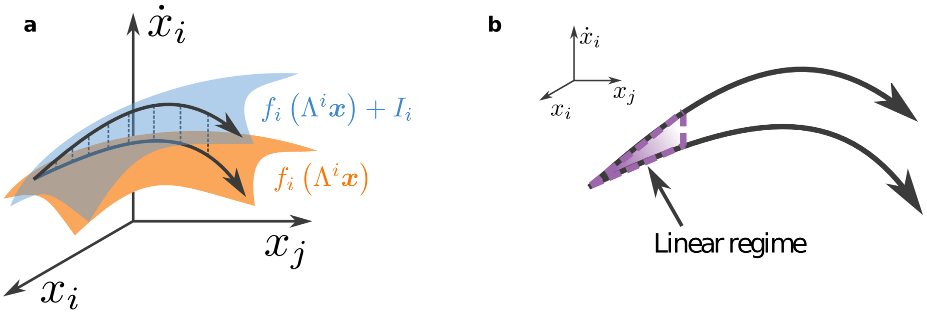

a
$\dot{x}_i$
$f_i\left(\Lambda^i \boldsymbol{x}\right) + I_i$
$f_i\left(\Lambda^i \boldsymbol{x}\right)$
x_j
x_i
b
$\dot{x}_i$
x_j
x_i
Linear regime

Figure 4.2. – Transient response dynamics and linear regime. **a**, Unperturbed and perturbed dynamics of unit i in the original network (orange) and perturbed replica (blue). The manifolds containing the dynamics are set by the f_i, the Λ^i and the driving signals I_{ik}. If the two systems start from the same initial condition, applying a constant driving signal on the replica will drive both dynamics apart with increasing time. **b**, However, for a time window close to the activation of the driving signal, the replica's dynamics may be linearly approximated around the original dynamics as explained in equation (4.3).

$$\dot{\boldsymbol{y}}_{ik} = \dot{\boldsymbol{x}}_i + \boldsymbol{g}_i \Gamma^i Y_k + I_{ik} \mathbf{1}_M, \tag{4.3}$$

where $\dot{\boldsymbol{y}}_{ik} = [\dot{y}_{ik,1}, \dot{y}_{ik,2}, \dots, \dot{y}_{ik,M}] \in \mathbb{R}^M$ and $\dot{\boldsymbol{x}}_i = [\dot{x}_{i,1}, \dot{x}_{i,2}, \dots, \dot{x}_{i,M}] \in \mathbb{R}^M$ are the rate of change of both k-th replica and original networks for a given time window and $\mathbf{1}_M$ is a row vector of size M full of ones . The matrices $\Gamma^i \in \mathbb{R}^{NM \times NM}$ and $Y \in \mathbb{R}^{NM \times M}$ are block diagonal matrices defined as

$$\Gamma^i := \begin{bmatrix} \Lambda^i & 0_{N\times N} & \dots & 0_{N\times N} \\ 0_{N\times N} & \Lambda^i & \dots & 0_{N\times N} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{N\times N} & 0_{N\times N} & \dots & \Lambda^i \end{bmatrix}, \tag{4.4}$$

and

$$Y := \begin{bmatrix} \boldsymbol{y}_{k,1} - \boldsymbol{x}_1 & \mathbf{0}_{N\times 1} & \dots & \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} & \boldsymbol{y}_{k,2} - \boldsymbol{x}_2 & \dots & \mathbf{0}_{N\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{N\times 1} & \mathbf{0}_{N\times 1} & \dots & \boldsymbol{y}_{k,M} - \boldsymbol{x}_M \end{bmatrix}, \tag{4.5}$$

where $\boldsymbol{y}_{k,m} = [y_{1k,m}, y_{2k,m}, \dots, y_{Nk,m}]^\mathsf{T} \in \mathbb{R}^N$ and $\boldsymbol{x}_m = [x_{1,m}, x_{2,m}, \dots, x_{N,m}]^\mathsf{T} \in \mathbb{R}^N$ are the network state of both k-th replica and original systems at time step m and $\mathbf{0}_{C\times S}$ is a matrix of size $C \times S$ full of zeros.

Perturbing K replicas yields the system of equations

$$\dot{\boldsymbol{y}}_i = \mathbf{1}_{1\times K} \otimes \dot{\boldsymbol{x}}_i + \boldsymbol{g}_i \Gamma^i Y + \boldsymbol{I}_i \otimes \mathbf{1}_{1\times M}, \tag{4.6}$$

where $\dot{\boldsymbol{y}}_i = [\dot{\boldsymbol{y}}_{i1}, \dot{\boldsymbol{y}}_{i2}, \dots, \dot{\boldsymbol{y}}_{iK}] \in \mathbb{R}^{KM}$ is a vector containing the K transient response dynamics, $Y = [Y_1, Y_2, \dots, Y_K] \in \mathbb{R}^{NM\times KM}$ is a matrix containing the differences between transient response and original dynamics for K different driving signals and $\boldsymbol{I}_i = [I_{i1}, I_{i2}, \ldots, I_{iK}] \in \mathbb{R}^K$ is a vector containing the driving signals used to generate each transient response dynamics. Here the operator $\otimes$ stands for Kronecker product.

Solving the equation (4.6) yields the vector of gradients $\boldsymbol{g}_i\Gamma^i$ as a function of solely measured data. At this point, one may naively assume that (4.6) is easily solvable once the number of transient responses is equal to the number of units in the network, $K = N$. Nonetheless, it is known that this problem can be numerically ill-conditioned [106] for large N. Thereby, results for $K = N$ may not be accurate nor reliable.

One may surpass this limitation by producing more transient responses than units in the network, $K > N$. This turns (4.6) into an overdetermined system of equations to which we can only find an approximate solution that minimizes the error function [22, 106]

$$E_i(\boldsymbol{g}_i\Gamma^i) = \left\| \dot{\boldsymbol{y}}_i - \mathbf{1}_{1\times K} \otimes \dot{\boldsymbol{x}}_i - \boldsymbol{g}_i\Gamma^i Y - \boldsymbol{I}_i \otimes \mathbf{1}_{1\times M}, \right\|_2^2 . \tag{4.7}$$

So, if we generate many replicas such that $K > N$, we may find an analytical approximation for $\boldsymbol{g}_i\Gamma^i$ from equation (4.6) as

$$\boldsymbol{g}_i\Gamma^i = (\dot{\boldsymbol{y}}_i - \mathbf{1}_{1\times K} \otimes \dot{\boldsymbol{x}}_i - \boldsymbol{I}_i \otimes \mathbf{1}_{1\times M})\, Y^{\dagger}, \tag{4.8}$$

where the † operator stands for Moore-Penrose pseudo-inverse, cf. Appendix A. It is important to note that (4.8) only relies on measured data, thus, requires no *a priori* information about the network intrinsic features, such as local dynamics and coupling functions.

4.2. Quantifying network connectivity

How can reconstructed gradients from equation (4.8) help in finding network connectivity? This depends on how the resulting entries of $\boldsymbol{g}_i\Gamma^i$ produced by (4.8) are analyzed. Firstly, let us arrange the entries of $\boldsymbol{g}_i\Gamma^i$ in a matrix $G^i \in \mathbb{R}^{M\times N}$ defined as

$$G^i := \begin{bmatrix} \frac{\partial f_{i,1}}{\partial x_1}\Lambda^i_{11} & \frac{\partial f_{i,1}}{\partial x_2}\Lambda^i_{22} & \cdots & \frac{\partial f_{i,1}}{\partial x_N}\Lambda^i_{NN} \\ \frac{\partial f_{i,2}}{\partial x_1}\Lambda^i_{11} & \frac{\partial f_{i,2}}{\partial x_2}\Lambda^i_{22} & \cdots & \frac{\partial f_{i,2}}{\partial x_N}\Lambda^i_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{i,M}}{\partial x_1}\Lambda^i_{11} & \frac{\partial f_{i,M}}{\partial x_2}\Lambda^i_{22} & \cdots & \frac{\partial f_{i,M}}{\partial x_N}\Lambda^i_{NN} \end{bmatrix}, \tag{4.9}$$

where $(\partial F_{i,m}/\partial x_j)$ is the partial derivative of f_i with respect to the activity of unit j evaluated at time step m. Therefore, the m-th row of G^i defines the gradient $\boldsymbol{\nabla} f_i$ evaluated at the m-th time step and the j-th column comprises the temporal behavior of $\partial f_i/\partial x_j$. Secondly, let us define the characterizer

$$\boldsymbol{\alpha}^i := \boldsymbol{L}(G^i) \in \mathbb{R}^N, \tag{4.10}$$

with $\boldsymbol{L} : \mathbb{R}^{M\times N} \to \mathbb{R}^N$ being an operator defined as

$$L_j(G^i) := \frac{1}{M}\sum_{m=1}^{M}\left|G^i_{mj}\right| + \frac{1}{M}\sum_{m=1}^{M}\left(G^i_{mj} - \mu^i_j\right)^2, \tag{4.11}$$

where $j \in \{1, 2, \dots, N\}$ and μ^i_j is the mean of the entries of the j-th column of G^i and is given by $\mu^i_j = M^{-1}\sum_{m=1}^{M} G^i_{mj}$. The operator (4.11) seeks to discern between existing links from non-existing links by (i) quantifying an absolute mean interaction coming from unit j to i (first term), and (ii) quantifying fluctuations around a mean interaction (second term). Finally, by choosing a threshold ψ^i, we may reconstruct the entries of the explicit dependency matrix as

$$\hat{\Lambda}^i_{jj} = H(\alpha^i_j - \psi^i). \tag{4.12}$$

where H is the Heaviside step function; $H(x) = 1$ for $x \geq 0$ and $H(x) = 1$ for $x < 0$.

4.3. Reconstructing networks of oscillators

To test the predictive power of our framework, we simulated networks of directionally coupled Kuramoto-like oscillators with coupling functions having two Fourier modes [87]

$$\dot{x}_i = \omega_i + \frac{1}{n_i}\sum_{j=1}^{N} J_{ij}\left[\sin\left(x_j - x_i - 1.05\right) + 0.33\sin\left(2\left(x_j - x_i\right)\right)\right]. \tag{4.13}$$

with n_i incoming connections per node. Here, the entries of J are given by $J = R \odot A$, where $A \in \{0,1\}^{N\times N}$ is a randomly-generated adjacency matrix and $\odot$ stands for entrywise matrix product. Both ω_i and R_{ij} are randomly drawn from uniform distributions in the respective intervals $\omega_i \in [-2, 2]$ and $R_{ij} \in [0.5, 1]$. Also, we simulated K perturbed replicas of (4.13) as

$$\dot{y}_{ik} = \omega_i + \frac{1}{n}\sum_{j=1}^{N} J_{ij}\left[\sin\left(y_{jk} - y_{ik} - 1.05\right) + 0.33\sin\left(2\left(y_{jk} - y_{ik}\right)\right)\right] + I_{ik}, \tag{4.14}$$

where $k \in \{1, 2, \dots, N\}$ is the replica index, $I_{ik} \in [-0.05, 0.05]$ is an external driving signal applied on the i-th unit of the k-th replica and

$$y_{ik}(t=0) = x_i(t=0) + \delta_{ik} \tag{4.15}$$

with δ_{ik} being randomly selected from a uniform distribution in the interval $\delta_{ik} \in [-0.025, 0.025]$, cf. Fig. 4.3 for one realization.

Assuming that simulations yield high-dimensional time series of a network $\forall\, i \in \{1, 2, \ldots, N\} : x_{i,m} = x_i(t_m)$, where $t_m = m\Delta t + t_0$ and Δt is sufficiently small such as to reliably estimate time derivatives $\dot{x}_i$ from measured time series, we calculate the time derivative $\dot{x}_{i,m}$ at the m-th time step as

$$\dot{x}_{i,m} = \frac{x_{i,m+1} - x_{i,m-1}}{2\Delta t}. \tag{4.16}$$

All simulations were performed in a time interval $t \in [0, 1]$ with $\Delta t = 0.1$, thus, $M = 10$.

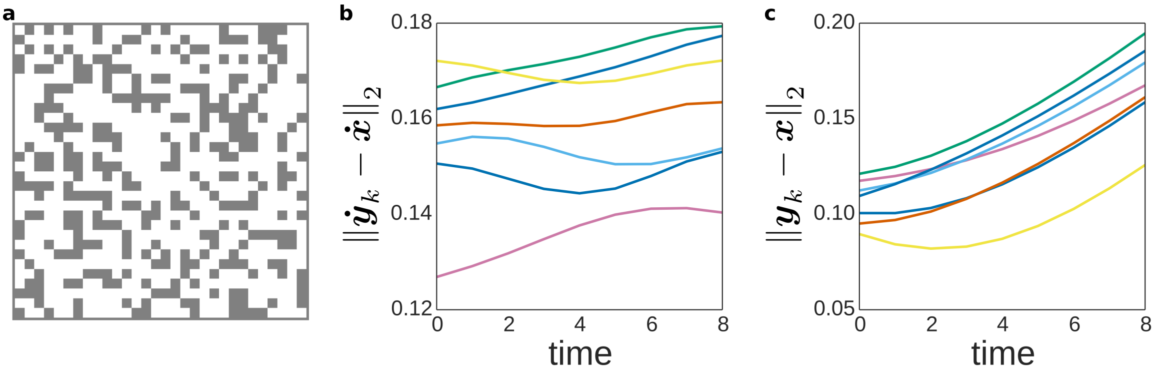


Figure 4.3. – External signals effectively drive replicas apart from the original system with increasing time. a, Adjacency matrix of a network of $N = 30$ and $n = 10$ Kuramoto-like oscillators (4.13). The matrix is coded as black for connections and white otherwise. (**b**,**c**), Distance between (**b**) the rate of change and (**c**) network state of seven distinct perturbed replicas (4.14) with respect an original system (4.13). The distances are measured in terms of the L_2-norm.

Does our simple approach truly reveal network connections? As explained in sec. 4.1 and sec. 4.2, our method recovers connections (i) by correctly reconstructing the gradient entries on the reference trajectory (4.8), and (ii) by quantifying the gradients variations through (4.11). These variations determine whether individual interactions with other units were active or not, cf. Fig. 4.4 and Fig. 4.5. Furthermore, our method is not restricted to cases in which one can interfere with the network dynamics, but, it may also be applied to any case where a local sampling can be recorded, because equation (4.8) is still valid when there are no external driving signals altering the network dynamics, $\forall\, i : I_i = 0$.

Yet, given that (4.8) is only valid for $K > N$, this approach may be impractical for studying real-world networks, which are often large in size N, and thereby, generating the necessary transient responses would involve a large (or even unrealistic) K number of experiments.

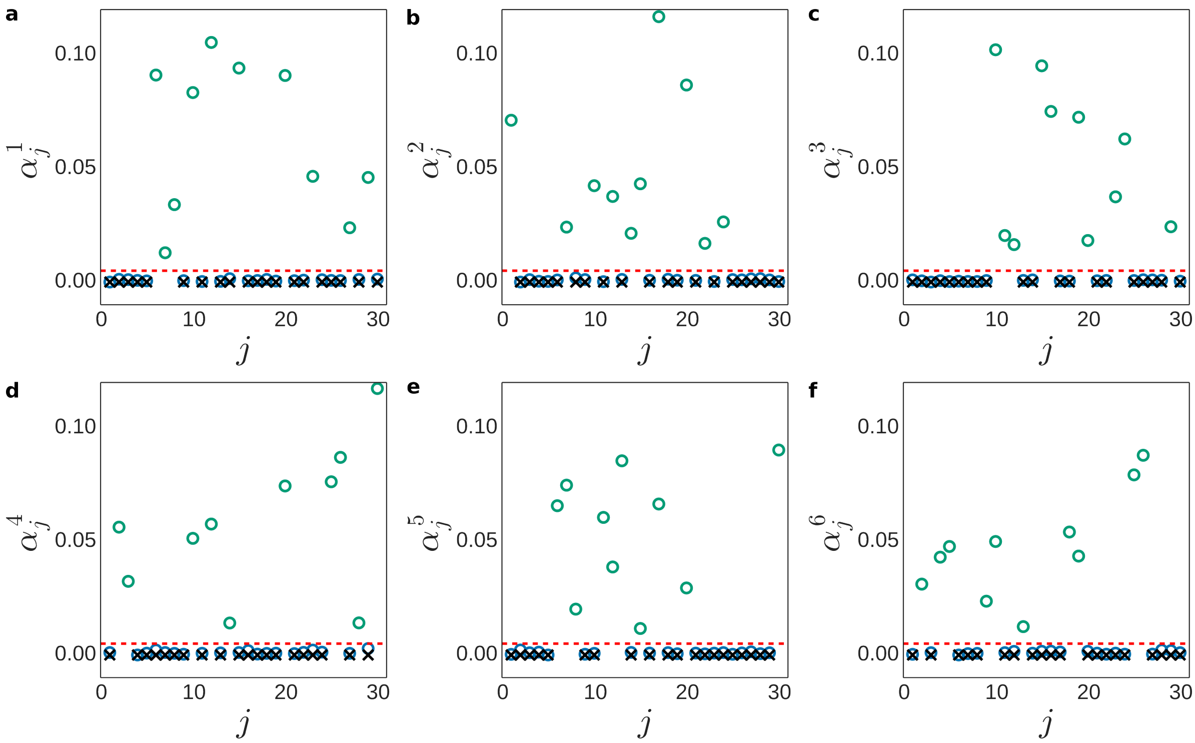


Figure 4.4. – Connectivity characterizers α^i separate existing connections from absent links. (**a**,**b**,**c**,**d**,**e**,**f**), Characterizers $\boldsymbol{\alpha}^i$ for $i \in \{1, 2, \ldots, 6\}$ of the network presented in Fig.4.3 with $K = 40$. The entry α_j^i ($\circ$) indicates how strongly unit j interacted with unit i in a given time window. Thus, bins close to zero represent absent connections, and conversely, bins far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links. The ($\times$) represent the actual absent links.

4.4. Reconstruction from few transient responses

Can we recover network connectivity even when the number of K independent transient responses is less than the number of units in the network, i.e. $K < N$? In many real-world cases, such as biological neural networks in the brain [11,107,108] or gene regulatory networks [109], a considerable number of possible incoming connections are not present. Instead, each unit i normally has a reduced number $n_i \ll N$ of incoming links.

Here, similarly to previous works [4,34,35,64,69,100], we also exploit the sparsity-inducing nature of the L_1-norm for selecting a particular solution from a family of possible solutions for $\boldsymbol{g}_i\Gamma^i$. The underlying reasoning for choosing a sparse-inducing norm is that if a network is sparse then many entries in $\boldsymbol{g}_i\Gamma^i$ are zero due to the non-existing explicit dependencies in Λ^i.

Specifically, if a system (4.6) is under-determined, the family of solutions for $\Gamma^i\boldsymbol{g}_i^\mathsf{T}$ may be parametrized as

$$\Gamma^i\boldsymbol{g}_i^\mathsf{T} = V\Sigma^\dagger U^\mathsf{T}\left(\dot{\boldsymbol{y}}_i - \mathbf{1}_{1\times K}\otimes\dot{\boldsymbol{x}}_i - \boldsymbol{I}_i\otimes\mathbf{1}_{1\times M}\right)^\mathsf{T} + W\boldsymbol{\zeta}_i, \tag{4.17}$$

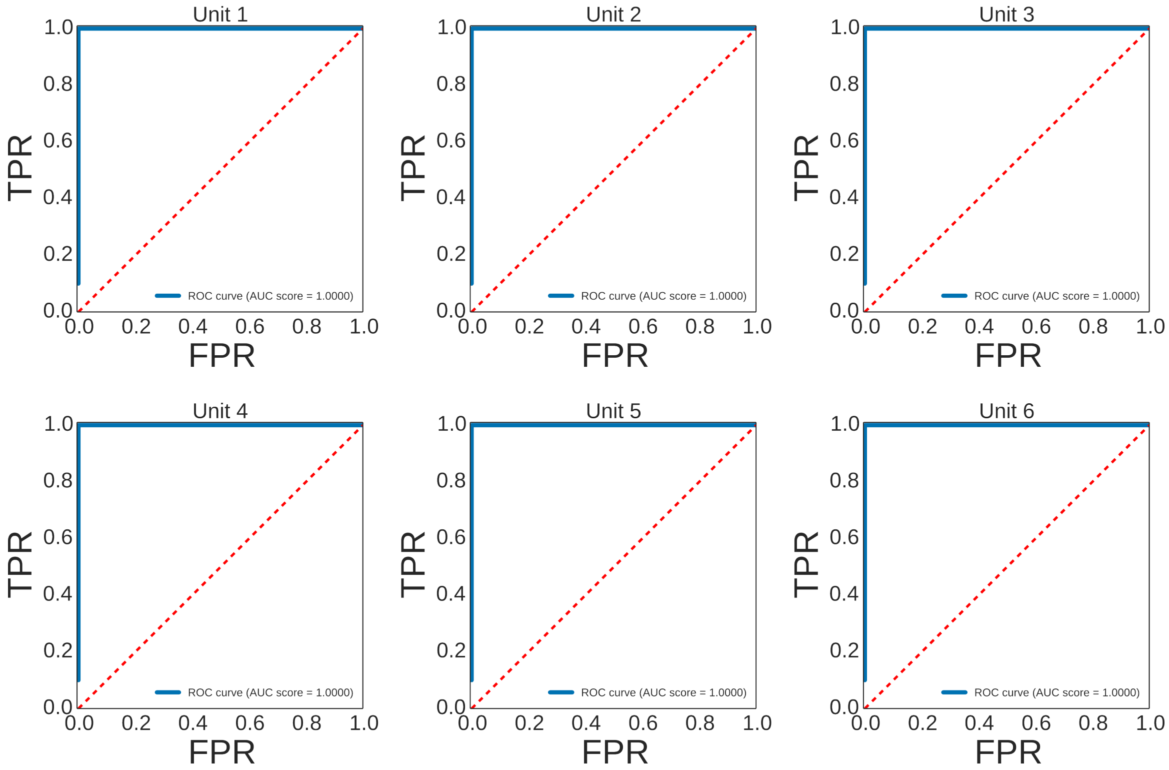


Figure 4.5. – Transient responses perfectly reveal network connections. ROC curves for characterizers shown in Fig.4.4. Results show perfect reconstructions (AUC score $=$ 1) for units $i \in \{1, 2, \ldots, 6\}$.

where $Y^\top = U\Sigma V^\top$ (singular value decomposition), $W \in \mathbb{R}^{NM\times M(N-K)}$ is a basis for the null-space[3] of $Y^\top$ and $\boldsymbol{\zeta}_i \in \mathbb{R}^{M(N-K)}$ is a vector that parametrizes the space of solutions for $\boldsymbol{g}_i\Gamma^i$, cf. Appendix B. Thus, different choices for $\boldsymbol{\zeta}_i$ yield different vectors $\boldsymbol{g}_i\Gamma^i$ that are consistent with equation (4.6).

Finally, by solving the optimization problem

$$\Gamma^i \boldsymbol{g}_i^\top := \min_{\boldsymbol{\zeta}_i} \left\| V\Sigma^\dagger U^\top \left(\dot{\boldsymbol{y}}_i - \mathbf{1}_{1\times K} \otimes \dot{\boldsymbol{x}}_i - \boldsymbol{I}_i \otimes \mathbf{1}_{1\times M}\right)^\top + W\boldsymbol{\zeta}_i \right\|_1 \tag{4.18}$$

with respect to the vector of parameters $\boldsymbol{\zeta}_i$, we reveal the entire connectivity of sparsely connected networks with fewer number of transient responses than units, cf. Fig. 4.6. Moreover, cf. Appendix B for an easy implementation to solve the optimization problem (4.18).

So, how does reconstruction from few transient responses scale with the network size? Our numerics suggest that the number $K_{0.95}$ of necessary transient responses for achieving reconstructions of AUC score larger than 0.95 scales sub-linearly with N (presumably as $K_{0.95} \sim \log(N)$), cf. Fig. 4.6. Specifically, this implies that the network connectivity can be reliably revealed even if the number of transient responses K is much smaller than the network size N.

[3] W may be easily extracted from the singular value decomposition of $Y^\top$, cf. [4, 22, 34, 35, 64, 69].

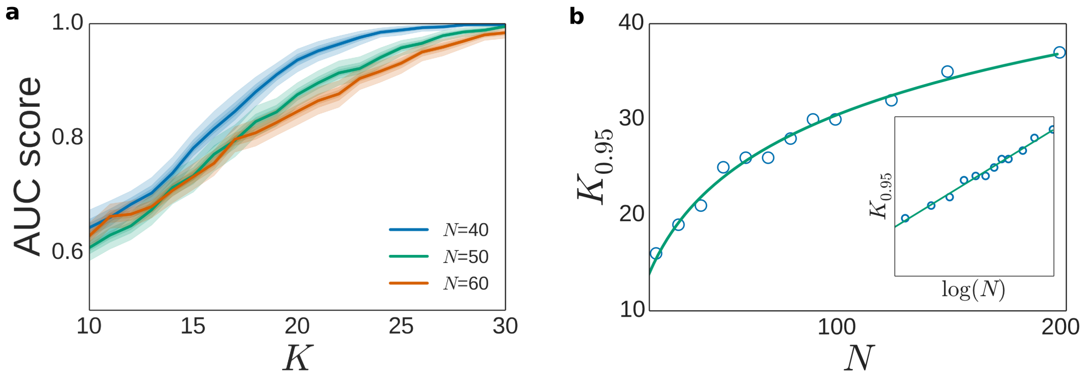


Figure 4.6. – Reconstructing sparsely connected networks with $K < N$ transient responses. a, Quality of reconstruction measured in AUC score for networks of Kuramoto-like oscillators (4.13) with $n_i = 10$ and $N \in \{40, 50, 60\}$ versus the number of transient-responses K. **b**, Minimum number of transient responses $K_{0.95}$ required for achieving AUC scores larger than 0.95 versus network size N with best logarithmic fit. The inset shows the same data with N in logarithmic scale. It is important to note that the number $\tilde{N}$ of unknowns per optimization problem is actually $\tilde{N} = MN$ with $M = 10$. However, given that the number of constraints also increases with M, results are presented with respect to N.

4.5. Locality of dynamical trajectories

In sec. 4.1, we introduced the locality condition $\|(\boldsymbol{x}, \dot{x}_i) - (\boldsymbol{y}_k, \dot{y}_{ik})\|_2$ for approximating the dynamics of replicas by equation (4.6). However, how strict is this condition? Do transient and original dynamics have to necessarily be proximal in order to reconstruct the network connectivity through (4.6)?

To answer these questions, we systematically varied initial conditions of replicas with respect to a parameter ν. Specifically, we selected the random part δ_{ik} in (4.15) from the uniform distribution

$$\delta_{ik} \in [-0.025 + \nu_{ik}, 0.025 + \nu_{ik}] , \tag{4.19}$$

where ν_{ik} is chosen accordingly to

$$\nu_{ik} = \begin{cases} -\nu & \text{if } l_{ik} \leq 0.5 \\ \nu & \text{if } l_{ik} > 0.5 \end{cases} , \tag{4.20}$$

and l_{ik} is drawn from the uniform distribution $l_{ik} \in [0, 1]$.

In previous sections, we inferred the connectivity of networks from proximal intial conditions, $\nu = 0$. Thereby, the distances between initial conditions of replicas and original systems were minimal. However, higher values of ν increase the magnitude of initial distances between trajectories, cf. Fig. 4.7a. Thus, given that external signals already drive the transient responses apart, one may expect that the approximation (4.6) loses its predictive power with increasing ν.

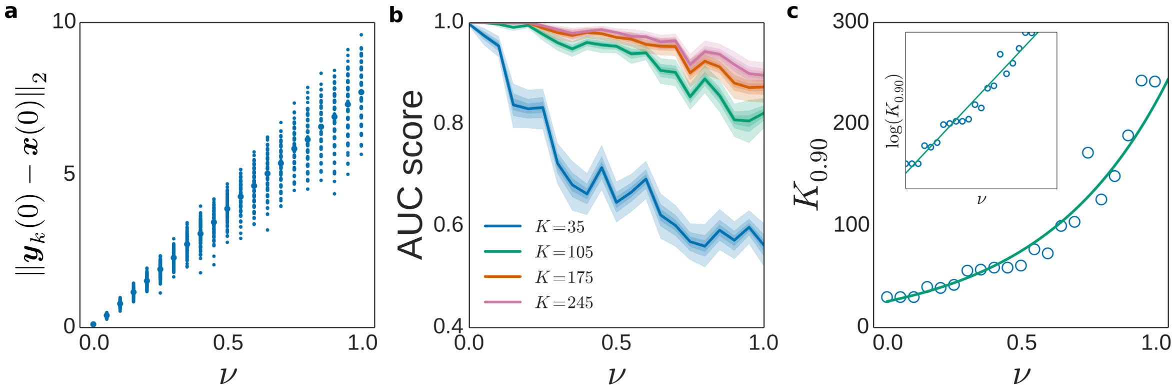


Figure 4.7. – More transient responses remedy the lack of locality. **a**, Difference between initial conditions of 300 replicas and a original network of $N = 30$ and $n = 10$ Kuramoto-like oscillators (4.13) versus ν. Each column of points represent the initial condition for different ν. Also, each column is composed of 300 points representing each individual replica. The distances are measured in terms of the L_2-norm. The parameter ν systematically increases the distance between dynamics. **b**, Quality of reconstruction for the same network versus ν for $K \in \{35, 105, 175, 245\}$. **c**, Minimum number of experiments $K_{0.90}$ required for achieving AUC scores larger than 0.90 versus ν with best exponential fit. The inset shows the same data with $K_{0.90}$ logarithmic scale.

Nonetheless, increasing the number of replicas remedies the lack of proximity between systems, cf. Fig. 4.7b. Furthermore, our numerics suggest that the necessary number of replicas $K_{0.90}$ for achieving reconstructions of AUC scores larger than 0.90 scales super-linearly with ν (presumably as $K_{0.90} \sim e^{\nu}$), cf. Fig. 4.7c. This results indicate that fair reconstructions are still feasible when the sampling lacks of locality, at the expense of having more replicas.

4.6. Discussion

In this chapter, we introduced a model-independent approach for revealing connections of complex networks from local samplings of their dynamics. While linearizing the network dynamics around a reference trajectory, we inferred the connectivity of networks by comparing nearby trajectories to the reference trajectory. Specifically, we proposed a mapping from nearby trajectories to Jacobian matrices evaluated at the reference trajectory. While focusing on the special case of having identical but perturbed replicas of a network, we demonstrated that these Jacobian matrices fully describe the network connectivity of the original system. Furthermore, we showed the potential applicability of this approach to reconstruct real-world networks by enhancing our method for reconstructing large and sparse networks with reduced number of nearby trajectories.

Previous works on reconstructing networks from local samplings have only focused on networks exhibiting stable dynamics [4, 6, 7, 63–67]. Externally driving these systems with different signals produces restrictions on the network connectivity, cf.

Ref. [22] for a comprehensive review. However, most of these approaches require a complete knowledge of network features, such as intrinsic and coupling functions. Moreover, these approaches require the network to be at or to admit to be driven to stable states [65–67].

Our framework generalizes the ideas exposed in [4, 6, 7, 63–67] to networks exhibiting more complex dynamics than simple stable states. Moreover, our method does not require a detailed pre-knowledge of local dynamics and coupling functions. And, despite that we focused here on examples in which one can perturb the network dynamics, our approach is not constrained by perturbation feasibility because it only relies on the locality of samplings. Thus, as long as the network dynamics is locally sampled, the connectivity is retrievable from network dynamics through our approach. Interestingly, exploiting this linearity of local samplings reveals the connectivity of complex networks with considerably shorter dynamical trajectories than in chapter 3.

Finally, the present study provides an explanation of how local samplings may determine the connections of networks independently of their dynamical features and collective behavior. Still, further efforts should be focused on reducing the number of nearby trajectories for achieving successful reconstructions. For instance, locally sub-sampling different regions of the dynamics space may help in determining the network structure if one focuses on the global properties induced by explicit dependencies matrices Λ^i. Anagolously to chapter 3, further efforts must also be oriented to extend our method to reconstruct networks where only a subset of units is measured. Furthermore, given the restricted length of dynamical trajectories (10 measurements), our approach rendered poorly when applied on stochastic dynamical trajectories. Thus, another promising direction for further efforts is to devise ways to complement our approach with established concepts from data filtering (e.g. Savitzky-Golay [110] and Kalman filtering [111]) to reconstruct network connectivities from noisy recordings.

Chapter 5

Network connectivity from average responses to external drivings

In this chapter, we complement the results of chapter 4 by presenting a simple geometric approach for revealing structural connections of networks displaying non-trivial collective dynamics such as periodic orbits or chaotic attractors. By applying external driving signals on identical replicas of a network, we infer the connections among units by comparing the centroids of original and driven collective behaviors in the dynamics space. Specifically, and similarly to chapter 4, we generate local samplings through small driving signals to demonstrate the direct relation between displacements of centroids of collective dynamics and the network connectivity. However, differently to chapter 4, here we focus on average responses measured on long samplings of the network dynamics. Furthermore, assuming that long samplings reveal orbits or attractors to a good extent, we compute the centroids from temporal averages of units' dynamics. Interestingly, given that we only focus on averaged quantities, the timestamps of temporal measurements is irrelevant. Thus, contrary to approaches described in chapter 3 and chapter 4, this method does not require units to be simultaneously recorded. Finally, we demonstrate the efficiency of our approach by successfully inferring connections of both sparse and non-sparse networks of coupled oscillators exhibiting non-trivial collective dynamics. This work has been done in collaboration with Mor Nitzan from the Department of Physics, Hebrew University, Jerusalem.

5.1. Mapping centroid displacements to network connectivity

As in previous chapters, we will represent networks in terms of generic dynamical systems (1.9), cf. chapter 1.

So, to understand how displacing centroids of periodic orbits or chaotic attractors may reveal the network connectivity, we will drive identical network replicas through external but small driving signals. Each driving condition will relocate the centroid of the dynamics in the dynamics space. Consequently, these displacements will be used to reconstruct the network connectivity, cf. Fig. 5.1.

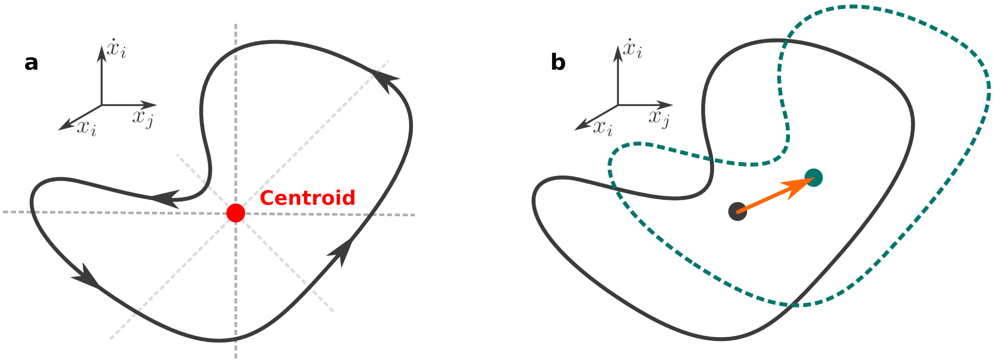


Figure 5.1. – Relocation of centroids of collective dynamics in the dynamics space through driving-response experiments. a, Non-trivial collective dynamics such as periodic orbits or chaotic attractors (black) may be characterized in the dynamics space through geometric centroids (red). **b**, Small external drivings shift centroids of stable non-trivial collective behaviors (black) to different positions (teal) in the dynamics space. The displacements (orange) depend on (i) driving signals and (ii) the network intrinsic features (e.g. local dynamics, coupling functions and connections). Thus, many centroid displacements under different driving conditions may yield information about the network connectivity.

So, how can we relate the displacement of centroids induced by external driving signals to the network connectivity? Firstly, let us define the temporal average $\langle y \rangle := \langle y(t) \rangle_t$ as

$$\langle y \rangle = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} y(t)\, dt. \tag{5.1}$$

Secondly, let us assume that we can drive M replicas of (1.9)

$$\dot{x}_{i,m} = f_i \left(\Lambda^i \boldsymbol{x}_m \right) + I_{i,m}(t) + \xi_{i,m}(t), \tag{5.2}$$

where $I_{i,m}$ for all $m \in \{1, 2, \ldots, M\}$ is the driving signal applied on the i-th unit of the m-th network replica, and $\forall\, i \in \{1, 2, \ldots, N\} : I_{i,0} \equiv 0$ represents the undriven original network. The vector $\boldsymbol{x}_m := [x_{1,m}, x_{2,m}, \ldots, x_{N,M}]^{\mathsf{T}} \in \mathbb{R}^N$ characterizes the state of the m-th replica and $\xi_{i,m}(t)$ represents external noise acting on the i-th unit of the m-th replica.

Thirdly, let us (i) describe the average dynamics of the network for each driving condition through the collection of vectors $\boldsymbol{z}_m := [z_{1,m}, z_{2,m}, \ldots, z_{N,m}]^{\mathsf{T}} \in \mathbb{R}^N$ where

$$z_{i,m} := \langle x_{i,m} \rangle\,, \tag{5.3}$$

and let us define the set of centroids $\mathcal{C}_{i,m}$ characterizing the average behavior of units in the dynamics space as

$$\mathcal{C}_{i,m} := [\boldsymbol{z}_m, \langle \dot{x}_{i,m} \rangle] \in \mathbb{R}^{N+1}. \tag{5.4}$$

Fourthly, given that in general $\langle \dot{x}_i \rangle \neq f_i\left(\Lambda^i \langle \boldsymbol{x} \rangle\right)$, thereby,

$$\langle \dot{x}_{i,m} \rangle = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \left[f_i\left(\Lambda^i \boldsymbol{x}_m\right) + I_{i,m}(t) + \xi_{i,m}(t) \right] dt. \tag{5.5}$$

So, assuming that there is a point $\boldsymbol{x}_0^*$ such that $\langle \dot{x}_{i,0} \rangle = f_i\left(\Lambda^i \boldsymbol{x}_0^*\right)$ and that driving signals $I_{i,m}(t)$ are sufficiently small such that $\left\|\mathcal{C}_{i,m} - \mathcal{C}_{i,0}\right\|_2 \ll 1$, we may calculate a first order approximation (5.5) around $\Lambda^i \boldsymbol{x}_0^*$ as

$$\langle \dot{x}_{i,m} \rangle \approx \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \left\{ f_i\left(\Lambda^i \boldsymbol{x}_0^*\right) + \boldsymbol{\nabla} f_i \Lambda^i \left[\boldsymbol{x}_m - \boldsymbol{x}_0^*\right] + I_{i,m}(t) + \xi_{i,m}(t) \right\} dt, \tag{5.6}$$

where $\boldsymbol{\nabla} f_i = \boldsymbol{\nabla} f_i\left(\Lambda^i \boldsymbol{x}_0^*\right)$. In particular, despite that in general $\boldsymbol{x}_m$ may not be close to $\boldsymbol{x}_0^*$, if signals $I_{i,m}(t)$ are sufficiently small then $\boldsymbol{x}_m$ and $\boldsymbol{x}_0^*$ are in average close to each other. Thus, time-averaging the response of units to driving signals yields

$$\langle \dot{x}_{i,m} \rangle \approx f_i\left(\Lambda^i \boldsymbol{x}_0^*\right) + \boldsymbol{\nabla} f_i \Lambda^i \left[\boldsymbol{z}_m - \boldsymbol{x}_0^*\right] + \langle I_{i,m} \rangle , \tag{5.7}$$

where we have assumed that $\xi_{i,m}(t)$ represents external white noise, thereby, $\langle \xi_{i,m} \rangle = 0$ for a long sampling. Nonetheless, given that f_i and Λ^i are unknown, one cannot compute $\boldsymbol{x}_0^*$ from $\langle \dot{x}_{i,0} \rangle$. Therefore, (5.7) cannot be directly employed for inferring the entries of $\boldsymbol{\nabla} f_i \Lambda^i$.

However, by selecting a reference centroid $\mathcal{C}_{i,r}$, we may rewrite (5.7) as

$$\langle \dot{x}_{i,m} \rangle - \langle \dot{x}_{i,r} \rangle \approx \boldsymbol{\nabla} f_i \Lambda^i \left[\boldsymbol{z}_m - \boldsymbol{z}_r\right] + \langle I_{i,m} \rangle - \langle I_{i,r} \rangle , \tag{5.8}$$

as long as $\left\|\mathcal{C}_{i,m} - \mathcal{C}_{i,r}\right\|_2 \ll 1$. So, if the number of experiments $M \gg N$, we may solve (5.8) for $\boldsymbol{\nabla} f_i \Lambda^i$ as

$$\boldsymbol{\nabla} f_i \Lambda^i \approx \boldsymbol{y}_i Z^{\dagger}, \tag{5.9}$$

where $\boldsymbol{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,M}] \in \mathbb{R}^{1\times M}$, $y_{i,m} := \langle \dot{x}_{i,m} \rangle - \langle \dot{x}_{i,r} \rangle - \langle I_{i,m} \rangle + \langle I_{i,r} \rangle$, $Z = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_M] \in \mathbb{R}^{N\times M}$, $\boldsymbol{Z}_m := [\boldsymbol{z}_m - \boldsymbol{z}_r] \in \mathbb{R}^N$ and the † operator represents the Moore-Penrose pseudo-inverse, cf. Appendix A. Thus, equation (5.9) demonstrates the existing relation between local displacements of centroids and the network connectivity.

5.2. Reconstructing networks of oscillators

To test the framework described here, we simulated directed networks of 2-dimensional Fitz Hugh-Nagumo oscillators [78] defined as

$$\begin{aligned} \dot{x}_{i,m}^1 =& x_{i,m}^1 - \frac{1}{3}\left(x_{i,m}^1\right)^3 - x_{i,m}^2 + 0.7 + \frac{1}{n_i}\sum_{j=1}^{N} J_{ij}\left(\dot{x}_{j,m}^1 - \dot{x}_{i,m}^1\right) + I_{i,m}^1 + \xi_{i,m}^1 \\ \dot{x}_{i,m}^2 =& 0.08\left(x_{i,m}^1 + 0.7 - 0.8x_{i,m}^2\right) + I_{i,m}^2 + \xi_{i,m}^2 \end{aligned} , \tag{5.10}$$

with n_i incoming connections per node and constant driving signals $I^d_{i,m}$ drawn from a uniform distribution defined in the interval $I^d_{i,m} \in [-0.25, 0.25]$, cf. Fig. 5.2. Here, the entries of J are given by $J = R \odot A$, where $A \in \{0,1\}^{N \times N}$ is the adjacency matrix of the network and $\odot$ stands for entry-wise-matrix product. The elements R_{ij} are also randomly drawn from a uniform distribution in the interval $R_{ij} \in [0.5, 1]$. Additionally, the $\xi^d_{i,m}$ represent external white noise acting on the d-th component of unit i in the m-th replica.

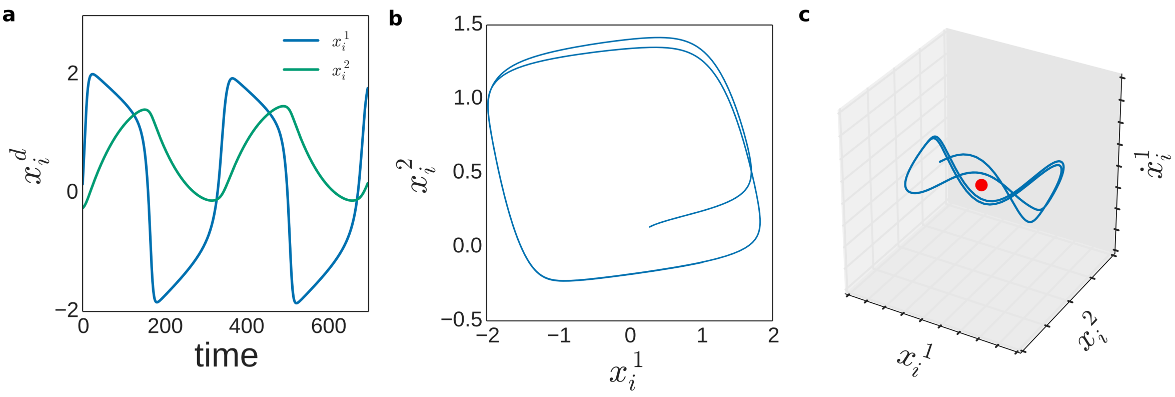


Figure 5.2. – Dynamics of Fitz Hugh-Nagumo oscillators. a, Temporal dynamics. **b**, Spatial dynamics. **c**, Representation in the dynamics space. The red dot represents the centroid of the dynamics.

As in previous chapters, we calculate time derivatives $\dot{x}_{i,k}$ at the k-th time step as

$$\dot{x}_{i,k} = \frac{x_{i,k+1} - x_{i,k-1}}{2\Delta t}. \tag{5.11}$$

Also, simulations were performed in a time interval $t \in [0, 70]$ with $\Delta t = 0.1$.

Most importantly, given that we are dealing with higher-dimensional units, here we consider all $d \in \{1, 2, \ldots, D\}$ dimensions of each unit as additional units, thus $\boldsymbol{x} \in \mathbb{R}^{DN}$. Also, we measure the network connectivity from gradients $\boldsymbol{\nabla} f_i \Lambda^i$ by means of connectivity characterizers $\boldsymbol{\alpha}^i \in \mathbb{R}^N$ defined as

$$\alpha^i_j := \frac{1}{D}\left(\sum_{d=1}^{D} \left|\frac{\partial f_i}{\partial x^d_j} \Lambda^i_{k_{jd}k_{jd}}\right|\right)^2. \tag{5.12}$$

where $j \in \{1, 2, \ldots, N\}$.

Finally, by choosing a threshold ψ_i, we reconstruct the entries of the explicit dependency matrix Λ^i as

$$\hat{\Lambda}^i_{jj} = H(\alpha^i_j - \psi^i), \tag{5.13}$$

where H is the Heaviside step function; $H(x) = 1$ for $x \geq 0$ and $H(x) = 1$ for $x < 0$.

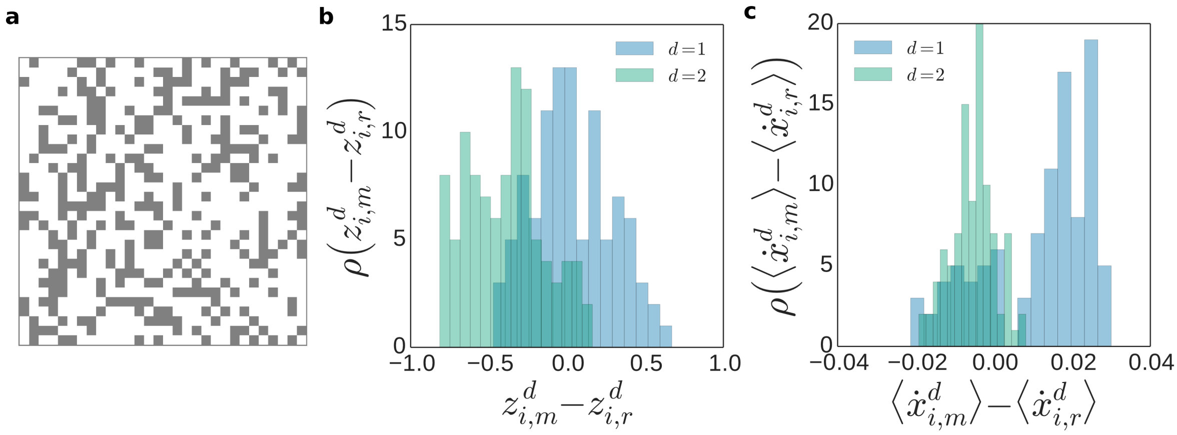


Figure 5.3. – External driving signals effectively displace centroids in the dynamics space. a, Adjacency matrix of a network of $N = 30$ and $n_i = 10$ Fitz Hugh-Nagumo oscillators (5.10). The matrix is coded as black for connections and white otherwise. (**b**,**c**), Histograms for variations between (**b**) the average network state and (**c**) the average rate of change with respect to a reference centroid.

So, how does our geometric approach perform? Reconstruction of networks of Fitz Hugh-Nagumo oscillators (5.10) indicate that centroid displacements correctly map the network connectivity. Specifically, these displacements uniquely determine the presence of individual interactions in the network, cf. Fig. 5.4. Furthermore, given that here we only exploit information coming from averaged quantities, the timestamps of temporal recordings is unimportant. Instead, recordings must capture the spatial structure of the collective dynamics in the dynamics space. Hence, this method does not require units to be simultaneously recorded.

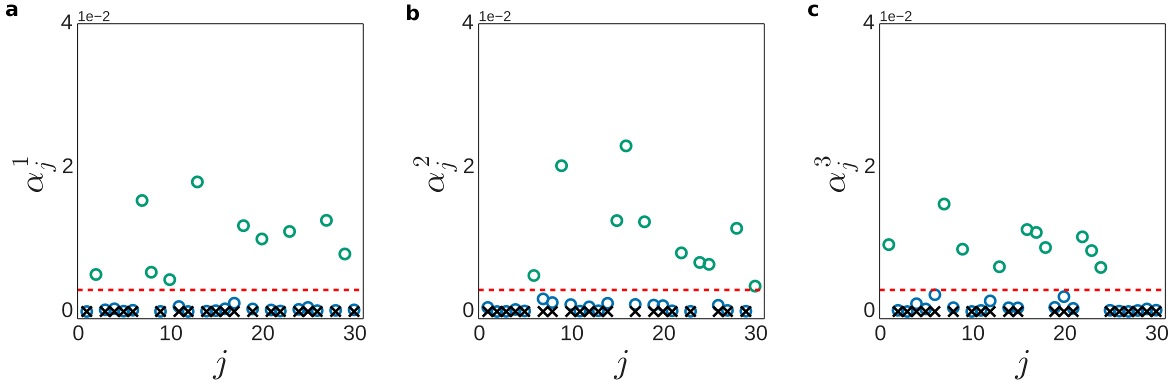


Figure 5.4. – Connectivity characterizers α^i separate existing connections from absent links. (**a**,**b**,**c**), Characterizers $\boldsymbol{\alpha}^i$ for $i \in \{1, 2, 3\}$ of the network presented in Fig.4.3 with $M = 70$. The entry α^i_j ($\circ$) indicates how strongly unit j interacted with unit i in a given time window. Thus, points close to zero represent absent connections, and conversely, points far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links. The ($\times$) represent the actual absent links.

Additionally, this method offers a simple way to handle noisy recordings. Noisy recordings induce an uncertainty on the actual network state. However, this uncertainty may be averaged out when computing centroids with increasing sampling times.

Yet, given that extensive samplings may be experimentally costly or unfeasible, here we instead studied how noisy recordings coming from finite and fixed sampling times affect the quality of reconstruction.

Reconstruction of Fitz Hugh-Nagumo oscillators reveal that the quality of reconstruction decreases with increasing noise strengths. However, this effect may be mitigated by employing greater number of centroids, cf. Fig. 5.5a. Also, our numerics suggest that the necessary number $M_{0.95}$ of displacements for achieving reconstructions of AUC scores larger than 0.95 scales as $M_{0.95} \sim e^{\lambda}$ with the noise strength λ, cf. Fig. 5.5b.

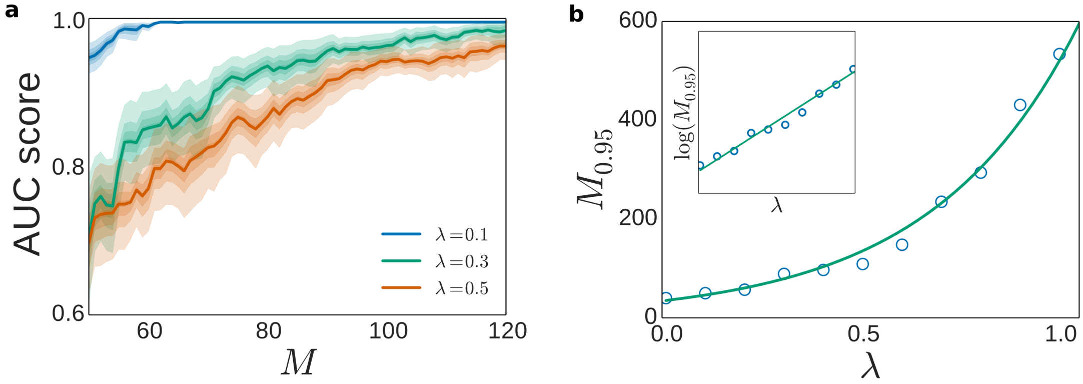


Figure 5.5. – Reconstruction from noisy measurements. (**a**,**b**) Reconstruction of networks of $N = 30$ and $n_i = 10$ Fitz Hugh-Nagumo oscillators (5.10). **a**, Area under the curve (AUC) score versus M. Reconstruction improves with M. This indicates that increasing the number of displacements mitigate the errors induced by noisy measurements. **b**, Minimum number of displacements $M_{0.95}$ required for achieving AUC scores larger than 0.95 versus noise strength λ with the best exponential fit. The inset shows the same data with $M_{0.95}$ in logarithmic scale.

Reconstructing networks of fixed size but with different number of incoming connections per node revealed that reconstruction scales with the number of incoming connections, cf. Fig. 5.6b. Specifically, we found that the number of displacements $M_{0.95}$ for achieving successful reconstructions increases sub-linearly (presumably as $M_{0.95} \sim \log(n_i)$) with the number of incoming connections. This indicate that reconstruction is feasible for denser networks with relatively few extra displacements.

5.3. Reconstruction from few centroid displacements

Given that (5.9) is only valid for $M > N$ and that one has to sample the dynamics well enough such that centroids can be accurately estimated, this method may be impractical to study real-world networks, which are usually large in size N, hence,

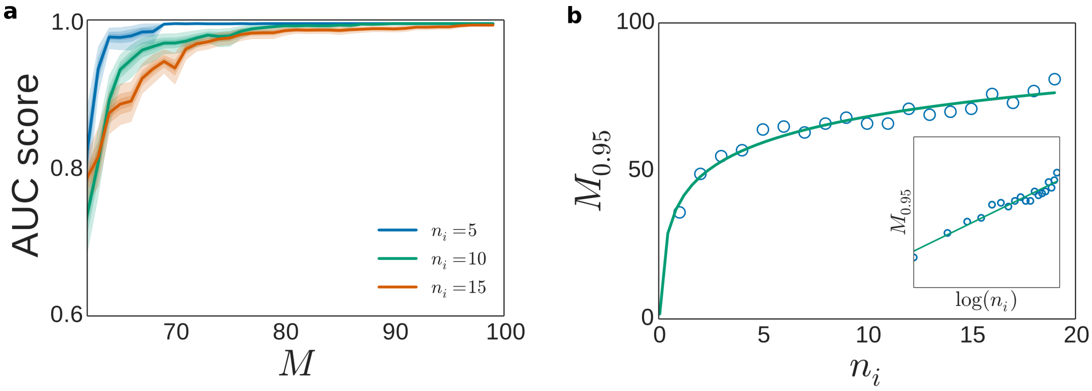


Figure 5.6. – Reconstruction scales with the number of incoming connections. a, Quality of reconstruction measured in AUC score for networks of Fitz Hugh-Nagumo oscillators (5.10) with $n_i = 10$ and $N \in \{5, 10, 15\}$ versus the number of centroid displacements M. **b**, Minimum number of displacements $M_{0.95}$ required for achieving AUC score larger than 0.95 versus number of incoming connections n_i with best logarithmic fit. The inset shows the same data with n_i in logarithmic scale. All networks were of size $N = 30$.

producing the necessary number M of displacements may be unrealistic. Here, similarly to chapter 4, we minimize the L_1-norm for selecting a particular solution from a family of possible solutions for $\boldsymbol{\nabla} f_i \Lambda^i$. Thus, if a system (5.8) is under-determined,

$$\Lambda^i \boldsymbol{\nabla} f_i^\mathsf{T} = V \Sigma^\dagger U^\mathsf{T} \boldsymbol{y}_i^\mathsf{T} + W \boldsymbol{\zeta}_i, \tag{5.14}$$

where $Z^\mathsf{T} = U \Sigma V^\mathsf{T}$, $W \in \mathbb{R}^{N \times (N-M)}$ is a basis for the null-space of Z^T and $\boldsymbol{\zeta}_i \in \mathbb{R}^{(N-M)}$ is a vector that parametrizes the space of solutions for $\boldsymbol{\nabla} f_i \Lambda^i$, cf. Appendix B.

Solving the optimization problem

$$\Lambda^i \boldsymbol{\nabla} f_i^\mathsf{T} := \min_{\boldsymbol{\zeta}_i} \left\| V \Sigma^\dagger U^\mathsf{T} \boldsymbol{y}_i^\mathsf{T} + W \boldsymbol{\zeta}_i \right\|_1 \tag{5.15}$$

with respect to $\boldsymbol{\zeta}_i$ entirely reveals the connectivity of sparse networks with few centroid displacements, cf. Fig. 4.6a.

Our numerics also suggest that the number $M_{0.95}$ of necessary displacements for achieving reconstructions of AUC score larger than 0.95 scales sub-linearly with N (presumably as $K_{0.95} \sim \log(N)$), cf. Fig. 4.6b. This indicates that the network connectivity can be accurately revealed with much less centroid displacements than the number of units of the network.

5.4. Discarding variations on time derivatives

In sec. 5.1, we showed that centroid displacements may reveal network connections as long as $\left\|\mathcal{C}_{i,m} - \mathcal{C}_{i,r}\right\|_2 \ll 1$. Yet, if driving signals are sufficiently small, the variations

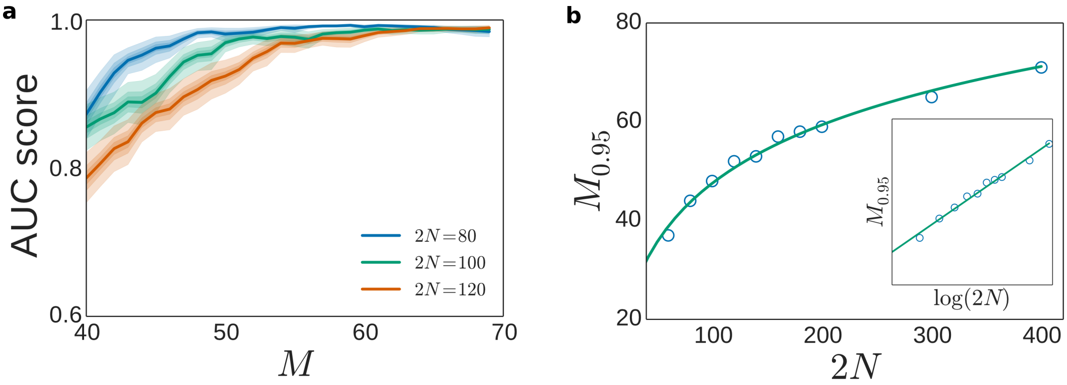


Figure 5.7. – **Reconstructing sparsely connected networks with $M < N$ centroid displacements.** **a**, Quality of reconstruction measured in AUC score for networks of Fitz Hugh-Nagumo oscillators (5.10) with $n_i = 10$ and $N \in \{40, 50, 60\}$ versus the number of centroid displacements M. **b**, Minimum number of displacements $M_{0.95}$ required for achieving AUC scores larger than 0.95 versus network size N with best logarithmic fit. The inset shows the same data with N in logarithmic scale.

$\langle \dot{x}_{i,m} \rangle - \langle \dot{x}_{i,r} \rangle$ may appear small when compared to $[\boldsymbol{z}_m - \boldsymbol{z}_r]$, cf. Fig. 5.3. Thus, can one discard the variations in the time derivatives in equation (5.8) and still achieve successful reconstructions? To answer this question, we reconstructed networks of Fitz Hugh-Nagumo oscillators (5.10) considering and discarding time derivatives' variations, $\langle \dot{x}_{i,m} \rangle - \langle \dot{x}_{i,r} \rangle \neq 0$ and $\langle \dot{x}_{i,m} \rangle - \langle \dot{x}_{i,r} \rangle := 0$, respectively, for several $\Delta t \in [0.1, 3.0]$.

Increasing Δt induces systematic errors in the estimation of time derivatives $\dot{x}_i$. Specifically, greater Δt enlarge the temporal distance between consecutive network states. Thus, one would naively expect that reconstruction results worsen with increasing Δt. Surprisingly, sampling rates Δt have little effect on reconstruction results, cf. Fig. 5.8a. This means that reconstruction is still feasible with coarser sampling rates.

Also, reconstructions revealed that results considering and discarding time derivatives' variations do not dramatically differ from each other, cf. Fig. 5.8a. Thus, to further study this, we measured the relative distance between reconstructions results as

$$\gamma := 1 - \frac{\mathrm{AUC}_d}{\mathrm{AUC}_c}, \tag{5.16}$$

where AUC_c and AUC_d represent the AUC score of considering and discarding time derivatives' variations, respectively. Thereby, $\gamma > 0$ if $\mathrm{AUC}_c > \mathrm{AUC}_d$ and $\gamma < 0$ if $\mathrm{AUC}_c < \mathrm{AUC}_d$.

Despite that considering time derivatives' variations often yield more accurate results, relative distances between reconstructions are rather small, cf. Fig. 5.8b. Thus, confirming our initial guess that small driving signals induce only small variations $\langle \dot{x}_{i,m} \rangle - \langle \dot{x}_{i,r} \rangle$. Interestingly, this additional approximation permits to sample the network dynamics without caring for an appropriate Δt to accurately compute $\dot{x}_i$.

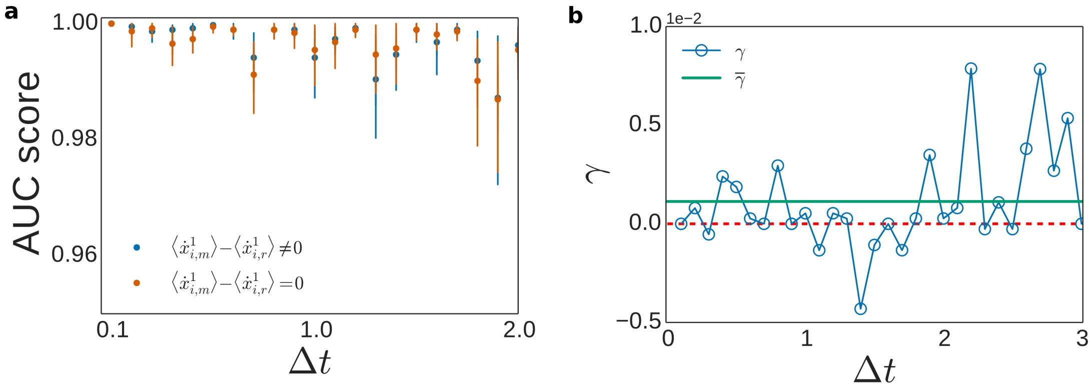


Figure 5.8. – Sampling rates and discarding the variations in time derivatives have little effect. Reconstruction of networks of Fitz Hugh-Nagumo oscillators (5.10) with $n_i = 10$, $N = 30$ and $M = 70$. **a**, Quality of reconstruction measured in AUC score with error bars versus sampling rate Δt. **b**, Relative difference γ versus sampling rate Δt with average relative difference $\bar{\gamma}$.

5.5. Discussion

Here, we proposed a geometric approach for revealing connections of complex networks from average responses to external drivings. By computing centroids of collective dynamics in the dynamics space, we reconstructed the network connectivity from displacements induced on such centroids by the external driving signals. Specifically, we introduced a linear mapping from centroid displacements to network connections. Additionally, based on that driving signals are sufficiently small, we further approximated this mapping by discarding the estimation of time derivatives. Interestingly, this provides looser experimental conditions for sampling the network dynamics by not requiring a temporal high-resolution sampling. Instead, this method requires an extensive spatial sampling of the network dynamics in the dynamics space such that centroids can be accurately calculated. Furthermore, we also demonstrated the possible applicability of this method for inferring the connectivity of real-world networks by extending our approach to infer the connections of sparse networks from few centroid displacements.

In spirit, the strategy employed in this chapter is similar to that of chapter 4. Applying small driving signals yields a local sampling of the average dynamics of networks exhibiting non-trivial collective behavior. However, we consider that the strongest feature of this approach is how it handles noisy recordings. Given that we focus on average responses to small drivings, one may average out noisy signals acting on the dynamics by either (i) sampling over longer periods of time or (ii) by employing more centroid displacements. Thus, this approach enhances the results already shown in chapter 4 by focusing in a different set of variables, the centroid of the network dynamics.

However, on the downside, this approach is constrained to perturbation feasibility. Specifically, it requires (i) to interfere with the network dynamics, and (ii) to know how strongly one effectively drives the network. Moreover, it also requires an extensive sampling of the network dynamics for computing a good estimate of centroids. Thus, all these issues are limiting factors that one should account for when applying this approach in experimental setups.

Finally and similarly to chapter 3 and chapter 4, further efforts should focus on reconstructing networks from only a subset of its units.

Chapter 6

Network connectivity from indirect measures of network dynamics

In this chapter, we extend our ongoing theory for reconstructing networks from their dynamics by demonstrating that network connections may also be revealed from indirect measures of network dynamics. Representing the network dynamics in dynamical observables (different variables than the original variables where interactions occur) may provide better dynamical features for inferring connections. To explain this concept, we introduce a model-independent approach for inferring connections of networks displaying *collectively-locked dynamics,* such as networks of synchronized and phase-locked oscillators. Specifically, exploiting their collectively-locked feature, (i) we select a indirect measure that also evolves in time with the network, and (ii) we represent the network dynamics with respect to this measure. Particularly, here we employ the *average over all units* as our indirect measure and represent the network dynamics in terms of deviations from this average. This transformation reduces the dynamics of collectively-locked networks to simple relaxations to steady states. Thus, by applying external driving signals to pull networks out of the steady states, we generate local samplings of the network dynamics and extract few network states from the relaxation dynamics. Employing these extracts, we reconstruct the network interactions through linearized models centered around the steady states. Finally, to demonstrate the efficiency of our approach, we infer the network connectivity of complex networks displaying collectively-locked dynamics purely from recorded dynamics.

6.1. Deviations from the average network activity as an example of indirect measures

Denoting the difference between units' activity as $\Delta_{ij}(t) := x_j(t) - x_i(t)$, we define a *collectively-locked dynamics* on a network as

$$\forall\, i, j \in \{1, 2, \dots, N\} \,:\, \dot{\Delta}_{ij} = 0. \tag{6.1}$$

This implies that the difference in activity $\Delta_{ij}(t)$ for any pair of units (i, j) is constant in time. Thus, all units evolve in time with a common collective rate of change $\dot{x}_i = \dot{x}_j = \omega$, cf. Fig. 6.1a. This phenomenon is known in coupled oscillators' literature as *phase-locked dynamics* [64, 89]. However, given that in this chapter we also work with synchronized higher-dimensional oscillators, we will refer to it as collectively-locked dynamics, including synchronization as a particular case where $\forall\, i, j \in \{1, 2, \dots, N\} \,:\, \Delta_{ij} = 0$.

In particular, the activity of networks exhibiting collectively-locked dynamics may be reduced to quantities that evolve together in time with the network. Particularly, the average network activity $\overline{x}(t) \in \mathbb{R}$ defined as

$$\overline{x}(t) := \frac{1}{N} \sum_{i=1}^{N} x_i(t), \tag{6.2}$$

is a quantity that may accurately describe the current state of the network because it jointly evolves in time with the network activity, cf. Fig. 6.1b.

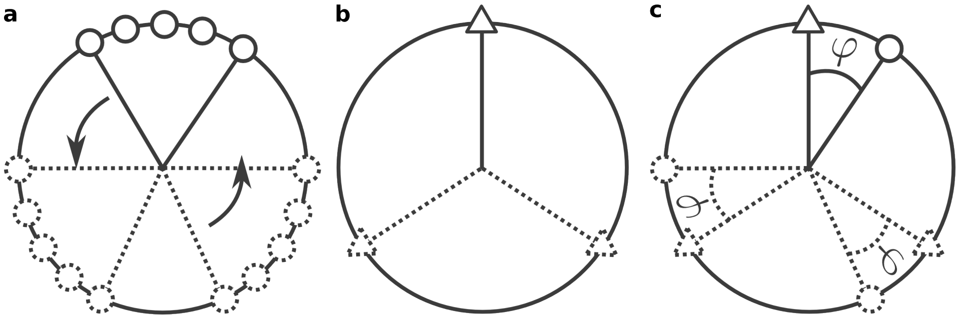


Figure 6.1. – Average network activity comprises the time evolution of collectively-locked networks. a, Units ($\circ$) in networks displaying collectively-locked dynamics move together with the same rate of change, thereby the difference between their states remain constant in time. **b**, This collective behavior may be characterized by the average network activity $\overline{x}(t)$ ($\triangle$), which jointly evolves in time with the network. **c**, The collectively-locked feature may, thereby, be represented on units in terms of deviations φ from the average network activity.

Furthermore, using this average network activity (6.2), one may transform this collectively-locked dynamics in a steady state by focusing only in deviations $\varphi_i(t) := x_i(t) - \overline{x}(t)$ from the average activity. Specifically, given that in the collectively-locked dynamics the difference between a unit's and the average network activity remains constant in time, cf. Fig. 6.1c, any perturbation that relaxes back to such collectively-locked dynamics may be seen as a relaxation to a steady state in $\varphi_i(t)$, where $\forall\, i \;:\; \lim_{t\to\infty} |\varphi_i(t) - \varphi_i^*| = 0$ and $\boldsymbol{\varphi}^* := [\varphi_1^*, \varphi_2^*, \ldots, \varphi_N^*]^\mathsf{T} \in \mathbb{R}^N$ represents the steady state of the transformed dynamics, cf. Fig. 6.2.

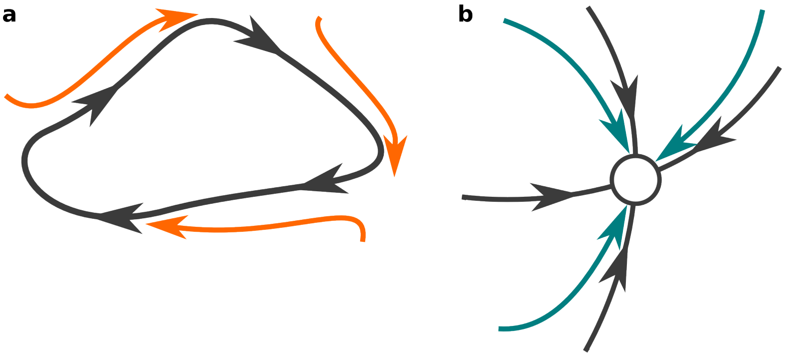


Figure 6.2. – Collectively-locked dynamics are equivalent to steady states. **a**, Different perturbations (orange) relaxing back to a collectively-locked dynamics converge at different locations on the dynamics. **b**, However, if one focuses on deviations from the network average activity, these perturbations (teal) behave as relaxations to a common steady state. This provides simpler conditions for reconstructing networks from local samplings, as shown in sec.6.2.

So, we represent the dynamics of collectively-locked dynamics in terms of deviations from the average network activity as

$$\dot{\varphi}_i = h_i(\Lambda^i \boldsymbol{\varphi}), \tag{6.3}$$

where $\dot{\varphi}_i$ represents the rate of change of the deviations from the average activity for unit i, $h_i : \mathbb{R}^N \to \mathbb{R}$ is a function that determines the dynamics of the deviations of i and $\boldsymbol{\varphi} := [\varphi_1, \varphi_2, \ldots, \varphi_N]^\mathsf{T} \in \mathbb{R}^N$ is a vector containing the deviations of every unit in the network.

6.2. Mapping deviations from the average network activity to network connectivity

To understand how deviations from the average activity reveal the connectivity of collectively-locked networks, we will (i) perturb the network dynamics and (ii) track how the deviations from the average activity relax back to their steady states, cf. Fig. 6.2. Specifically, we will perturb the networks through external signals to generate local samplings around the steady states. Finally, we will reconstruct the network connectivity from a simplified version of equation (4.8) for local samplings around steady states.

So, what do the deviations from the average activity tell us about the connectivity? Assuming that we can apply K different perturbations on the network, we may write the dynamics for each $k \in \{1, 2, \dots, K\}$ condition as

$$\dot{\varphi}_{ik} = h_i(\Lambda^i \boldsymbol{\varphi}_k), \tag{6.4}$$

where $\dot{\varphi}_{ik}$ and $\boldsymbol{\varphi}_k := [\varphi_{1k}, \varphi_{2k}, \dots, \varphi_{kN}]^{\mathsf{T}} \in \mathbb{R}^N$ are the rate of change of i and vector of deviations following the k-th perturbation.

If perturbations are sufficiently small such as not to kick the network out of the basin of attraction of the steady state, we calculate a first order approximation (6.4) around $\Lambda^i \boldsymbol{\varphi}^*$ as

$$\dot{\varphi}_{ik} = \boldsymbol{\nabla} h_i \Lambda^i \left[\boldsymbol{\varphi}_k - \boldsymbol{\varphi}^*\right], \tag{6.5}$$

where $\boldsymbol{\nabla} h_i := \left[\frac{\partial h_i}{\partial \varphi_1}, \frac{\partial h_i}{\partial \varphi_2}, \dots, \frac{\partial h_i}{\partial \varphi_N}\right] \in \mathbb{R}^N$ and $\boldsymbol{\varphi}^*$ represents the steady state of the transformed system.

Here, the entries of $\boldsymbol{\nabla} h_i \Lambda^i = \left[\frac{\partial h_i}{\partial \varphi_1}\Lambda^i_{11}, \frac{\partial h_i}{\partial \varphi_2}\Lambda^i_{22}, \dots, \frac{\partial h_i}{\partial \varphi_N}\Lambda^i_{NN}\right] \in \mathbb{R}^N$ reveal network connections. Therefore, by extracting M network states from the each of the K relaxation dynamics, we may construct the overdetermined system of equations

$$\dot{\boldsymbol{\varphi}}_i = \boldsymbol{\nabla} h_i \Lambda^i Y, \tag{6.6}$$

where $\dot{\boldsymbol{\varphi}}_i := [\dot{\boldsymbol{\varphi}}_{i1}, \dot{\boldsymbol{\varphi}}_{i2}, \dots, \dot{\boldsymbol{\varphi}}_{iK}] \in \mathbb{R}^{KM}$ with $\dot{\boldsymbol{\varphi}}_{ik} := [\dot{\varphi}_{ik,1}, \dot{\varphi}_{ik,2}, \dots, \dot{\varphi}_{ik,M}] \in \mathbb{R}^M$ and $Y := [\Phi_1, \Phi_2, \dots, \Phi_K] \in \mathbb{R}^{N \times KM}$ is a vector of matrices

$$\Phi_k := \left[\boldsymbol{\varphi}_{k,1} - \boldsymbol{\varphi}^*, \boldsymbol{\varphi}_{k,2} - \boldsymbol{\varphi}^*, \dots, \boldsymbol{\varphi}_{k,M} - \boldsymbol{\varphi}^*\right] \in \mathbb{R}^{N \times M} \tag{6.7}$$

where $\boldsymbol{\varphi}_{k,m} \in \mathbb{R}^N$ represents the m-th network state of the k-th relaxation. We may solve (6.6) for $\boldsymbol{\nabla} h_i \Lambda^i$ as

$$\boldsymbol{\nabla} h_i \Lambda^i = \dot{\boldsymbol{\varphi}}_i Y^{\dagger}, \tag{6.8}$$

where the † operator represents the Moore-Penrose pseudo-inverse, cf. Appendix A. Finally, in order to assess the presence of links, we first measure the connectivity characterizers $\boldsymbol{\alpha}^i \in \mathbb{R}^N$ defined as

$$\alpha^i_j := \frac{1}{D}\left(\sum_{d=1}^{D}\left|\frac{\partial f_i}{\partial x^d_j}\Lambda^i_{k_{jd}k_{jd}}\right|\right)^2. \tag{6.9}$$

where $j \in \{1, 2, \dots, N\}$, and second, we reconstruct the entries of Λ^i as

$$\hat{\Lambda}^i_{jj} = H(\alpha^i_j - \psi^i), \tag{6.10}$$

where ψ^i is a threshold and H is the Heaviside step function; $H(x) = 1$ for $x \geq 0$ and $H(x) = 1$ for $x < 0$.

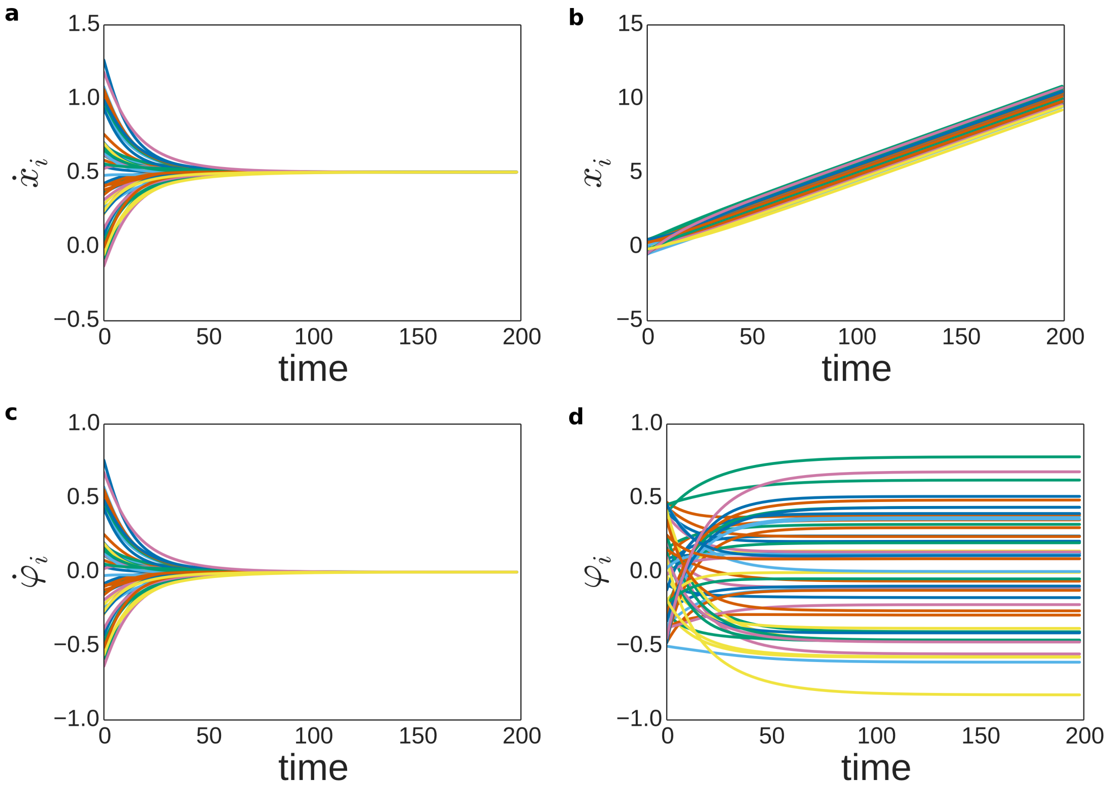


Figure 6.3. – Constant collective frequencies on original variables are equivalent to steady states in transformed variables. Simulation of a network of $N = 50$ and $n_i = 25$ Kuramoto oscillators (6.11). (**a**,**b**), Time derivatives $\dot{x}_i$ and oscillators' states x_i for all $i \in \{1, 2, \ldots, N\}$ in time. The dynamics of all oscillators evolve towards a common collective frequency in time. (**c**,**d**), Time derivatives $\dot{\varphi}_i$ and deviations from the average network activity φ_i for all $i \in \{1, 2, \ldots, N\}$ in time. When observed on the transformed dynamics, the oscillators look like approaching to a steady state.

6.3. Reconstructing networks of collectively-locked oscillators

To test the performance of this approach, we simulated directed networks of phase-locked Kuramoto oscillators [86, 89]

$$\dot{x}_{i,k} = \omega_i + \frac{1}{n_i} \sum_{j=1}^{N} J_{ij} \sin\left(x_{j,k} - x_{i,k}\right), \tag{6.11}$$

where ω_i is the natural frequency of unit i, n_i is the number of incoming connections per node and the entries J_{ij} of $J \in \mathbb{R}^{N\times N}$ are given by $J = R \odot A$, where $\odot$ stands for entry-wise-matrix product and $A \in \{0, 1\}$ is an adjacency matrix. Both the ω_i and the R_{ij} are randomly drawn from uniform distributions on the respective intervals $\omega_i \in (0, 1]$ and $R_{ij} \in [0.5, 1]$.

Also, to assess the generality of this approach, we simulated directed networks of

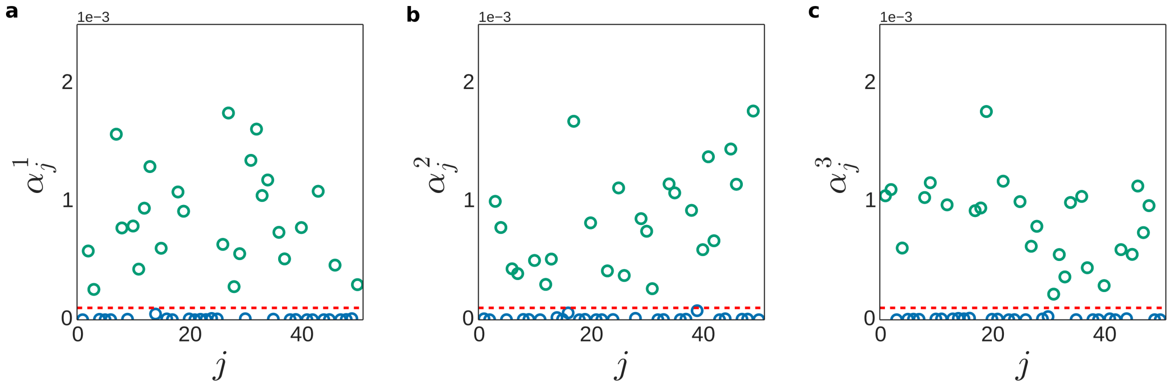


Figure 6.4. – Connectivity characterizers α^i separate existing connections from absent links. (**a**,**b**,**c**), Characterizers $\boldsymbol{\alpha}^i$ for $i \in \{1, 2, 3\}$ of the network presented in Fig.6.3 with $K = 70$. The entry α_j^i ($\circ$) indicates how strongly unit j interacted with unit i in a given time window. Thus, bins close to zero represent absent connections, and conversely, bins far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links.

synchronized Rössler oscillators [79] in periodic and chaotic regimes. Each oscillator is defined as

$$\begin{aligned}
\dot{x}_{i,k}^1 &= -x_{i,k}^2 - x_{i,k}^3 + \frac{1}{n_i}\sum_{j=1}^{N} J_{ij}\left(x_{j,k}^1 - x_{i,k}^1\right), \\
\dot{x}_{i,k}^2 &= x_i^1 + a x_{i,k}^2, \\
\dot{x}_{i,k}^3 &= b + x_{i,k}^3\left(x_{i,k}^1 - c\right), \qquad (6.12)
\end{aligned}$$

where n_i is the number of incoming connections to unit i. The entries of J were set as before. The periodic and chaotic dynamics were generated with parameters $(a, b, c) = (0.2, 1.7, 4.0)$ and $(a, b, c) = (0.2, 1.7, 13.0)$, respectively.

It is important to mention that for emulating perturbations on the stable dynamics, we employed random initial conditions for each of the $k \in \{1, 2, \ldots, K\}$ relaxation dynamics.

As in previous chapters, we calculate time derivatives $\dot{x}_{i,m}$ at the m-th time step as

$$\dot{x}_{i,m} = \frac{x_{i,m+1} - x_{i,m-1}}{2\Delta t}. \qquad (6.13)$$

All simulations were performed in a time interval $t \in [0, 20]$ with $\Delta t = 0.1$ but only the first $M = 5$ time points were selected to reconstruct the networks.

So, how does the dynamics of our toy models look like when observed on the deviations from average network activity? Simulations of Kuramoto oscillators (6.11) show that the network relax back to a non-zero collective frequency, cf. Fig. 6.3a and Fig. 6.3b. Therefore, all oscillators eventually evolve in time with a constant and common rate of change. However, when the same dynamics is transformed

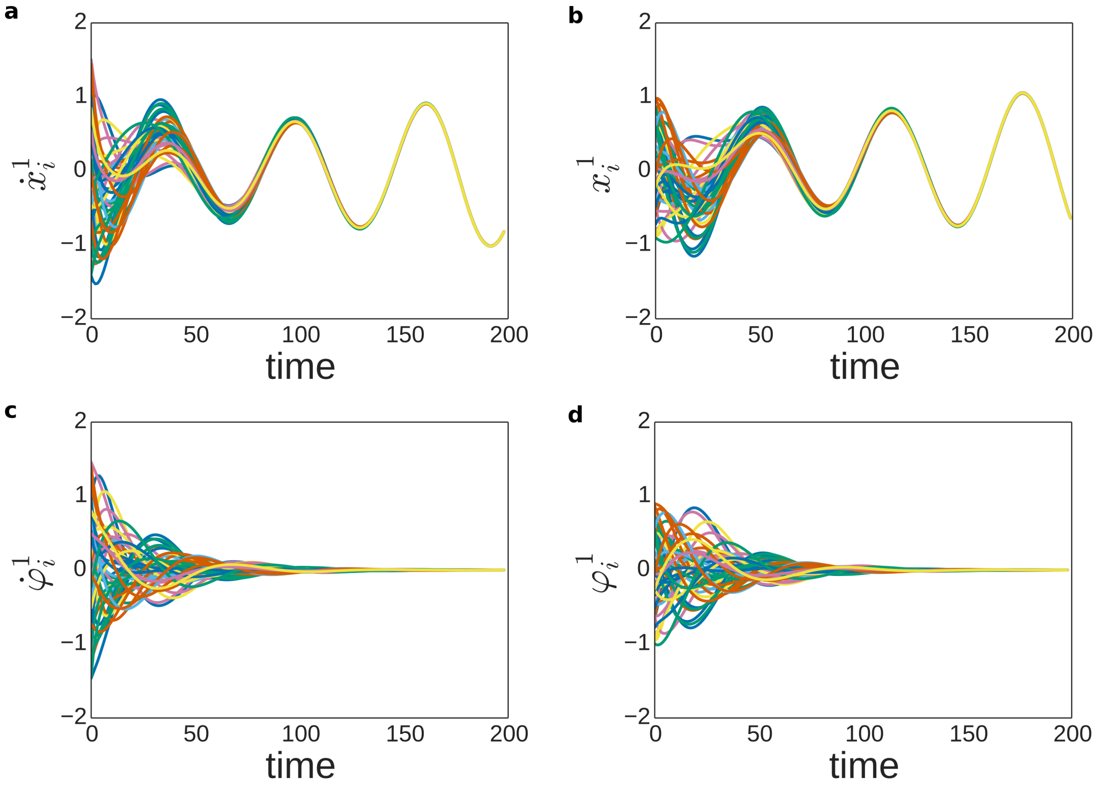


Figure 6.5. – Periodic and synchronized dynamics are reduced to steady state dynamics in transformed variables. Simulation of a network of $N = 50$ and $n_i = 25$ Rössler oscillators (6.12). (**a**,**b**), Time derivatives $\dot{x}_i^1$ and oscillators' states x_i^1 for all $i \in \{1, 2, \ldots, N\}$ of the first component in time. The dynamics of all oscillators evolve towards a common synchronized state in time. Differently to Fig.6.3, here the rate of change is no longer constant in time. (**c**,**d**), Time derivatives $\dot{\varphi}_i^1$ and deviations from the average network activity φ_i^1 for all $i \in \{1, 2, \ldots, N\}$ in time. As in Fig.6.3, when observed on the transformed dynamics, the oscillators look like approaching to a steady state.

into deviations from the average network activity, the units rather than evolving to a common collective frequency, they evolve to a steady state. So, eventually, the network will stop evolving in time in these transformed variables, cf. Fig. 6.3c and Fig. 6.3d.

Reconstructions of Kuramoto oscillators (6.11) indicate that this novel, yet intuitive, change of variables is capable to reveal network connections, cf. Fig. 6.4. Specifically, the deviations from the average network activity provide a common framework on which all relaxations to the stable dynamics can be compared, thereby, providing the ideal scenario for a local sampling, cf. Fig. 6.2b.

But, what if we were dealing with a more challenging dynamics? what if units evolve with a common, but variable, rate of change? Simulations of Rössler oscillators (6.12) show that oscillators synchronize to a common state that evolves in a periodic manner in time, cf. Fig. 6.5a and Fig. 6.5b. Thus, all oscillators eventually perform in an identical but non-trivial manner. Yet again, if we look at the deviations

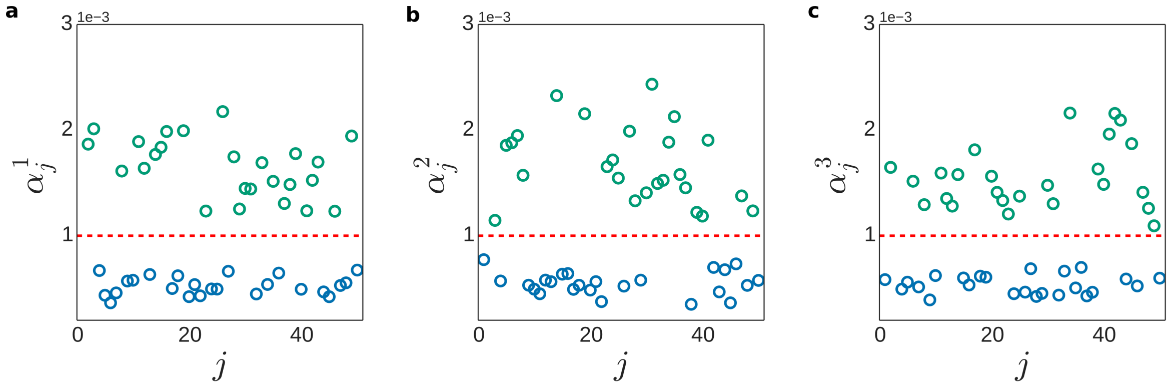


Figure 6.6. – Connectivity characterizers α^i separate existing connections from absent links. (**a**,**b**,**c**), Characterizers $\boldsymbol{\alpha}^i$ for $i \in \{1, 2, 3\}$ of the network presented in Fig.6.5 with $K = 70$. The entry α^i_j ($\circ$) indicates how strongly unit j interacted with unit i in a given time window. Thus, bins close to zero represent absent connections, and conversely, bins far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links.

from the average network activity, the oscillators seem to relax towards a steady state, cf. Fig. 6.5c and Fig. 6.5d. Furthermore, reconstructions of Rössler oscillators (6.12) also demonstrate that deviations from the average network activity reveal the network connectivity even if the average network activity is evolving in time in a nontrivial manner, cf. Fig. 6.6. Interestingly, we observed identical results for networks exhibiting a chaotic and synchronized dynamics, cf. Fig. 6.7 and Fig. 6.8. This demonstrates that despite the dynamical complexity of networks, local samplings of the deviations from the network average activity safely reveal the connectivity of networks exhibiting a collectively-locked dynamics.

6.4. Discussion

Transforming the network dynamics into a new set of variables may provide an ideal scenario for inferring connections from local samplings. Particularly, in this chapter, we demonstrated this concept on networks displaying collectively-locked dynamics. So, by representing these type of dynamics in terms of deviations from the average network activity, we transformed the collectively-locked dynamics into relaxations to steady states. Then, by kicking networks out of the steady states with random perturbations, we generated local samplings from the relaxation dynamics. Finally, using a linearized model centered at the steady states, we safely revealed the physical links of collectively-locked networks solely from their dynamics.

Previous works on inferring the connectivity of networks of coupled oscillators have been focused on three strategies. The first, inference from statistical similarity measures [105, 112, 113], recovers links among oscillators from statistical dependency measures applied on units' time series. And, although successful, most of

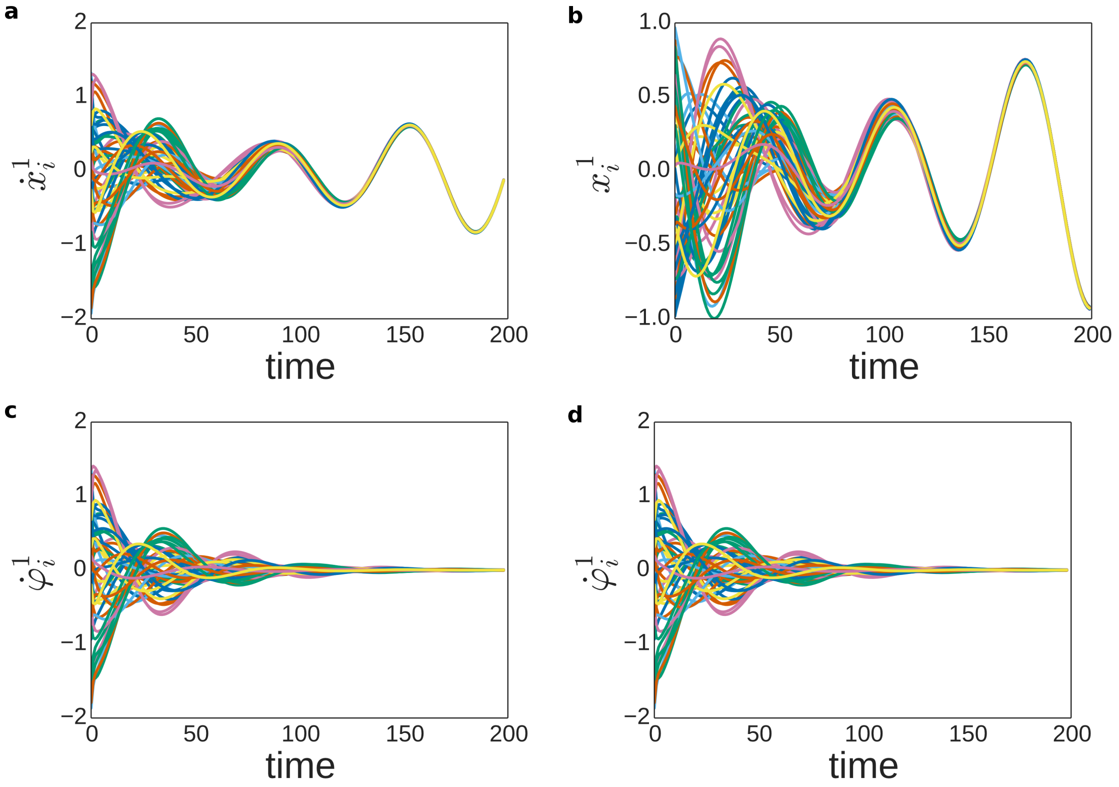


Figure 6.7. – Chaotic and synchronized dynamics are also reduced to steady state dynamics in transformed variables. Simulation of a network of $N = 50$ and $n_i = 25$ Rössler oscillators (6.12). (**a**,**b**), Time derivatives $\dot{x}_i^1$ and oscillators' states x_i^1 for all $i \in \{1, 2, \ldots, N\}$ of the first component in time. The dynamics of all oscillators evolve towards a common synchronized state in time. Differently to Fig.6.5, here the network dynamics is chaotic. (**c**,**d**), Time derivatives $\dot{\varphi}_i^1$ and deviations from the average network activity φ_i^1 for all $i \in \{1, 2, \ldots, N\}$ in time. As in Fig.6.3, when observed on the transformed dynamics, the oscillators look like approaching to a steady state.

these approaches start to fail when the size of the network and the number of incoming connections per node increases. Therefore, their applicability is reduced to relatively small networks (except for [112]). The second, inference from model fitting [34, 35, 69], recovers connections by fitting pre-imposed models for nodal dynamics to recorded time series. However, despite that model fitting approaches scale comparatively better than similarity measures alternatives for larger and denser networks, one still needs to know in detail the mechanisms generating the network dynamics (i.e. local dynamics and coupling functions). This constraints their applicability to only a few cases where a strong *a priori* knowledge is available. The third (and most related to our approach), inference from driving the network dynamics [64, 67], reconstructs links from driving steady states of networks of oscillators. Particularly, these approaches (i) are model-independent and (ii) scalable to larger networks. Nonetheless, these either demand (i) an accurate control of driving signals or (ii) a precise estimation of driving strengths.

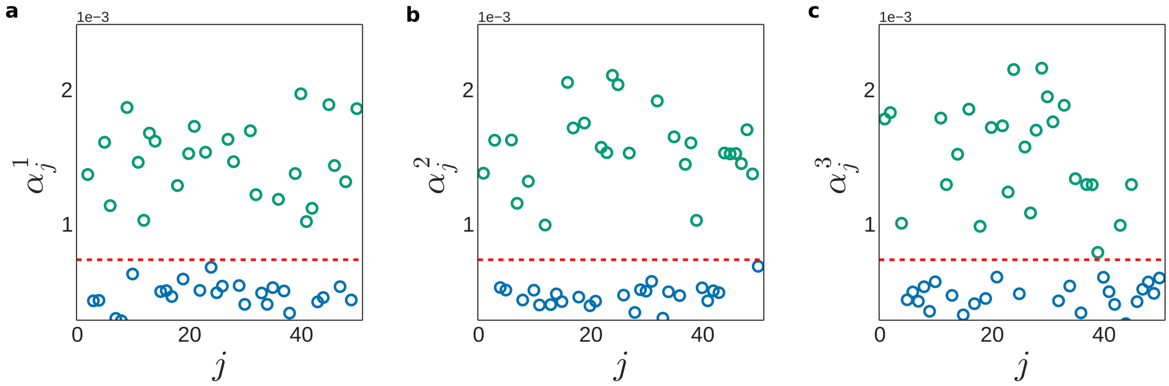


Figure 6.8. – Connectivity characterizers α^i separate existing connections from absent links. (**a**,**b**,**c**), Characterizers $\boldsymbol{\alpha}^i$ for $i \in \{1, 2, 3\}$ of the network presented in Fig.6.7 with $K = 70$. The entry α^i_j ($\circ$) indicates how strongly unit j interacted with unit i in a given time window. Thus, bins close to zero represent absent connections, and conversely, bins far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links.

Our approach enhances the ideas exposed in [64, 67]. By transforming the dynamics of collectively-locked networks into steady state dynamics, one can generate local samplings by simply kicking the networks out of steady states. Thus, neither an accurate control nor a precise estimation of driving signals is longer needed for estimating links. Furthermore, this approach can also be generalized for sparse networks under a sub-sampling of the network dynamics using L_1-norm minimizations as those used in chapter 4 and chapter 5. This ensures its applicability on real-life networks which are often large in size and sparsely connected. Although here we restricted our study to only collectively-locked networks, the present study explains how indirect measures of network dynamics may reveal physical links of complex networks. Moreover, this concept will prove useful in chapter 7 when reconstructing the structural connectivity of networks of spiking neurons from observables of their dynamics. Specifically, we will reconstruct such networks from spike trains of their units even though the network dynamics of these systems takes place on their membrane potentials.

Chapter 7

Network connectivity from discrete events in time

In this chapter, we demonstrate that also discrete events in time may reveal structural connections of complex networks. By proposing a novel representation of network dynamics in terms of discrete events, we infer the connectivity of networks from local samplings of their dynamics. Specifically, assuming linearity around a reference event, we infer the network connectivity by comparing the reference to similar events. Interestingly, this approach is not limited by perturbation feasibility nor requires networks to be at stable states. We restrict ourselves to just measure discrete events in time. Furthermore, here we employ the linearity of local samplings to demonstrate that similar events are elicited by similar causes. Finally, we demonstrate the efficiency of our approach by successfully reconstructing networks of Leaky Integrate-and-Fire (LIF) neurons from their spike trains under different dynamical conditions. This work has been done in collaboration with Dimitra Maoutsa from the Faculty of Mathematics and Computer Science of University of Goettingen, whom I co-supervised.

7.1. Event representation of network dynamics

To understand how one can represent a network dynamics in terms of discrete events in time, we will focus in this section on networks of spiking neurons. In particular, we will only focus on the qualitative behavior of neurons, and, we refer the reader to [13] for further details on neurons functioning. Also, it is worth mentioning that the ideas and results exposed in this section apply to other types of interconnected systems producing discrete events in time.

In the following, we will present two sections where we explain how to represent the dynamics of networks of spiking neurons in terms of discrete events in time. The first, an explanatory section, contains all the important concepts and ideas of our framework applied in very simple setups. The second formalizes our ideas to more complex and general setups.

7.1.1. Explanatory case

Broadly speaking, neurons are biological cells capable of generating sudden electrical discharges or *spikes* whenever their internal dynamics crosses a threshold [13]. When assembled in groups, neurons transmit spikes to other neurons through a special type of directed connections known as synapses. Consequently, this transmission of spikes affects the internal dynamics of neurons, thereby, delaying or advancing the onset of spikes in the network. Although very simple, this dynamics gives rise to many interesting and non-trivial collective behaviors [17].

For illustrative purposes, firstly, let us assume we are given an isolated neuron i with an internal dynamics described by $\dot{V}_i(t) = f_i\left(V_i(t)\right)$ with $V_i(t) \in \mathbb{R}$ representing the membrane potential of i. In the absence of interactions or driving signals the membrane potential remains at a constant resting state value $V_i(t) = c_i$. However, if the membrane potential crosses a threshold (due to interactions or drivings) at time t, $V_i(t) \geq \theta_i$, the neuron fires a spike $s_i(t)$ at t and is reset to its resting state.

Secondly, let us drive the neuron to fire using an external driving signal, $\dot{V}_i(t) = f_i\left(V_i(t)\right) + I_i$, where $I_i \in \mathbb{R}$ represents the driving signal. Particularly, let us assume we inject a constant electrical current across the neuron's membrane. The current increases the membrane potential until it reaches the threshold and then is reset to its resting state. In an idealized setup, this dynamics keeps occurring at a regular frequency as long as the constant input current remains active, cf. Fig. 7.1.

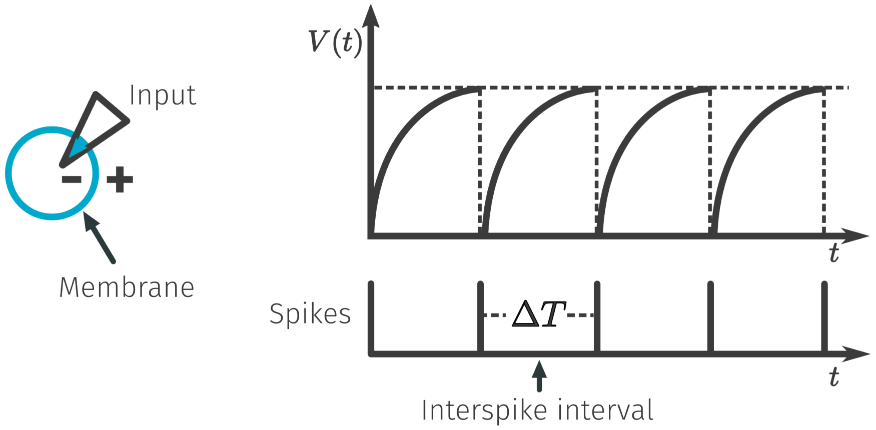


Figure 7.1. – Dynamics of an isolated neuron. Injecting a sufficiently large constant current across the neuron's membrane increases the membrane potential. The shape of this increase is determined by the internal parameters of the neuron, cf. [13] for further details. When the membrane potential reaches a threshold, the neuron fires an electrical pulse and its membrane potential is reset to a resting state value.

We may characterize this periodic dynamics in terms of the *interspike interval* $\Delta T_i \in \mathbb{R}$. This quantity measures the duration of time between two consecutive spikes

$s_i(t_{i,m})$ and $s_i(t_{i,m+1})$, thereby, it is defined as

$$\Delta T_{i,m} := t_{i,m+1} - t_{i,m} \tag{7.1}$$

where $t_{i,m}$ is the time instant at which the m-th spike was generated, so it follows that $t_{i,m+1} - t_{i,m} > 0$, cf. Fig. 7.1.

Now, let us couple a neuron j to neuron i, where $\dot{V}_j(t) = f_j\left(V_j(t)\right) + I_j$. Neuron j has also a constant input, thereby, it fires at a constant frequency, cf. Fig. 7.2. Given that j is connected to neuron i, the spikes generated by j affect the internal dynamics of i, cf. Fig. 7.2. Consequently, modifying the natural occurrence of spikes induced by I_i on neuron i, and leading to variations on ΔT_i, cf. Fig. 7.2.

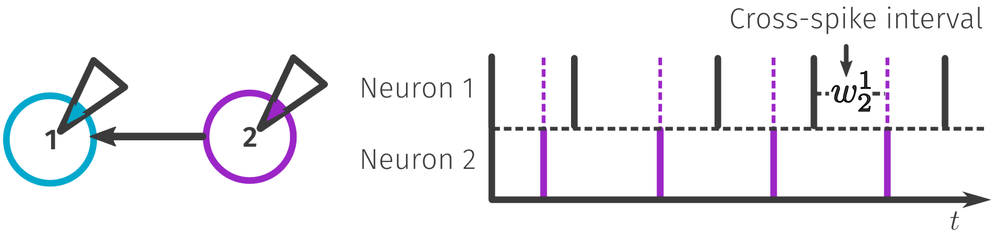


Figure 7.2. – Dynamics of coupled neurons. Given that the second neuron has no input from the first, it fires at a constant frequency due to the constant input current. However, the spikes produced by neuron 2 arrive at neuron 1 and modify its internal dynamics. This causes variations on the constant interspike interval described in Fig.7.1. Thus, the interspike interval of neuron 1 comprises information about the incoming connections of 1.

Therefore, one may naively assume that

$$\Delta T_{i,m} = h_i\left(w^i_{j,m} + \tau_{ij}\right), \tag{7.2}$$

where $h_i : \mathbb{R} \to \mathbb{R}$ is a function that determines the length of $\Delta T_{i,m}$ depending on the input spikes, $w^i_{j,m} \in \mathbb{R}$ is the m-th cross-spike interval from j to i and it is defined as

$$w^i_{j,m} := t_{j,p} - t_{i,m}, \tag{7.3}$$

with $t_{j,p}$ being the p-th spike generated by neuron j where $t_{j,p} + \tau_{ij} \in (t_{i,m}, t_{i,m+1})$, and $\tau_{ij} \in \mathbb{R}$ accounts for a time delay from neuron j to i.

As the next step, let us assume that we are given a network of size N. Additionally, we will assume (for now) that neurons $j \in Q(i)$ where $Q(i) = \{1, 2, \ldots, N\} \setminus \{i\}$, can only fire (at most) once per interspike interval. Therefore, if $t_{j,p} + \tau_{ij} \in (t_{i,m}, t_{i,m+1})$ then $t_{j,p+1} + \tau_{ij} \in (t_{i,m+1}, t_{i,m+2})$ for all $j \in Q(i)$.

So generalizing (7.2) for the aforementioned assumptions leads to

$$\Delta T_{i,m} = h_i\left(\Lambda^i\left(\boldsymbol{w}^i_m + \boldsymbol{\tau}_i\right)\right), \tag{7.4}$$

where $h_i : \mathbb{R}^N \to \mathbb{R}$ still determines the duration of interspike intervals, $\boldsymbol{w}^i_m := \left[w^i_{1,m}, w^i_{2,m}, \ldots, w^i_{N,m}\right]^\mathsf{T} \in \mathbb{R}^N$ is the vector of cross-spike intervals that occurred during the m-th interval and $\boldsymbol{\tau}_i := [\tau_{i1}, \tau_{i2}, \ldots, \tau_{iN}]^{\mathsf{T}} \in \mathbb{R}^N$ represents the vector of time delays from input neurons. Here, as usual, Λ^i represents the explicit dependency matrix and it determines which neurons are affecting the dynamics of i, cf. chapter 1. Formally, equation (7.4) assigns a specific interspike interval to a specific vector of inputs for neuron i.

Therefore, we propose to represent the dynamics of neuron i in a higher dimensional *event space* $\mathcal{E}_i \subset \mathbb{R}^{(N+1)}$, where each realization of neuron i's dynamics is given by events defined as

$$e_{i,m} := \left[\boldsymbol{w}^i_m + \boldsymbol{\tau}_i, \Delta T_{i,m}\right]^{\mathsf{T}} \in \mathbb{R}^{(N+1)}, \tag{7.5}$$

where m indicates the m-th interspike interval, cf. Fig. 7.3. In other words, we describe the dynamics of single neurons in terms of their interspike intervals and the causes of those intervals. Such representation will be of great help in sec. 7.2 for revealing synaptic connections from discrete events.

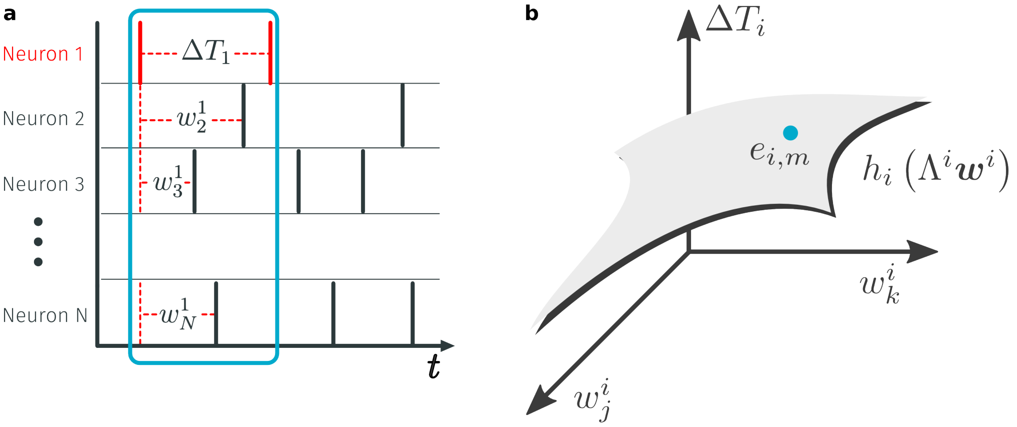


Figure 7.3. – Events and event space of a neuron. **a**, Schematics of an event for neuron 1. Given an interspike interval of neuron 1, an event is formed by such interspike interval and all the input spikes that may have lead neuron 1 to fire the last spike of the interval. **b**, General schematics of an event space for neuron i. Each event represents a point on the manifold spanned by equation (7.4). Furthermore, one may consider each event as single state of neuron i. Thus, we may use this kind of representations to describe the dynamics of neuron i.

7.1.2. General case

What if neurons $j \in Q(i)$ fire more than once per interspike interval? Defining $K_i \in \mathbb{R}$ as the maximum number of spikes generated by any neuron $j \in Q(i)$ in a sampling of $M+1$ interspike intervals, we may rewrite (7.4) as

$$\Delta T_{i,m} = h_i\left(\Lambda^i\left(W^i_m + \boldsymbol{\tau}_i \mathbf{1}_{K_i}\right)\right), \tag{7.6}$$

where $h_i : \mathbb{R}^{N\times K_i} \to \mathbb{R}$, $\mathbf{1}_{K_i}$ is a row vector of size K_i full of ones and $W_m^i \in \mathbb{R}^{N\times K_i}$ is a matrix of cross-spike intervals constructed as

$$W_m^i = \begin{bmatrix} w_{11,m}^i & w_{12,m}^i & \cdots & w_{1K_i,m}^i \\ w_{21,m} & w_{22,m} & \cdots & w_{2K_i,m}^i \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1,m}^i & w_{N2,m}^i & \cdots & w_{NK_i,m}^i \end{bmatrix}. \tag{7.7}$$

The entry $w_{jk,m}^i$ represents the k-th cross-spike interval generated by the j-th neuron during the m-th interspike interval. Thus, the columns of W_m^i indicate when neurons fired for the first time, second time, and so on, until K_i-th time during a specific interspike interval of i. Furthermore, from now on we will refer to the k-th column

$$\boldsymbol{w}_{k,m}^i := \left[w_{1k,m}^i, w_{2k,m}^i, \ldots, w_{Nk,m}^i\right]^\mathsf{T} \in \mathbb{R}^N \tag{7.8}$$

of (7.7) as the k-th firing profile of neuron i during the m-th interspike interval.

We may also extend the definition (7.5) of events in the event space for this general case as

$$e_{i,m} := \left[\text{vec}\left(W_m^i + \boldsymbol{\tau}_i \mathbf{1}_{K_i}\right), \Delta T_{i,m}\right]^\mathsf{T} \in \mathbb{R}^{(NK_i+1)}, \tag{7.9}$$

where the operator vec : $\mathbb{R}^{S\times R} \to \mathbb{R}^{SR}$ stands for vectorization of a matrix and it transforms a matrix into a column vector [114].

7.2. Mapping discrete events to network connectivity

How can we map discrete events in time of a network to its connectivity? Specifically, how can we relate the onset of apparently spontaneous and discrete network activity to structural connections? Let us assume we can generate a local sampling of $M+1$ events in the event space, cf. Fig. 7.6a. Selecting a reference event $\mathcal{E}_{i,r}$ which is closest to all other events in the sampling, thereby, $e_{i,r} := \min_{e_{i,s}} \sum_m \|e_{i,m} - e_{i,s}\|_2$ with $m, s \in \{1, 2 \ldots, M+1\}$, we may locally approximate the model for interspike intervals (7.6) in the event space around $e_{i,r}$ as

$$\Delta T_{i,m} = \Delta T_{i,r} + \text{tr}\left(\left(\frac{\partial h_i}{\partial W^i}\right)^\mathsf{T} \Lambda^i \left[W_m^i - W_r^i\right]\right), \tag{7.10}$$

where $(\partial h_i / \partial W^i) \in \mathbb{R}^{N\times K_i}$ is a matrix derivative and stands for $(\partial h_i/\partial W^i) = (\partial h_i/\partial W^i)\left(\Lambda^i\left(W_r^i + \boldsymbol{\tau}_i\right)\right)$ and tr$(\cdot)$ stands for the trace operator. Interestingly, the delay vector $\boldsymbol{\tau}_i$ vanishes in this local representation because $[W_m^i + \boldsymbol{\tau}_i - W_r^i - \boldsymbol{\tau}_i] =$

$[W^i_m - W^i_r]$. Furthermore, the matrix derivative has the following structure

$$\left(\frac{\partial h_i}{\partial W^i}\right)^{\mathsf{T}} = \begin{bmatrix} \frac{\partial h_i}{\partial W^i_{11}} & \frac{\partial h_i}{\partial W^i_{21}} & \cdots & \frac{\partial h_i}{\partial W^i_{N1}} \\ \frac{\partial h_i}{\partial W^i_{12}} & \frac{\partial h_i}{\partial W^i_{22}} & \cdots & \frac{\partial h_i}{\partial W^i_{N2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_i}{\partial W^i_{1K_i}} & \frac{\partial h_i}{\partial W^i_{2K_i}} & \cdots & \frac{\partial h_i}{\partial W^i_{NK_i}} \end{bmatrix}, \tag{7.11}$$

thereby,

$$\left[\left(\frac{\partial h_i}{\partial W^i}\right)^{\mathsf{T}} \Lambda^i \left[W^i_m - W^i_r\right]\right]_{kk} = \sum_{j=1}^{N} \frac{\partial h_i}{\partial W^i_{jk}} \Lambda^i \left[w^i_{jk,m} - w^i_{jk,r}\right], \tag{7.12}$$

where $[\cdot]_{kk}$ stands for the k-th diagonal entry. Rewriting (7.12) in terms of firing profiles (7.8) yields

$$\left[\left(\frac{\partial h_i}{\partial W^i}\right)^{\mathsf{T}} \Lambda^i \left[W^i_m - W^i_r\right]\right]_{kk} = \boldsymbol{\nabla} h_{i,k} \Lambda^i \left[\boldsymbol{w}^i_{k,m} - \boldsymbol{w}^i_{k,r}\right], \tag{7.13}$$

with $\boldsymbol{\nabla} h_{i,k} := \left[\frac{\partial h_i}{\partial W^i_{1k}}, \frac{\partial h_i}{\partial W^i_{2k}}, \ldots, \frac{\partial h_i}{\partial W^i_{Nk}}\right] \in \mathbb{R}^N$ representing the gradient of the function at the k-th firing profile. Therefore, substituting (7.13) in (7.10) yields

$$\Delta T_{i,m} = \Delta T_{i,r} + \sum_{k=1}^{K_i} \boldsymbol{\nabla} h_{i,k} \Lambda^i \left[\boldsymbol{w}^i_{k,m} - \boldsymbol{w}^i_{k,r}\right]. \tag{7.14}$$

In particular, equation (7.14) explains that, locally, we may approximate the inter-spike interval of neuron i in terms of linear combinations of the firing profiles. Yet, how can this linear model for the dynamics of neuron i in the event space help us in reconstructing its set of incoming connections? Let us rewrite (7.14) as

$$\Delta T_{i,m} = \Delta T_{i,r} + \boldsymbol{g}_i \Gamma^i \boldsymbol{y}_m, \tag{7.15}$$

where $\boldsymbol{g}_i := [\boldsymbol{\nabla} h_{i,1}, \boldsymbol{\nabla} h_{i,2}, \ldots, \boldsymbol{\nabla} h_{i,K_i}] \in \mathbb{R}^{NK_i}$ contains the gradients for each firing profile, $\Gamma^i \in \mathbb{R}^{NK_i \times NK_i}$ is a block diagonal matrix defined as

$$\Gamma^i := \begin{bmatrix} \Lambda^i & 0_{N\times N} & \ldots & 0_{N\times N} \\ 0_{N\times N} & \Lambda^i & \ldots & 0_{N\times N} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{N\times N} & 0_{N\times N} & \ldots & \Lambda^i \end{bmatrix}, \tag{7.16}$$

with $0_{N\times N}$ being a square matrix of size N full of zeros, and

$$\boldsymbol{y}_m := \left[\boldsymbol{w}^i_{1,m} - \boldsymbol{w}^i_{1,r}, \boldsymbol{w}^i_{2,m} - \boldsymbol{w}^i_{2,r}, \ldots, \boldsymbol{w}^i_{K_i,m} - \boldsymbol{w}^i_{K_i,r}\right]^{\mathsf{T}} \in \mathbb{R}^{NK_i}. \tag{7.17}$$

Accounting for all other M events in the sampling, we may construct the linear system of equations

$$\boldsymbol{d}_i = \boldsymbol{g}_i \Gamma^i Y, \tag{7.18}$$

where $\boldsymbol{d}_i := [\Delta T_{i,1} - \Delta T_{i,r}, \Delta T_{i,2} - \Delta T_{i,r}, \dots \Delta T_{i,M} - \Delta T_{i,r}] \in \mathbb{R}^M$ contains the differences with respect to the reference interval and $Y := [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_M] \in \mathbb{R}^{NK_i \times M}$. Particularly, given that

$$\boldsymbol{g}_i \Gamma^i = \left[\boldsymbol{\nabla} h_{i,1} \Lambda^i, \boldsymbol{\nabla} h_{i,2} \Lambda^i, \dots, \boldsymbol{\nabla} h_{i,K_i} \Lambda^i \right] \in \mathbb{R}^{NK_i} \tag{7.19}$$

we may employ the vector $\boldsymbol{g}_i \Gamma^i$ to infer the incoming connections of neuron i in terms of recorded events only.

Moreover, if $M > NK_i$ we may find an approximate solution for $\boldsymbol{g}_i \Gamma^i$ as

$$\boldsymbol{g}_i \Gamma^i = \boldsymbol{d}_i Y^\dagger, \tag{7.20}$$

where the operator † stands for the Moore-Penrose pseudo-inverse, cf. Appendix A.

Finally, we propose to measure the strength of connections through connectivity characterizers $\boldsymbol{\alpha}^i \in \mathbb{R}^N$ defined as

$$\alpha_j^i := \frac{\partial h_i}{\partial W_{j1}^i} \Lambda_{jj}^i, \tag{7.21}$$

where $j \in \{1, 2, \dots, N\}$. Although we recover the gradients $\boldsymbol{\nabla} h_{i.k}$ at every $k \in \{1, 2, \dots, K_i\}$, the first gradient $\boldsymbol{\nabla} h_{i.1}$ may already map the existence or absence of connections.

Therefore, by selecting a threshold ψ^i, we reconstruct the entries of the explicit dependency matrix Λ^i as

$$\hat{\Lambda}_{jj}^i = H(\alpha_j^i - \psi^i), \tag{7.22}$$

where H is the Heaviside step function; $H(x) = 1$ for $x > 0$ and $H(x) = 0$ for $x < 0$.

7.3. Reconstructing networks of spiking neurons

To test our approach, we simulated networks of pulse-coupled Leaky Integrate-and-Fire (LIF) neurons [17,34,35,40]

$$\tau \dot{V}_i = -V_i + R_i I_i + \tau \sum_{j=1}^{N} \sum_{m \in \mathbb{Z}} J_{ij} \delta \left(t - t_{j,m} - \tau_{ij} \right), \tag{7.23}$$

where δ is a delta function, $\tau = 20\,\mathrm{ms}$ is the membrane time constant, $\tau_{ij} = 1.5\,\mathrm{ms}$ is a time delay from neuron j to i, I_i is the injected current on i and the elements J_{ij} represent the synaptic connections. The values for I_i and J_{ij} are varied in the following examples depending on the desired dynamics. Additionally, we added a refractory period $\tau_{\text{ref}} = 2\,\text{ms}$ to all neurons. All simulations were performed with Neural Simulation Tool NEST [115] with a resolution $\delta t = 0.1\,\text{ms}$.

We firstly tested our method on networks of LIF neurons with only inhibitory synapses, $J_{ij} < 0$. Spikes coming from inhibitory connections decrease the membrane potential, thus, they delay the onset of the next spike. We chose this simple example because, by increasing the amount of inhibition in the system, $|J_{ij}|$, one may easily shift from a *regular dynamics* to an *irregular dynamics*, cf. Fig. 7.5a. In particular, in the regular regime, neurons fire in an almost constant manner, while in the irregular regime, they do not, cf. Fig. 7.4a and Fig. 7.7a.

So, does our simple and intuitive approach reveal synaptic connections of coupled neurons? Reconstructions of networks of purely inhibitory LIF neurons (7.23) in a regular regime show that local samplings in the event space correctly reveal the synaptic connectivity. Particularly, this simple combined strategy of representing the network dynamics in the event space followed by a local sampling in this space uniquely determines the existence of synaptic connections, cf. Fig. 7.4. Furthermore, increasing the number of events in the local sampling eases the burden of distinguishing between existing from absent connections, cf. Fig. 7.4b, Fig. 7.4c and Fig. 7.4d. Interestingly, given that we only exploit local information in the event space, the timestamps of temporal recordings are irrelevant.

But, what about more complicated dynamics? For instance, does our method recover synaptic connections from less regular dynamics? To systematically study this, let us first define the coefficient of variation for neuron i as

$$CV(i) := \frac{\sqrt{S^{-1}\sum_{s=1}^{S}\left(\Delta T_{i,s} - \mu(i)^2\right)}}{\mu(i)}, \tag{7.24}$$

where $\mu(i) = S^{-1}\sum_{s=1}^{S}\Delta T_{i,s}$ and S is the number of events in the sampling. The coefficient of variation (7.24) measures the regularity neuron i's firing. Thus, smaller values indicate more regular and larger values more irregular dynamics.

Systematic reconstructions of networks of LIF neurons (7.23) indicate (i) that more inhibition induces more irregular dynamics, and (ii) that our approach starts failing to recover synaptic connections with more irregular dynamics, cf. Fig. 7.5.

This is because more irregular dynamics sample the event space in less local manners, cf. Fig. 7.6. Specifically, an irregular dynamics samples different regions of the event space. Thus, local approximations of the form (7.10) are no longer valid unless one forces a local sampling by imposing vicinity conditions on events, cf. Fig. 7.6b. For instance, an example of a vicinity condition would be selecting only the M closest events to the reference such that one produces a local sampling. Yet, achieving the necessary number of events for recovering connections may require to observe the network for longer periods of time.

So, does enforcing a local sampling on an irregular dynamics employing longer samplings in time work at all? Reconstruction of purely inhibitory networks of LIF

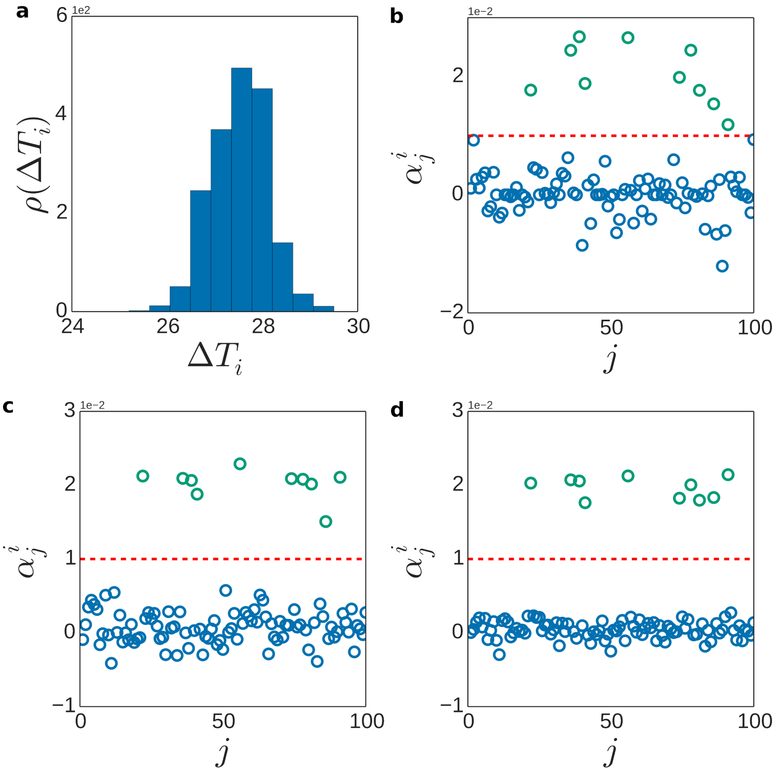


Figure 7.4. – Regular dynamics reveal synaptic connections. Reconstruction of networks of LIF neurons (7.23) of $N = 100$ with purely inhibitory connections $J_{ij} = -0.3$ mV for all $i, j \in \{1, 2, \ldots, N\}$ and a connection probability $\gamma = 0.1$ for a simulation time of 50 s. The input currents were randomly selected from the uniform distribution $I_i \in [1.4, 1.6]$ pA for all $i \in \{1, 2, \ldots, N\}$. **a**, Histogram of ΔT_i for a randomly selected neuron i. (**b**,**c**,**d**), Characterizers $\boldsymbol{\alpha}^i$ for i with $M \in \{120, 300, 600\}$, respectively. The entry α^i_j ($\circ$) indicates how strongly unit j interacted with unit i. Thus, points close to zero represent absent connections, and conversely, points far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links.

neurons (7.23) in an irregular regime show that this strategy of enforcing local samplings in the event space from longer samplings in time reveal connections, cf. Fig. 7.7b, Fig. 7.7c and Fig. 7.7d. Therefore, this confirms the importance of local samplings in the event space for revealing structural connections of spiking neurons purely from their spike trains.

Furthermore, reconstructions of networks of LIF neurons (7.23) having excitatory and inhibitory connections also demonstrate that local samplings accurately reveal synaptic connections even if there are different types of connections present in the network, cf. Fig. 7.8. Remarkably, our simple approach distinguishes between inhibitory and excitatory connections. This is because whereas input spikes coming from inhibitory connections delay the onset of the subsequent spike, those coming from excitatory connections advance it. In other words, inhibitory inputs increase the duration of interspike intervals and the excitatory inputs reduce it. Therefore, strengths $\alpha^i_j > 0$ for inhibitory and $\alpha^i_j < 0$ for excitatory inputs, cf. Fig. 7.8.

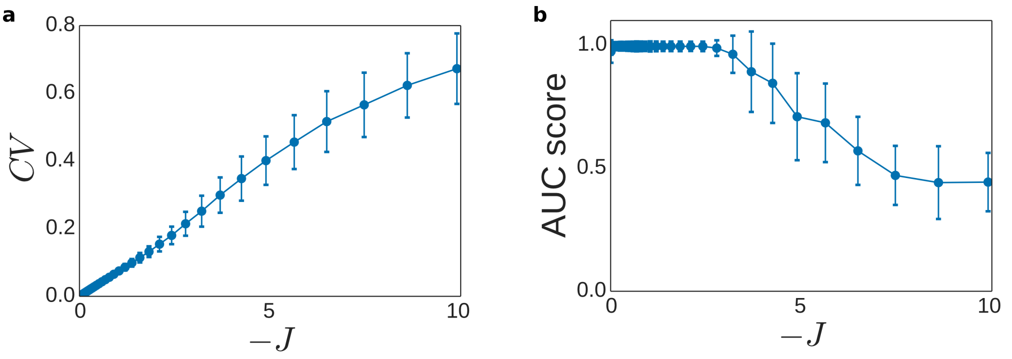


Figure 7.5. – **Does our method fail on irregular dynamics?** Reconstruction of networks of LIF neurons (7.23) of $N = 100$ with purely inhibitory connections with a connection probability $\gamma = 0.1$ for a simulation time of 50 s. **a**, Coefficient of variation (7.24) versus amount of inhibition in the system. Increasing the inhibition consistently drives the neuron to a more irregular dynamics. **b**, Quality of reconstruction measured in AUC score versus the amount of inhibition in the system. This suggests that more irregular dynamics are less likely to reveal synaptic connections through local samplings.

Reconstructing networks of fixed size with different connection probabilities γ revealed that denser connectivities have no effect on the necessary number of events for achieving successful reconstructions, cf. Fig. 7.9. This demonstrates that our approach may even recover complicated connectivity structures employing relatively few recordings.

So far, we have assumed that we know the actual delays τ_{ij} for all $i, j \in \{1, 2, \dots, N\}$. But, what if we have no accurate estimations for neurons' synaptic delays? Despite that delays vanish from our local sampling condition (7.10), they are still important for computing events (7.9) from spike trains. Inaccurate delays may induce a mismatching between interspike intervals and input spikes, thereby, causing a misclassification of events.

Interestingly, systematic reconstructions of networks of LIF neurons (7.23) show that our approach still recovers synaptic connections to a great extent in the presence of inaccurate estimations of τ_{ij}, cf. Fig. 7.10. Moreover, this lack of accuracy is alleviated by using more events when reconstructing the network, cf. Fig. 7.10.

Nonetheless, by defining the error function

$$E_2(\tau_{ij}) = \frac{\|\boldsymbol{d}_i - \boldsymbol{g}_i \Gamma^i Y\|_2}{\max \|\boldsymbol{d}_i - \boldsymbol{g}_i \Gamma^i Y\|_2}, \tag{7.25}$$

one may methodically study the effect of selecting different delays for the same data. Basically, different delays produce different events, and therefore, they produce different matrices Y, cf. (7.9) and (7.17).

So, further analysis indicates that the error function (7.25) shows a global minimum approximately at the actual delay, cf. Fig. 7.10. This implies that our local sampling

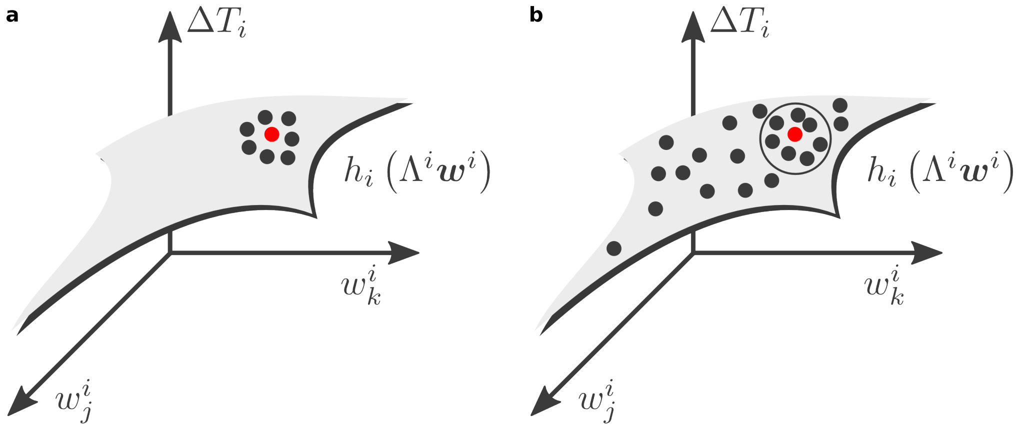

a
ΔT_i
$h_i\left(\Lambda^i \boldsymbol{w}^i\right)$
w_k^i
w_j^i
b
ΔT_i
$h_i\left(\Lambda^i \boldsymbol{w}^i\right)$
w_k^i
w_j^i

Figure 7.6. – Regular and irregular dynamics, different types of samplings in the event space. **a**, Local sampling of the event space. A regular dynamics induces a local sampling in the event space because firings occur in a quasi-periodic manner, thus, all events are qualitatively similar. **b**, Global sampling of the event space. An irregular dynamics, instead, samples different parts of the event space. Therefore, local sampling constraints will not reveal connections in this type of sampling unless one imposes vicinity constraints (e.g. closest events to the reference).

strategy may also be used to estimate the synaptic delays from spike trains if one combines it with gradient descent approaches.

Still, it is important to clarify that we are recovering connections from local samplings in the event space of units. Thus, our reconstruction results strongly depend on the selected reference event. For instance, let us consider an event where a spike coming from a pre-synaptic input arrives during the refractory period of the post-synaptic neuron. Given that the post-synaptic neuron is at the refractory period, the incoming spike does not alter the internal dynamics of the neuron. This means that concerning the post-synaptic neuron, there is no net effect coming from the pre-synaptic neuron. Therefore, this pre-synaptic neuron will appear as an absent connection if one reconstructs the set of incoming connections at that specific event.

7.4. Discussion

In this chapter, we introduced a general approach for reconstructing structural connections of networks from discrete events in time. While expressing the network dynamics in terms of discrete events, we proposed to reconstruct the structural connectivity from local samplings in event spaces. Interestingly, local samplings in these spaces generate linear conditions for network connections in terms of timestamps of discrete events. Specifically, by selecting a reference event in the event space, we developed a linear mapping for structural connections employing nearby events to a reference. Furthermore, we demonstrated that this linear mapping reliably recovers structural connections by reconstructing the synaptic connectivity of networks of LIF neurons from their spike trains alone under different dynamical conditions. Re-

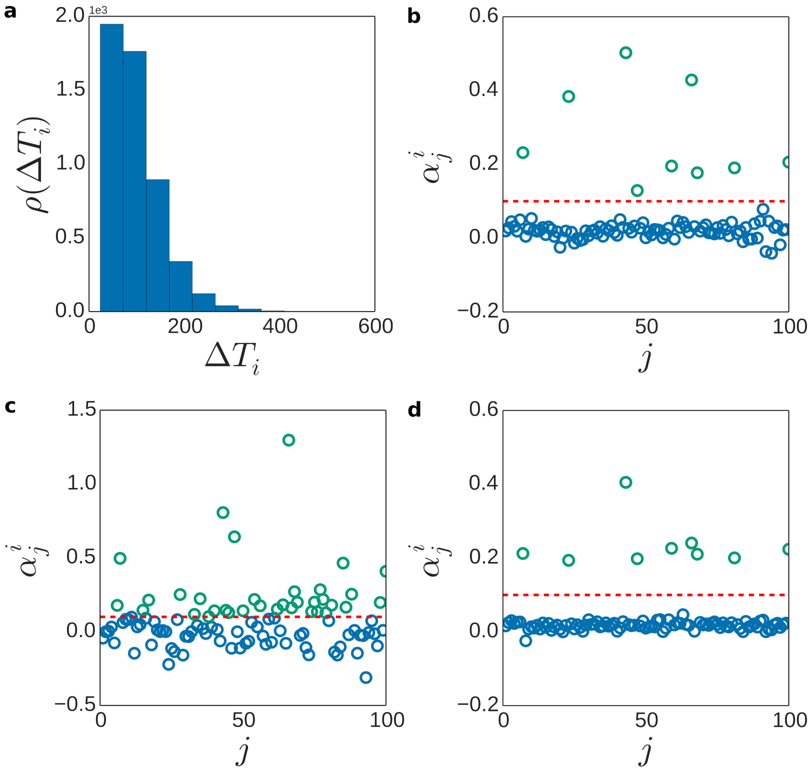


Figure 7.7. – Irregular dynamics reveal synaptic connections. Reconstruction of networks of LIF neurons (7.23) of $N = 100$ with purely inhibitory connections $J_{ij} = -8.0$ mV for all $i, j \in \{1, 2, \ldots, N\}$ and a connection probability $\gamma = 0.1$ for a simulation time of 500 s. The input currents were randomly selected from the uniform distribution $I_i \in [1.4, 1.6]$ pA for all $i \in \{1, 2, \ldots, N\}$. **a**, Histogram of ΔT_i for a randomly selected neuron i. (**b**,**c**,**d**), Characterizers $\boldsymbol{\alpha}^i$ for i with $M \in \{120, 300, 600\}$, respectively. The entry α_j^i ($\circ$) indicates how strongly unit j interacted with unit i. Thus, points close to zero represent absent connections, and conversely, points far from zero indicate existing physical links. The dashed lines illustrate optimal thresholds ψ^i for distinguishing between existing and absent links.

markably, our approach is sensitive to different types of interactions (e.g. inhibition and excitation), it is not constrained by the number of incoming connections and it employs relatively short samplings.

Particularly, previous studies on non-correlation-based methods for inferring synaptic connections from spike trains have focused on reconstruction schemes employing simple models that mimic the dynamics of networks of spiking neurons [34, 35, 40, 116]. This technique resembles that of [69], where the authors propose known functions for representing the network dynamics. Yet, in real-life networks, these specific functions and mechanism are not at hand. Moreover, imposing inaccurate mechanisms to model the network dynamics may lead to erroneous reconstructions. On the other hand, correlation-based methods (which in general are model independent) construct causal relationships from statistical dependencies between recorded spike trains [36, 47, 117]. And, although successful, this approaches not only require extensive samplings, but they only work on sparsely connected networks. This limits

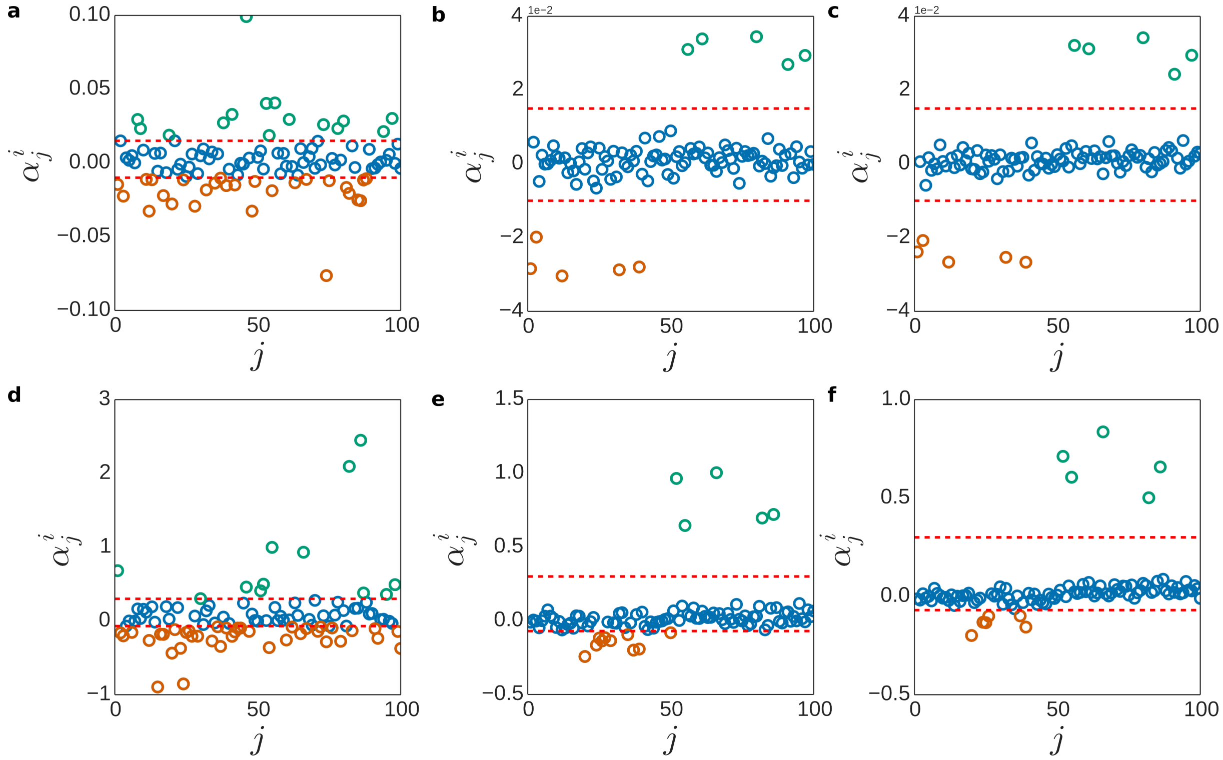


Figure 7.8. – Local samplings in the event space distinguish inhibition from excitation. (upper row), Reconstruction of networks of LIF neurons (7.23) of $N = 100$ with $N_{exc} = 50$ excitatory and, $N_{inh} = 50$ inhibitory connections in a regular regime, $J_{exc} = 0.4$ mV and $J_{inh} = -J_{exc}$, and a connection probability $\gamma = 0.1$ for a simulation time of 50 s. The input currents were randomly selected from the uniform distribution $I_i \in [1.2, 1.4]$ pA for all $i \in \{1, 2, \ldots, N\}$. (**b**,**c**,**d**), Characterizers $\boldsymbol{\alpha}^i$ for a randomly selected i with $M \in \{120, 300, 600\}$, respectively. The entry α_j^i ($\circ$) indicates how strongly unit j interacted with unit i. (lower row), Reconstruction of networks of LIF neurons (7.23) of $N = 100$ with $N_{exc} = 50$ excitatory and , $N_{inh} = 50$ inhibitory connections in an irregular regime, $J_{exc} = 2.0$ mV and $J_{inh} = -5J_{exc}$, and a connection probability $\gamma = 0.1$ for a simulation time of 500 s. The input currents were randomly selected from the uniform distribution $I_i \in [1.02, 1.04]$ pA for all $i \in \{1, 2, \ldots, N\}$. (**b**,**c**,**d**), Characterizers $\boldsymbol{\alpha}^i$ for a randomly selected i with $M \in \{120, 300, 600\}$, respectively.

their applicability on more challenging network structures.

In contrast, our method is intrinsically model independent, thus it only relies on recorded spike trains. Moreover, our method is not constrained to only sparsely connected networks, but it can also reveal connections of denser networks. In addition, it can also reveal connections even if estimations for time delays are substantially inaccurate if the sampling is increased. Nonetheless, although here we only focused on networks of spiking neurons, the concepts and methods introduced in this study may be applied to other systems where the network dynamics is also determined by discrete events.

The present study demonstrates that discrete events in time may reveal physical network connections if they are properly mapped. Yet, given that in this chapter we only simulated oscillatory neurons for generating local samplings, future work

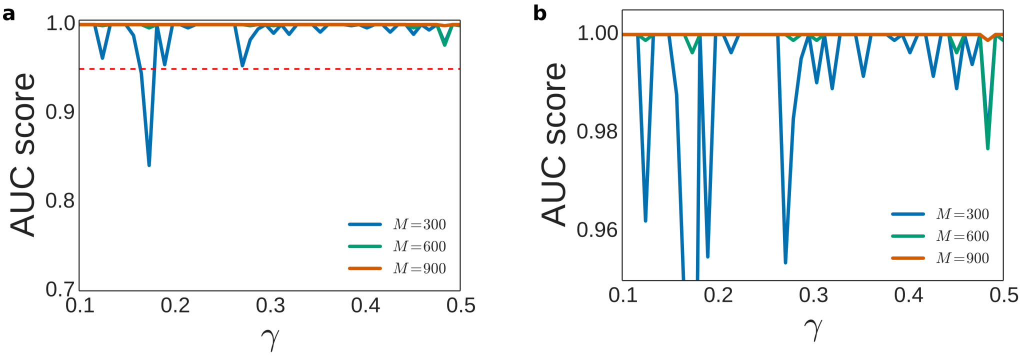


Figure 7.9. – Local samplings reveal synaptic connections of denser networks with no extra effort. (**a**,**b**), Quality of reconstruction measured in AUC score versus connection probability γ for networks of LIF neurons (7.23) of $N = 100$ with $N_{exc} = 50$ excitatory and, $N_{inh} = 50$ inhibitory connections in a regular regime, $J_{exc} = 0.4$ mV and $J_{inh} = -J_{exc}$, and a simulation time of 50 s. The input currents were randomly selected from the uniform distribution $I_i \in [1.2, 1.4]$ pA for all $i \in \{1, 2, \ldots, N\}$. The horizontal dashed line represents an AUC score $= 0.95$.

must be directed to testing the preditive power of our approach on networks where neurons' inputs are irregular in time (e.g. poissonian spike trains as inputs). In particular, irregular inputs may induce even more irregular samplings on the event spaces than those generated by the oscillators employed in this chapter. Consequently, one might need to observe and record the network dynamics for longer periods of time to achieve successful reconstructions. In addition, further efforts must be focused on reducing the number of events for successful reconstructions. Finally, a promising direction is to combine the event space representations introduced in this chapter with the Algorithm for Revealing Network Interactions (ARNI) from chapter 3. Such combination would enhance our framework by discarding local samplings, and thus, allowing to use all recordings in the sampling (e.g. in an irregular dynamics).

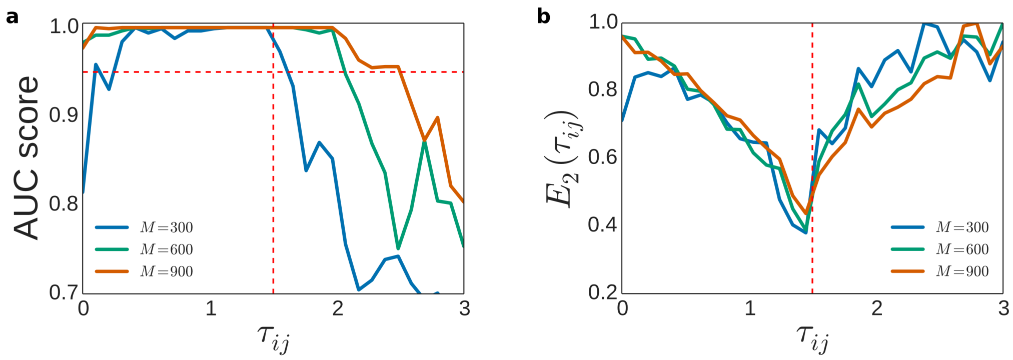


Figure 7.10. – Local samplings employing inaccurate synaptic delays still reveal connections. Reconstruction of networks of LIF neurons (7.23) of $N = 100$ with $N_{exc} = 50$ excitatory and, $N_{inh} = 50$ inhibitory connections in a regular regime, $J_{exc} = 0.4$ mV and $J_{inh} = -J_{exc}$, and a connection probability $\gamma = 0.1$ for a simulation time of 50 s. The input currents were randomly selected from the uniform distribution $I_i \in [1.2, 1.4]$ pA for all $i \in \{1, 2, \dots, N\}$. **a**, Quality of reconstruction measured in AUC score versus synaptic delays. The horizontal dashed line represents an AUC score $= 0.95$. The vertical dashed line indicates the actual delay $\tau^*_{ij} = 1.5$ ms for all $i, j \in \{1, 2, \dots, N\}$. **b**, Error function (7.25) versus synaptic delays. The vertical dashed line indicates the actual delay $\tau^*_{ij} = 1.5$ ms for all $i, j \in \{1, 2, \dots, N\}$.

Conclusions

In this thesis, we studied the network dynamics as an inverse problem from a general standpoint. While relying on concepts and tools coming from nonlinear dynamics and linear algebra, we introduced a collection of physics-inspired inverse approaches to explain a myriad of fundamental questions about networks, not yet proposed at the time at which this thesis was written. For instance, we resolved how disparate networks may generate identical dynamics in time and we provided an explicit mathematical model that links all this networks. Also, we demonstrated that (in general) a network dynamics alone may reveal the network structural connections, without needing to define additional parameters and functions, and regardless of the type of network and dynamics. Thus, the content and results of this thesis are applicable to any kind of interconnected systems that may be described by networks of dynamical systems.

In the first part of the thesis (chapter 1), we portrayed the limitations of mainstream representations of network dynamics in terms of coupled dynamical systems having pairwise interactions. Also, we proposed an alternate representation of network dynamics that solves the structural limitations of mainstream approaches. In particular, this representation constitutes the starting point of all other chapters.

In the second part (chapter 2), we explained how disparate networks perform identical dynamics in time. Specifically, we developed a mathematical framework for finding the family of networks that generates a common collective dynamics. By detaching the contributions to network dynamics coming from structure and function, we analytically parametrized all the possible ways that a collection of generic interconnected units have to interact to reproduce a specific dynamics. In addition, we demonstrated that interactions may be optimized for specific network structures and still achieve pre-defined dynamics.

Our results extend the ideas exposed in [26, 27] on the design of networks of spiking neurons where interactions occur at discrete times to networks of continuously coupled units. Interestingly, our work constitutes (to our knowledge) one the few first studies describing the interplay between the structure and function of complex networks from an analytical perspective. In particular, we demonstrated the existence of a general *give-and-take* mechanism between the structure and function of complex networks. Moreover, such mechanism may be exploited for network design for specific function.

In the third part (chapter 3), we formulated a general approach for inferring structural connections of networks from time series alone. Specifically, we proposed a decomposition of units' dynamics in terms of network interactions. Particularly, we demonstrated that such decompositions reveal physical links between units. Furthermore, we introduced an efficient algorithm for retrieving the network structure from these decompositions. In addition, we found that sufficiently long observations and compositions of several short observations of units' dynamics are equivalent means for revealing network structural connections.

Up-to-date theory on inferring structural connections from time series guarantees that network connections are retrievable if a prior knowledge of the actual local and coupling functions underlying the network dynamics is available [22, 68–70, 73, 74]. In this thesis, based on the limitation that in numerous experimental setups a prior knowledge is infeasible, we went a step further and demonstrated that such prior knowledge is not necessary for revealing the structure of networks. Instead, we showed that time series of network units carry all the necessary information for determining the network structure.

In the fourth part (chapter 4), we introduced a model-independent concept for inferring connections of networks from local samplings of their dynamics. While focusing on the special case of having identical but perturbed replicas of a network, we demonstrated the applicability of this concept by revealing the connectivity of networks from comparisons between nearby and reference dynamical trajectories. Specifically, we derived a mapping from nearby trajectories to Jacobian matrices evaluated at the reference trajectory. These Jacobian matrices are of great importance because they fully describe the network connectivity of the original system. Interestingly, this concept may be applied in a broad variety of situations as shown in chapter 5, chapter 6 and chapter 7.

In the fifth part (chapter 5), we developed a geometric framework for reconstructing connections of networks from average responses to external drivings. Specifically, we formulated linear mappings from displacements of centroids of non-trivial collective dynamics to network structural connections. In particular, given that here we focused on centroids of network dynamics, chaotic and noisy dynamics are readily filtered out when computing centroids through the averaging of network dynamics. Furthermore, this contribution extends the results showed in the previous part and reaffirms the importance of local samplings for inferring the structural connections of networks.

In the sixth part (chapter 6), we further studied the versatility of local samplings for determining the structural connectivity of networks from their dynamics. In particular, by representing the network dynamics in an alternate non-inertial reference frame, we demonstrated that indirect measures of the network dynamics may provide ideal scenarios for revealing network connections from dynamics through local samplings.

Current research on network inference from responses to drivings demonstrated that, under stability conditions, one may systematically interfere with the network dynamics and recover the network physical connections if the network is at simple stable states, such as steady states or periodic orbits [4, 6, 7, 63, 64]. However, little is known about reconstructing networks that do not display simple stable dynamics. For instance, it was proposed in [65–67] that driving networks to steady states through many steady state stabilizations may ease the inference of connections, even if the undriven network shows no relaxation to a steady state.

The framework built up in this thesis through chapter 4, chapter 5 and chapter 6 generalizes the ideas exposed in [4, 6, 7, 63–67] to networks exhibiting more complex dynamics than simple stable states. In general, we demonstrated that local samplings (formally, measurements with linear structure) of quantities that reflect the interconnectedness of network units (e.g. units' activity, average activity or deviations from the average activity) may reveal network connections.

In the last part (chapter 7), while focusing on networks of spiking neurons and their spike trains, we developed an event-based framework for representing the network dynamics in terms of discrete events in time. Interestingly, local samplings on these event-based representations generate linear conditions for network connections in terms of timestamps of discrete events. Furthermore, we demonstrated that these linear conditions map structural connections by reconstructing the synaptic connectivity of networks of LIF neurons from their spike trains alone under different dynamical conditions.

Previous works on inferring synaptic connections from spike trains have focused on two main strategies: (i) assuming models for the evolution of membrane potential of neurons [34,35,40,116], and (ii) applying correlation-based methods that infer causal relationships between recorded spike trains [36, 47, 117]. However, such strategies only work on controlled conditions, such as prior knowledge of intrinsic and coupling functions, prior knowledge of physiological parameters, long recordings and sparsity, among others. In this thesis, by combinining concepts from the preceding chapters, we developed a new strategy for inferring synaptic connections from spike trains. A strategy that makes no assumption nor requires a prior knowledge of physiological parameters or functions underlying the network dynamics. Thus, we consider that our contribution may be a great addition to the state of the art of this field.

Extensions of the work presented in this thesis open further novel perspectives about other inverse problems on networks. Future work still needs to completely understand the give-and-take mechanism described in chapter 2 and [24]. For instance, developing clear links between the family of networks generating a specific dynamics and the functional forms of coupling functions may provide a comprehensive framework for designing networks for specific function. Also, formulating optimization schemes based on the results of chapter 2 for increasing (i) the robustness of networks to dynamical perturbations, or (ii) the adaptability to structural perturbations may lead to interesting applications in real-world networks, such as in power grids where one wants to preserve a specific stable dynamics [50–53]. Also, additional efforts should be focused on devising ways to reduce the amount of data necessary to reveal network structural connections. For instance, locally sub-sampling different network states may help in determining the network structure if one focuses on the global topological properties of the network. In addition, it is necessary to extend our results to networks where only a subset of units is accessible for measurement. Furthermore, another promising direction for future research is to study the feasibility of extending our framework to predict the dynamics of networks based on previously recorded dynamics. Moreover, we also foresee that combinations of the methods presented in this thesis may even provide valuable means for accessing and recovering other quantities and network parameters of interest from the network dynamics alone.

Finally, we hope that our contribution to this rapidly growing field of inverse problems on networks marks a tipping point in the community to progressively move away from model based (or better said, intuition based) inverse approaches to more data driven (and unbiased) inverse approaches.

Appendix A

Multiple linear regression and L_2-norm minimization

Given the linear relation

$$\boldsymbol{y} = \boldsymbol{\beta} X + \boldsymbol{\varepsilon}, \tag{A.1}$$

where $\boldsymbol{y}^\mathsf{T} \in \mathbb{R}^M$ are the values for a dependent variable, $\boldsymbol{\beta}^\mathsf{T} \in \mathbb{R}^N$ is a vector of unknown coefficients, $X \in \mathbb{R}^{N \times M}$ is the set of realizations for the N-independent variables and $\boldsymbol{\varepsilon}^\mathsf{T} \in \mathbb{R}^M$ is a vector of random errors normally distributed with mean $\mu = 0$ and constant variance σ^2 [118]; How can one estimate $\boldsymbol{\beta}$ by some $\hat{\boldsymbol{\beta}}$ such that the difference between the predicted $\boldsymbol{\beta} X$ and measured values $\boldsymbol{y}$ is minimized? If one employs L_2-norm for measuring the error between predictions and measurements, we may pose the problem as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{\beta} X\|_2^2, \tag{A.2}$$

which is also known as the linear least squares method [118].

Due to its convexity properties, the optimization problem (A.2) has a global minimum. Thus, minimizing (A.2) yields

$$\forall i \in \{1, \ldots, N\} \ : \ \frac{\partial}{\partial \beta_i} \left[(\boldsymbol{y} - \boldsymbol{\beta} X)(\boldsymbol{y} - \boldsymbol{\beta} X)^T \right] \bigg|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}} = 0. \tag{A.3}$$

$$\begin{aligned} \frac{\partial}{\partial \beta_i} \left[\boldsymbol{y}\boldsymbol{y}^T - 2\boldsymbol{\beta} X \boldsymbol{y}^T - \boldsymbol{\beta} X (\boldsymbol{\beta} X)^T \right] \bigg|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}} &= 0, \\ \Leftrightarrow \frac{\partial \boldsymbol{\beta}}{\partial \beta_i} \left(X \boldsymbol{y}^T - X (\boldsymbol{\beta} X)^T \right) \bigg|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}} &= 0, \end{aligned}$$

which in turn provides the best approximation

$$\hat{\boldsymbol{\beta}} = \boldsymbol{y}X^T(XX^T)^{-1}, \tag{A.4}$$

of $\boldsymbol{\beta}$ according to the linear least squares method [118]. Furthermore, the matrix

$$X^\dagger = X^T(XX^T)^{-1}, \tag{A.5}$$

is also known as the Moore-Penrose pseudo-inverse.

Appendix B

Singular value decomposition and L_1-norm minimization

Formally, from the fundamental theorem of linear algebra [119], any rectangular matrix $A \in \mathbb{R}^{M\times N}$ may be decomposed into the product of three matrices as

$$A = U\Sigma V^{\mathsf{T}}, \tag{B.1}$$

where $U \in \mathbb{R}^{M\times M}$ and $V \in \mathbb{R}^{N\times N}$ are unitary matrices and $\Sigma \in \mathbb{R}^{M\times N}$ is a rectangular diagonal matrix containing the *singular values of* A. This decomposition is known as the *Singular Value Decomposition* (SVD) of matrix A.

The SVD is of particular interest when finding the family of solutions to an underdetermined system of equations,

$$A\boldsymbol{y} = \boldsymbol{b}, \tag{B.2}$$

where $\boldsymbol{b} \in \mathbb{R}^{M}$ a vector of known values and $\boldsymbol{y} \in \mathbb{R}^{N}$ is a vector of unknowns. Specifically, one can use the SVD (B.1) to find a particular solution for $\boldsymbol{y}$ from (B.2) as

$$\boldsymbol{y}_p = V\tilde{\Sigma}U^{\mathsf{T}}\boldsymbol{b}, \tag{B.3}$$

where $\tilde{\Sigma} \in \mathbb{R}^{N\times M}$ is the Moore-Penrose pseudo-inverse of Σ and is defined as

$$\tilde{\Sigma} = \Sigma^{\mathsf{T}}\left(\Sigma\Sigma^{\mathsf{T}}\right)^{-1}. \tag{B.4}$$

Equation (B.3) only represents a particular solution from the set of all possible solutions for (B.2). Nevertheless, the general solution for (B.2) is given by the particular solution (B.3) plus a linear combination of vectors in the nullspace of A,

$$\boldsymbol{y} = V\tilde{\Sigma}U^{\mathsf{T}}\boldsymbol{b} + W\boldsymbol{\zeta}, \tag{B.5}$$

where $W \in \mathbb{R}^{M\times(N-M)}$ is a basis for the nullspace of A and $\boldsymbol{\zeta} \in \mathbb{R}^{(N-M)}$ is a vector of open parameters. This means that any choice for $\boldsymbol{\zeta}$ produces a vector $\boldsymbol{y}$ that satisfies (B.2).

If one assumes that $\boldsymbol{y}$ is sparse, i.e. that only few of its entries are different zero, one may search for that vector $\boldsymbol{\zeta}$ that maximizes the number of zero entries in $\boldsymbol{y}$. We may achieve this by solving the optimization problem [4, 34, 35, 64]

$$\arg\min_{\boldsymbol{\zeta}} \|V\tilde{\Sigma}U^T\boldsymbol{b} + W\boldsymbol{\zeta}\|_1, \tag{B.6}$$

employing the Barrowdale and Roberts algorithm [120].

B.1. L_1-norm minimization as a linear program

Interestingly, the optimization problem

$$\min_{\boldsymbol{y}} \|A\boldsymbol{y} - \boldsymbol{b}\|_1\,, \tag{B.7}$$

where $A \in \mathbb{R}^{M\times N}$, $\boldsymbol{y} \in \mathbb{R}^N$ and $\boldsymbol{b} \in \mathbb{R}^M$, (B.7) may be also posed as the linear program [121]

$$\min_{\boldsymbol{s}} \mathbf{1}^{\mathsf{T}}\boldsymbol{s} \quad \text{s.t.} \quad \begin{array}{c} A\boldsymbol{y} - \boldsymbol{b} \preceq \boldsymbol{s} \\ A\boldsymbol{y} - \boldsymbol{b} \succeq -\boldsymbol{s} \end{array}, \tag{B.8}$$

where $\mathbf{1} \in \mathbb{R}^M$ is a vector of ones and $\preceq$ and $\succeq$ denote entry-wise comparison and $\boldsymbol{s}$ is an auxiliary variable. To solve (B.8), we solve the second linear program

$$\min_{x} \tilde{\boldsymbol{c}}^{\mathsf{T}}\boldsymbol{x} \quad \text{s.t.} \quad \tilde{A}\boldsymbol{x} = \tilde{\boldsymbol{b}}, \tag{B.9}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{s} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A & -I \\ -A & -I \end{bmatrix}, \quad \tilde{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{b} \\ -\boldsymbol{b} \end{bmatrix}, \quad \tilde{\boldsymbol{c}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \tag{B.10}$$

and $I \in \mathbb{R}^{M\times M}$ and $\mathbf{0} \in \mathbb{R}^N$ are the identity matrix and a vector of zeroes, respectively. The advantage of posing problem (B.7) as (B.9) is that the latter can be easily solved in a standard way by implementing any solver for linear programs (e.g. the `linprog` function in MATLAB [122]).

Bibliography

[1] Euler L 1741 *Comment. Acad. Sci. Petropolitanae* **8** 128–140

[2] Schwender J (ed) 2009 *Plant Metabolic Networks* (Springer Science & Business Media) ISBN 0387787453

[3] Newman M 2010 *Networks: An Introduction* (New York: Oxford University Press) ISBN 0191500704

[4] Yeung M K S, Tegner J, Collins J J, Tegnér J and Collins J J 2002 *Proc. Natl. Acad. Sci. USA* **99** 6163–6168

[5] de Jong H 2002 *J. Comput. Biol.* **9** 67–103

[6] Tegner J, Yeung M K, Hasty J and Collins J J 2003 *Proc. Natl. Acad. Sci. USA* **100** 5944–5949

[7] Gardner T S, di Bernardo D, Lorenz D and Collins J J 2003 *Science (80-.).* **301** 102–5

[8] Levine M and Davidson E H 2005 *Proc. Natl. Acad. Sci. USA* **102** 4936–4942

[9] di Bernardo D, Thompson M J, Gardner T S, Chobot S E, Eastwood E L, Wojtovich A P, Elliott S J, Schaus S E and Collins J J 2005 *Nat. Biotechnol.* **23** 377–383

[10] Faith J J, Hayete B, Thaden J T, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins J J and Gardner T S 2007 *PLoS Biol.* **5** 54–66

[11] Deco G, Jirsa V K and McIntosh A R 2011 *Nat. Rev. Neurosci.* **12** 43–56

[12] Park H J and Friston K 2013 *Science (80-.).* **342** 1238411

[13] Dayan P and Abbott L 2001 *Theoretical Neuroscience* (Cambridge: MIT Press)

[14] Strogatz S H 2001 *Nature* **410** 268–76

[15] Mirollo R E and Strogatz S H 1990 *SIAM J. Appl. Math.* **50** 1645–1662

[16] Winfree A T 2013 *The Geometry of Biological Time* (Springer Science & Business Media) ISBN 1475734840

[17] Brunel N *J. Comput. Neurosci.* **8** 183–208

[18] Drachman D A 2005 *Neurology* **64** 2004–5

[19] Huberman B A and Adamic L A 1999 **401** 131

[20] Marbach D, Prill R J, Schaffter T, Mattiussi C, Floreano D and Stolovitzky G 2010 *Proc. Natl. Acad. Sci. U. S. A.* **107** 6286–91

[21] Marbach D, Costello J C, Küffner R, Vega N M, Prill R J, Camacho D M, Allison K R, Kellis M, Collins J J and Stolovitzky G 2012 *Nat. Methods* **9** 796–804

[22] Timme M and Casadiego J 2014 *J. Phys. a-Mathematical Theor.* **47** 343001 1–36

[23] Tarantola A 2005 *Inverse Problem Theory and Methods for Model Parameter Estimation* (SIAM) ISBN 0898715725

[24] Casadiego J and Timme M 2015 Network Dynamics as an Inverse Problem *Math. Technol. Networks* (*Springer Proceedings in Mathematics & Statistics* vol 128) ed Mugnolo D (Cham: Springer International Publishing) pp 39–48 ISBN 978-3-319-16618-6

[25] Prinz A A, Bucher D and Marder E 2004 *Nat. Neurosci.* **7** 1345–52

[26] Memmesheimer R M and Timme M 2006 *Phys. D* **224** 182–201

[27] Memmesheimer R M and Timme M 2006 *Phys. Rev. Lett.* **97** 188101

[28] Cotterell J and Sharpe J 2010 *Mol. Syst. Biol.* **6** 425

[29] Schaerli Y, Munteanu A, Gili M, Cotterell J, Sharpe J and Isalan M 2014 *Nat. Commun.* **5** 4905

[30] Slusarczyk A L, Lin A and Weiss R 2012 *Nat. Rev. Genet.* **13** 406–420

[31] Cameron D E, Bashor C J and Collins J J 2014 *Nat. Rev. Microbiol.* **12** 381–90

[32] Schadt E E 2009 *Nature* **461** 218–23

[33] Barabási A L, Gulbahce N and Loscalzo J 2011 *Nat. Rev. Genet.* **12** 56–68

[34] Van Bussel F, Kriener B and Timme M 2011 *Front. Comput. Neurosci.* **5** 3

[35] van Bussel F 2010 *Topological Optimization in Network Dynamical Systems* Ph.D. thesis Georg-August-Universitaet Goettingen

[36] Pernice V and Rotter S 2013 *J. Stat. Mech. Theory Exp.* **2013** P03008

[37] Barzel B and Barabási A L L 2013 *Nat. Biotechnol.* **31** 720–725

[38] Guo X, Zhang Y, Hu W, Tan H and Wang X 2014 *PLoS One* **9** e87446

[39] Steinway S N, Biggs M B, Loughran T P, Papin J A and Albert R 2015 *PLOS Comput. Biol.* **11** e1004338

[40] Makarov V A, Panetsos F and de Feo O 2005 *J. Neurosci. Methods* **144** 265–79

[41] MacKay D 2003 *Information Theory, Inference and Learning Algorithms* (Cambridge: Cambridge University Press)

[42] Schreiber T 2000 *Phys. Rev. Lett.* **85** 461–464

[43] Granger C W J 1969 *Econometrica* **37** 424–438

[44] Ladroue C, Guo S, Kendrick K and Feng J 2009 *PLoS One* **4** e6899

[45] Ta H, Yoon C, Holm L and Han S 2010 *BMC Syst. Biol.* **4** 70

[46] Friston K J 2011 *Brain Connect.* **1** 13–36

[47] Quinn C J, Coleman T P, Kiyavash N and Hatsopoulos N G 2011 *J. Comput. Neurosci.* **30** 17–44

[48] Hlinka J, Hartman D and Palus M 2012 *Chaos* **22** 33107

[49] Feizi S, Marbach D, Médard M and Kellis M 2013 *Nat. Biotechnol.* **31** 726–33

[50] Witthaut D and Timme M 2012 *New J. Phys.* **14** 083036

[51] Rohden M, Sorge A, Timme M and Witthaut D 2012 *Phys. Rev. Lett.* **109** 064101

[52] Witthaut D and Timme M 2013 *Eur. Phys. J. B* **86** 377

[53] Manik D, Witthaut D, Schäfer B, Matthiae M, Sorge A, Rohden M, Katifori E and Timme M 2014 *Eur. Phys. J. Spec. Top.*

[54] Meloni S, Perra N, Arenas A, Gómez S, Moreno Y and Vespignani A 2011 *Sci. Rep.* **1** 62

[55] Gauvin L, Panisson A, Cattuto C and Barrat A 2013 *Sci. Rep.* **3** 3099

[56] Chen D B, Xiao R and Zeng A 2014 *Sci. Rep.* **4** 6108

[57] Smilkov D, Hidalgo C A and Kocarev L 2014 *Sci. Rep.* **4** 4795

[58] Timme M, Wolf F and Geisel T 2002 *Phys. Rev. Lett.* **89** 258701

[59] Wiley D A, Strogatz S H and Girvan M 2006 *Chaos* **16** 15103

[60] Jahnke S, Memmesheimer R M and Timme M 2008 *Phys. Rev. Lett.* **100** 48102

[61] Memmesheimer R M 2010 *Proc. Natl. Acad. Sci. USA* **107** 11092–11097

[62] Menck P J, Heitzig J, Marwan N and Kurths J 2013 *Nat. Phys.* **9** 89–92

[63] Timme M 2006 *Europhys. Lett.* **76** 367–373

[64] Timme M 2007 *Phys. Rev. Lett.* **98** 224101

[65] Yu D and Parlitz U 2010 *Phys. Rev. E* **82** 26108

[66] Yu D 2010 *Automatica* **46** 2035–2040

[67] Yu D and Parlitz U 2011 *PLoS One* **6** e24333

[68] Yu D, Righero M and Kocarev L 2006 *Phys. Rev. Lett.* **97** 188701

[69] Shandilya S G and Timme M 2011 *New J. Phys.* **13** 013004

[70] Wang W X, Yang R, Lai Y C, Kovanis V and Harrison M A F 2011 *Europhys. Lett.* **94** 48006

[71] Wang W X, Yang R, Lai Y C, Kovanis V and Grebogi C 2011 *Phys. Rev. Lett.* **106** 154101

[72] Wang W X X, Lai Y C C, Grebogi C and Ye J 2011 *Phys. Rev. X* **1** 21021

[73] Han X, Shen Z, Wang W X and Di Z 2015 *Phys. Rev. Lett.* **114** 028701

[74] Su R Q, Wang W X and Lai Y C 2012 *Phys. Rev. E* **85** 065201

[75] Ashwin P and Timme M 2005 *Nonlinearity* **18** 2035–2060

[76] Strogatz S H 2014 *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* ISBN 0813349117

[77] Kapitaniak T and Bishop S R 1999 *The Illustrated Dictionary of Nonlinear Dynamics and Chaos* ISBN 0471983233

[78] Faghih R T, Savla K, Dahleh M A and Brown E N 2010 *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.* **2010** 4116–9

[79] Rössler O 1976 *Phys. Lett. A* **57** 397

[80] Karlebach G and Shamir R 2008 *Nat. Rev. Mol. Cell Biol.* **9** 770–80

[81] Leenheer P D, Angeli D and Sontag E D 2006 *J. Math. Chem.* **41** 295–314

[82] Pearson K J and Zhang T 2013 *Graphs Comb.* **30** 1233–1248

[83] Bretto A 2013 *Hypergraph Theory: An Introduction* ISBN 3319000802

[84] Korsch H J and Jodl H J 1999 *Chaos: a program collection for the PC, Band 1* ISBN 3540638938

[85] Skardal P S, Taylor D and Sun J 2014 *Phys. Rev. Lett.* **113** 144101

[86] Acebrón J, Bonilla L, Pérez Vicente C, Ritort F and Spigler R 2005 *Rev. Mod. Phys.* **77** 137–185

[87] Hansel D, Mato G and Meunier C 1993 *Phys. Rev. E* **48** 3470–3477

[88] Meyer C D 2000 *Matrix Analysis and Applied Linear Algebra* (Society for Industrial and Applied Mathematics)

[89] Strogatz S H 2000 *Phys. D Nonlinear Phenom.* **143** 1–20

[90] Yu D and Parlitz U 2008 *Phys. Rev. E* **77** 66221

[91] Bruex A, Kainkaryam R M, Wieckowski Y, Kang Y H, Bernhardt C, Xia Y, Zheng X, Wang J Y, Lee M M, Benfey P, Woolf P J and Schiefelbein J 2012 *PLoS Genet.* **8** e1002446

[92] Yosef N, Shalek A K, Gaublomme J T, Jin H, Lee Y, Awasthi A, Wu C, Karwacz K, Xiao S, Jorgolli M, Gennert D, Satija R, Shakya A, Lu D Y, Trombetta J J, Pillai M R, Ratcliffe P J, Coleman M L, Bix M, Tantin D, Park H, Kuchroo V K and Regev A 2013 *Nature* **496** 461–8

[93] Roy S and Kundu T K 2014 *IUBMB Life* **66** 100–9

[94] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning* Springer Series in Statistics (New York, NY: Springer New York)

[95] Yuan M and Lin Y 2006 *J. R. Stat. Soc. Ser. B (Statistical Methodol.* **68** 49–67

[96] Eldar Y C and Mishali M 2009 *IEEE Trans. Inf. Theory* **55** 5302–5316

[97] Kowalski M 2009 *Appl. Comput. Harmon. Anal.* **27** 303–324

[98] Eldar Y C, Kuppinger P and Bolcskei H 2010 *IEEE Trans. Signal Process.* **58** 3042–3054

[99] Duarte M F and Eldar Y C 2011 *IEEE Trans. Signal Process.* **59** 4053–4085

[100] Napoletani D and Sauer T D 2008 *Phys. Rev. E* **77** 26103

[101] Fawcett T 2006 *Pattern Recognit. Lett.* **27** 861–874

[102] Paulsson J 2004 *Nature* **427** 415–8

[103] Kaern M, Elston T C, Blake W J and Collins J J 2005 *Nat. Rev. Genet.* **6** 451–64

[104] Bryan K and Leise T 2013 *SIAM Rev.* **55** 547–566

[105] Tirabassi G, Sevilla-Escoboza R, Buldú J M and Masoller C 2015 *Sci. Rep.* **5** 10829

[106] Stoer J and Bulirsch R 1993 *Introduction to Numerical Analysis* (New York: Springer)

[107] Ohiorhenuan I E, Mechler F, Purpura K P, Schmid A M, Hu Q and Victor J D 2010 *Nature* **466** 617–21

[108] Bullmore E and Sporns O 2012 *Nat. Rev. Neurosci.* **13** 336–49

[109] Leclerc R D 2008 *Mol. Syst. Biol.* **4** 213

[110] Savitzky A and Golay M J E 1964 *Anal. Chem.* **36** 1627–1639

[111] Moore J 1968 *Proc. IEEE* **56** 871–871

[112] Ren J, Wang W X X, Li B and Lai Y C C 2010 *Phys. Rev. Lett.* **104** 058701

[113] Prignano L and Díaz-Guilera A 2012 *Phys. Rev. E* **85** 036112

[114] Magnus J R and Neudecker H 1988 *Matrix differential calculus with applications in statistics and econometrics* ISBN 0471915165

[115] Gewaltig M O and Diesmann M 2007 *Scholarpedia* **2** 1430

[116] Zaytsev Y V, Morrison A and Deger M 2015 *J. Comput. Neurosci.* **39** 77–103

[117] Gerwinn S, Macke J H and Bethge M 2010 *Front. Comput. Neurosci.* **4** 12

[118] Yan X and Su X G 2009 *Linear Regression Analysis: Theory and Computing* (Toh Tuck: World Scientific Publishing)

[119] Strang G 1993 *Am. Math. Mon.* **100** 848–855

[120] Barrodale I and Roberts F D K 1974 *Commun. ACM* **17** 319–320

[121] Boyd S and Vandenberghe L 2009 *Convex Optimization* (Cambridge: Cambridge University Press)

[122] MATLAB7 *(Natick, MA Mathworks Inc.)*

Acknowledgments

Since I began my path in science approximately 10 years ago, I had only one goal in mind: to generate substantial knowledge in a challenging scientific field. Because of this, I chose to work in this field. And today, after 4 years, I am happy to say that I have contributed (at least a bit) to this exciting and challenging field of complex networks. Nevertheless, I did not get to this stage on my own. Thus, in the following lines, I would like to thank all those persons that directly or indirectly helped me reach my goal.

I would like to thank my supervisor Prof. Dr. Marc Timme for giving me the opportunity to pursue a PhD in his group. But, especially, I would like to thank you Marc for your unmeasurable support. You are one of the main reasons that I am here today writing these lines.

I would like to thank the members of my thesis committee Prof. Dr. Ulrich Parlitz and Prof. Dr. Reiner Kree, without your understanding and support I would certainly not be here.

I would also like to thank my parents Jose and Carmen and my siblings Luiselena and Edicson for their unconditional love and support. There is no day (not even a moment) in which you are not present in my mind. This work is dedicated to you, because, in the end, I am just a linear combination of you all.

Special thanks are due to my former colleagues at the Clinical Neurophysiology Department. Catarina, Giorgi, Rafael and Olga, thank you very much for giving me the courage to pursue my dream. Without your support, I would not be here.

Many thanks to my colleagues at the Network Dynamics Group of the Max Planck Institute for Dynamics and Self-Organization. Needless to say, the Network Dynamics Group felt like a second home to me. Also, I would like to give special thanks to Fabio, Gunter, Martin and Hinrich for all those TT training sessions. You built the TT machine that I am today. In addition, I also want to thank Hauke and Dimitra for all those funny discussions in the Masters room and for letting me learn how to supervise great students.

I would like to thank the staff of the GGNB and PBCS offices. But especially, I want to thank Kirsten, Antje and Tina for their understanding and support over these years.

And, where would I be without my friends? In the end, no man is an island. So, many many thanks to all those great people that I met along these 4 years. I would love to name you all, but you are simply way too many!

I would also like to give special thanks to my Göttinger Colombuddies John, Chepe, Melanie, Pedro and Angela and many others (sorry you are also too many)! Nothing but good memories come to my mind whenever I think of you, well except the hangovers.

Special thanks to my Venezuelan friends Amanda, Anelise, Johnny, and especially, Nairim and Frank, for providing me with *un pedacito de mi tierra bien lejos de esa vaina!* You became my second family, and I really hope that our paths cross again (hopefully many times) in the future.

And, last but not least, my partner in adventures, my damsel in distress, my Silvia. I could spend days writing words of gratitude to you, and such will not be enough to express how thankful I am for all your support, especially that of the last weeks of October 2015. There is no doubt that thanks to you and only you I am here finishing this stage (a very important stage) of my life.