Università di Genova

Corso di Dottorato in Ingegneria Elettrica
Dipartimento di Ingegneria Navale, Elettrica, Elettronica
e delle Telecomunicazioni

Parameter identification in networks of dynamical systems

Valentina Baruzzi

Supervisor: Prof. Marco Storace
PhD Coordinator: Prof. Mario Marchesoni

XXXV ciclo – 2022

Abstract

Mathematical models of real systems allow to simulate their behavior in conditions that are not easily or affordably reproducible in real life. Defining accurate models, however, is far from trivial and there is no one-size-fits-all solution. This thesis focuses on parameter identification in models of networks of dynamical systems, considering three case studies that fall under this umbrella: two of them are related to neural networks and one to power grids. The first case study is concerned with central pattern generators, i.e. small neural networks involved in animal locomotion. In this case, a design strategy for optimal tuning of biologically-plausible model parameters is developed, resulting in network models able to reproduce key characteristics of animal locomotion. The second case study is in the context of brain networks. In this case, a method to derive the weights of the connections between brain areas is proposed, utilizing both imaging data and nonlinear dynamics principles. The third and last case study deals with a method for the estimation of the inertia constant, a key parameter in determining the frequency stability in power grids. In this case, the method is customized to different challenging scenarios involving renewable energy sources, resulting in accurate estimations of this parameter.

Contents

Introduction

Describing physical entities through mathematical models has always been a fundamental goal of numerous areas of scientific research. With the exponential increase of available computational power, having access to accurate mathematical models of physical systems allows to simulate their behavior in conditions that are not easily or affordably reproducible in real life. However, defining mathematical models that are complex enough to reliably approximate the physical systems' behavior, but also simple enough to have a clear physical interpretation and a feasible computational implementation is far from trivial. There is no *one size fits all* solution, and each case study needs a tailored approach. Numerous case-specific guidelines for system modeling have been provided throughout the scientific literature. For example, in a 1980 study by Chua [1], two general approaches to electrical device modeling are presented: the physical approach and the black-box approach. The physical approach is applicable when the device physics and operating mechanisms are well-understood and consists of four basic steps. In step 1, the important physical variables, phenomena, and operating mechanisms should be identified through a careful analysis of the device physics; the internal structure of the device should then be partitioned into as many distinct units as possible, such that each one can be modeled separately. In step 2, the relevant physical equations which relate the internal physical variables of each unit to their inputs and outputs are formulated. In step 3 the formulated equations should be solved by making the necessary simplifying assumptions and lumping together the separate units when possible. Step 4 consists of synthesizing the nonlinear network, by defining the links between the final identified units. The black-box approach should instead be used when the device is so complex that a physical approach is not feasible or would be impractical. For example, biological circuit models (such as the well-known Hodgkin-Huxley neuron model) where physical mechanisms at the molecular level were poorly understood, or VLSI (very large scale integration) circuits which contain thousands of components. The black-box approach is also divided into four main steps: experimental observations, mathematical modeling, model validation, and nonlinear network synthesis. Step one consists of probing the device with different testing signals to extract as many general qualitative properties as possible (for example a hysteresis loop or subharmonic response under certain conditions). In step 2 a mathematical model capable of simulating most of the observed properties should be synthesized; since there is no systematic method to accomplish this task, the design of an appropriate mathematical structure is described as an “art”, that can count on some guidelines and clues about prescribed properties of certain known nonlinear systems. Step 3 consists of analyzing the derived mathematical model to determine whether it is indeed capable of approximately reproducing the measured data and observed properties; both a quantitative validation (comparing the simulated response to the measured response) and a qualitative validation (a mathematical proof that the model does exhibit all the qualitative properties observed experimentally) should be carried out. Step 4 is equivalent to the last step of the physical approach, and consists of synthesizing the nonlinear network.

System modeling techniques have evolved throughout the decades, both in terms of white-box approaches (based on knowledge about the physics of the system), black-box approaches (based on statistical information from the data), and grey-box approaches (a combination of both). Black-box techniques have particularly benefited from machine learning advances, using powerful tools such as neural networks and Gaussian processes, leading to countless real-life applications [2]. In 2014 Grieves introduced the concept of *digital twin* [3] as a virtual representation of a complex physical asset in the digital space for the purpose of closely characterizing the operations of the original physical process or system. Now, the digital twin concept has been widely adopted at the industrial scale, becoming one of the enabling technologies of *Industry 4.0* [4, 5].

In recent years, the study of networks of dynamical systems in particular has garnered growing interest due to their widespread presence and importance in various fields, including physics, biology, engineering, and the social sciences. Network science was defined as a new field of basic research during the 2005 U.S. National Research Council [6], after the attention towards complex networks generated in the scientific community thanks also to pioneering studies by Watts & Strogatz [7], Barabasi & Albert [8], and Girvan & Newman [9]. Networks of dynamical systems consist of interconnected units, each governed by its own set of rules or dynamics. At first, the efforts of the scientific community were focused on understanding network topology and its structural properties; concepts like smallworldness, scale-free property, modularity and fractality arose one after the other, and various network models were proposed to mathematically describe the architecture and evolution of real-world networks [10]. One of the key features of networks of dynamical systems is their ability to exhibit complex emergent behaviors, which are often difficult to predict or control, arising from the interactions between their constituent units [11]. A fundamental goal of this field of research is thus understanding how the interplay between the structure of the network and the dynamics of the units determines the collective behavior of the network system [12]. In recent years, a shift from the structural analysis of networks to the simulation of their collective dynamics has been possible thanks to unprecedented computing power, big data sets and new computational modeling techniques; this has provided a bridge between the dynamics of individual nodes and the emergent properties of macroscopic networks [10].

Many real-world processes take place in network structures: the movement of money through an economy, the circulation of electricity through a power grid, the evolution of populations in an ecosystem, the flow of electric signals through a network of neurons. Examples of such networks are displayed in Fig. 1. These and many other phenomena of scientific interest are best approximated by models of dynamical systems interconnected by an appropriate set of links, forming a network [12]. Dealing with fields as disparate as the social sciences, physics, biology, computer science and applied mathematics, made it difficult for the scientific community to find common ground, agree on definitions and reconcile and appreciate the different approaches that each field had adopted to study networks. Integrating knowledge and methodologies from graph theory, statistical physics, computer science, and sociology is still a work in progress, presenting all the difficulties and traps inherent in interdisciplinary work [10].

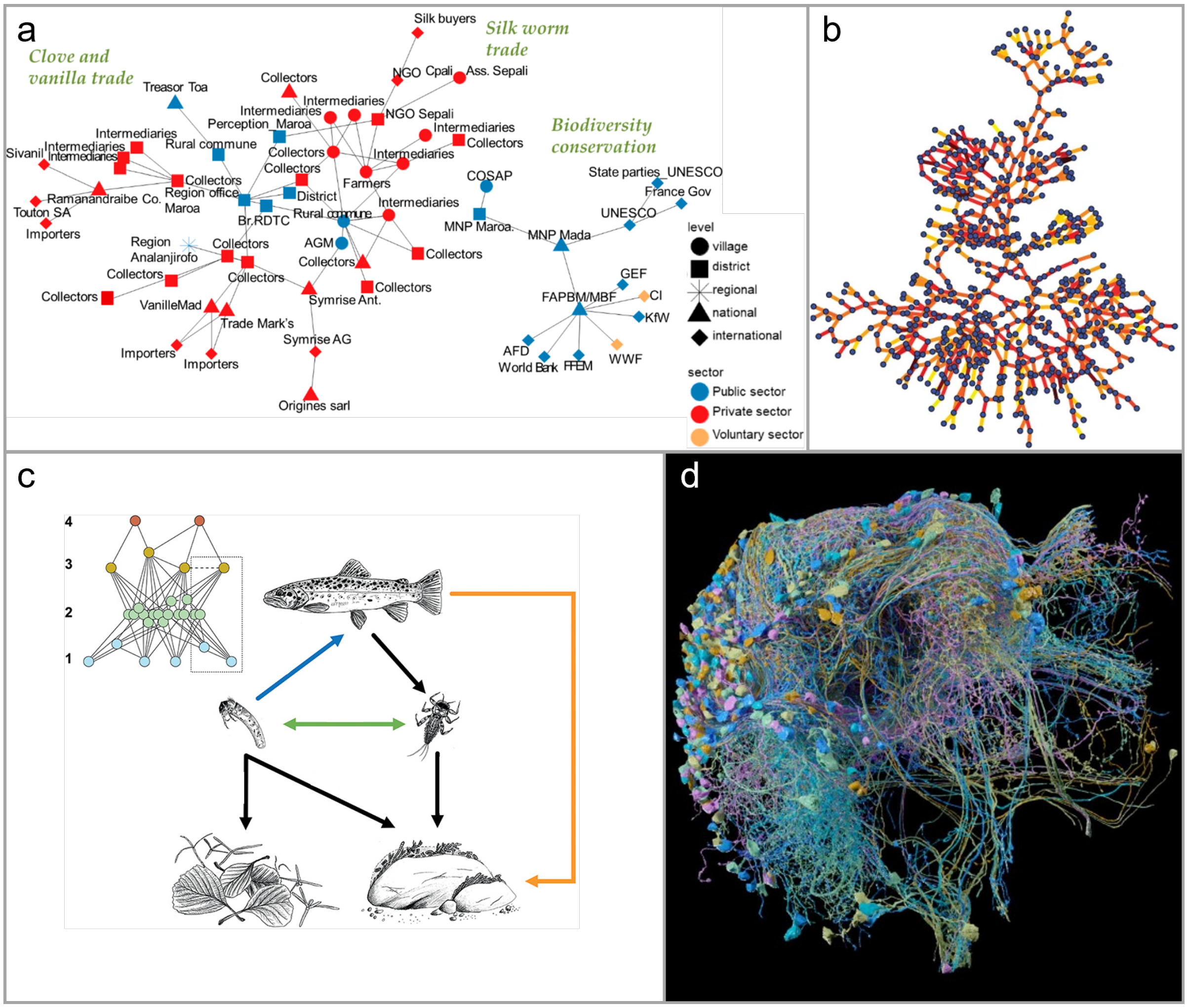


Figure 1 ***(previous page)*****:** (a) Network graph of the flow of money in Mahalevona village (Madagascar) representing two domains: economic (clove and vanilla), and environmental (biodiversity conservation and silkworm trade) [13]. (b) Representation of the physical network of transmission lines for the power grid of Northern Italy. The color scale of the lines indicates the link weights, ranging from yellow to red to black, defined as the absolute value of the corresponding admittance. Adapted from [14]. (c) A simplified representation of biotic interactions in a stream food web. Black arrows represent trophic interactions with top-down effects on resources. The blue arrow represents a trophic interaction with bottom-up effects on the consumer. The orange arrow represents an indirect consequence of trophic interactions, whereas the green arrow represents non-trophic interactions. The inset shows the complete network of species at the site of which the large scheme is an extract or simplification as shown by the dotted box. Blue nodes (trophic level 1) represent basal resources, green nodes (trophic level 2) represent invertebrates, and orange (trophic level 3), and red nodes (trophic level 4) represent fish. Trophic interactions in the network are exemplified by continuous lines and non-trophic interactions by the dashed lines. Adapted from [15]. (d) The largest synaptic-level connectome ever reconstructed of a portion of the central brain (referred to as the hemibrain) of the fruit fly, Drosophila melanogaster [16, 17].

Depending on the network that is being modeled, individual units can exhibit linear or nonlinear dynamics. Nonlinear dynamics are prevalent in real-world scenarios and can exhibit a wide range of behaviors, including stable states, periodic oscillations, and chaotic dynamics. While each nonlinear model has its own unique features and challenges, they can all be studied under the theory of nonlinear dynamical systems, a well-developed branch of mathematics and physics [18].

One of the key challenges when modeling networks of dynamical systems is to accurately determine both the values of the parameters that govern the behavior of the individual units and the topology and strength of the connections between units. These aspects have a fundamental impact on the collective behavior of the system and are often difficult to measure directly or to estimate with standard optimization techniques. For example, inferring causal relations between network nodes (i.e., inferring the directed graph topology) from time series measurements is an ill-posed mathematical problem, where typically an infinite number of potential solutions can reproduce the given data. Decades of research effort have produced a large and diverse set of mathematical methods, from the Nobel prize winning work of Granger [19, 20], that introduced the concept of *Granger causality*, to *convergent cross mapping* [21] (a statistical test for a cause-and-effect relationship based on the theory of dynamical systems and suitable for cases where causal variables have synergistic effects), to recent techniques involving the use of recurrent neural networks [22]. However, all these methods have limitations in accurately reproducing causal network structures, especially for networks of nonlinear dynamical systems [21, 23, 24, 25, 26, 27], and whether it is possible to formulate a consistent and unifying theory of causality for time-dependent systems remains an open question [28]. Inferring causal relations from time series measurements is not the only approach that has been proposed to recover network topology. Other examples include synchronization-based techniques [29, 30, 31], methods based on perturbations [32] and driving the system to a fixed point [33]. Each approach, however, has its own drawbacks [34, 35]; generally, systems exhibiting complex dynamics, such as asynchronous chaotic activity, bifurcations, multistability or other prevalent features of high-dimensional, nonlinear systems, prevent network reconstruction by the aforementioned methods.

Therefore, when designing or dealing with networks of nonlinear dynamical systems, it is important to carefully consider the problem of parameters and topology identification and, when necessary, develop tailored methods for accurately identifying the parameters' values.

This thesis focuses on parameter identification in networks of dynamical systems considering three case studies, two of which concern neural networks and one concerning power grids. These seemingly unrelated types of dynamical networks have proven to exhibit apparent qualitative similarities [36].

The first case study is concerned with central pattern generators (CPGs): they are neural networks that autonomously produce rhythmic patterns of activity, such as those seen in walking, breathing, and other types of movement. These patterns are generated by the interaction of neurons within the CPG and do not require external input [37]. The study of CPGs is important for several reasons: understanding the neural basis of these behaviors can provide insight into the underlying mechanisms of motor control and may have applications in fields such as robotics and rehabilitation; CPGs are also an interesting model system for studying how the nervous system can generate complex patterns of activity without relying on input from the senses or higher brain centers. Moreover, CPGs are present in many different types of animals, from invertebrates to humans, which means that studying them can provide insight into the evolutionary history of rhythmic behaviors and the neural mechanisms that underlie them. The first chapter of this thesis deals with the design of a biologically plausible CPG model for quadruped locomotion, an example of a biological network where each node is represented by a realistic neuron model. It also illustrates custom parameter identification strategies that have been developed to tune the numerous parameters of this model, to ensure its correct behavior. The results presented in this chapter have been published in [38] and [39].

The second case study is in the context of brain networks: deriving reliable models that can reproduce certain human brain functions under specific conditions is the subject of broad interest from the scientific community [40, 41, 42]. One of the pillars of the research in this field is the growing availability of neuroimaging techniques, which allow, on the one hand, the estimation of anatomical connectomes and, on the other hand, the detection of patterns of activity across brain areas, often referred to as functional connectivity. The second chapter of this thesis tries to bridge functional and anatomical neural connectivity through nonlinear dynamics and network theory, aiming to build brain models based on the imaging data of individual subjects. In particular, it details a method to derive the weights of the connections between brain areas utilizing both data-driven and model-based approaches. In this case, each network node is represented by a neural mass model [43], a mathematical model that represents a group of neurons as a single entity, or “neural mass”. These models are used to study the collective behavior of large populations of neurons and to understand how they contribute to brain function. Neural mass models can be used to simulate a wide range of brain phenomena, including the generation of brain waves, the spread of neural activity through the brain, and the interactions between different brain areas. The results presented in this chapter are illustrated in [44].

The third and last case study focuses on power grids, and in particular on the role of the inertia constant: conventionally, the inertia constant is a measure of the rotational inertia of the synchronous generators connected to the grid and is used to describe the ability of the generators to maintain their speed of rotation under changing load conditions. The inertia constant is important in determining the stability of the power grid: a high inertia constant helps to prevent frequency oscillations and other types of instability in the power grid [45]. To ensure that the power grid remains stable as the percentage of renewable energy sources increases, it is necessary to implement control systems that provide synthetic inertia, to maintain a sufficiently high inertia constant. In this scenario, the inertia constant is time-varying, and it becomes important to track its value in real-time during normal operating conditions to ensure the smooth operation of the power grid. The third chapter of this thesis presents a method for the online estimation of the inertia constant in power grids. It discusses the general algorithm framework and how it can be adapted to be effective in different scenarios. The results presented in this chapter have been published in [46], [47] and [48].

Chapter 1

Central Pattern Generators

Personal contribution — Sections 1.1 and 1.2 present known material and concepts that serve to frame the original contributions detailed in the rest of the chapter. The generalized half-center oscillator (section 1.3) and the quadruped central pattern generator (section 1.4) models, along with the related parameter identification strategies and results, are original material. In Section 1.5, well-known global optimization algorithms are applied to the novel central pattern generator model.

1.1 What are central pattern generators?

Central pattern generators (CPGs) are small neural circuits that can autonomously produce various rhythmic patterns of neural activity, also in the absence of sensory feedback or inputs from higher areas [37]. CPGs underlie many behavioral rhythmic activities and they bear a fundamental function in both invertebrate and vertebrate animals as they determine multi-phase locomotion: the innate motor behavior that requires sequential activation of body muscles in a coordinated way. Although CPGs can produce rhythms without sensory inputs, sensory feedback is important to adapt and fine-tune the activity of the CPG to the environment [49]. Moreover, their activity is modulated through the influence of hierarchically higher areas, which can, for example, prompt transitions between gaits [50, 51, 52].

The concept of the CPG itself was first described in invertebrates [53, 54] and, shortly after, its theorized existence in the spinal cord of vertebrate animals [55] was confirmed by intracellular recordings [56, 57].

During the last few decades, a variety of molecular, genetic, pharmacological, and imaging studies have been conducted to understand the structure and function of the locomotor CPG [58, 59, 60]. The lamprey in particular has been extensively used to model and study locomotion [61, 62, 63], since its nervous system has a simple structure with very few neurons and its excised spinal cord can survive for days, but an array of different species (sea slug, leech, cockroach, stick insect, crustaceans, rat and mouse) has been investigated [64, 65, 66, 67].

Because the invasive methods used in reduced animal preparations are not viable to study humans, there is only indirect evidence of a human locomotor CPG; however, this evidence suggests that the general operational principles for the control of rhythmic movement found in other animals extend as well to humans. Ongoing understanding of locomotor CPGs in humans is important to design effective gait rehabilitation strategies in those with neurological injuries [68].

The study of CPGs is also relevant in developing bio-inspired engineering solutions: there are several interesting properties of CPGs that have been proven to be useful for the control of locomotion in robots. CPG-inspired controllers have been tested and used extensively for robot locomotion in the past two decades, with examples in various fields: insect-inspired hexapod and octopod robots [69], swimming robots [70], terrestrial snake robots [71], quadruped [72] and biped walking robots [73]. Since CPG models can produce stable rhythmic patterns by exhibiting limit cycle behavior, the modeled system is robust against transient perturbations, as the state variables rapidly return to the limit cycle provided that they are not pushed outside the basin of attraction and that the Floquet multiplicators lie significantly far from the unit circle. Well-implemented CPG models usually have few driving signals, therefore reducing the dimensionality of the control problem, and they typically produce smooth modulation of the trajectories even when the control parameters are abruptly changed, thus avoiding possible damage to the robot. CPGs are well suited for distributed organization, allowing delays reduction in the control loop. They also give the opportunity to easily integrate sensory feedback signals in the control mechanism [74] [75].

1.2 Modeling of CPGs

CPG models can be designed at different levels of abstraction, depending on the phenomena under investigation [74]. To study rhytmogenesis in small neural circuits or in single pacemaker neurons [76], detailed biophysical models are needed (such as Hodgkin-Huxley type models) to compute how ion pumps and ion channels influence membrane potentials [77].

Connectionist models that use simplified neuron models (such as LIF neurons) are useful to investigate how network properties influence rhythm generation and how different circuits synchronize via interneuron connections [78, 79, 80].

Something that is well established in dynamical systems theory is that the dynamics of populations of oscillatory centres depend mainly on the type and topology of couplings rather than on the local mechanisms of rhythm generation [81, 82]; for this reason, mathematical models of coupled nonlinear oscillators (representing the activity of a complete oscillatory centre) are ideal to study population dynamics and how inter-oscillator couplings and differences of intrinsic frequencies affect the synchronization and the phase lags within a population of oscillatory centres [83, 84, 85, 86, 87, 88]. Some extensively used oscillators are van der Pol, Stein and FitzHugh–Nagumo [87], Matsuoka oscillators [89, 90, 91] and phase oscillators [83, 85, 86, 92].

The process of designing a CPG model begins with the definition of the general architecture, including the number and the type of neurons or oscillators. The type and topology of couplings will then determine the conditions for synchronization between the oscillators and the resulting gaits. In the context of robot controllers, the waveform will determine the trajectories that will actually be performed by the joint angles during a cycle. Lastly, one has to define how important quantities (frequency, amplitude, phase lags for gait transition) depend on control parameters and how the feedback signal will affect the activity of the CPG. The process is not straightforward because the different aspects of CPG design are strongly interconnected and cannot be considered separately [74]. For this reason, there is no sound methodology to obtain a full understanding of how control parameters can tune quantities such as frequency, amplitude, phase lags or waveforms [75], and developing methods for parameter identification and tuning of CPG models is a relevant objective. Dynamical systems theory can offer useful principles to design CPGs and understand their functioning: in fact, the dependence of gait transition from high-level control parameters can be explored by performing bifurcation analysis. However, biologically-inspired models that are developed on the basis of physiological evidence are often too complex and contain too many neurons to be suitable for analysis based on tools within the framework of dynamical systems theory. It is also true that neurons in CPGs are often grouped in subsets that behave in synchrony and can be modeled by single neurons, allowing the reduction of large networks to their simplified cores. It is important to find a compromise between biological plausibility, allowing to reproduce gaits qualitatively and quantitatively, and simplicity of the system, allowing its analysis [93]. Once a simple enough structure for the network is obtained, a bifurcation parameter accounting for the descending pathways has to be included in neuron and synapse models. Brute force simulations can then be carried out to determine the nonlinear dependence of the network components on the bifurcation parameter.

1.2.1 Biological plausibility

As discussed, various approaches to the modeling of CPGs and CPG-inspired control systems have been explored in the last decades [75, 94]. Recently, new methods have been proposed to reduce large models of detailed neural networks to smaller CPG circuits, trading off biological plausibility and complexity of the model [95, 96, 97, 98, 99].

Although CPGs can function autonomously, one of the functions of inputs from hierarchically higher areas is to prompt transitions between gaits [50, 51, 52]. A single gait in a typical CPG model is obtained by fixing the connectivity. By contrast, to generate multiple gaits the CPG connections between constituent neurons are typically changed by acting on the synaptic weights to model the control action of the brainstem [94, 97, 98, 100]. The modulation from higher areas that controls the synchronization between the CPG neurons, and thus triggers gait switches, is conveniently integrated into CPG models by directly affecting the synaptic conductance strengths. However, in real CPGs changes in conductance values are the result of long-term synaptic plasticity, and therefore they are hardly a cause for quick gait switches, which can instead be accounted for more realistically by short-term neuromodulation. Indeed, most natural CPGs exhibit patterns of functional connectivity between neurons or synchronized clusters of neurons that can undergo spontaneous fluctuations and be highly responsive to perturbations, e.g., induced by sensory input or cognitive tasks, on a timescale of milliseconds or hundreds of milliseconds, respectively, thus ensuring robustness and stability. This short-term neuromodulation lacks in most CPG models.

1.3 The generalized half-center oscillator

One of the pivotal building blocks of many CPGs is the half-center oscillator (HCO). The HCO concept is widely used to model two synchronous pools of neurons reciprocally inhibiting each other to produce stable rhythmic alternation in animal locomotion [55, 101]. Before tackling complete locomotion CPG networks, this section focuses on HCOs to introduce novel biologically plausible modeling solutions.

This basic structure has been largely studied from both biological and nonlinear dynamics standpoints. Many previous works on the phasic/antiphasic synchronization of neurons or groups of neurons analyzed the influence of synaptic changes on circuit dynamics. For example, refs. [102, 103, 104, 105] consider HCO models where transitions between stable synchronous states occur through direct manipulations of the synaptic weights; in [106] a large database of HCO models is swept using a brute-force approach, without a focus on gait transitions. While the importance of an interplay between inhibitory and excitatory coupling has already been outlined [105], the thorough understanding of its functional role for determining multiple states or patterns in such neural networks and how transitions between them may stably occur remains yet insufficient.

Moreover, there is growing evidence that (i) post-synaptic potential (PSP) summation increasing with the spike frequency in the pre-synaptic cell is a crucial factor for the stable functioning of some CPGs [107, 108, 109, 110], while other experiments indicate that (ii) the activity of some synapses is barely affected by the spike frequency [94].

Based on this evidence, this section proposes a novel generalized half-center oscillator (gHCO) composed of two neurons or of two neural pools that are coupled reciprocally by excitatory synapses, in addition to the standard HCO's reciprocally inhibitory synapses. This original model warrants a more biologically plausible mechanism of short-term plasticity to implicitly control the phase lag between the gHCO cells by varying their spike frequency through sensory drive or external currents, rather than directly manipulating the synaptic conductance strengths. The proposed generalized half-center oscillator is shown in Fig. 1.1. It is made of two neurons or two neural pools, coupled by both excitatory (marked by a black circle) and inhibitory (marked by a black triangle) synapses.

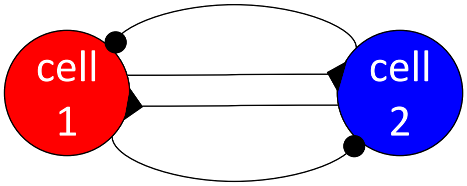


Figure 1.1: gHCO neural circuit with inhibitory (denoted with •) and excitatory (▶ ◀) synapses reciprocally coupling two oscillatory cells.

1.3.1 Design constraints

There are a few simple constraints that neurons and synapses must meet for the circuit to generate stably the desired rhythmic outcomes:

(a) both neurons are endogenous bursters;

(b) their spiking voltage range is above the hyperpolarized voltage within each burst (i.e., they do not *undershoot* [111]);

(c) their mean spike frequency can be controlled;

the gHCO bursters are coupled by both:

(d) slow synapses with PSP summation whose strength increases with the growing spike frequency in presynaptic cells;

(e) fast synapses without PSP summation.

In what follows, both gHCO cells are represented by the Hodgkin-Huxley (HH) type model of the thalamic reticular neuron [112, 113], defined by the following state equations:

$$\begin{cases} \dfrac{dV}{dt} = \dfrac{-I_T - I_L - I_{Na} - I_K - I_c + I^{syn}}{C} \\ \dfrac{dCa}{dt} = -\dfrac{kI_T}{2Fd} - \dfrac{K_T Ca}{Ca + K_d} \\ \dfrac{dy}{dt} = \dfrac{y^\infty - y}{\tau_y}, \quad y = \{h, m, n, m_T, h_T\} \end{cases} \tag{1.1}$$

where V is the membrane potential of the neuron, Ca is the intracellular calcium concentration and y is a generic gating variable; the equations describing the evolution of the ion currents I_T (calcium), I_{Na} (sodium), I_K (potassium), and I_L (leakage) and all the details on the model parameter values and meaning can be found in Appendix A.

This slow-fast model with seven state variables can exhibit endogenous bursting activity of alternating trains of fast action potentials with long quiescent intervals, as depicted in Fig. 1.2. The dynamics of the i-th neuron can be summarized as

$$\frac{d}{dt}\begin{bmatrix} V_i \\ \mathbf{y}_i \end{bmatrix} = \begin{bmatrix} -\sum_k I_k + I_i^{syn} \\ f(V_i, \mathbf{y}_i) \end{bmatrix}, \quad \text{where} \;\; i = 1, 2. \tag{1.2}$$

where $f(V_i, \mathbf{y}_i)$ is a vector function describing $\mathbf{y}_i$-dynamics. The term $\sum_k I_k$ includes intracellular currents and the control current I_c, which acts essentially on the spike frequency within bursts. For the given model, bursting activity occurs when $I_c \in [-0.43,\, 0.13]\frac{\mu A}{cm^2}$, with the mean intraburst interval decreasing from 15.36 ms to 4.13 ms. The term I_i^{syn} is the incoming mixed, excitatory/inhibitory synaptic current originating from the jth cell onto the ith, post-synaptic cell:

$$I_i^{syn} = g^{ex}(E^{ex} - V_i)s_j^{ex} + g^{in}(E^{in} - V_i)s_j^{in}, \tag{1.3}$$

where $E^{ex/in}$ are the reversal potentials for excitatory/inhibitory synapses and $0 \leq s_j^{ex/in} \leq 1$ is the activation or neurotransmitter release rate of the synapse, excitatory ($V_i < E^{ex}$) or inhibitory ($V_i > E^{in}$). For the slow synapses with PSP summation a first-order dynamic synapse [114, 115, 116] is employed. The dynamic evolution of its activation rate is governed by the following equation

$$\frac{ds_j}{dt} = \alpha\,(1 - s_j)f_\infty(V_j) - \beta s_j, \quad f_\infty(V_j) = \frac{1}{1 + e^{-\nu(V_j - \theta)}}, \tag{1.4}$$

where θ is the synaptic threshold, whereas α and β are coefficients weighting the raise and decay terms, respectively. To model the static synapses without PSP summation the fast threshold modulation paradigm [117] is employed, using the sigmoidal function: $0 \leq s_j = f_\infty(V_j) \leq 1$, with θ being below the spike-level.

Fig. 1.2 illustrates the contrasting properties of these synapse models, showing the bursting voltage traces V_1 (red) and V_2 (blue) and the synaptic activation dynamics, fast $s_2^{in}(t)$ (gray) and slow $s_2^{ex}(t)$ (black) at the edge of the I_c bursting interval. Observe that the neurotransmitter release rate $s_2^{in}(t)$ of the fast FTM synapse (i) is maximized as soon as the voltage $V_2(t)$ in the pre-synaptic cell overcomes the synaptic threshold θ^{in} (indicated by the grey lines in panels a,b), (ii) remains constant regardless of the spike frequency, and (iii) vanishes with the burst termination. In contrast, the low spike frequency (panels a,c) barely activates the slow synapse (see $s_2^{ex}(t)$) that at high spike frequency (panels b,d) exhibits the profound PSP build up; the ascending rate is ruled by α, and the exponential decay due to β starts after the voltage lowers below θ.

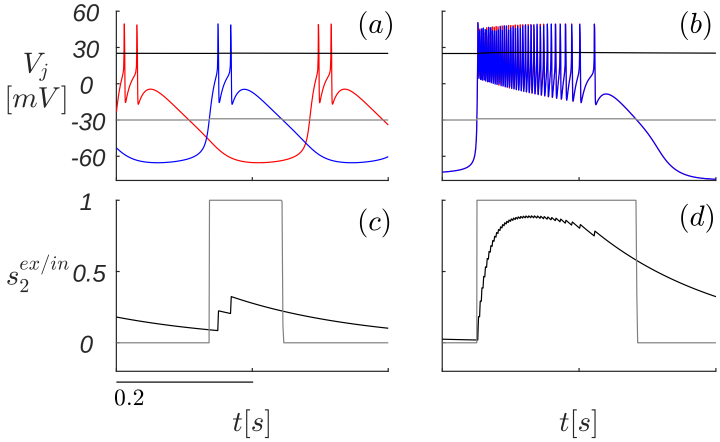


Figure 1.2: Asymptotic anti-phase (a) and synchronous (b) bursting voltage traces V_1 (red) and V_2 (blue) at $I_c = -0.43\ \frac{\mu A}{cm^2}$ and $0.13\ \frac{\mu A}{cm^2}$, resp., in gHCO (Eqs. (1.2), (1.3), (1.4)), superimposed with excitatory/inhibitory thresholds θ (horizontal lines) at 10 mV and -30 mV. (c, d) Synapse dynamics: fast modulatory $s_2^{in}(t)$ (gray) vs. slowly summating/decaying $s_2^{ex}(t)$ (black).

1.3.2 Identification of the gHCO parameters

The neuron and synapse models described by Eqs. (1.2), (1.3) and (1.4) are calibrated to physiologically plausible values to meet the above requirements (a)-(e) and to ensure a smooth and reversible transition from anti-phase to in-phase bursting occurring in the gHCO as the spike frequency changes due to I_c-variations.

To clarify things, the dynamics of the gHCO with fast FTM inhibitory and slow excitatory synapses are considered first; the corresponding synaptic thresholds are set at $\theta^{in} = -30$ mV and $\theta^{ex} = 25$ mV, respectively. As such, the inhibitory synapses without

PSP summation (de)-activate quickly and their strength remains constant during each burst regardless of the spike frequency. In contrast, the slow excitatory synapses exhibit PSP summation that becomes stronger with an increase in the spike frequency.

Fig. 1.2 shows that at the low end of the bursting region ($I_c = -0.43\ \frac{\mu A}{cm^2}$), near the transition to the hyperpolarized quiescence, the gHCO neurons oscillate in anti-phase with the smallest number of spikes per burst and lowest spike frequency (panels (a, c)), whereas on the opposite side ($I_c = 0.13\ \frac{\mu A}{cm^2}$) the neurons burst in phase with a larger number of spikes per burst and with much higher spike frequency (panels (b, d)). Changing the value of I_c changes the strength of the excitatory synapses, and hence the proportion between inhibition and excitation. The *phase lag* Δ between burst initiations in the neurons [118, 119, 120] allows to quantify the phase-locked states produced by the gHCO. The definition of Δ assumes that isolated or coupled neurons maintain relatively close temporal characteristics and each one evolves on a structurally stable periodic orbit in the state space of the corresponding model. The phase variable $\Phi_i(t)$, defined modulo 1, indicates the position on the periodic orbit of the ith neuron. Consequently, the phase lags between burst initiations in a network of two neurons can be described by the state variable $\Delta = \Phi_2 - \Phi_1$. The time evolution of this state variable, being quite complex due to nonlinear interactions, can be determined through numerical simulations, in which $\Phi_i(t)$ is reset to 0 when the voltage V_i increases above some synaptic threshold V_{th} at times $t_i^{(q)}$. The synaptic threshold is set at $V_{th} = -30$ mV and the phase lags between coupled cells are computed in a discrete set of time instants as

$$\Delta^{(q)} = \frac{t_2^{(q)} - t_1^{(q)}}{t_1^{(q)} - t_1^{(q-1)}}, \quad \text{mod } 1 \tag{1.5}$$

In the case of synchronous or in-phase bursters, $\Delta = 0$ (or, equivalently, $\Delta = 1$); when they burst in alternation, with $\Delta = 0.5$, they are in anti-phase. The intermediate values of Δ correspond to all the other possible delays between the gHCO cells.

The bifurcation analysis of the system described by Eqs. (1.2), (1.3) and (1.4) was carried out using the computational toolbox CEPAGE [121].

Since the gHCO should transition from anti-phase regime to in-phase regime varying I_c, the proportion between inhibition and excitation should be significantly different for the two values of I_c at the edges of its range. To this end, the goal is to have maximum difference in the mean values of s_i^{ex} (over one period) at the two extreme values of I_c, i.e., -0.43 $\frac{\mu A}{cm^2}$ (anti-phase pattern) and 0.13 $\frac{\mu A}{cm^2}$ (in-phase bursting).

The numerical values of θ^{ex}, α and β are identified according to this principle, running a set of simulations over a grid of parameter values: $\theta^{ex} = \{10, 25\}$ mV, 10 evenly spaced values of $\alpha \in [0.05, 1]$ kHz and 10 evenly spaced values of $\beta \in [0.005, 0.1]$ kHz. The considered values of θ^{ex} indicate voltage levels representative of two different conditions: at $\theta^{ex} = 10$ mV each spike appears broader, i.e. V_j stays above θ^{ex} for a longer time window; at $\theta^{ex} = 25$ mV each spike appears narrower, i.e. V_j stays above θ^{ex} for a shorter time period. The parameter setting that provides the maximum difference in the mean values of s_i^{ex} for the two extreme values of I_c is chosen (see Appendix A for numerical values). The synaptic conductances $g^{in/ex}$ are set to obtain anti-phase synchronization for low spike frequency, a condition in which the mean value of s_i^{ex} is minimum, and in-phase synchronization for high spike frequency, a condition in which the mean value of s_i^{ex} is maximum.

The results are summarized in Fig. 1.3, and reveal the dependence of the phase lag Δ on the I_c-current, and hence explicitly on the spike frequency within bursts. As expected, at low I_c-values between -0.43 $\frac{\mu A}{cm^2}$ and -0.40 $\frac{\mu A}{cm^2}$, the fast reciprocal inhibition within the gHCO dominates and makes its neurons burst in alternation with $\Delta = 0.5$. As the I_c-current is increased, the spike frequency raises, which in turn makes the slow excitatory synapses sum up faster and stronger on average. With larger I_c values, the reciprocal excitation gradually prevails over the reciprocal inhibition, which gives rise to the smooth onset of the stable in-phase bursting in the gHCO. This is revealed in the bifurcation diagram with a characteristic pitchfork shape of the dependence of the phase lag Δ on the I_c-current.

It has to be noted that this diagram has been obtained by making a multi-shooting for each parameter value. This is a direct indication that there is no hysteresis and therefore the absence of multi-stability or the coexistence of anti- and in-phase bursting for the same parameter values, and that the transition between activity rhythms is continuous and reversible.

Observe that the maximal synaptic conductances $g^{in/ex}$ in Eq. (1.3) once set are not changed, and the transition is solely determined by the gradual increase/decrease of the mean s_i^{ex}-value caused by the spike frequency variations in the gHCO neurons.

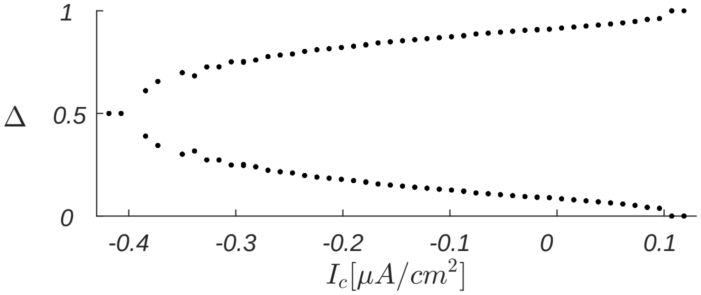


Figure 1.3: Bifurcation diagram showing how the phase lag Δ between the gHCO neurons is affected by the current I_c; here, 30 initial Δ-values were sampled evenly between 0.05 and 0.95 for each of the 50 I_c-values.

1.3.3 Constraints on neuron and synapse models are essential

This section aims to illustrate, also through counterexamples, that the defined requirements (a)-(e) on the neuron and synapse models are essential for the successful design of the proposed gHCO concept.

If the bursting condition (a) is broken, the approach is no longer applicable. Two neurons, spiking in isolation, can burst in alternation due to reciprocal inhibition, but not through reciprocal excitation, which makes both even more synchronously depolarized with a higher frequency.

If the neurons undershoot (condition (b)), which is typical for elliptical bursters [122] (see Fig. 1.4(a)), it is not feasible to set θ^{in} as previously explained to warrant evenly constant activation s_i^{in} for the duration of the burst. Indeed, this choice can result in less robust dynamics of the gHCO, due to inhibition-excitation competition (see Fig. 1.4(b)).

Condition (c), outlining the importance of being able to control spike frequency and not only burst duration of the pre-synaptic cell, is quite crucial for stable gHCO functions. To point out its significance, the exponential integrate-and-fire (eIF) neuron model [124] is employed; an external current I_{ext} primarily controls the burst duration with insignificant spike-frequency variations, as shown in Fig. 1.5(a). In this scenario, the activation of both inhibitory and excitatory synapses is mainly determined by the burst duration in the eIF-neurons, and thus I_{ext}-variations can only cause proportional changes in the average excitatory s_i^{ex}- and inhibitory s_i^{in}-values. As a result, neither inhibition nor excitation can solely dominate and produce the expected solo stable anti-phase or in-phase bursting patterns within the given I_{ext}-range, as shown in Fig. 1.5(b). Conversely, changing the parameter g_e of the eIF neuron model significantly modifies the spike frequency, and the

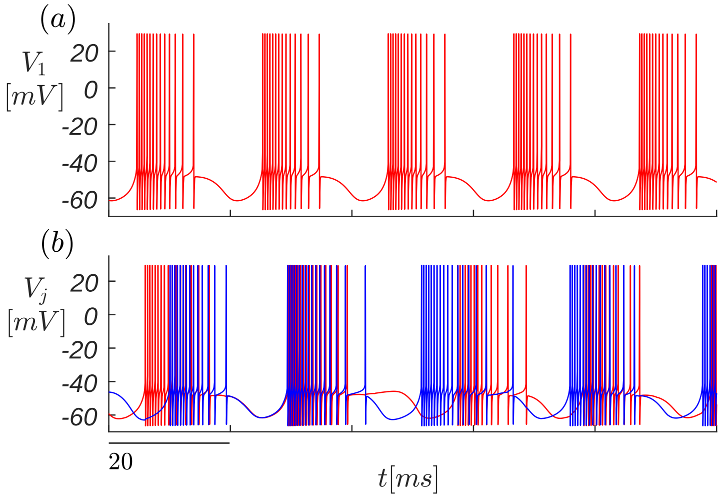


Figure 1.4: (a) Asymptotic bursting voltage trace with undershoot produced by the Plant neuron model [123, 122]. (b) Voltage traces produced by the gHCO with two coupled Plant neurons. Model equations and parameters are provided in Appendix A.

corresponding bifurcation diagram has the characteristic pitchfork shape shown in Fig. 1.6, as expected. However, the parameter g_e is a conductance, and thus is not a realistic control parameter, according to the previously discussed guidelines.

Condition (d) follows (c), as the synaptic threshold θ for the slow synapses has to be within the spike voltage range of the pre-synaptic neuron, and the dynamics is to be slow enough to allow $s_i(t)$ to grow and the synapse to exhibit PSP summation.

Condition (e) guarantees that the activation of the fast synapse does not exhibit PSP summation and hence does not change due to spike frequency variations in the pre-synaptic neuron.

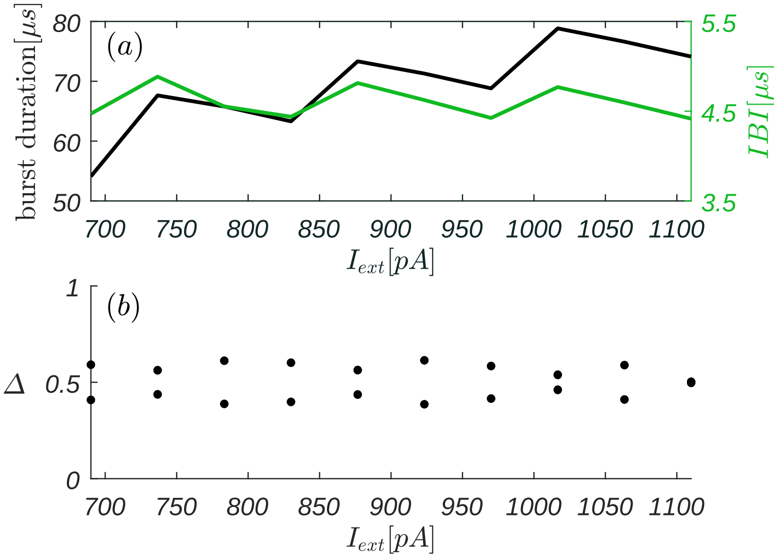


Figure 1.5: (a) Mean values (over 5 s) of the IBI (green line) and the burst duration (black line) plotted against I_{ext} for the exponential IF model [124]. (b) Corresponding bifurcation diagram for the phase lag Δ between the cells in the gHCO, in which each cell is an exponential IF-model. Model equations and parameters are provided in Appendix A.

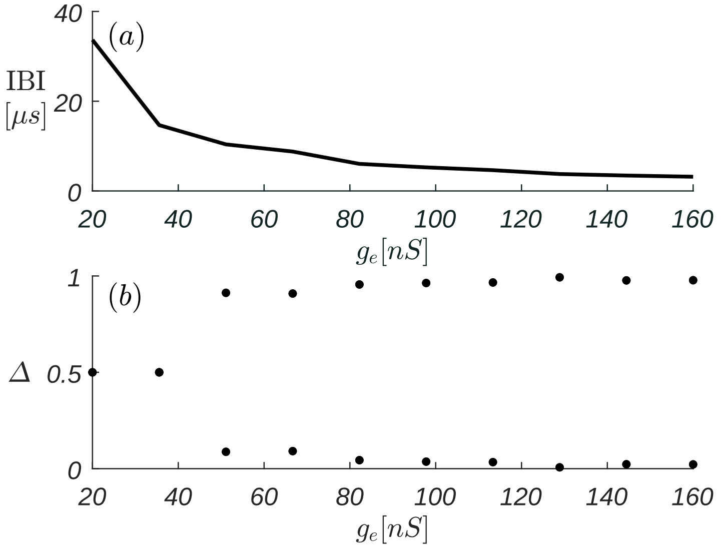


Figure 1.6: (a) Mean values (over 5 s) of the IBI plotted against g_e for the exponential IF-model [124]. (b) Corresponding bifurcation diagram for the phase lag Δ between the cells in the gHCO, in which each cell is an exponential IF model. Model equations and parameters are provided in Appendix A.

1.3.4 The gHCO as a building block for locomotion CPGs

As the objective of designing the gHCO is to use it as a building block for a quadruped CPG, both the phase lags and the burst frequency should be consistent with target quadruped gaits. For instance, in left-right alternation of the mouse locomotion, a phase lag $\Delta = 0.5$ occurs at low burst frequencies (walk and trot gaits), whereas a phase lag Δ close to 0 (or to 1, equivalently) occurs at high burst frequencies (gallop and bound gaits) [125, 126, 100]. Recall that the thalamic reticular neuron model in isolation exhibits high-frequency bursting at small I_c-values and slow bursting at greater I_c-values. Therefore, for the gHCO built with such models to produce in-phase/anti-phase synchronization at high/low burst frequencies for the desired gaits, the time scale of the synapses in its circuitry should be swapped: slow inhibitory synapses with PSP summation and FTM-fast excitatory ones without PSP summation.

1.4 A minimalistic biologically plausible CPG model

This section focuses on quadruped locomotion and explores the possibility of using the gHCO in the design of a minimalist CPG circuit that can stably produce gaits observed in four-limb animals.

The forms of locomotion that can be observed in animals vary in many ways in terms of exhibited gaits and posture, owing to the interplay of mechanical sensors/actuators (musculoskeletal system) and control (neural system). Mice, for example, can walk, trot, gallop or bound [125, 126]. This means that there must be mechanisms to change the synchronization patterns between the limbs and the limb segments during locomotion. So, it is reasonable to assume that rather than having multiple dedicated CPGs for such functions, there is a single multi-functional or multi-modular CPG circuit that is versatile to coordinate all desired motor patterns [127].

The schematic diagram in Fig. 1.7 is meant to illustrate two alternative strategies

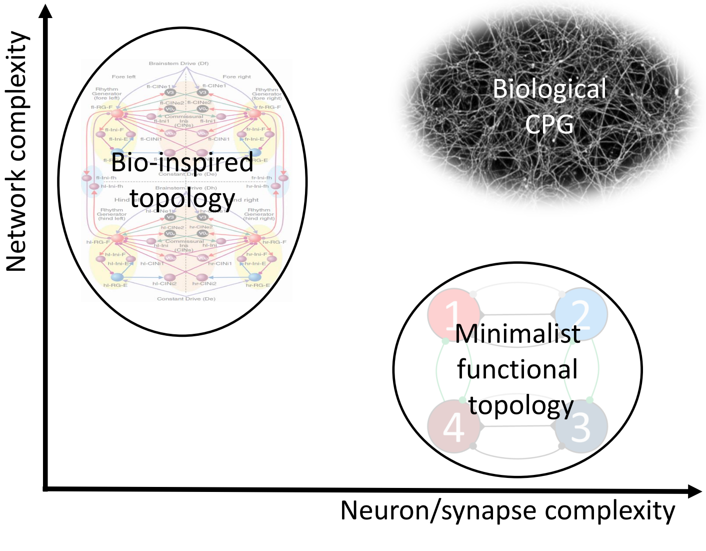


Figure 1.7: Comparison diagram of two alternatives to model a biological CPG: (left) detailed network architecture with simple functional units [128]); (right) simplest design with biologically plausible neural and synaptic models.

based on the interplay between network complexity and nonlinear dynamics of individual cells and synaptic properties. The aim is to design a CPG with the simplest network architecture (i.e., minimalist functional topology), containing only two gHCOs properly connected, by finding the parameters of cells and synapses through an optimization algorithm that accounts for physiological parameter ranges.

The design method for the novel proposed model is based on three simple cornerstones: (i) short timescale gait transitions are triggered by short-term plasticity mechanisms; (ii) the CPG model is able to reproduce basic quadruped gaits with characteristic qualities; (iii) key model parameters can be calibrated through bifurcation analysis to meet specific gait features, following the guidelines already explored in [100].

According to these premises, the proposed four-cell CPG model, where each neural unit controls the flexor regulating the swing phase of the corresponding limb, has a pair of fore neurons and a pair of hind neurons that represent essentially two gHCOs, whose cells are coupled by mixed synapses with different time scales. Firstly, table 1.1 introduces the abbreviations for the various types of synapses employed in the 4-cell CPG model. The four-cell CPG model is shown in Fig. 1.8: as in previous sections, each cell is modeled by the thalamic reticular burster. The dynamics of the membrane potential V_i and of the voltage-dependent state variables $\mathbf{y}_i$ are described in Eq. (1.2) (see Appendix A for a list of all fixed parameters and a description of their biological relevance).

The term I_i^{syn} groups together the incoming synaptic currents, and in this case is

Table 1.1: Synapse types

synapse type	abbreviation	symbol
fast excitatory	E	▶
slow inhibitory	S	●
delayed fast inhibitory	D	●
fast inhibitory	F	🟢

defined as

$$\begin{aligned} I_i^{syn} = \sum_{j=1}^{4} [g_{ij}^E (E^E - V_i(t)) s_j^E + g_{ij}^S (E^S - V_i(t)) s_j^S + \\ + g_{ij}^D (E^S - V_i(t-\delta)) s_j^D + g_{ij}^F (E^S - V_i(t)) s_j^F], \end{aligned} \tag{1.6}$$

where $E^{E/S}$ represents the reversal potentials for excitatory/inhibitory synapses, $0 \leq s_j^k(V_i, t) \leq 1$ $(k = E, S, D, F)$ describes the neurotransmitter release rate of the synapse, g_{ij}^k $(k = E, S, D, F)$ is the maximal synaptic conductance or weight, while δ is the time delay imposed for D-type synapses. The S-synapses are described by the first-order kinetic model in Eq. (1.4). The fast E, D and F-type synapses are modeled using the fast threshold modulation (FTM) approach [129, 117], $s_j^k = f_\infty^k(V_j)$ $(k = E, D, F)$.

What is the rationale behind choosing these synapses? Some of them result from the functional reduction of the detailed, 40-cell mouse locomotion CPG proposed in

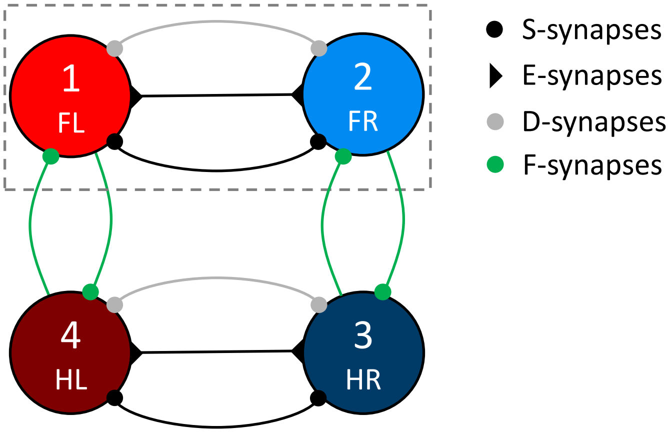


Figure 1.8: CPG circuit of four coupled cells labeled as follows: FL/FR – fore-left/right, HL/HR – hind-left/right. The dashed box includes the gHCO that governs the forelimbs. The following symbols ▶, ●, • and • denote, resp., the excitatory E-, and inhibitory S-, F- and D-synapses (see the table above). Each CPG cell controls flexor muscles regulating the swing phase of a limb, while D-synapses simulate the actions of the neural populations (not explicitly represented in the given model) controlling the extensor muscles. The fore and hind gHCOs are coupled through ipsilateral F-synapses.

[128] down to the 4-cell circuit initially introduced in [98]. In particular, the F-synapses were introduced to replace inhibitory interneuron populations that do not introduce a significant delay, and the D-synapses simulate the delayed action of the neuron groups controlling the extensor muscles. Moreover, the nonlinear interplay of both fast-excitatory and slow-inhibitory synapses is pivotal to devising this (simple yet functional and flexible) CPG network: S- and E-synapses are crucial to model the effect of short-term synaptic plasticity, as explained in the previous sections detailing the gHCO mechanism.

1.4.1 Quadruped gaits

The goal is to fit the key parameters of the proposed minimalistic 4-cell CPG so that it can generate all four mouse gaits: walk (W), trot (T), gallop (G), and bound (B). Each gait has specific quantitative features [125, 126] such as the frequency (f) and duty cycle (d) of each rhythmic pattern driving a limb and the phase lags Δ_{12}, Δ_{13}, Δ_{14} between the driving signal generated by the reference cell 1 and the ones produced by the other three cells. These features are summarized in table 1.2.

Table 1.2: Quadruped gaits features [125, 126]

gait	f [Hz]	d	Δ_{12}	Δ_{13}	Δ_{14}
walk (W)	[2 4]	<0.4	0.5	0.75	0.25
trot (T)	[4 9]	[0.4 0.51]	0.5	0	0.5
gallop (G)	[9 10]	>0.51	0.1	0.5	0.6
bound (B)	[10 12]	>0.51	0	0.5	0.5

The spatiotemporal patterns of each mouse gait are depicted in Fig. 1.9. Note again that walk and trot require the fore (and hind) cells to burst in anti-phase at low burst frequencies, whereas during bound they should become synchronized with higher burst frequencies. The duty cycle d is calculated according to a threshold V_t (grey horizontal line in Fig. 1.10) as the ratio between the time interval in which $V_i(t) > V_t$ within a period (red interval) and the period (red plus blue intervals).

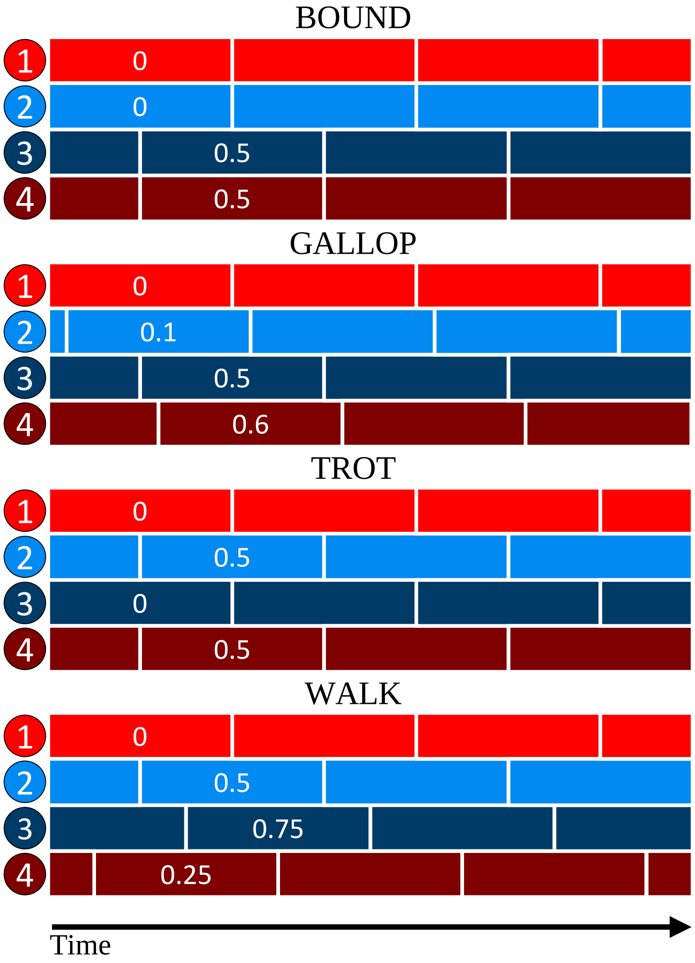


Figure 1.9: Spatio-temporal patterns of the four mouse gaits (with the colors matching the cells in Fig. 1.8): bound, gallop, trot and walk with the characteristic phase lags (as listed in table 1.2) between the reference cell 1 and the other three cells of the CPG.

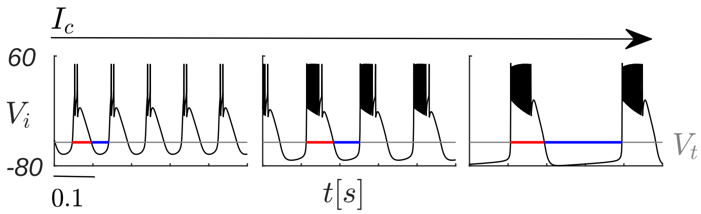


Figure 1.10: Voltage traces of the isolated burster at three different I_c-values: -0.43 $\frac{\mu A}{cm^2}$ (left panel), -0.15 $\frac{\mu A}{cm^2}$ (center), 0.13 $\frac{\mu A}{cm^2}$ (right). The threshold V_t (grey horizontal line) is used to calculate the duty cycle d as the ratio between the red interval and the period (red plus blue intervals).

1.4.2 Parameter identification strategy

The objective is to devise a CPG model that can exhibit intrinsic short-term synaptic plasticity in the form of facilitation (without any changes in synaptic weights) and is able to generate quadruped gaits and smooth gait transitions when triggered by the external drive I_c. To achieve this goal, several cellular and synaptic parameters that stably yield the four quadruped gaits with the realistic features listed in table 1.2 need to be identified. This task is challenging, mainly due to the high number of parameters and their interplay and to the constraints imposed on the overall model performance.

This section defines the strategy (summarized in Fig. 1.11) that further exploits the design methods proposed in [100] to systematically calibrate both neuronal and synaptic parameters. This strategy includes several steps discussed in detail below. In brief, the initial steps from 0 through 5 in Fig. 1.11 deal with the steady-state behaviors of the network, while possible transitions (transient dynamics) are examined in the final step 6. It should be quite clear from the beginning that, due to its high dimensionality, the complex problem of parameter setting cannot be addressed by a monolithic brute-force approach based on performing exhaustive search.

Therefore, the original problem should be hierarchically subdivided into consecutive simpler steps. However, the dynamics of the complete CPG network results from the

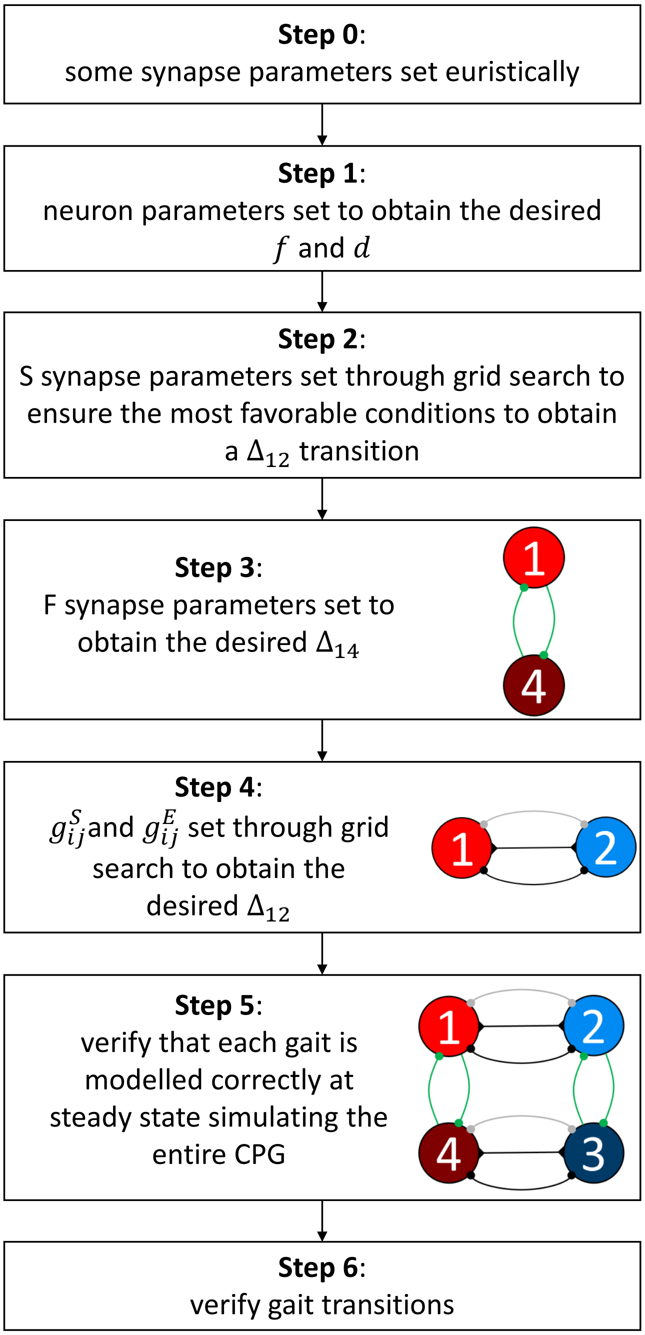

Step 0:
some synapse parameters set euristically
Step 1:
neuron parameters set to obtain the desired f and d
Step 2:
S synapse parameters set through grid search to ensure the most favorable conditions to obtain a Δ_{12} transition
Step 3:
F synapse parameters set to obtain the desired Δ_{14}
1
4
Step 4:
$g_{ij}^{S}$and g_{ij}^{E} set through grid search to obtain the desired Δ_{12}
1
2
Step 5:
verify that each gait is modelled correctly at steady state simulating the entire CPG
1
2
4
3
Step 6:
verify gait transitions

Figure 1.11: Flowchart summarising the design strategy proposed: steps 0 through 5 focus on the steady-state dynamics of the network, while step 6 verifies the transitions between gaits.

interplay of the dynamics of the four individual cells coupled by mixed synapses with different time scales. Therefore, given that the correct behavior of the complete CPG is not guaranteed *a priori*, the previously found results should not be tabulated, but revisited and verified *a posteriori*.

Step 0 – fixing the key synaptic parameters: this reduces the dimension of the set of parameters to be optimized, based on preliminary considerations (see Appendix A for the set values). This includes the time delay δ of the D-synapses, which is set to be half of the bursting period of the presynaptic neuron since it is assumed that flexor and extensor cells are activated in anti-phase. The synaptic thresholds θ^E, θ^D, θ^F of all fast synapses are set the same so that their strength does not change with spike frequency variations. The parameter ν is set the same for all synapse types. Also the weights g_{ij}^D of the delayed D-synapses, which are an order of magnitude weaker than g_{ij}^S of the S-synapses according to [100], are set all equal. In addition, following [100], the ratios g_{41}^F/g_{14}^F and g_{32}^F/g_{23}^F (i.e., the ratios of the weights of fore-to-hind and hind-to-fore coupling) are set the same.

Step 1 – single cell: its temporal scaling is performed through the parameter ξ so that the values of the burst frequency f, obtained by varying I_c in the burst region, span the range required to model all gaits as listed in table 1.2. The reference voltage V_t, used to calculate the duty cycle d, is consequently adjusted to obtain the desired values of d. The result of this step is a pair of diagrams (see Fig. 1.12) showing how f (bottom panel) and d (top panel) are controlled by the drive I_c across all four gaits, labeled as B[ound], G[allop], T[rot] and W[alk] in the color-coded intervals.

Step 2 – S-type synapses: their key parameters, exponential factors α, β and synaptic threshold θ^S, are identified through grid search to achieve maximum difference in mean synaptic activation $\bar{s}_j^S$ (evaluated over the burst period) at the endpoints of the I_c range ($[-0.43, 0.13]\ \frac{\mu A}{cm^2}$). This simple constraint ensures the most favorable conditions to obtain the transition of the phase lag Δ_{12} from in-phase to anti-phase as I_c is varied, provided that the weights g_{ij}^S and g_{ij}^E (set in step 4) are properly balanced. The result is illustrated in Fig. 1.13.

Step 3 – combined fore and hind gHCOs: they are coupled pair-wise through cells 1 and 4, and cells 2 and 3 by the inhibitory F-synapses. Consider ΔI_c as the difference between the I_c values driving the fore and hind cells: $I_{c1/2} + \Delta I_c = I_{c4/3}$, respectively. The bursting dynamics of the sub-network (cells 1 and 4 or, equivalently cells 2 and 3) are simulated over a dense grid of I_c and ΔI_c values (within the ranges determined in step 1), matching the weights g_{ij}^F (whose ratio is fixed in step 0) to identify the correct fore/hind phase lags (Δ_{14} and Δ_{23}) of the targeted gaits. The outcome of this simulation step is a set of (I_c, ΔI_c) pairs that yield the desired Δ_{14}-values for each gait within the proper I_c intervals, as shown by the color map in Fig. 1.14. With this map, a function

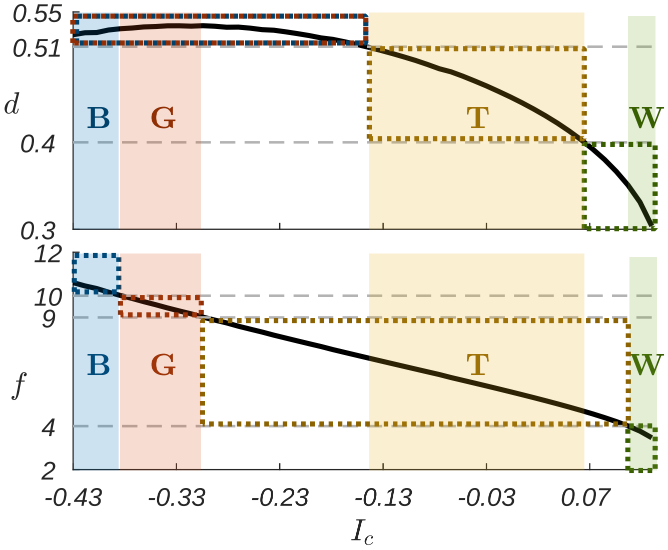


Figure 1.12: Duty-cycle d and bursting frequency f (Hz) of the neuron model (to be compared to d and f in table 1.2) against I_c ($\mu A/cm^2$). In both panels, the horizontal dashed grey lines mark the d or f ranges corresponding to each gait (see table 1.2); the dotted rectangles mark the corresponding I_c intervals for d and f. Observe that there is no threshold separating B and G duty-cycle intervals as their ranges overlap. Colored areas highlight the I_c intervals suitable (in terms of frequency and duty cycle) for each gait, as obtained from the intersection of the conditions for both features: B (blue), G (red), T (yellow), and W (green).

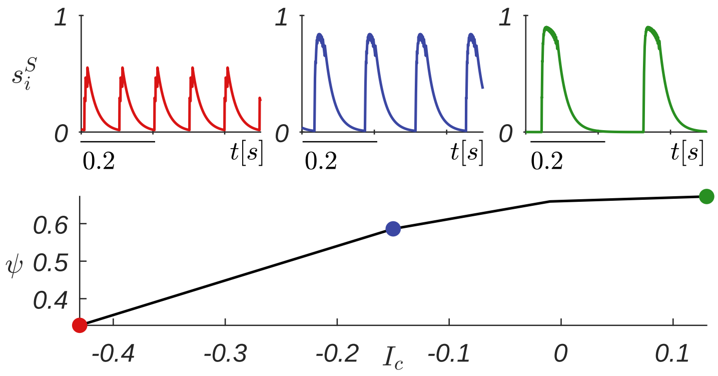


Figure 1.13: Bottom: the ratio ψ between mean inhibition/excitation influx through the S- and E-synapses, plotted against the external drive I_c ($\mu A/cm^2$). Top insets showing steady-state time plots of $s_j^S(t)$ at I_c=-0.43 I_c ($\mu A/cm^2$ (left panel), -0.15 (center) and 0.13 I_c ($\mu A/cm^2$ (right).

$\Delta I_c(I_c)$ can be defined through a piecewise-linear interpolation of the identified (I_c, ΔI_c) pairs. Such a function is used to properly drive the control action through I_c for all the CPG cells and obtain the desired Δ_{14}.

Step 4 – intra-gHCO synapses: their balanced weights g_{ij}^S and g_{ij}^E are identified through a dense grid search so that they guarantee the desired steady-state Δ_{12} phase lag for each gait within the I_c intervals (determined in step 1).

Step 5 – full CPG model: it is simulated with the parameter values determined in the previous steps to verify that, as a whole, it produces all established gaits regardless of the initial conditions of the four cells.

Step 6 – gait transitions: they are verified while I_c is varied.

1.4.3 Results of parameter identification and modeled gaits

The bifurcation analysis of the CPG model was performed using the toolbox CEPAGE [121], with the parameter values summarized in Appendix A.

Step 3 allows to identify the synaptic parameters to stably maintain the desired phase lag Δ_{14} near the target values in the bound, trot, and walk regions. It has to be noted, however, that the value of Δ_{14} for the gallop gait, which should be near 0.6, was not as accurately reproduced as the targeted phase lags for the other gaits. This is evident in Fig. 1.14, showing the piecewise-linear function $\Delta I_c(I_c)$ (in red) that yields the correct Δ_{14}-values for bound, trot, and walk. An extended ΔI_c range was also examined, finding that higher differences in the I_c values for fore and hind cells would cause self-sustained and irregular oscillations of Δ_{14} instead.

According to step 4, a grid search over g^S_{ij} and g^E_{ij} values was performed to determine which weight combinations give the desired Δ_{12} for each gait. Because modeling the gallop gait is precluded by the result of step 3, the focus is on modeling the three remaining gaits: bound, trot and walk. For each of the corresponding I_c intervals identified in step 1, a fixed I_c-value representative of the gait is examined, by prioritizing structural stability of the rhythm generation, i.e., by choosing I_c sufficiently far from adjacent intervals or from transition values. For each weight pair and identified I_c-value, the gHCOs were simulated with initial conditions close to 0 and 0.5 (resp., in-phase and anti-

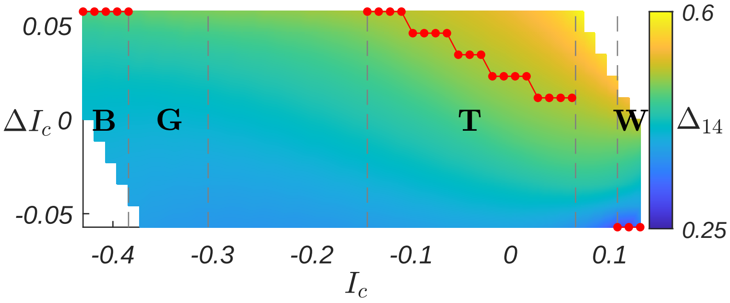


Figure 1.14: Color map of the phase lag Δ_{14} over a grid of I_c ($\mu A/cm^2$) and ΔI_c ($\mu A/cm^2$) values, for fixed g^F_{ij} (see Appendix A for numeric values). Red dots mark (I_c, ΔI_c) pairs that yield Δ_{14} values closest to the desired ones for bound, trot and walk in the proper I_c intervals. Appropriate values of Δ_{14} to model gallop are not achieved in the corresponding I_c interval. Red solid lines denote the piecewise-linear function $\Delta I_c(I_c)$ that interpolates the found (I_c, ΔI_c) pairs for bound, trot and walk.

phase states) using the following validation criteria: a g_{ij}^S-g_{ij}^E pair is considered valid if it warrants a Δ_{12} phase lag close enough to the tabulated values (see table 1.2) regardless of initial conditions. Among all such combinations, the (g_{ij}^S,g_{ij}^E) pair that corresponds to the best bifurcation diagram was selected, guaranteeing the target Δ_{12} values for the gaits, and no multistability for the broadest stretch in the I_c-intervals determined in step 1. This bifurcation diagram is shown in Fig. 1.15. It evidences that the I_c range corresponding to the bistable CPG lies almost completely outside of the parameter intervals determined in step 1, thus barely interfering with the network functionality in the trot region. On the other hand, the presence of multistability is indicative of the hysteresis associated with gait transitions, which is a well-known phenomenon in animal locomotion reported in numerous experimental studies [130, 131]. As an additional consideration, note that no weight combinations yielded the desired Δ_{12}-value for gallop, unlike other gaits, confirming the difficulty in modeling this gait.

The complete CPG was simulated next to test its rhythm generation, according to step 5. One can observe from Figs. 1.16-1.18 that the proposed CPG-circuit produces, respectively, bound, trot and walk correctly with the desired Δ_{12}, Δ_{13} and Δ_{14} phase lags.

The phase lags between all cells for all gaits are in agreement with the results obtained

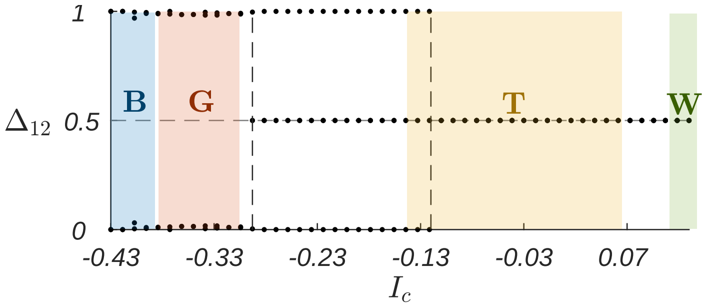


Figure 1.15: Bifurcation diagram that shows the phase lag Δ_{12} at steady-state of the cells in the gHCO against I_c ($\mu A/cm^2$). Colored rectangles highlight the I_c intervals suitable for each gait determined in step 1: B (blue), G (red), T (yellow), and W (green). The vertical dashed lines indicate the parameter range where the CPG circuit becomes bi-stable.

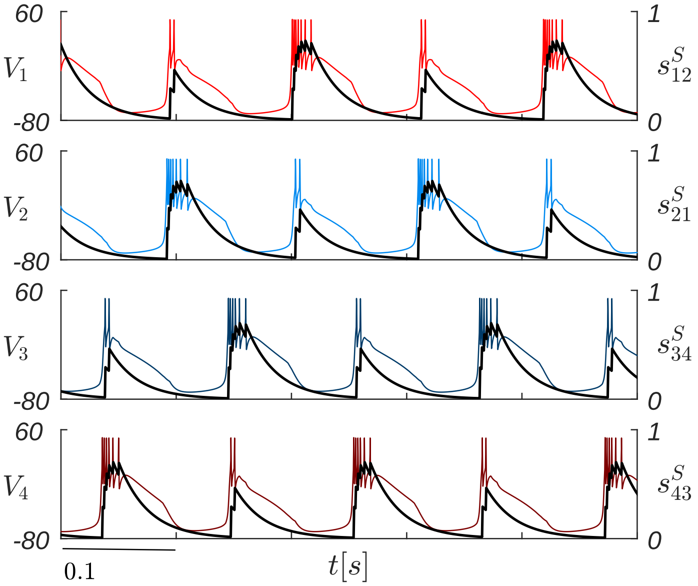


Figure 1.16: Steady-state bound. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s_i^S of the efferent S-synapse (black lines).

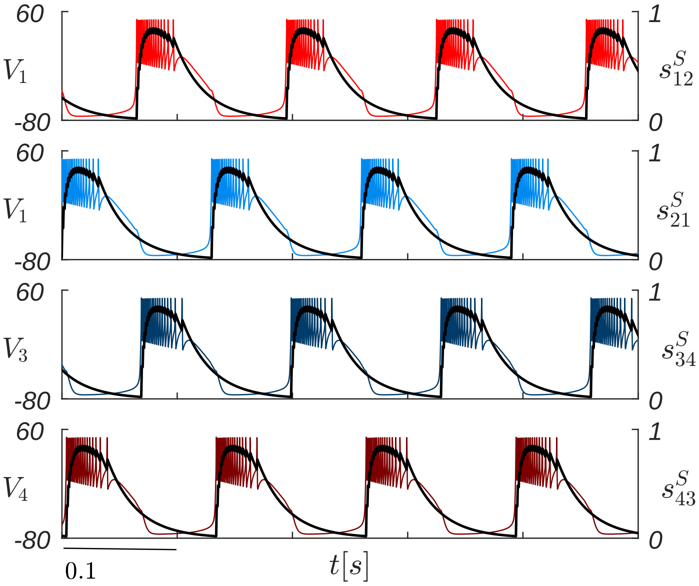


Figure 1.17: Steady-state trot. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s_i^S of the efferent S-synapse (black lines).

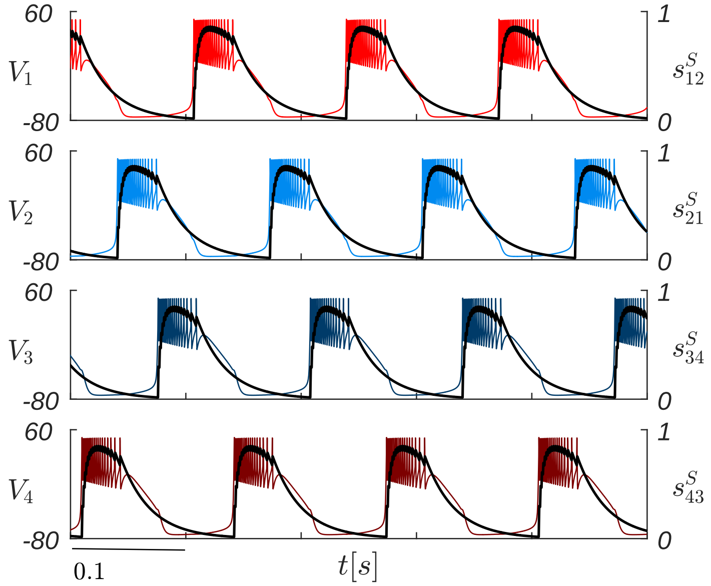


Figure 1.18: Steady-state walk. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s_i^S of the efferent S synapse (black lines).

in steps 3 and 4, confirming the reliability of the proposed design method. Note that, due to reciprocal interactions, the corresponding frequency f and duty cycle d of the network cells differ from those recorded in the isolated ones for the same I_c values. Needless to say, stronger synaptic coupling results in greater f and d deviations from the values in isolation. Note, however, that lower synaptic weights would not satisfy the requirements imposed in steps 3 and 4. The shift from the expected value of d can be at least partially corrected *a posteriori* by acting on the reference voltage V_t used to calculate d, which does not influence any other aspect in the CPG dynamics.

Lastly, according to step 6, the transient behavior of the CPG was verified by varying I_c and ΔI_c following the piece-wise linear function (shown in red) in Fig. 1.14 and observing the transitions between gaits. As can be seen in Fig. 1.19, all transitions happen promptly (in less than 2s) and smoothly. However, when transitioning from trot to walk, observe that, during a time transient in the order of tens of seconds after the transition, neurons 1 and 2 occasionally skip a burst, momentarily disrupting the

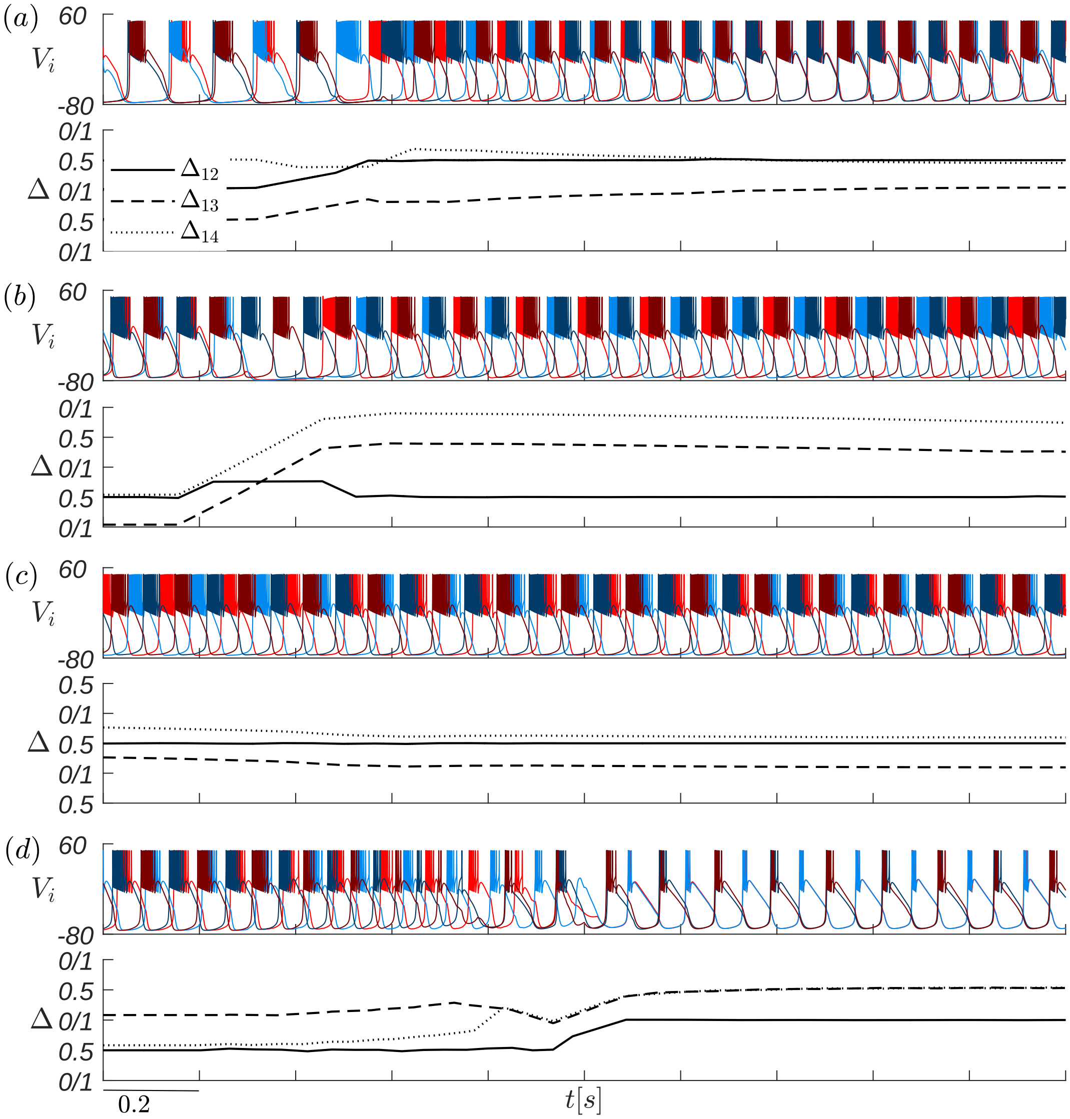


Figure 1.19: Membrane voltage V_i (i=1,2,3,4) of the four neurons (colors as in Fig. 1.8) and phase lags Δ_{1j} (j=2 solid line, j=3 dashed line, j=4 dotted line) during the transitions between bound and trot (a), trot and walk (b), walk and trot (c), trot and bound (d). The transitions are obtained by varying the values of I_c and ΔI_c following the red piecewise-linear function shown in Fig. 1.14.

alternation sequence, as shown in Fig. 1.20. To identify the cause of this behavior, the influence of inhibitory and excitatory synapses was examined separately. It was found that the same phenomenon still occurs with the E-type synapses silenced, for weights of the S-type synapses g_{ij}^S in the order of magnitude determined in step 4. Such behavior was not observed for values of g_{ij}^S decreased by more than two orders of magnitude. This again suggests that lower synaptic weights would be less problematic in terms of any unwanted behavior caused by the interplay of the different dynamics within the network.

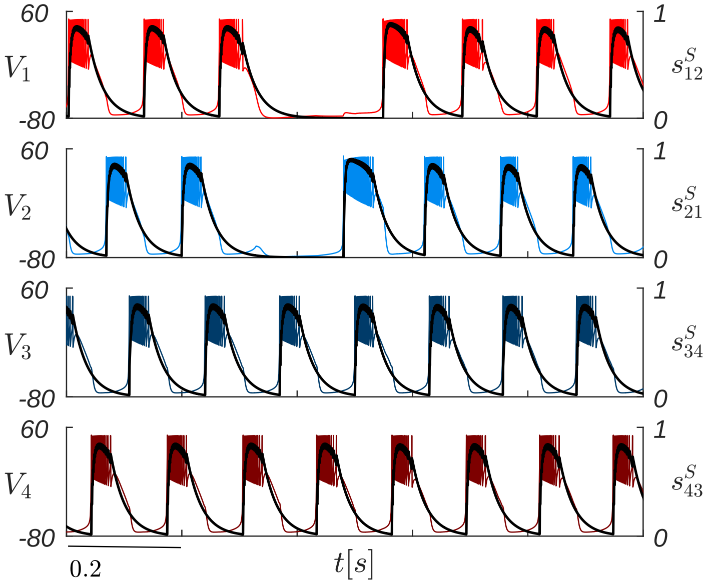


Figure 1.20: Fragment of the time interval where neurons 1 and 2 skip a burst after the transition from trot to walk. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s_i^S of the efferent S synapse (black lines).

1.4.4 Alternative synapse model

A modified version of the first-order synapse [132] was also tested to model slow inhibitory synapses. The dynamics of its activation is governed by the following equation

$$\frac{ds_j}{dt} = \alpha\, s_j\, (1 - s_j) f_\infty(V_j) - \beta s_j \tag{1.7}$$

compared to Eq. 1.4, there is an additional multiplicative s_j term that warrants a greater contrast in the mean s_i-values corresponding to the low and high ends of the bursting I_c-range for the given neuron model, a favorable property in the gHCO mechanism. The ratio ψ between mean inhibition and excitation varying I_c (obtained from step 2 of the parameter identification strategy) using this modified synapse is shown in Fig. 1.21. Note the difference with respect to Fig. 1.13: in this case, the inhibitory synapse is essentially silent for low values of I_c.

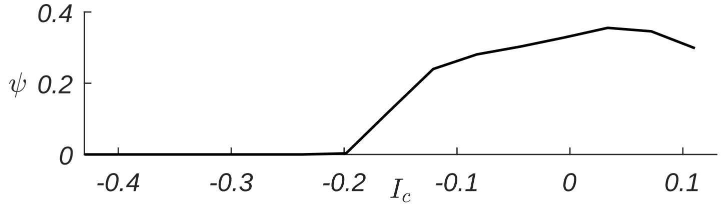


Figure 1.21: The ratio ψ between mean inhibition/excitation influx through the S- and E-synapses, plotted against the external drive I_c ($\mu A/cm^2$).

The bifurcation diagram resulting from step 4 of the parameter identification strategy (Sec. 1.4.2) is shown in Fig. 1.22. Note that, compared to the bifurcation diagram in Fig. 1.15, the CPG does not become bi-stable for any value of I_c.

The resulting gaits are summarized in Figs. 1.23, 1.24, and 1.25. Notice that, in this case, longer time intervals (1 second) are shown, to better showcase the activation sequence for the walking gait. The modified version of the first-order synapse allows obtaining a walking gait that better conforms with its characteristic frequency and duty cycle.

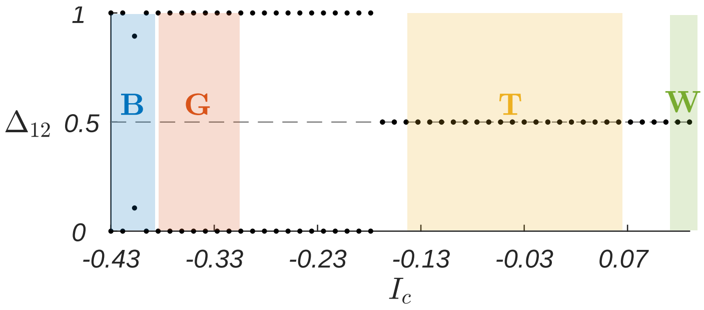


Figure 1.22: Bifurcation diagram that shows the phase lag Δ_{12} at steady-state of the cells in the gHCO against I_c ($\mu A/cm^2$). Colored rectangles highlight the I_c intervals suitable for each gait determined in step 1: B (blue), G (red), T (yellow), and W (green).

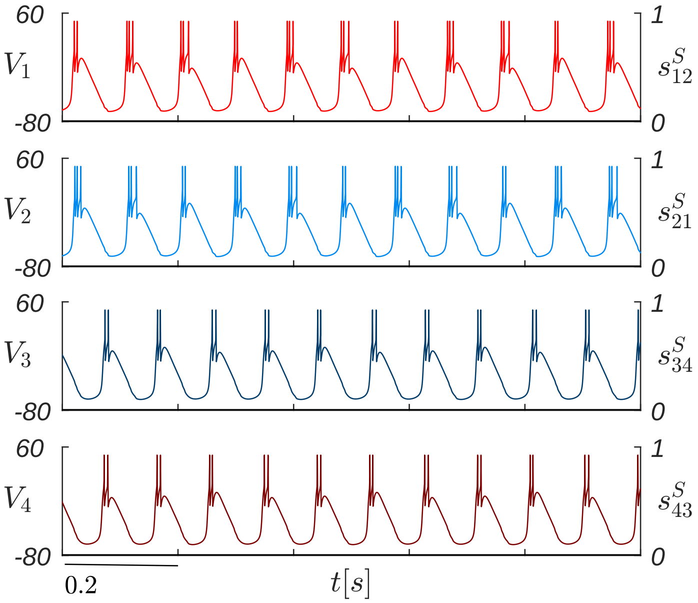


Figure 1.23: Steady-state bound. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s^S_i of the efferent S synapse (black lines).

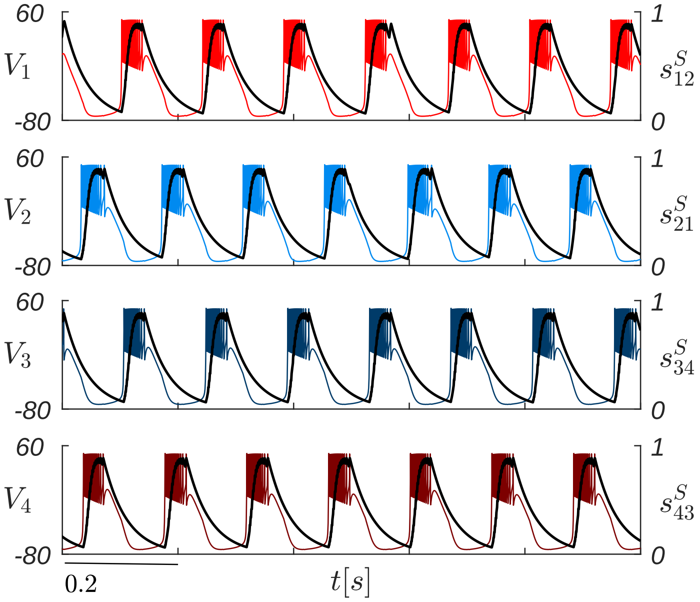


Figure 1.24: Steady-state trot. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s^S_i of the efferent S synapse (black lines).

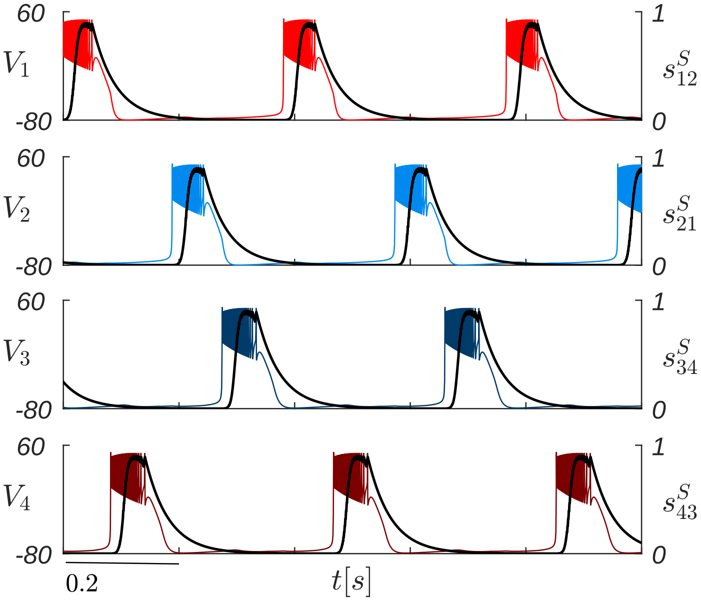


Figure 1.25: Steady-state walk. Membrane voltage V_i (mV) of each of the four neurons (colors as in Fig. 1.8) and synaptic activation s^S_i of the efferent S synapse (black lines).

1.4.5 Alternative parameter identification strategy

The strategy described in Sec. 1.4.2 is not recursive in nature. Recursion can be nonetheless integrated into this learning strategy by replacing steps 4 and 5 with steps 4B and 5B as in Fig. 1.26. Starting from null gHCO synaptic weights (g_{ij}^S=g_{ij}^E=0), in step 4B the weights g_{ij}^S (identical, to maintain the gHCO symmetry) are increased until the CPG is able to produce the desired gaits with Δ_{12} close to 0.5 (trot and walk, in the considered case) for sample values of I_c and ΔI_c representative of the gaits, regardless of initial conditions.

If the obtained CPG is not able to model the desired gaits with Δ_{12} close to 0 (bound, in this case), step 5B is performed. In step 5B the weights g_{ij}^E (identical) are increased until the CPG is able to produce the desired gaits with Δ_{12} close to 0 for sample values of I_c and ΔI_c representative of the gaits, regardless of initial conditions. If the new CPG is not able to model the desired gaits with Δ_{12} close to 0.5, step 4B is repeated. The steps are re-iterated until all gaits are modeled correctly. Since the success of the strategy is not guaranteed *a priori*, the iterations should be stopped when the weights reach unrealistically high values.

Generally, the outcomes of both the original and the alternative strategy are equivalent, since both strategies will yield a gHCO weight pair g_{ij}^S, g_{ij}^E that leads to excitation prevailing at low spike frequency (causing the neurons of the gHCO to synchronize) and inhibition prevailing at high spike frequency (causing the neurons to alternate). Steps 4B and 5B are less computationally demanding than the grid search in step 4 of the original strategy but do not span the complete parameters domain, possibly leading to a local minimum. Moreover, the grid search in step 4 can produce more than one valid weight pair, giving the possibility of selecting the most favorable pair based on additional considerations on the bifurcation diagram. For example, one could select the bifurcation diagram where no multi-stability is present for the longest stretch in the I_c intervals determined in step 1.

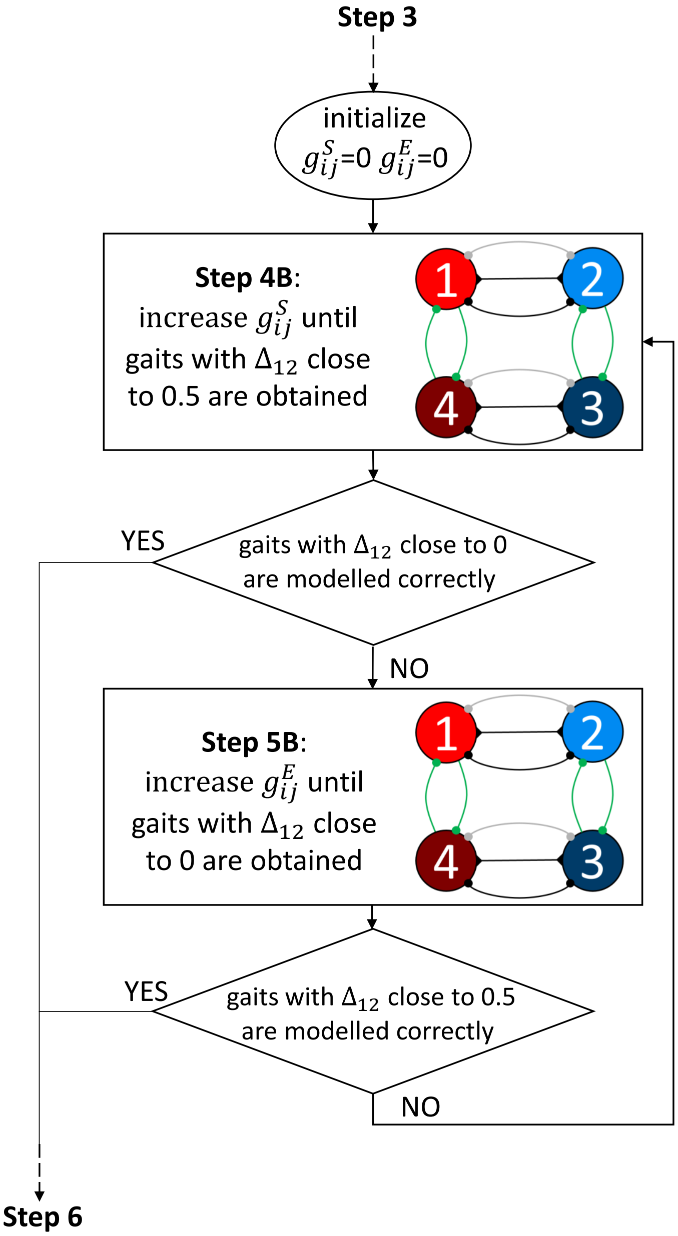

Step 3
initialize
g_{ij}^S=0 g_{ij}^E=0
Step 4B:
increase g_{ij}^S until
gaits with Δ_{12} close
to 0.5 are obtained
1
2
4
3
YES
gaits with Δ_{12} close to 0
are modelled correctly
NO
Step 5B:
increase g_{ij}^E until
gaits with Δ_{12} close
to 0 are obtained
1
2
4
3
YES
gaits with Δ_{12} close to 0.5
are modelled correctly
NO
Step 6

Figure 1.26: Recursive flowchart that summarizes the alternative steps 4B and 5B.

1.5 Parameter identification through global optimization

The custom optimization strategy proposed section 1.4.2 hierarchically subdivides the high-dimensional problem of parameter setting into consecutive simpler steps, each one giving as output the optimal subset of parameter values to be used as a footprint for the following step. However, since the dynamics of the final CPG network depends on the interplay of all model elements, a global optimization algorithm could reach parameter combinations not explored by the previously proposed strategy. The first step is to define a cost function to be minimized.

1.5.1 Cost function

It is evident that the problem of tuning the parameters of the CPG model so that it can successfully reproduce quadruped gaits cannot be codified in an analytic form. Since the goal is to obtain quadruped gaits with certain characteristics in terms of phase lags, frequency and duty cycle (see table 1.2), the cost function can be defined by comparing the target values of these characteristics with the values obtained by simulating the CPG with a certain parameter set. In particular, the cost function is defined as follows: for a candidate solution x (set of parameters), the CPG is simulated for values of the control current I_c compatible with the desired gaits, until it reaches steady-state. To ensure that transitions between gaits would be successful as well, each gait is simulated for initial conditions close to adjacent gaits. From each simulation, the membrane potentials of the four neurons V_1, V_2, V_3, V_4 are considered; if any of the membrane potentials exhibit abnormalities (i.e. anything different from regular bursting) the cost function is assigned an arbitrarily large value and subsequent simulations with the current parameter set x are not performed. Otherwise, the phase lags Δ_{12}, Δ_{13}, Δ_{14}, the frequency f and the duty cycle d are extracted.

The cost function $\epsilon(x)$ is then calculated as:

$$\epsilon(x) = \sum_{\mathcal{G}} w^{\mathcal{G}} \epsilon^{\mathcal{G}}(x) \tag{1.8}$$

with $w^{\mathcal{G}}$ weighting factor and superscript $\mathcal{G} = \{B, G, T, W\}$ indicating the gait (bound, gallop, trot, and walk, respectively). Each of the terms $\epsilon^{\mathcal{G}}$ is obtained as the sum of 3 components:

$$\epsilon^{\mathcal{G}}(x) = w_{\Delta}\epsilon^{\mathcal{G}}_{\Delta}(x) + w_f \epsilon^{\mathcal{G}}_f(x) + w_d \epsilon^{\mathcal{G}}_d(x) \tag{1.9}$$

where

- $\epsilon^{\mathcal{G}}_{\Delta}(x)$ is the mean squared difference between the values of Δ_{12}, Δ_{13} and Δ_{14} extracted from each simulation and their target values; this term is weighted by a factor w_{Δ}.

- $\epsilon^{\mathcal{G}}_f(x)$ is (i) zero if f falls within its target ranges for all simulations or (ii) the absolute value of the normalized mean difference between f and the nearest end of its target range; this term is weighted by a factor w_f.

- $\epsilon^{\mathcal{G}}_d(x)$ is (i) zero if d falls within its target ranges for all simulations or (ii) the absolute value of the normalized mean difference between d and the nearest end of its target range; this term is weighted by a factor w_d.

Defining the cost function as explained above allows weighing appropriately the different contributors: for instance, w_{Δ} should be much greater than w_f and w_d, since each gait is primarily defined by its phase lags Δ_{12}, Δ_{13} and Δ_{14}, which determine the limb movement pattern. Moreover, the *a priori* knowledge about the difficulty in reproducing the gallop gait can be taken into account by choosing w^G smaller in comparison with w^B, w^T and w^W.

The obtained cost function is thus discontinuous, non-differentiable, and highly nonlinear. Global optimization strategies suitable to solve this problem are based on particle swarm and genetic algorithms. Particle swarm optimization (PSO) [133] is a computational method that finds the global minimum of a cost function by considering a population of candidate solutions, i.e., particles; the particles move around the search space according to simple rules that regulate the particle's position and velocity, inspired by bird flocking, fish schooling, and swarming theory in particular. Each particle's movement is influenced by both its local best-known position and the global best-known positions in the search space, which are updated as the search progresses, moving the swarm toward the best solutions. PSO makes no assumptions about the problem being optimized and can search very large spaces of candidate solutions. It also does not calculate the gradient of the problem being optimized. The peculiarity and strength of PSO lie in the delicate balance between conservative testing of known regions versus risky exploration of the unknown.

Genetic algorithms (GA) [134] are an optimization method based on natural selection, the process that drives biological evolution. They involve a population of potential solutions that undergoes a series of genetic operations, such as crossover and mutation, to generate new solutions. The fitness of each solution is evaluated, and the fittest solutions are selected for "reproduction". Over successive generations, the population "evolves" toward an optimal solution.

Another aspect to take into account is that the cost function is expensive to evaluate since it requires multiple simulations of the CPG model; for this reason, surrogate optimization (SO) [135] is another candidate optimization technique, since it allows carrying out the optimization process using a small number of function evaluations. SO works by iteratively fitting a surrogate model to the cost function using data from function evaluations and then using the surrogate model to guide the optimization process.

1.5.2 Optimization results

The search for the minimum of the cost function is performed using the following Matlab functions, that implement, respectively, PSO, GA and SO algorithms:

- *particleswarm*: it implements the algorithm described in [133, 136, 137]. The algorithm initializes the positions (each corresponding to a candidate parameter set x) and velocities of the swarm's particles and repeats the following steps at each iteration: (i) it evaluates the cost function at each particle position and determines the best location, (ii) it chooses new velocities based on the current velocity, the particles' individual best locations, and the best locations of their neighbors, and (iii) it updates the particle locations.

- *ga*: it implements the algorithm described in [138]. The algorithm creates an initial population (with each individual corresponding to a candidate parameter set x) and, at each iteration, (i) it evaluates the fitness value of each individual of the current population and it creates the next generation by (ii) performing either single individual mutation or individuals—crossover on individuals with high fitness values and by (iii) sampling a small group of individuals from those with low fitness values (the so-called elite individuals).

- *surrogateopt*: it implements the algorithm described in [135]. The algorithm attempts to find the global minimum of a cost function using a small number of function evaluations. At each iteration, the algorithm alternates between two phases: (i) evaluation of the cost function at random points (with coordinates corresponding to a candidate parameter set x) and construction of the surrogate by interpolating a radial basis function through said points; (ii) search for the minimum by sampling the surrogate at random points and rating them based on their value and their distances from points where the objective function has been evaluated; the objective function is evaluated at the best candidate point and the surrogate is updated.

All synaptic weights were included in the set of parameters to be optimized, since they are fundamental in determining the CPG behavior, along with the values of ΔI_c for each gait. Therefore, $x = \{g_{12}^{S},\, g_{12}^{E},\, g_{41}^{F},\, g_{14}^{F},\, g_{12}^{D},\, \Delta I_c^{BOUND},\, \Delta I_c^{GALLOP},\, \Delta I_c^{TROT},\, \Delta I_c^{WALK}\}$, with $n_x = 9$ elements. All other parameters were set as follows, taking into account the symmetry of the CPG model:

- g^S_{21}, g^S_{34} and g^S_{43} equal to g^S_{12};
- g^E_{21}, g^E_{34} and g^E_{43} equal to g^E_{12} ;
- g^F_{32} equal to g^F_{41};
- g^F_{23} equal to g^F_{14};
- other parameters as in table A.2 of Appendix A.

The optimization algorithms were initialized with the default options suggested for $n_x = 9$: a swarm of 90 particles for *particleswarm*, a population of 200 individuals for *ga*, and a set of 20 sample points to create the surrogate function for *surrogateopt*. The explorable parameter space was constrained by imposing lower and upper bounds for each parameter in x, allowing value ranges that produce a biologically plausible behavior of the CPG model. The result of the custom optimization strategy proposed in section 1.4.2 was taken into account by setting one instance x^*_0 of the initial candidate parameter set to the obtained parameter values (listed in table A.2 of Appendix A). The remaining instances of the initial candidate parameter set, i.e. 89 particle positions for *particleswarm*, 199 individuals for *ga*, and 19 sample points' coordinates for *surrogateopt*, were set randomly within the bounds. Cost function weights w^G and w_Δ were set to 0.01 and 200, respectively, while all other cost function weights were set to 1.

All algorithms were able to find parameter sets x which yielded values of the cost function $\epsilon(x)$ lower than $\epsilon(x^*_0)$, with GA reaching the lowest value, as summarized in table 1.3.

Table 1.3: Values of the cost function $\epsilon(x)$ as resulting from each optimization strategy

	custom optimization strategy	PS	GA	SO
$\epsilon(x)$	4.0932	3.4907	1.8053	2.9670

At the same time, none of the parameter sets x found by the global optimization algorithms drifted significantly from $\epsilon(x_0^*)$, as illustrated in Fig. 1.27. This suggests that there are no regions of the parameter space distinct from the one identified by the custom optimization strategy where the gaits are correctly reproduced. The global optimization algorithms, however, can further refine the parameter set x, reaching lower values of $\epsilon(x)$.

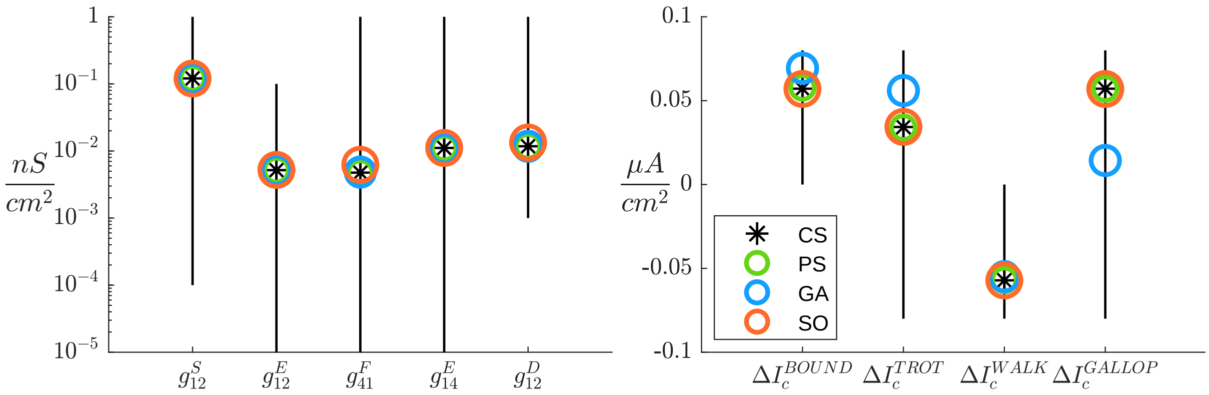


Figure 1.27: Parameter sets x found by the custom optimization strategy (CS, black stars), the PS optimization (green circles), the GA optimization (blue circles), and the SO algorithm (orange circles). The black lines indicate the range between the lower and the upper bounds constraining the explorable parameter space.

Fig. 1.28 shows the impact of this refinement on the obtained gaits. None of the parameter sets found by the algorithms produced an acceptable gallop gait, which is thus not shown. The lower values of $\epsilon(x)$ reached by the global optimization algorithms, however, come at the cost of a significantly higher computational load. Considering the running time for the custom optimization strategy T_{CS}[1], the running time for the global optimization algorithms is, in increasing order: $3.53 \cdot T_{CS}$ for the PS optimization, $7.61 \cdot T_{CS}$ for the SO algorithm and $24.50 \cdot T_{CS}$ for the GA optimization. It has to be noted that these values are only indicative of the computational cost of the specific runs, and

[1] 9 hours and 38 minutes on Intel®Xenon(R) CPU E5-1620 v2 @ 3.70Ghz x 8

give a general idea of the different orders of magnitude. Even when maintaining all other conditions unaltered, both the run time and the performance of the global optimization algorithms can be impacted by the initialization of the particles' positions, population individuals, or points' coordinates, and likely worsened if the instance x_0^* of the initial candidate parameter set is not known *a priori*. The high computational cost, however, precludes a systematic analysis over multiple runs.

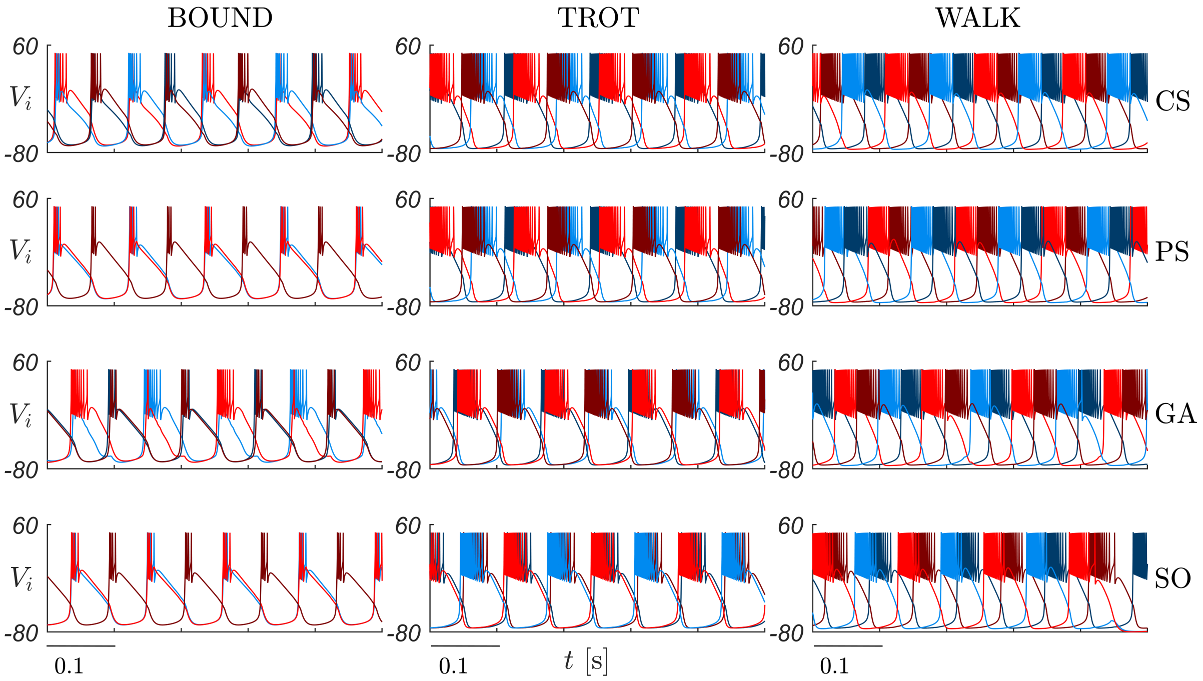


Figure 1.28: Gaits obtained by simulating the CPG with the parameter sets x found by the custom optimization strategy (CS), the PS optimization, the GA optimization, and the SO algorithm.

1.6 Concluding remarks

The developed gHCO model with a short-term plasticity mechanism is able to account for short-timescale gait transitions induced by sensory input or cognitive tasks. In this model, spike-frequency dependent synapses can themselves dynamically control the rhythmic outcomes of the gHCO from in-phase bursting to anti-phase bursting and vice versa, without changing time constants and conductances.

The gHCO concept led to the formulation of a design strategy to devise CPG models according to a minimalist-functional-topology approach (see Fig. 1.7), using a simple network structure while focusing on realistic modeling of nonlinear cellular and synaptic dynamics to account for short-term synaptic plasticity. The resulting CPG model can reproduce the sought locomotion patterns of the mouse rather well and could have potential in robotics applications. Indeed, engineering modern mobile robots requires knowledge of a section of the theoretical biology that is related to the dynamics of CPGs [74, 75]. Generally speaking, the choice between a bio-inspired-topology and a minimalist-functional-topology approach (or an intermediate recipe with a different mix of levels of abstraction for network structure and elements) is driven by the specific application or designer interest.

The proposed design strategy relies on tuning the parameters of a model with fixed topology through extensive simulations and optimizations. The proposed custom strategy hierarchically subdivides the high-dimensional problem of parameter setting into consecutive simpler steps, each one giving as output the optimal subset of parameter values to be used as a footprint for the following step. Dividing the problem into steps, not only makes it approachable in terms of dimensionality reduction, but also gives insight into the role of each sub-system and its interplay with other model elements. The success of each step depends of course on the neuron and synapse models' adequacy in fulfilling the requirements for the target gaits. Notice that, however, the dynamics of the final CPG network depends on the interplay of all model elements. For this reason, different global optimization methods have also been tested for tuning the model parameters; the results suggest that the custom optimization strategy is efficient in finding the region of the parameter space where the gaits are correctly reproduced and that the strategy does not overlook other suitable parameter space regions. The global optimization algorithms were able to refine the set of parameters, given *a priori* knowledge about the result of the custom optimization strategy, at the cost of a much higher running time.

When the parameters are properly tuned, the proposed bio-inspired 4-cell CPG is indeed able to model gait transitions in a biologically plausible way, through varying external currents only, while synaptic conductances and time constants remain fixed. Following the proposed design strategy, three desired gaits (bound, trot, and walk) out of the four standard mouse gaits were reproduced correctly, as well as smooth and prompt transitions between them. To remedy the inability of modeling gallop, which confirms the difficulty in obtaining the corresponding phase lag Δ_{12} using fast excitatory synapses and slow inhibitory synapses in the given CPG, one option would be to consider a network configuration where each cell is composed of a spiking and a bursting neuron to factually decouple the intra-burst spike frequency and the burst frequency of the cell. Then, fast inhibitory synapses and slow excitatory synapses in such a CPG could provide the desired phase lag Δ_{12} more affordably, as suggested by the results discussed in section 1.3 (see, in particular, Fig. 1.3).

Chapter 2

Brain Networks

Personal contribution — The entire chapter can be considered as original. Nonetheless, known findings, models and algorithms are utilized throughout all sections to frame and build the proposed method, which aims to derive connection weights between brain areas that are compatible with both the structural and the functional connectivity observed experimentally.

2.1 Bridging functional and anatomical neural connectivity through nonlinear dynamics

The Holy Grail of computational neuroscience is to derive reliable models that, under specific conditions, can reproduce experimentally observed human brain functions, as witnessed by many current joint international efforts in this direction [40, 41, 42]. One of the pillars of the research in this field is the growing availability of neuroimaging techniques, which allow the detection of patterns of activity across neural units that encode objects, concepts, or states of information [139]. Various techniques based on imaging data have been developed to extract functional and structural connectivity matrices of the human/animal cortex, but the way brain function is shaped by the underlying anatomical substrate is far from understood [140]. Moreover, anatomical connectomes estimated from experimental data – e.g., from diffusion magnetic resonance imaging, dMRI – are always subject to a degree of uncertainty. A second research path that is attracting a growing interest is how to employ network theory and nonlinear dynamics to build brain models based on imaging data [141, 142, 143, 144, 145, 146, 147, 148].

The orchestrated activity of neural populations has been postulated to be one of the key mechanisms underlying brain functions [149]. When synchronization is observed between brain areas, it is approximate, due to the heterogeneity of physical parameters and to the unavoidable presence of noise. At a macroscopic level, we see the emergence of regular oscillations of the local field potential (LFP) in specific brain areas, including one or more populations, such as in the case of the cortical rhythms in the cerebral cortex [150]. One of the most common ways to non-invasively detect the presence of coherence among brain areas is to analyze the correlation between the blood oxygen level-dependent (BOLD) signals measured in these areas through functional magnetic resonance imaging (fMRI).

This chapter proposes a method to reconcile structural connectivity matrix data, derived from dMRI, and synchronous clusters data, that arise from fMRI.

2.1.1 Summary of the proposed method

The approach is to model the *topology* and the *dynamics* from which the resting activity patterns of the cortical network emerge. In particular, with regards to the topology, the level of uncertainty on the anatomical connectome (dMRI data) of a cortical network is reduced by leveraging both fMRI data and tools borrowed from the theories of nonlinear dynamics and complex networks; the original network topology is modified to make it compatible with the existence of the synchronous clusters observed experimentally. With regards to the dynamics, each area is modeled as a Wilson-Cowan oscillator [151], which has been shown to be reasonably realistic to simulate cortical dynamics [152]; the node dynamics are analyzed to ensure the stability of the synchronous clusters. Moreover, the corresponding BOLD signal is modeled through a nonlinear dynamical system [153, 141], in order to get macroscopic data that can be directly compared with the fMRI measurements in terms of cluster composition. The main novelty element is the connection established between anatomical and functional data, based on the study of cluster synchronization through the Master Stability Function (MSF) approach [154, 155]. Nonlinear dynamics provides tools for studying exact cluster synchronization; when applying these tools to experimental data, for which synchronization is only approximate, it is necessary to introduce the simplifying assumption of homogeneous nodes to derive meaningful results. The heterogeneity will be reintroduced in the final (optimized) model in the form of noise and mismatched node parameters, to both increase its degree of accuracy and verify the robustness of the obtained results. Hence, data-driven (the original connectivity matrices derived from MRI data) and model-based (the dynamical network model) approaches are used complementary.

Figure 2.1 illustrates the main elements of the proposed method, which starts from functional and structural connectomes derived from fMRI and dMRI scans, respectively. The functional data evidences the presence of approximate synchronization (namely, high correlation) between the neural activity of specific brain areas. The areas with high correlation are assumed to be part of the same *synchronous cluster*. The objective is to obtain network models with different granularity in terms of clusters, that are compatible with the clusters evidenced by the functional connectivity analysis, in order to reduce the high uncertainty on the weights of the structural connectome. For each resolution level, this is done as follows:

- node dynamics is added to the topology defined by the structural connectome through a neural mass model [43]. The method proposed remains valid independent of the particular choice of the neural mass model and of its parameters.

- The network is chosen to be homogeneous in the nodes to focus the analysis on exact cluster synchronization.

- The concept of equitable clusters [156, 157] and the MSF approach [154] are used to optimize the weights of the structural connectome with the aim of making the network converge to the clusters of highly functionally-correlated areas; this is done by changing the weights as little as possible from their experimentally measured values, also taking into account their uncertainty.

At this point, one can choose the model with the most suitable resolution level, according to the specific objective, and establish approximate synchronization by reintroducing

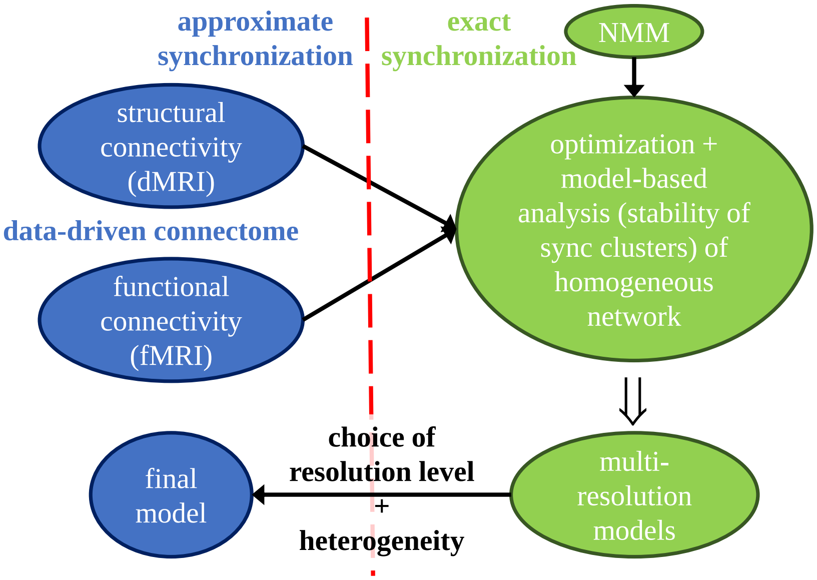

approximate synchronization
exact synchronization
NMM
structural connectivity (dMRI)
data-driven connectome
functional connectivity (fMRI)
optimization + model-based analysis (stability of sync clusters) of homogeneous network
final model
choice of resolution level
+
heterogeneity
multi-resolution models

Figure 2.1: Summary of the proposed method. It starts (top-left ovals) from data-driven structural and functional connectomes, derived from dMRI and fMRI scans. It uses a model-based method (top-right ovals) to bridge functional and structural connectivity. The reference (homogeneous) network model has an initial topology defined by the structural connectome and node dynamics imposed by a neural mass model (NMM). It employs a multi-resolution approach, by analyzing cluster partitions at different granularities (bottom-right oval). For each resolution level, the uncertainty on the connection weights is reduced by optimizing them to enforce the existence of the corresponding cluster synchronous solution. One resolution level is chosen, and the corresponding model is made heterogeneous, thus obtaining again approximate synchronization (bottom-left oval).

heterogeneity either on the connection weights or in the nodes. This leads to the final network model, in which both functional and anatomical connectivity are incorporated. The model should be able to reproduce the correlation in the BOLD signals observed experimentally in fMRI data (i.e., the approximate synchronization between brain areas) with connection weights that differ from the dMRI connectivity matrix by an extent comparable with the uncertainty introduced by the measuring process.

In the following sections, the method steps are detailed through a case study.

2.1.2 Topology and dynamics

The method is applied to functional and structural connectomes obtained from 10 fMRI and 10 dMRI scans, executed across a month, of one healthy adult at resting state, taken from the Neurodata MRI Cloud database [158]. A weighted and undirected network graph with $N = 48$ nodes is considered (see Fig. 2.2), each representing a cortical area according to the Harvard-Oxford Cortical Structural Atlas (RRID:SCR_001476) (see Appendix B for the complete list of node names and numbering). Edges represent long-range connections between the cortical areas. This graph can be described by a symmetric structural connectivity matrix A_0, whose entries $a_{0_{ij}}$ are the connection weights. A_0 is built by normalizing and averaging the 10 structural connectomes, since anatomical connectivity at this scale does not vary significantly in a subject in short time periods [159, 160]. Information on the uncertainty of each weight is stored in a matrix Σ_{A_0}, where each entry is the variance of the weight among the 10 normalized connectomes.

By assigning dynamics to each node, the network of N coupled neural oscillators is described by the following general set of equations ($i, j = 1, \dots, N$):

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{F}(\boldsymbol{x}_i(t)) + \boldsymbol{\Gamma}\left(\boldsymbol{x}_i, \sigma \sum_j a_{ij} \boldsymbol{G}(\boldsymbol{x}_j(t))\right) \tag{2.1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^m$ are the state variables of node $i = 1, \dots, N$, $\boldsymbol{F} : \mathbb{R}^m \to \mathbb{R}^m$ describes the dynamics of an isolated node and $\boldsymbol{\Gamma} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ describes the coupling between

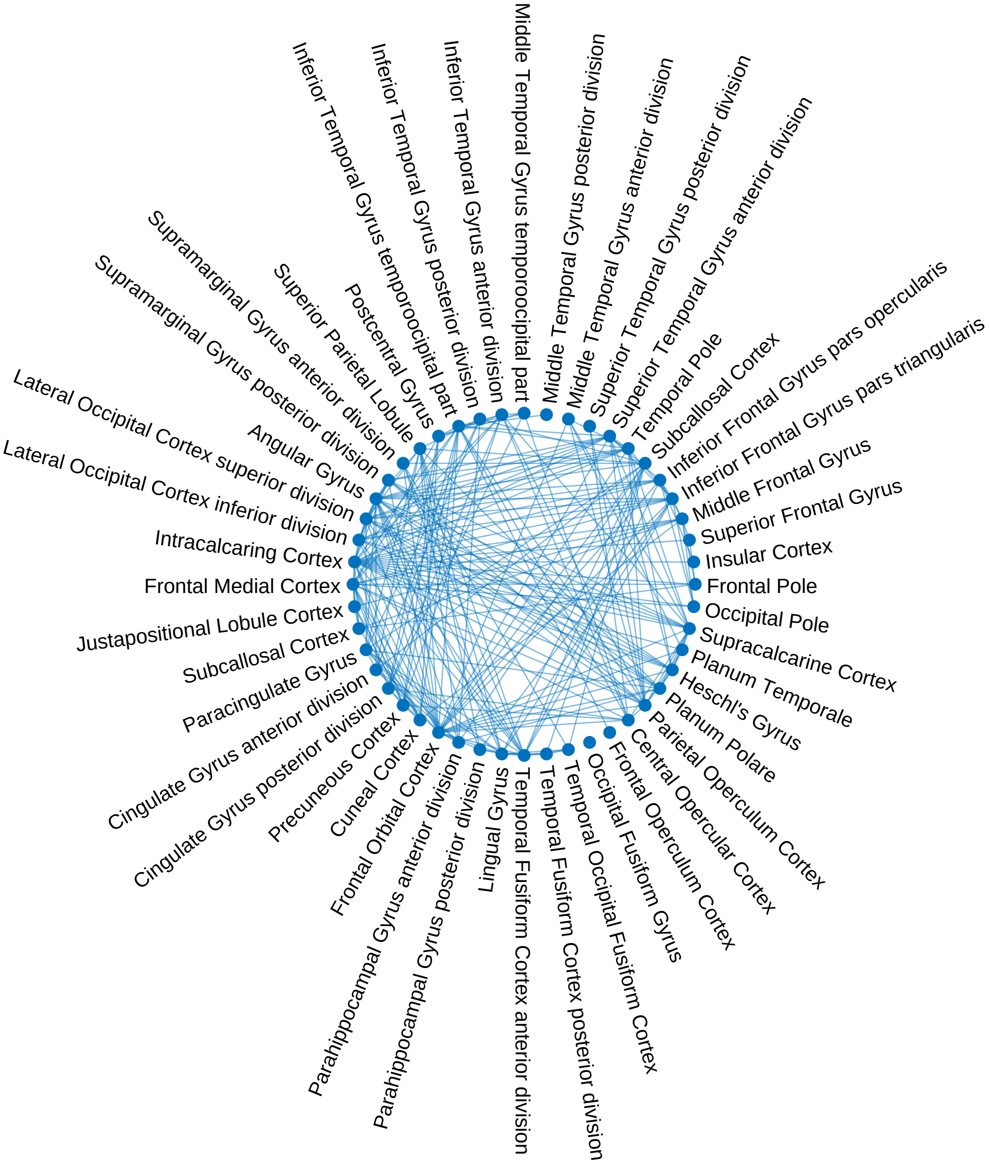

Middle Temporal Gyrus temporooccipital part
Middle Temporal Gyrus posterior division
Middle Temporal Gyrus anterior division
Superior Temporal Gyrus posterior division
Superior Temporal Gyrus anterior division
Temporal Pole
Subcallosal Cortex
Inferior Frontal Gyrus pars opercularis
Inferior Frontal Gyrus pars triangularis
Middle Frontal Gyrus
Superior Frontal Gyrus
Insular Cortex
Frontal Pole
Occipital Pole
Supracalcarine Cortex
Planum Temporale
Heschl's Gyrus
Planum Polare
Parietal Operculum Cortex
Central Opercular Cortex
Frontal Operculum Cortex
Occipital Fusiform Gyrus
Temporal Occipital Fusiform Cortex
Temporal Fusiform Cortex posterior division
Temporal Fusiform Cortex anterior division
Lingual Gyrus
Parahippocampal Gyrus posterior division
Parahippocampal Gyrus anterior division
Frontal Orbital Cortex
Cuneal Cortex
Precuneous Cortex
Cingulate Gyrus posterior division
Cingulate Gyrus anterior division
Paracingulate Gyrus
Subcallosal Cortex
Justapositional Lobule Cortex
Frontal Medial Cortex
Intracalcaring Cortex
Lateral Occipital Cortex inferior division
Lateral Occipital Cortex superior division
Angular Gyrus
Supramarginal Gyrus posterior division
Supramarginal Gyrus anterior division
Superior Parietal Lobule
Postcentral Gyrus
Inferior Temporal Gyrus temporooccipital part
Inferior Temporal Gyrus posterior division
Inferior Temporal Gyrus anterior division

Figure 2.2: Scheme of the network graph described by the structural connectivity matrix A_0; for clarity, only the edges corresponding to the highest weights (first quartile) are displayed; nodes are named according to the Harvard-Oxford Cortical Structural Atlas (see Appendix B for node numbering).

nodes, where a_{ij} is the strength of the coupling from node j to node i. The parameter σ controls the overall strength of the connections.

The functional connectomes refer to the same 48 brain areas and are given in the form of correlation matrices, whose entries describe the pair-wise temporal correlation between the activity of two brain areas.

2.2 Hierarchical clustering

From each of the 10 functional connectomes, given in the form of correlation matrices, it is possible to identify the clusters of nodes that exhibit coherent activity with a hierarchical clustering approach, as commonly done in the literature [161, 162, 163, 164]. Given a correlation matrix X, the corresponding dissimilarity matrix can be defined as $D = \mathbf{1}_N - X$ (where $\mathbf{1}_N$ is an $N \times N$ matrix with all entries equal to 1). D is converted to vector form with Matlab's *squareform* function and fed to Matlab's *linkage* function with the *'complete'* option (which implements the farthest neighbor method) to perform the hierarchical clustering.

The result can be visualized as a dendrogram, an example of which is shown in Fig. 2.3. Depending on the level ℓ at which the dendrogram is cut horizontally, the corresponding clustering is described by the sub-trees originating from each branch cut by the horizontal line: for example, in panel b the red line cuts the dendrogram at the level corresponding to 13 clusters. A certain variability across the dendrograms obtained from different fMRI sessions is expected. A set of clustering configurations, describing the functional connectivity of the subject at different granularities, is identified by selecting a set of levels $\mathcal{L}^* = \{\ell_k^*\}$ of the dendrogram derived from a chosen correlation matrix X^*, each corresponding to k clusters.

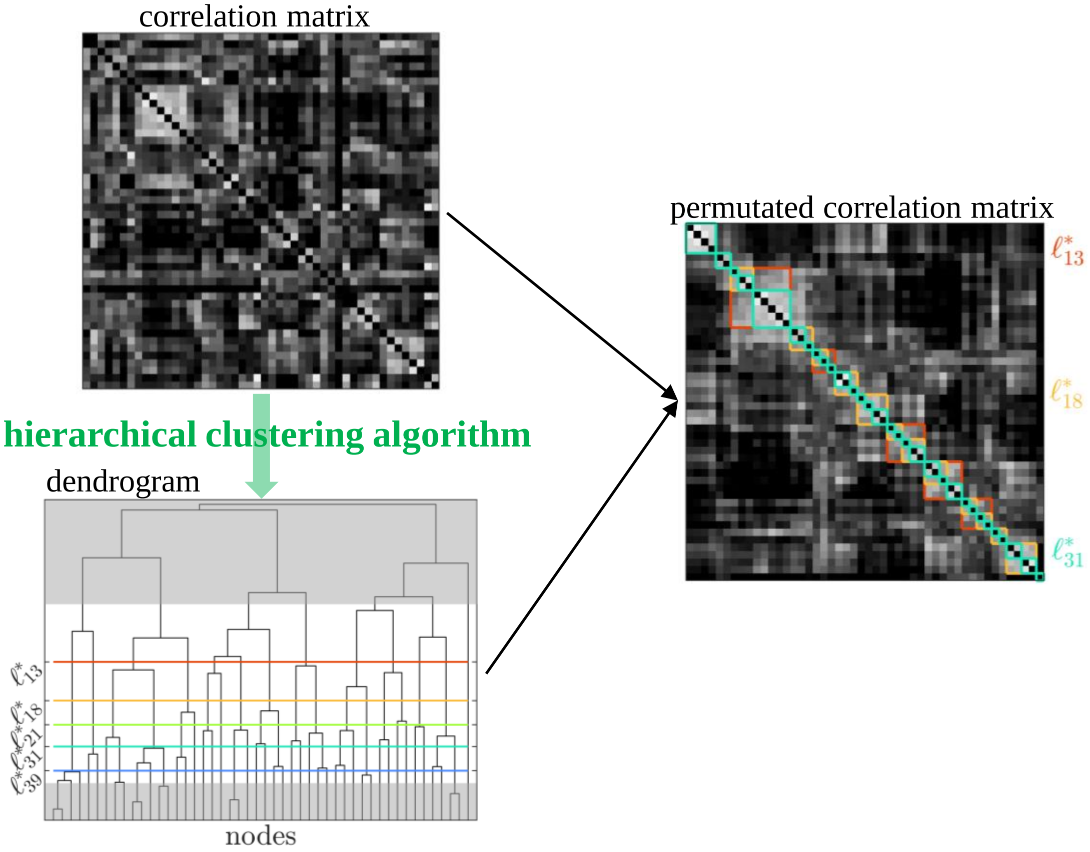


Figure 2.3: Given a correlation matrix, it is possible to identify the clusters of nodes that exhibit coherent activity with a hierarchical clustering approach. The obtained hierarchical clustering can be visualized in the form of a dendrogram, where nodes appear on the horizontal axis according to a permutation that places nodes belonging to the same cluster adjacent to one another. Depending on the level ℓ at which the dendrogram is cut horizontally, the corresponding clustering is described by the sub-trees originating from each branch cut by the line: for example, the red line cuts the dendrogram at the level corresponding to 13 clusters. The correlation matrix can be re-arranged based on the obtained permutation so that the clusters can be evidenced along the diagonal. This figure shows the dendrogram associated with X^*. The levels ℓ^*_k are marked by colored horizontal lines. Grey areas correspond to trivial partitions with too few or too many clusters (see also panel a of Fig. 2.4); the sets of clusters corresponding to levels ℓ^*_{13} (red), ℓ^*_{18} (orange), and ℓ^*_{31} (teal) are highlighted on the permutated correlation matrix.

2.2.1 Identification of the target partition

For each level ℓ of the dendrograms obtained from the hierarchical clustering of the correlation matrices, a cumulative (across sessions) similarity index Ψ_1 is calculated based on the Fowlkes and Mallows comparison measure B [165], thus obtaining the curve $\Psi_1(\ell)$ shown in panel a of Fig. 2.4. Specific levels ℓ_k^* (corresponding to k clusters) are selected as local maxima of this curve (colored dots), discarding local maxima that are not in ascending order and trivial partitions with too few or too many clusters (grey areas). The aim of this selection is to describe with the best accuracy, but at different scales, the functional connectivity of the subject, based on the consistency of the corresponding clustering across the 10 fMRI sessions.

The clusters corresponding to each level ℓ_k^* are established by computing a second cumulative (across the selected levels) similarity index Ψ_2, shown in panel b of Fig. 2.4. Ψ_2 gives an indication of how much the clustering obtained from a session is similar to the clusterings from all other sessions. Therefore, the fMRI session corresponding to the maximum value of Ψ_2 is chosen as the most representative of the subject. In the illustrating example, session 10 has the highest Ψ_2 (red dot). The selected correlation matrix is referred to as X^*.

The Fowlkes and Mallows method [165] allows to compare two hierarchical partitions by calculating the comparison measure B, which ranges from 0 (no matching between the two partitions) to 1 (perfect matching between the two partitions). This comparison measure is the key element to compute the two similarity indices, Ψ_1 and Ψ_2. Ψ_1 is calculated for each level ℓ as

$$\Psi_1 = \frac{1}{N_s} \sum_{i=1}^{N_s} B_i \tag{2.2}$$

where B_i is the comparison measure between a single pair of functional connectivity matrices (corresponding to different fMRI sessions on the same subject) and $N_s = \sum_{k=1}^{n_s-1} k$ is the total number of possible pairwise comparisons between the sessions. In the illustrating example, there are $n_s = 10$ sessions, so $N_s = 45$.

The similarity index Ψ_2 is calculated for each session s as

$$\Psi_2 = \frac{1}{n_\ell} \sum_{\ell \in \mathcal{L}^*} \frac{1}{n_s} \sum_{\substack{k=1 \\ k \neq s}}^{n_s} B_k \tag{2.3}$$

with B_k comparison measure between session s and session k and n_ℓ number of elements in $\mathcal{L}^*$. For the illustrating example, $\mathcal{L}^* = \{\ell^*_{13}, \ell^*_{18}, \ell^*_{21}, \ell^*_{31}, \ell^*_{39}\}$, i.e. $k \in \{13, 18, 21, 31, 39\}$. The level ℓ^*_{13} (ℓ^*_{39}) corresponds to the coarsest (finest) scale of description of the brain areas' correlations.

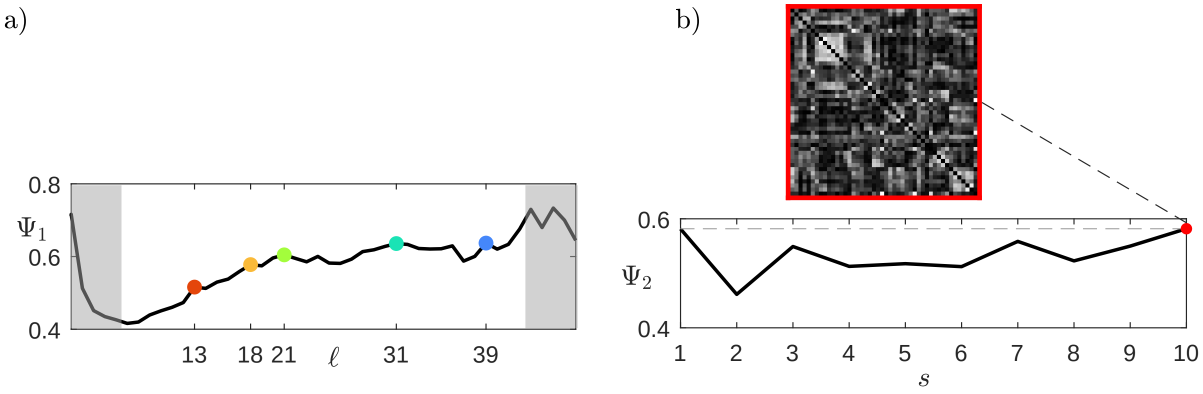


Figure 2.4: (a) Cumulative (across the 10 fMRI sessions) similarity index Ψ_1, based on the Fowlkes and Mallows comparison measure, for each dendrogram level ℓ. Ψ_1 measures the consistency of the corresponding clustering across the fMRI sessions. Specific levels ℓ^*_k (corresponding to k clusters) are selected as local maxima of this curve (colored dots, same color code as Fig. 2.3), discarding local maxima that are not in ascending order and trivial partitions with too few or too many clusters (grey areas, corresponding to those of Fig. 2.3). (b) Cumulative (across levels ℓ^*_k) similarity index Ψ_2 for each of the 10 sessions. Ψ_2 indicates of how much the clustering obtained from a session is similar to the clusterings from all other sessions. The fMRI session corresponding to the maximum value of Ψ_2 (red dot, corresponding to the correlation matrix X^* shown above) is chosen as the most representative of the subject.

2.3 Compatibility of the structural and functional connectivity matrices

The main modeling assumption is that a network with homogeneous nodes, for which exact synchronization within each cluster is obtained, can be used as a reference. On the one hand, this could seem unrealistic, because in a real neuronal network each area is expected to be unique, i.e., from a modeling standpoint, to have its own parameter values, as the corresponding neuronal populations are heterogeneous with respect to multiple physical properties. On the other hand, the identification of these heterogeneities (and corresponding model parameters) from measured data is highly nontrivial and depends on the quality and quantity of data, the noise inherent to measurements of biological systems, and the complexity of the chosen model. For these reasons, heterogeneity is initially removed from the network – thus dealing with *exact cluster synchronization* and focusing on the basic phenomenon stripped out of any other effect – to re-introduce it later, in the form of noise and heterogeneous node parameters.

Given these premises, is the structural connectivity matrix A_0 compatible with the clustering observed in the functional connectivity matrix X^*? To verify this, the network nodes are modeled as Wilson-Cowan neural masses [151] and the BOLD signal associated with each node is obtained by using the Balloon–Windkessel hemodynamic model of Friston and Harrison [153].

2.3.1 The Wilson-Cowan neural mass model

The dynamics of the i-th node are described by the equations:

$$\begin{cases} \tau_E \dot{E}_i = -E_i + \frac{1}{1+e^{-c(w_{EE}E_i - w_{IE}I_i + P + \eta_i(t) + E_i^{syn} - \theta)}}, \\ \tau_I \dot{I}_i = -I_i + \frac{1}{1+e^{-c(w_{EI}E_i - \theta)}}, \end{cases} \tag{2.4}$$

with

$$E_i^{syn} = \sigma \sum_j a_{ij} E_j, \tag{2.5}$$

where the state variables $E_i(t)$ and $I_i(t)$ are the fraction of excitatory and inhibitory neurons firing per unit time at instant t, respectively; couplings are modeled as in [141, 166], without considering delays for ease of presentation. However, the approach can be successfully adapted to account for connection delays, as will be detailed in section 2.6. The weights w_{uv}, with $u, v = \{E, I\}$, describe the intra-population strength of connection from neuron type u to v; c and θ are, respectively, the gain and the threshold of the sigmoid; P is a spontaneous excitatory background input; $\eta_i(t)$ is a white gaussian noise signal, kept null unless otherwise stated; τ_E, τ_I control the timescales of the first-order kinetics; E_i^{syn} is the synaptic input to node i, which is determined by the structural connectivity matrix entries a_{ij} and the fraction of active excitatory cells in each j-th pre-synaptic population. Eq. (2.5) follows from the fact that long-range connections between cortical areas are only excitatory [167].

The parameters are set as in [166]: $w_{EE} = 3.5$, $w_{IE} = 2.5$, $w_{EI} = 3.75$, $c = 4$, $\theta = 1$, $P = 0.34$. P is chosen so that the isolated node converges to an equilibrium point corresponding to a low activity state, but coupled nodes converge to limit cycles even for small coupling weights. Time constants are set to τ_E =0.002 s and τ_I =0.004 s, so that their ratio is as in [166], but the nodes exhibit oscillations around 40 Hz as in [141]. For a fixed value of P, the convergence of the steady-state trajectory to a limit cycle or to an equilibrium point depends on the value of the synaptic input E_i^{syn}, which in turn is influenced by the overall strength of the connections σ in Eq. (2.5). The range of σ is chosen so that the nodes exhibit oscillatory behavior, in accordance with the local field potential oscillations observed in cortical activity. In the illustrating example, the nodes exhibit oscillatory behavior for $\sigma \in [0.0001, 0.2]$. For $\sigma > 0.2$ a growing portion of nodes 'saturates', converging to an equilibrium point corresponding to a high activity state.

2.3.2 The Balloon-Windkessel hemodynamic model

The Balloon–Windkessel hemodynamic model of Friston and Harrison relates neural activity to perfusion changes. The model is described by the equations:

$$\begin{cases} \dot{s}_i = z_i - \kappa s_i - \gamma(f_i - 1), \\ \dot{f}_i = s_i, \\ \tau\dot{\nu}_i = f_i - \nu_i^{\frac{1}{\alpha}}, \\ \tau\dot{q}_i = f_i\frac{1-(1-\rho)^{\frac{1}{f_i}}}{\rho} - \nu_i^{\frac{1}{\alpha}}\frac{q_i}{\nu_i}, \end{cases} \tag{2.6}$$

where s_i is the vasodilatory signal, which increases according to the neuronal activity z_i of the i-th region (in our case $z_i(t) = E_i(t)+I_i(t)$), and is subject to autoregulatory feedback; f_i is the inflow, ν_i is the blood volume and q_i is the deoxyhemoglobin content. Parameter $\alpha = 0.32$ is the Grubb's exponent [168], $\rho = 0.34$ is the resting oxygen extraction fraction and the other biophysical parameters are set to $\kappa = 0.65$ per s, $\gamma = 0.41$ per s and $\tau = 0.98$ s, as per the mean values reported in [153]. The BOLD signal is taken to be a static nonlinear function of ν_i and q_i:

$$y_i = V_0(7\rho(1 - q_i) + 2(1 - \frac{q_i}{\nu_i}) + (2\rho_i - 0.2)(1 - \nu_i)) \tag{2.7}$$

where $V_0 = 0.02$ is the resting blood volume fraction [153].

2.3.3 Simulation procedure

For each level ℓ_k^* the following procedure is carried out:

1. The network dynamics and the corresponding BOLD signals are simulated for an array of values of σ (see Eq. (2.12)), repeating the procedure 30 times to account for the effects of different initial conditions.

2. For each simulation, the correlation matrix of the BOLD signals is calculated, discarding transient data. The resulting correlation matrix can be compared with the functional connectivity matrix X^*.

3. The same hierarchical clustering that was applied on X^* is applied on each of the correlation matrices, obtaining a clustering with k clusters (corresponding to level ℓ_k^*). Each obtained partition is compared with the target partition associated with X^*, and the Fowlkes and Mallows matching index B between the two partitions is computed [165].

The initial conditions for all network simulations are set according to the following criteria: (i) give the same initial conditions to the nodes belonging to the same target cluster, to encourage the eventual emergence of said cluster (ii) perturb slightly these initial conditions to ensure that the state variables do not get stuck on an unstable orbit. For this reason, the initial conditions for the state variables E_i and I_i of the nodes belonging to the same target cluster are defined as a common mean value, set randomly at each trial, plus Gaussian noise with a small standard deviation of 10^{-5}. Initial conditions for the state variables of the BOLD model are simply set randomly at each trial, regardless of the target cluster configuration. Network simulations are performed with the *ode45* Matlab ODE solver. The correlation matrices are calculated using Matlab's *corrcoef* function, which returns an N-by-N matrix with correlation coefficients of variable pairs as off-diagonal entries.

In the illustrative example, the comparison measure B is averaged over the 30 random trials (and thus denoted as $\bar{B}$) and is found to be low (always below 0.3) for all levels ℓ_k^* and for all values of σ, as shown in Fig. 2.5. This highlights that using the information contained in the structural connectivity matrix A_0 'as is' to model synaptic weights generates results that are incompatible with the observed functional connectivity. This finding is in agreement with other studies that have questioned the reliability of using connectivity matrices describing fiber density as a quantitative indication of synaptic weights [169].

The following section shows how it is possible to intervene on the structural connectivity matrix A_0 to make it compatible with the observed functional connectivity.

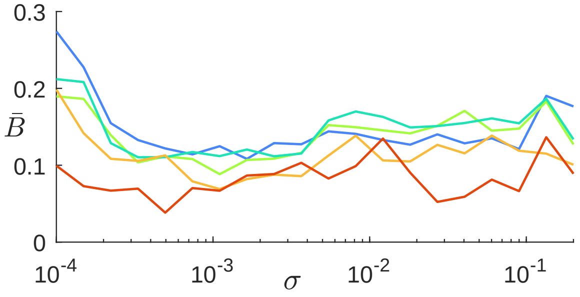


Figure 2.5: Comparison measure $\bar{B}$ between the cluster partition obtained from network simulations with synaptic weights a_{0ij} (entries of A_0) and the target cluster partition associated with X^*, averaged over 30 random trials for different values of σ. Different colors refer to different levels ℓ_k^* (same color code as in Fig. 2.3 and 2.4).

2.4 Optimization of the structural connectivity matrix

As a first approximation, the identified clusters of nodes with coherent activity can be viewed as synchronized clusters. There are two separate questions addressed in the following sections, one is the existence of a cluster-synchronous solution and the other one is its stability. The existence of a cluster-synchronous solution requires that, for proper initial conditions, groups of nodes synchronize. This solution is also stable if, under small perturbations, the system state goes back to the same synchronous clusters.

Given a graph with N nodes, a partition $P = \{C_1, C_2, ..., C_k\}$ of the graph is defined as a subdivision of its node set into k clusters, each composed of n_p nodes, such that (i) there are no empty clusters, (ii) the clusters comprise all nodes and (iii) the clusters are pairwise disjoint, i.e. $\sum_{p=1}^{k} n_p = N$. Each cluster can be identified through the labels of its constituting nodes and a given color. A network graph can admit many different colorings, each one describing a different partition. The existence of a particular set of self-sustained synchronized clusters in a network graph with homogeneous nodes can be ruled out or not based on the network topology, independently from the dynamical model of the nodes; in particular, the existence of an equitable partition for a given network is a necessary condition for the existence of a cluster synchronous solution [170, 171]. At this point, only the existence of a cluster synchronous solution is considered, not its stability; the possibility to converge to such a solution depends also on the node dynamics, i.e. the functions $\boldsymbol{F}$ and $\boldsymbol{\Gamma}$ [155].

A partition (and the corresponding coloring) is equitable if all nodes with color p receive the same overall input from the nodes of color q, for $p, q = 1, ..., k$. For each level ℓ_k^*, given the set of target clusters, the goal is to obtain a structural connectivity matrix A_k for which they constitute an equitable partition. A_k is calculated starting from A_0 and solving a quadratic optimization problem. The cost function takes into consideration the element-wise quadratic difference between A_0 and A_k and also the level of uncertainty associated with each entry of the A_0 matrix, stored in the matrix Σ_{A_0}, so that weights that are more reliable are less likely to be changed significantly.

Matrix Ξ_k is defined as the element-wise square difference between A_k and A_0, i.e., $\Xi_{k_{ij}} = (a_{k_{ij}} - a_{0_{ij}})^2$. Fig. 2.6 (top panels) shows that the entries of the matrix Ξ_k are comparable with the entries of the matrix Σ_{A_0}. Moreover, the entries of the matrix Ξ_k are smaller as the number of clusters increases because the optimization process changes less the original matrix A_0. This result means that the optimized matrices A_k are still strongly based on dMRI experimental data and do not differ from the original connectomes more than what can be justified by the uncertainty introduced by the measurement process. Bottom panels show that the square difference between the entries of the matrices A_0 and A_k distributes similarly to the entries of the matrix Σ_{A_0}, i.e., the higher the uncertainty of a specific weight, the larger the change introduced by the optimization algorithm.

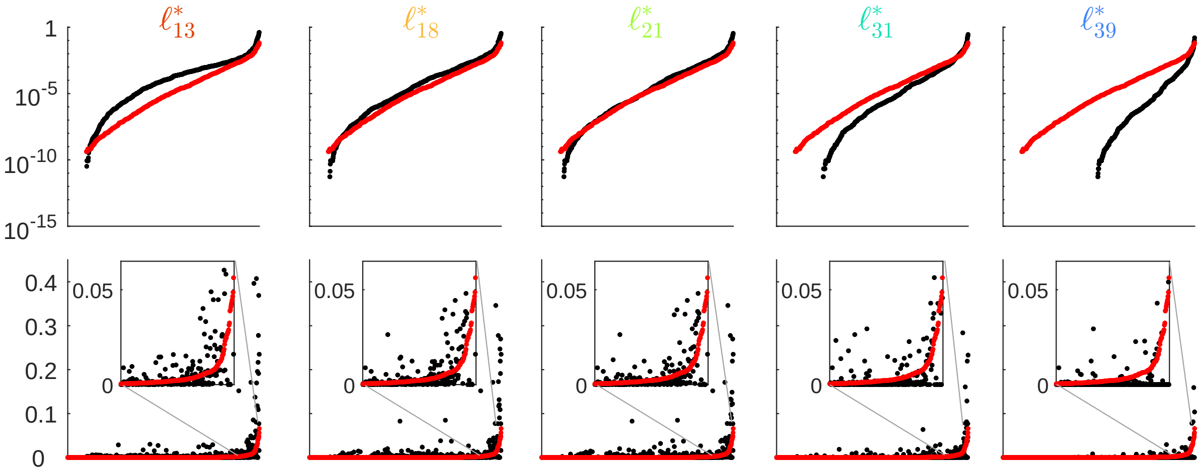


Figure 2.6: Entries of the matrices Σ_{A_0} (red dots) and Ξ_k (black dots), for all levels ℓ_k^*. Top panels: entries of both Σ_{A_0} and Ξ_k are ordered from the smallest to the largest and are graphed on a semi-logarithmic scale. Bottom panels: entries of Σ_{A_0} are ordered from the smallest to the largest and entries of Ξ_k are displayed following the same permutation of the indices, graphed on a linear scale.

2.4.1 Optimization algorithm

The structural connectivity matrix A_0 is derived from diffusion MRI data, and, as such, it has positive entries, is symmetrical, and has null diagonal entries. The optimized matrix A_k is also constrained to (i) have positive entries, (ii) be symmetric, and (iii) have null diagonal entries, to maintain the original features of the structural connectivity matrix. From the variance matrix Σ_{A_0}, a 'reliability' matrix $\hat{\Sigma}_{A_0}$ is defined as $\hat{\Sigma}_{A_0} = 100 \cdot (\max(\Sigma_{A_0}) - \Sigma_{A_0} + \epsilon)$, so that the entries with low variance have high reliability and vice versa; ϵ is an arbitrarily small quantity that has the only function of avoiding zero entries. $\hat{\Sigma}_{A_0}$ is included in the cost function so that the more reliable an entry is, the less likely it is to be changed significantly in the optimization process to find A_k.

The optimization process is described in the following: the columns of A_k are stacked in a vector x; the columns of the matrix obtained multiplying entry by entry A_0 by $\hat{\Sigma}_{A_0}$ are stacked in a vector α; a matrix H is defined as the N^2-size diagonal matrix with $\hat{\Sigma}_{A_0}$

entries on its diagonal; a matrix $M_1x = 0$ codifies the equitable partition conditions and a matrix $M_2x = 0$ codifies constraints (ii) and (iii). The optimal vector x (i.e., the matrix A_k) is found by solving the following optimization problem, with a quadratic objective function and linear constraints of equality and inequality:

$$\min_x \frac{1}{2} x^T H x - \alpha^T x \tag{2.8}$$

s.t.

$$\begin{aligned} x &> 0 \\ M_1 x &= 0 \\ M_2 x &= 0. \end{aligned} \tag{2.9}$$

This quadratic programming problem is numerically solved with the Matlab function *quadprog*.

2.5 The optimized model

The stability of the exact cluster synchronous solution can be studied by using the Master Stability Function (MSF) approach [154] to determine the intervals of σ values for which the synchronous clusters for each level ℓ_k^* are stable.

2.5.1 Stability analysis

A network of N coupled Wilson-Cowan neural oscillators can be described by the general set of equations (2.12) with $m = 2$ state variables:

$$
\begin{aligned}
\boldsymbol{x}_i &= \begin{bmatrix} E_i \\ I_i \end{bmatrix}, \\
\boldsymbol{F}(\boldsymbol{x}_i) &= \begin{bmatrix} \frac{1}{\tau_E}(-E_i) \\ \frac{1}{\tau_I}(-I_i + \frac{1}{1+e^{-c(w_{EI}E_i-\theta)}}) \end{bmatrix}, \\
\boldsymbol{\Gamma}(\boldsymbol{x}_i, \sigma \sum_j a_{0ij} G(x_j)) &= \begin{bmatrix} \frac{1}{\tau_E} \frac{1}{1+e^{-c(w_{EE}E_i - w_{IE}I_i + P + E_i^{syn} - \theta)}} \\ 0 \end{bmatrix}, \\
\boldsymbol{G}(\boldsymbol{x}_j) &= \begin{bmatrix} E_j \\ 0 \end{bmatrix}.
\end{aligned}
$$

The cluster synchronization state can be denoted as $\boldsymbol{x}_i(t) = \boldsymbol{s}_p(t)$, where node i belongs to cluster $\mathcal{C}_p$. Small variations $\boldsymbol{w}_i(t) = \boldsymbol{x}_i(t) - \boldsymbol{s}_p(t)$, stacked in the state perturbation vector $\boldsymbol{W}(t)$, are introduced to investigate the stability of the synchronous state and their linearized variational equations are derived (see section B.2 in Appendix B for details).

It is desirable to find a coordinate transformation that separates *as much as possible* the perturbation modes in the stability analysis, thus allowing a reduction of the stability problem in lower-dimensional sub-problems. To this end, the coordinate transformation introduced in [172, 173, 174], based on simultaneous block diagonalization (SBD), is performed through the canonical transformation matrix $T = \begin{pmatrix} T_\parallel \\ T_\perp \end{pmatrix}$ [175].

The $k \times N$ submatrix $T_{\parallel}$ is associated with the directions *along* the synchronization manifold and the corresponding perturbations do not influence the stability of the synchronized clusters. The $(N-k) \times N$ submatrix $T_{\perp}$ is associated with the directions *transverse* to the synchronization manifold and the evolution of the variational equation along these directions determines the stability of the synchronized clusters.

Transverse perturbations are denoted as $\boldsymbol{\eta}_{\perp}(t) = (T_{\perp} \otimes I_m)\boldsymbol{W}(t)$ (where $\otimes$ is the Kronecker product and I_m is a $m \times m$ identity matrix) and their variational equation is:

$$\dot{\boldsymbol{\eta}}_{\perp}(t) = \rho_1(\{\boldsymbol{s}_p(t)\})\boldsymbol{\eta}_{\perp}(t) + \rho_2(\{\boldsymbol{s}_p(t)\})\boldsymbol{\eta}_{\perp}(t) \tag{2.10}$$

where the set $\{\boldsymbol{s}_p(\cdot)\}$ collects all the synchronous solutions corresponding to the k clusters and ρ_1 and ρ_2 are time-varying matrices defined in section B.2 of Appendix B. The Lyapunov exponents for each $\boldsymbol{\eta}_{\perp}$ component are calculated and the σ intervals where the cluster synchronous solution is stable are determined as those where the MSF (i.e., the maximum Lyapunov exponent) is negative.

2.5.2 Robustness to noise and parameter mismatches

The assumption of exact cluster synchronization is necessary to derive meaningful results, but is restrictive and unrealistic when referring to experimental data from the activity of brain areas. For this reason, it should be verified that the obtained stability intervals of σ values hold also for perturbed versions of A_k that produce approximate synchronization (for small perturbation of the matrix A_k, this was studied in [171]).

The compatibility check, as described in steps 1-3 of section 2.3.3, is repeated on the optimized structural connectivity matrices A_k perturbed by additive Gaussian white noise with mean 0 and standard deviation σ_A. This analysis tests for up to which level of noise on the connection weights the optimized structural connectivity matrix remains compatible with the target clusters, derived from the functional connectivity matrix, in the considered σ intervals. The results for all levels ℓ_k^* are shown in Fig. 2.7, along with the intervals of σ values for which the synchronous clusters for each level are stable according to the MSF approach (red lines). As expected, the average comparison measure $\bar{B}$ between the target partition (derived from experimental data) and the partition obtained by simulating the network with the optimized and perturbed structural connectivity matrices A_k becomes lower as σ_A grows. It can be observed that $\bar{B}$ is above the maximum value obtained by simulating the network with the original structural con-

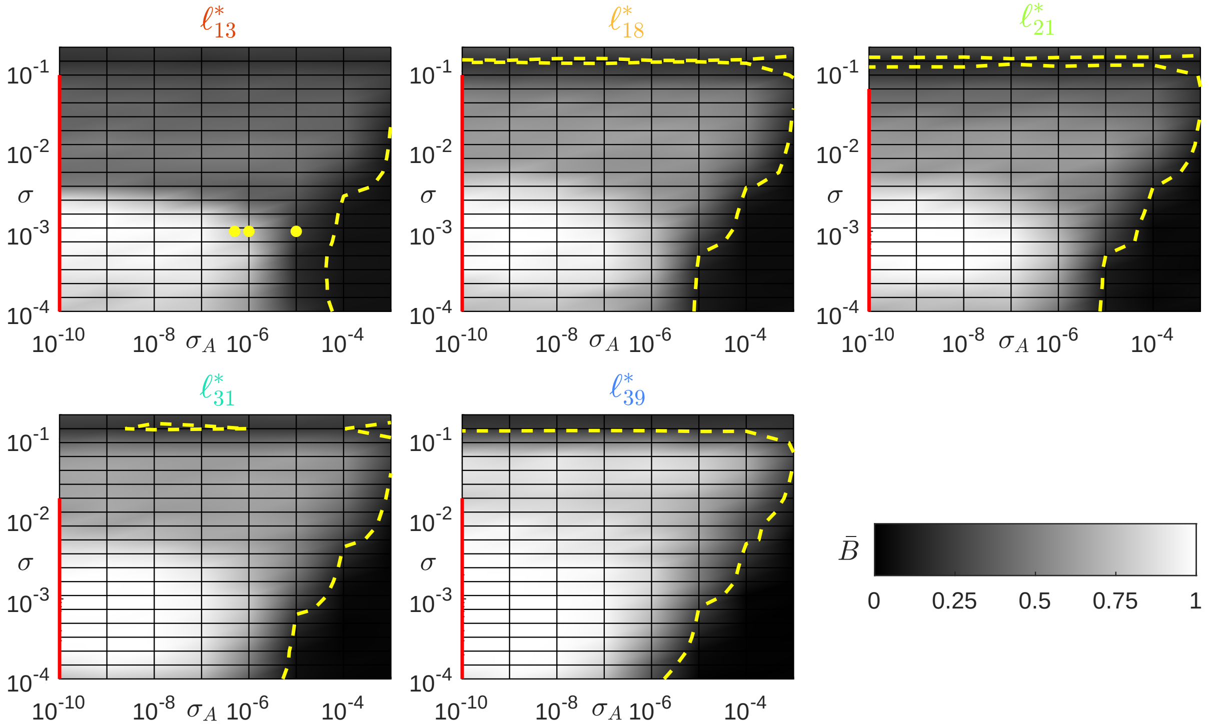


Figure 2.7: Comparison (through the average comparison measure $\bar{B}$) between the target cluster partition (derived from experimental data) and the partition obtained from network simulations with optimized matrices A_k perturbed by Gaussian noise, for each level ℓ^*_k; σ_A on the abscissa denotes the noise standard deviation. Red bars on the vertical axis highlight the σ (synaptic strength) for which the synchronous clusters are stable according to the MSF approach. Dashed yellow lines are the level curves delimiting the regions where $\bar{B}$ is higher than the maximum value obtained by simulating the network with the original structural connectivity matrix A_0 for the same level. Yellow dots mark the values of σ and σ_A used in Fig. 2.8.

nectivity matrix A_0 in the large region enclosed within the yellow dashed curve. In this region, the optimized network (with connectivity matrix A_k) behaves in better accordance with the observed functional connectivity than the original network (with connectivity matrix A_0). Dark regions that fall outside the curve for low values of σ_A are always beyond the stability interval identified through the MSF approach. This indicates that the results are robust to noise on the connection weights, also comparing the numerical values of σ_A with the fact that the entries of A_k have a mean of 0.029 and a standard deviation of 0.065, with minimum 0 and maximum 0.952. The raster plots and BOLD signals corresponding to specific values of σ_A and $\bar{B}$ (yellow dots in Fig. 2.7) are shown in Fig. 2.8: the progressive desynchronization of clusters is apparent in both the raster plots and the BOLD signals as σ_A increases and $\bar{B}$ decreases.

A similar analysis was carried out by adding a white Gaussian noise term $\eta_i(t)$ with standard deviation σ_η to every node's excitatory subpopulation input (see Eq. (2.4)). Panel a of Fig. 2.9 illustrates the results obtained for ℓ^*_{21} and $\sigma = 10^{-3}$: similarly, $\bar{B}$ holds values close to 1 until σ_η remains under a certain threshold. Note that $\bar{B}$ is higher than the value obtained by simulating the network with A_0 for the same level and the same σ value, for standard deviation σ_η up to about 10^{-2}, which approaches the amplitude of the node input without noise; this means that the optimized structural connectivity matrix A_k appears to be in better accordance with the observed functional connectivity than A_0, even for a relatively high noise level. This provides evidence of the robustness of the obtained results.

The robustness of the model was also tested by introducing heterogeneous node parameters. In particular, the model parameters that represent connection weights between inhibitory and excitatory subpopulations within each node (see Eq. (2.4)) were sampled from Gaussian distributions with standard deviation σ_w. Results are shown in panel b of Fig. 2.9: also in this scenario, the optimized structural connectivity matrix A_k is in better accordance with the observed functional connectivity than A_0, up to a relatively high value of σ_w.

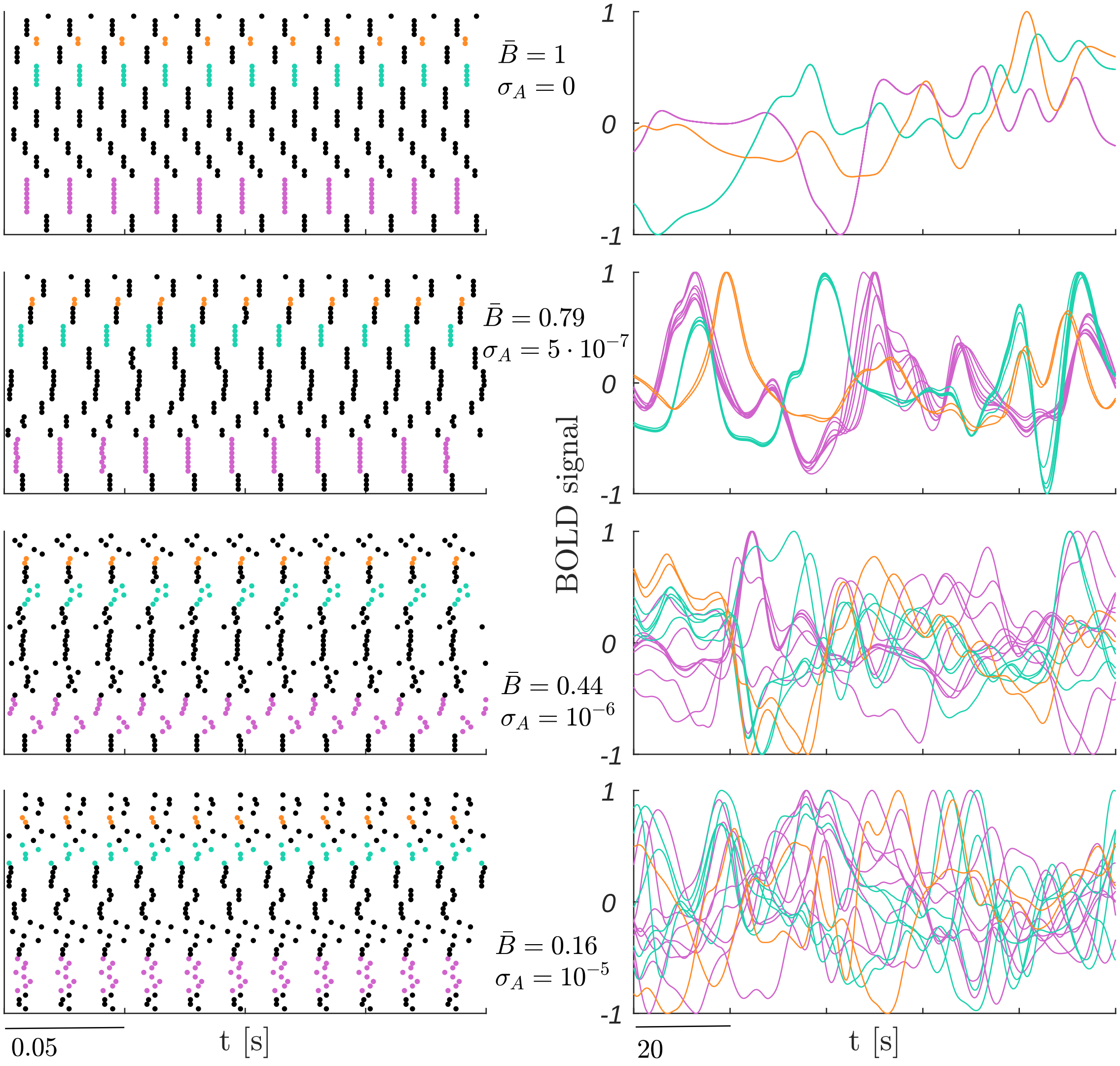


Figure 2.8: Raster plots (left) and normalized BOLD signals (right) for level ℓ^*_{13}, for different values of noise standard deviation σ_A and for fixed $\sigma = 10^{-3}$ (see yellow dots in Fig. 2.7). Points in the raster plots denote the peaks of excitatory subpopulation activity of each node, with nodes belonging to the same cluster represented as adjacent to each other. The normalized BOLD signals correspond to nodes in three sample clusters, chosen to showcase clusters of different sizes (pink: 8 nodes, teal: 5 nodes, orange: 2 nodes). Values of σ_A increase from top to bottom and correspond to decreasing values of $\bar{B}$.

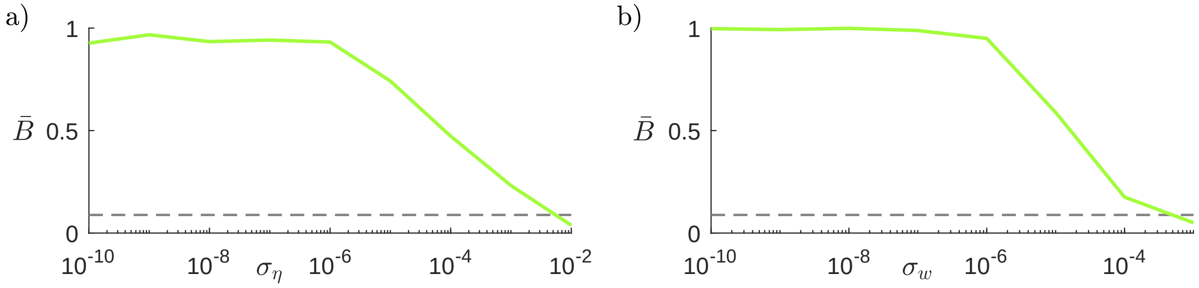


Figure 2.9: (a) Comparison (through the average comparison measure $\bar{B}$) between the target cluster partition and the partition obtained from network simulations with connectivity matrix A_{21} and a white Gaussian noise term $\eta_i(t)$ added to every node's excitatory subpopulation input; the result is obtained for $\sigma = 10^{-3}$. σ_η is the noise signal standard deviation (see Eq. (2.4)). The gray dashed line marks the value of $\bar{B}$ obtained by simulating the network with the original structural connectivity matrix A_0 and without noise for the same level and same σ value (Fig. 2.5). (b) Comparison (through the average comparison measure $\bar{B}$) between the target cluster partition and the partition obtained from network simulations with connectivity matrix A_{21} and heterogeneous nodes. Heterogeneity is introduced by sampling the parameters w_{EE}, w_{IE}, w_{EI} (see Eq. (2.4)) from Gaussian distributions with means $\mu_{w_{EE}} = 3.5$, $\mu_{w_{IE}} = 2.5$ and $\mu_{w_{EI}} = 3.75$ and standard deviation σ_w. The result is obtained for $\sigma = 10^{-3}$. The gray dashed line marks the value of $\bar{B}$ obtained by simulating the network with the original structural connectivity matrix A_0 and homogeneous nodes for the same level and same σ value.

2.6 Introducing delayed synaptic connections

The proposed method can be generalized to account for delays in the synaptic connections between brain areas. To demonstrate how, a delay τ_{ij} was assigned to each connection between nodes i and j, i.e., to each entry of the structural connectivity matrix A_0. The delay matrix was calculated by multiplying an $N \times N$ distance matrix by a propagation velocity of 1.5 m/s [176]; the distance matrix collects the pair-wise 3D euclidean distances between nodes, specifically between the centers of brain areas according to the Harvard-Oxford Cortical Structural Atlas. When delays are introduced, the synaptic current in the Wilson-Cowan neural mass model is re-defined as

$$E_i^{syn}(t) = \sigma \sum_j a_{ij} E_j(t - \tau_{ij}), \tag{2.11}$$

To optimize the structural connectivity matrix and carry out the stability analysis with the MSF approach, the delays τ_{ij} are quantized over L values τ^l. This choice is motivated by two main reasons: (i) the computational cost of the MSF approach increases linearly with increasing L, (ii) the degrees of freedom available to the optimization algorithm decrease as L increases, as detailed in the following. The network of N neural oscillators, coupled through L kinds of links (each one characterized by a delay τ^l), can then be described by the following general set of equations ($i, j = 1, \dots, N, \quad l = 1, \dots, L$):

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{F}(\boldsymbol{x}_i(t)) + \boldsymbol{\Gamma}\left(\boldsymbol{x}_i(t), \sigma \sum_l \sum_j a_{ij}^l \boldsymbol{G}(\boldsymbol{x}_j(t - \tau^l))\right) \tag{2.12}$$

In this case, $L = 3$ kinds of connections are considered, and delays τ_{ij} are quantized over 3 values. To carry out the optimization, the structural connectivity matrix A_0 is split into L matrices A_0^l, each corresponding to a link with delay τ^l. In particular, entries of A_0^l corresponding to different delays are set to zero and $\sum_l A_0^l = A_0$. Each matrix A_0^l is optimized individually, thus obtaining matrices A_k^l, according to the definition of equitable partitions for graphs with L different kinds of links [177]. The final optimized structural connectivity matrix for level k is obtained as $\sum_l A_k^l = A_k$. This process is summarized in Fig. 2.10. For high values of L, extremely sparse matrices A_0^l would be obtained, thus reducing the degrees of freedom for the optimization algorithm. This is due to the fact that zero entries are constrained at zero by the optimization algorithm.

Recall that matrix Ξ_k is defined as the element-wise square difference between A_k and A_0, i.e., $\Xi_{k_{ij}} = (a_{k_{ij}} - a_{0_{ij}})^2$. Panel a of Fig. 2.11 shows that the entries of the matrix Ξ_k (exemplified for ℓ_{21}^*) are still comparable with the entries of the matrix Σ_{A_0}, also when A_k is obtained as explained above. Moreover, panel b shows that the square difference between the entries of the matrices A_0 and A_k (exemplified for ℓ_{21}^*) distribute similarly to the entries of the matrix Σ_{A_0}.

The stability analysis is carried out by generalizing the MSF formalism to account for links with L different delays τ^l (see section B.3 in Appendix B for details). The cluster synchronous solution for level ℓ_{21}^* is stable for $\sigma \in [10^{-4}, 0.1341]$. The raster plot

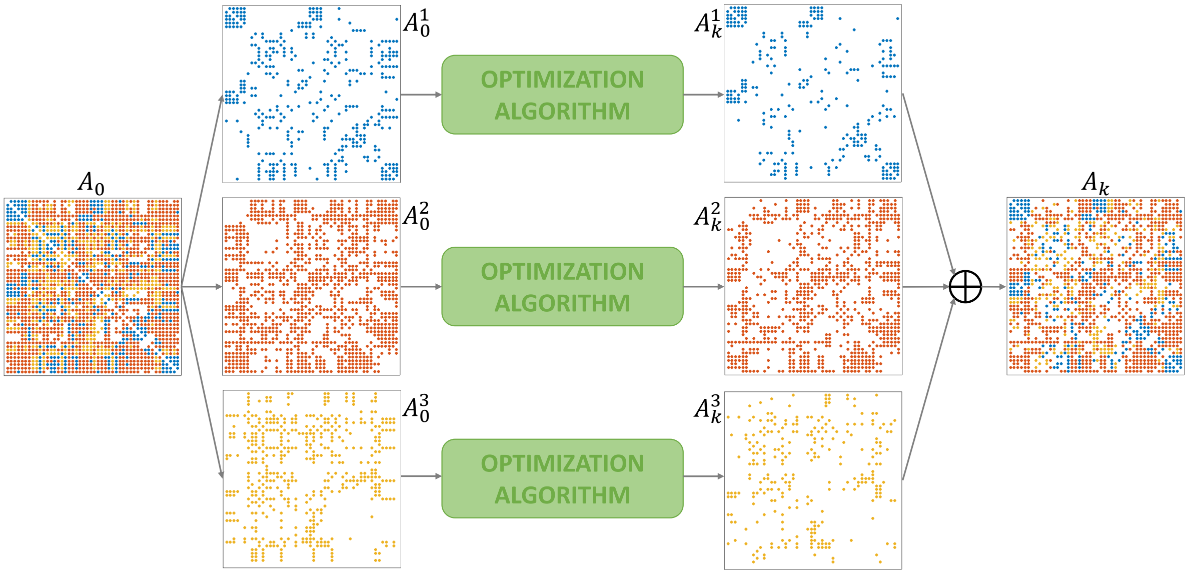


Figure 2.10: The structural connectivity matrix A_0 is split into 3 matrices A_0^1, A_0^2 and A_0^3, corresponding to link kinds with delay $\tau^1 = 0.0204$ s, $\tau^2 = 0.0518$ s and $\tau^3 = 0.0832$ s, respectively. Each matrix A_0^l is optimized individually, obtaining matrices A_k^1, A_k^2 and A_k^3 (in this example, k=21). The final optimized structural connectivity matrix for level k is obtained as $\sum_l A_k^l = A_k$.

for $\sigma = 10^{-3}$ and $L = 3$, with delays quantized as in panel a of Fig. 2.12, is shown in panel c of the same figure, evidencing the synchronization between the nodes in each cluster. Heterogeneity in delay values was reintroduced *a posteriori* to test the robustness of the result obtained for $L = 3$. In particular, the delays were quantized over 50 values (panel b), which overlap almost perfectly with the original values of τ_{ij}. τ_{ij} values were not used as is because of the excessive computational burden. The corresponding raster plot is shown in panel d: as can be observed, heterogeneity in the delay values does not compromise the synchronization of the nodes within each cluster but influences the phase difference between clusters [178]. This analysis evidences that, from a theoretical standpoint, the proposed method can be generalized to account for heterogeneous delays in the network; however, the computational cost of simulating the network dynamics for fully heterogeneous delays becomes prohibitive.

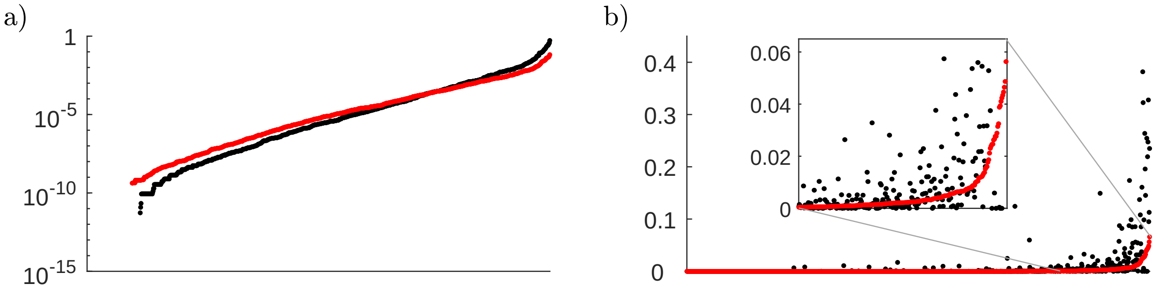


Figure 2.11: (a) Entries of the matrices Σ_{A_0} (red dots) and Ξ_k (black dots), for level ℓ^*_{21}. Entries of both Σ_{A_0} and Ξ_k are ordered from the smallest to the largest and are graphed on a semi-logarithmic scale. (b) Entries of the matrices Σ_{A_0} (red dots) and Ξ_k (black dots), for level ℓ^*_{21}. Entries of Σ_{A_0} are ordered from the smallest to the largest and entries of Ξ_k are displayed following the same permutation of the indices, graphed on a linear scale.

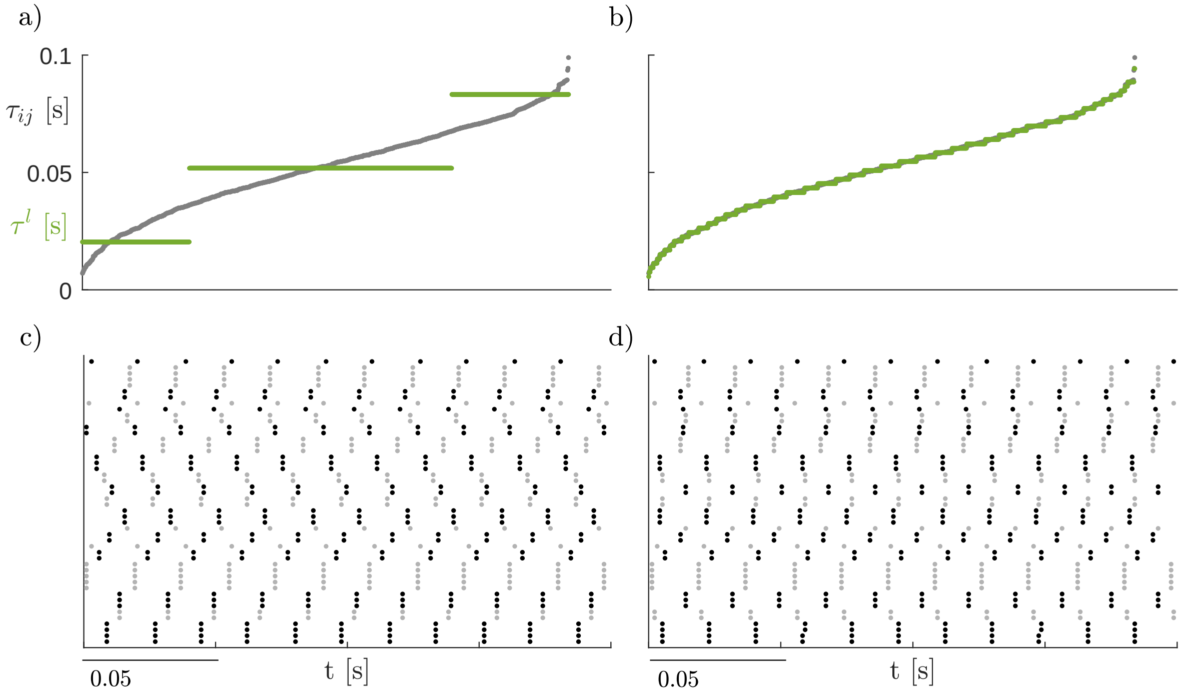


Figure 2.12: (a) Delays τ_{ij} sorted in ascending order (gray dots) and delays τ^l quantized over 3 values (green lines). (b) Delays τ_{ij} sorted in ascending order (gray dots) and delays τ^l quantized over 50 values (green lines). (c) Raster plot for level ℓ^*_{21}, for fixed $\sigma = 10^{-3}$ and delays quantized over 3 values. Points in the raster plots denote the peaks of the excitatory subpopulation activity of each node, with nodes belonging to the same cluster represented as adjacent to each other; clusters are represented by alternating black and gray colors to facilitate visualization. (d) Raster plot for level ℓ^*_{21}, for fixed $\sigma = 10^{-3}$ and delays quantized over 50 values.

2.7 Concluding remarks

The proposed method, based on nonlinear dynamics, aims to get information about the structural connectivity of a subject based on both dMRI and fMRI data. Some key points and assumptions are discussed in deeper detail in the following to evidence the flexibility of the proposed approach.

Types of data – fMRI data were used to derive the functional connectivity matrices and the associated clustering. As an alternative, also other kinds of data providing similar information (e.g. EEG, MEG) could be used [179].

Hierarchical approach and clustering – Hierarchical clustering is an established method to identify clusters of nodes in brain networks [143, 161, 162, 163, 164], due to the hierarchical modularity exhibited by the human brain [180]. Identifying the most appropriate number of clusters from a dendrogram, however, is not straightforward. A multi-level approach was carried out in the proposed example and the levels of interest ℓ_k^* were selected based on considerations derived by the experimental data. In the absence of multiple scans for each subject or *a priori* knowledge, it is also possible to use other metrics to determine the optimal number of clusters, such as the silhouette score [181] or the gap statistic [182]. In principle, one could also consider all levels, but this would greatly increase the computational cost of the method.

Directed vs undirected graphs – The proposed approach could be easily generalized to directed connections [177]. In this case, however, the maximum size of the network to be numerically studied would be lower, due to the higher computational complexity of the MSF approach in the non-symmetric case, as detailed in [177].

Dynamics of the nodes – In the proposed example, nodes described by the Wilson-Cowan neural mass model [151] were used, with couplings as in [141, 166] and parameters as in [166]. As an alternative, one could employ next-generation neural mass models able to exactly reproduce the macroscopic dynamics of heterogeneous spiking neural networks [183, 184, 185, 186, 187]. This would allow using heterogeneous parameters that are identified with high precision. The MSF step can be taken provided that the considered neural mass model falls under the proposed formalism, or the formalism is appropriately generalized. The node homogeneity hypothesis needed to carry out the optimization of the structural connectivity matrix and the stability analysis would require the use of average parameters, under the assumption that parameters that characterize distinct brain areas are not too different [188].

Cost function of the optimization process – The cost function can be customized, e.g., to impose that some elements of the connection matrix are not changed or to weigh differently the uncertainty about the entries of A_0.

Subjects – The proposed results have been obtained with data measured on healthy subjects in resting state. For non-healthy subjects, specific neural mass models should be used (e.g., the epileptor [189] for epilepsy, or the model proposed in [190] for Parkinson's disease), but the proposed approach remains valid, *mutatis mutandis*. For healthy subjects exposed to a specific stimulus, an external input representing said stimulus should be added to the model. This could allow distinguishing between stimulus responses and ongoing activity [191].

Proposed example vs. global picture – Neuroscientists monitor brain activity in many ways, using a variety of recording techniques in subjects under a plethora of health, neurophysiological and mental conditions. This complex and variegate scenario requires the integration of insights across diverse datasets to understand brain functions [192]. This chapter proposed a method to integrate the information provided by dMRI and fMRI, with the aim of reducing the uncertainty about the anatomical neural connectivity of a specific subject at resting state. The analysis carried out has been restricted to a limited set of clusterings: an extensive analysis for all levels ℓ_k^* would have been prohibitive in terms of computational cost. While this is just a first step, the proposed method can pave the way towards a better interplay of nonlinear dynamics, complex networks, and neurophysiological data. In turn, this can lead to better brain models and a deeper understanding of brain functions. Indeed, the study of dynamic functional connectivity reveals that brain activity switches between a set of states, where a finite number of clustering are identifiable [193, 160]. The proposed method could be generalized by modifying the structural connectivity optimization process so that the optimized matrix admits all observed clusterings. The stability analysis could then be carried out for each synchronous cluster solution, and simulations with different initial conditions could give information on the basin of attraction of each one, thus providing a model able to fully reproduce the observed dynamic functional connectivity.

Chapter 3

Power Grids

Personal contribution — Section 3.1 presents known material and concepts that serve to substantiate the need for inertia estimation algorithms in power grids. The analysis and the results detailed in the rest of the chapter are original material, although an initial version of the inertia estimation algorithm was taken from the literature.

3.1 The role of inertia in power grids

In recent years the presence of renewable energy sources (RES), like photovoltaic and wind power plants, has significantly increased in power grids [194]. This is also due to environmental policies, the fact that fossil fuel reserves are limited and that alternative solutions are necessary. Including renewable energy sources in power systems, however, poses challenges in terms of system stability. When the grid is at steady state, the electrical frequency is maintained at its reference value (for example 50 Hz). However, if a power imbalance occurs, the synchronous generators change their rotational speed causing a deviation of electrical frequency from its nominal value. Large deviations may cause instability in the grid and lead to faults. Each synchronous generator, and also the power grid as a whole, is characterized by the inertia constant, which is a measure of the system's ability to counteract the frequency changes triggered by external disturbances, as exemplified in Fig. 3.1. The inertia constant is thus a key factor in determining system stability [45].

By definition, the mechanical rotational speed of synchronous generators is directly coupled with the electrical frequency. So, in a power system composed only of synchronous generators, the source of inertia is the kinetic energy of the rotating masses. In this condition, the inertia constant of the power grid remains steady over the long term and thus it is a reliable reference for frequency stability control [195].

On the other hand, renewable energy sources are interfaced with the grid through power converters, which have no intrinsic inertia. Photovoltaic plants have no rotating masses and in wind generators the rotational speed of the turbine is not coupled with the electrical frequency. This reduces the global inertia of the system, thus compromising stability and reliability [196, 197]. To combat the decrease in system inertia and effectively integrate RES into power grids, various control strategies able to provide synthetic inertia have been developed [198, 199]. Even so, the power that can be provided by renewable sources is intermittent by nature, as is clear when talking about photovoltaic plants and wind turbines. This means that the controllers cannot always provide constant synthetic inertia, making the global system inertia time-varying. Consequently, the need arises for methods able to continuously estimate the inertia constant of the system during normal operating conditions [200]. Numerous strategies have been proposed, but many rely on

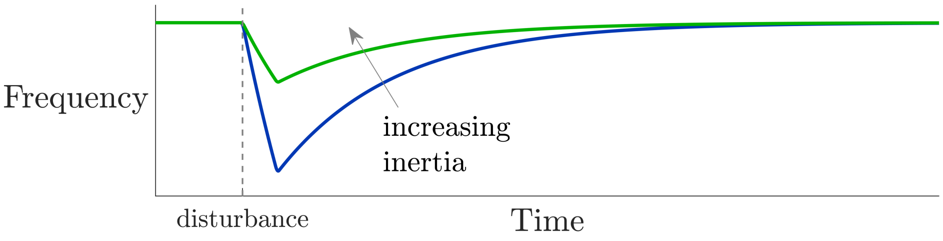


Figure 3.1: A larger value of inertia constant causes smaller frequency deviations in response to disturbances.

post-fault data [201, 202, 203, 204, 205] or injection of probing signals [206], making them not suitable for continuous inertia estimation. Some methods to estimate inertia under normal operating conditions use statistical models trained on historical data [207, 208], which are often not easily accessible.

The focus should thus be directed towards algorithms able to estimate the inertia constant in a continuous fashion, under normal operating conditions, relying on ambient measurements that can be collected through commonly used phasor measurement units (PMUs) [209]. The method proposed in [210] fits this category: it exploits the colored noise on bus voltages, line currents, and power flows, caused by the random fluctuations of the power consumption of the loads, and utilizes their covariance matrix to solve an optimization problem that fits such measurements to the synchronous machine classical model. This technique allows estimating both the inertia provided by synchronous machines (along with their equivalent damping) and the synthetic inertia provided by controllers of converter-interfaced generators that emulate the behavior of synchronous machines (along with their droop). This method relies on measurements in time periods of at least 15 minutes, a time scale that is compatible with short-term dispatch and adjustments markets.

The method considered in the following sections also relies on ambient measurements and it is based on the estimation of a linear model, relating the active power fluctuations to the frequency fluctuations on a bus, and the subsequent inertia constant extraction as a parameter of the identified model. The idea for this approach was first proposed in [211]. Compared to [210], this method considers shorter time scales (tens of seconds) that are compatible with weather changes that could influence the synthetic inertia contribution of RES controllers.

3.2 General pipeline of the inertia constant estimation method

Under normal operating conditions, the frequencies and powers in a power grid vary slightly around their steady-state values (denoting the operating point of the grid), mainly due to random fluctuations in the loads. The dynamics of a synchronous generator can be described by the well-known *swing equation* [45, 212]:

$$2H\dot{\omega} = P_m - P_e - D\left(\omega - \omega_0\right) \tag{3.1}$$

which relates the normalized mechanical power P_m and electrical power P_e – expressed in *per units* (p.u.) – to the normalized rotor angular speed ω (p.u.), through the inertia constant H [s], the damping coefficient D (dimensionless) and the normalized rated synchronous speed ω_0 (p.u.). Eq. (3.1) can be rewritten in terms of the variations $\Delta\omega$, ΔP_m and ΔP_e of frequency and powers with respect to their steady-state values. If small stochastic load variations are considered, it can be assumed that P_m never changes (i.e. its variation is 0), whereas $\Delta\omega$ can be approximated by the variation of the electrical frequency Δf measured through PMUs at the generator bus. Therefore, Δf (output) and ΔP_e (input) behave according to the following equation:

$$2H\dot{\Delta f} = -\Delta P_e + D\Delta f \tag{3.2}$$

By applying a unit-step ΔP_e in $t = 0$, the time evolution of Δf for $t > 0$ with initial condition $\Delta f(0) = 0$ is $\Delta f(t) = \frac{1}{D}\left(e^{-\frac{D}{2H}t} - 1\right)$, then the generator inertia constant can be expressed as

$$H = -1/(2\sigma_0), \tag{3.3}$$

where $\sigma_0 \triangleq \dot{\Delta f}\Big|_{t=0}$ is the slope of $\Delta f(t)$ in $t = 0$.

With this approach, it is possible to estimate the inertia constant at each time step τ (update time), based on the measurements of Δf and ΔP_e collected within a moving time window of length W, as depicted in Fig. 3.2. The general estimation pipeline can be summarized as follows:

1. take measurements of $\Delta f(t_j)$ and $\Delta P_e(t_j)$ at times t_j, within the time window, with sampling time Δt;
2. preprocess data to facilitate the system identification;
3. apply a system identification algorithm to identify the linear system that given $\Delta P_e(t_j)$ in input returns $\Delta f(t_j)$ as output;
4. compute a metric that measures the estimation accuracy to accept or discard the estimated system;
5. compute the inertia constant from the estimated linear system;
6. slide the time window of a time step τ and repeat steps 1-6.

The same pipeline can be exploited to compute the equivalent inertia constant of the whole grid: for a system with N generators, each with inertia H_i and rated power $P_{0,i}$ $(i = 1, \ldots, N)$, the equivalent inertia constant is defined as

$$H_T = \frac{\sum_{i=1}^{N} H_i P_i}{\sum_{i=1}^{N} P_{0,i}} \tag{3.4}$$

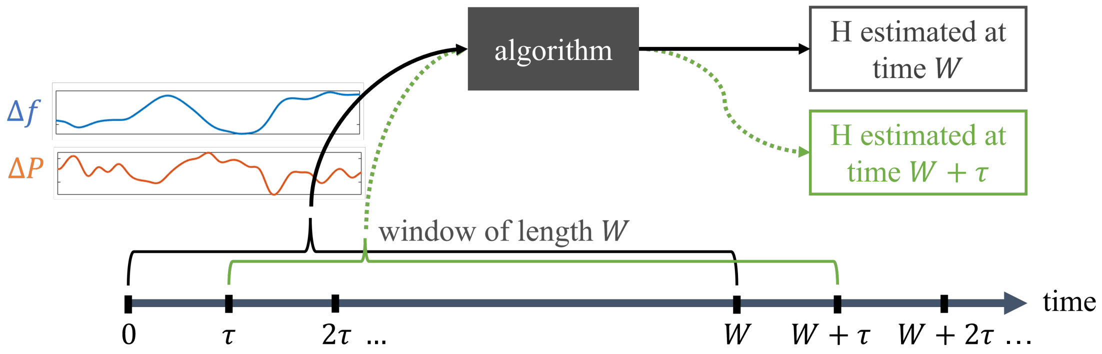


Figure 3.2: The inertia constant is estimated at each update time τ, based on the measurements of Δf and ΔP_e collected within a moving time window of length W.

It can be estimated by considering the frequency of the center of inertia

$$f_{COI} = \frac{\sum_{i=1}^{N} H_i f_i}{\sum_{i=1}^{N} H_i}, \tag{3.5}$$

and the total power

$$P_T = \frac{\sum_{i=1}^{N} P_i}{\sum_{i=1}^{N} P_{0,i}} \tag{3.6}$$

where f_i and P_i are the electrical frequency and active power measured through the PMUs at the bus connected to the i-th generator. During normal operation of the grid, both f_{COI} and P_T oscillate around their steady-state values, therefore the algorithm described above can be directly applied by considering Δf and ΔP_e as the difference between f_{COI} and its steady-state value, and P_T and its steady-state value, respectively.

3.3 Application 1: grid with photovoltaic power plant

3.3.1 Simulation framework

A slightly modified version of the well-known IEEE-14-bus power system, shown in Fig. 3.3, is used as a case study. All the generators are equipped with automatic voltage regulators. Eleven stochastic loads [213] were added on buses 2-6, 9-14, which absorb only active power $p_L(t) = n(t)P_{L,0}$, with nominal value $P_{L,0} = 0.1 MW$ where $n(t)$ is an Ornstein-Uhlenbeck's process such that $\dot{n} = \alpha(\bar{n} - n(t)) + b\zeta(t)$, where $\zeta(t)$ is Gaussian white noise with zero mean and variance equal to 1. The parameters are set to $P_{L,0} = 0.1$ MW, $n(0) = 0$, $\alpha = 0.5$, $b = 1$ and $\bar{n} = 0$. The purpose of this setup is to emulate realistic power fluctuations occurring in the power system.

A 2 MW PV power plant is connected to bus 11 (see Fig. 3.3). Behavioral models and controllers of DC/DC and DC/AC converters are used, as in [196]. In particular, the DC/AC converter controller provides synthetic inertia [198] by regulating its output power, in order to counteract frequency deviations, as synchronous generators do, and saturation blocks keep the capacitor voltage and current within security limits. Buses 1,

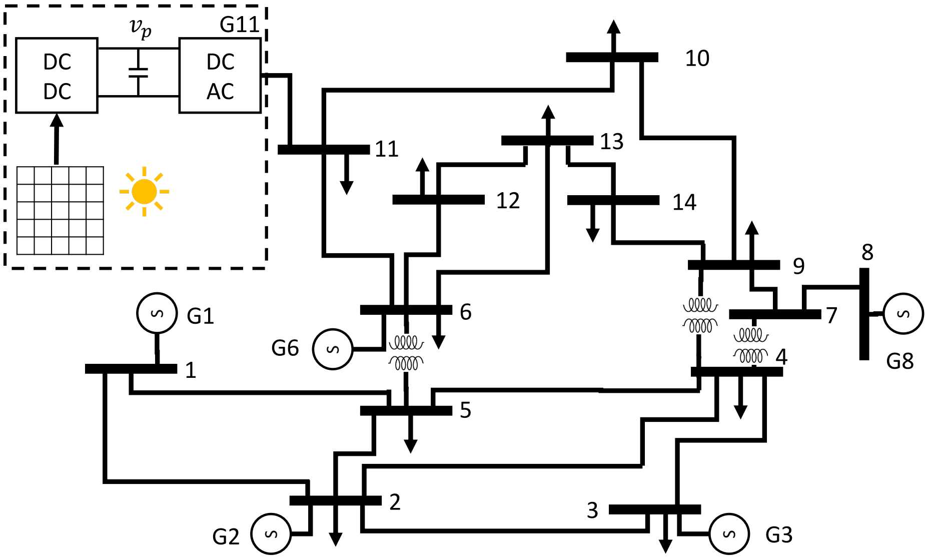


Figure 3.3: Scheme of the IEEE-14 network with a PV plant interfaced with the grid by means of a DC/DC and DC/AC converter.

2, 3, 6, 8, and 11 (connected to generators) are equipped with PMUs used to sample Δf and ΔP_e. Simulations of the considered power network are performed through the circuit simulator PAN [214, 215].

The real (estimated) inertia constant of the i-th generator ($i = 1, 2, 3, 6, 8, 11$) is denoted as H_i^* (H_i), whereas H_0^* (H_0) is the real (estimated) inertia constant of the whole grid. The inertia constants and rated powers of the generators in the IEEE 14 grid are listed in Tab. 3.1. The parameters of the PV controller have been set such

Table 3.1: Inertia and rated power of all generators in the IEEE14 grid.

	G1	G2	G3	G6	G8	G11
H_i^* [s]	5.148	6.540	6.540	5.060	5.060	1833
$P_{0,i}$ [MW]	615	60	60	25	25	2

that $H_{11}^* = 1833$s, resulting in a grid inertia constant $H_0^* = 10$s. The value of provided synthetic inertia depends on different factors and, in principle, any value of inertia can be synthesized, provided that some energy is always available. This justifies the high value of H_{11}^*. In the absence of the contribution of synthetic inertia given by the PV controller, the grid inertia constant is $H_0^* = 5.36$s.

3.3.2 Detailed pipeline and parameter tuning

Firstly, the standard IEEE-14 bus power system is simulated without the added complexity of the PV power plant, with the objective of tuning the pipeline steps to get the best inertia estimation performance on known H_i^* values. The grid is simulated for 1000 s with sampling time $\Delta t = 50$ ms. The inertia estimation is performed on $n_W = \lfloor \frac{1000-W+1}{\tau} \rfloor$ time-windows, with an update time $\tau = 1$ s. The signals Δf and ΔP_e are filtered with a 6-th order Butterworth low-pass filter with cut-off frequency 0.5 Hz, to preserve only frequency components that are significant in estimating the inertia constant [211]. The time window of length W is divided in two parts: the data in the first part of the time window (of length $\frac{3}{4}W$) is used to identify, through the n4sid algorithm [216], a linear system of order n with input $\Delta P_e(t_j)$ and output $\Delta f(t_j)$. The data in the rest of the window (of length $\frac{1}{4}W$) is used to validate the identified system, by computing the *fitting ratio*

$$\nu = \sum_j \left[\widehat{\Delta f}(t_j) - \Delta f(t_j)\right]^2 \Bigg/ \sum_j [\Delta f(t_j)]^2 \tag{3.7}$$

being $\widehat{\Delta f}(t_j)$ the output of the identified system. A value close to 0 indicates that the estimation is good. If ν is larger than a given threshold $\tilde{\nu}$ (fixed at $\tilde{\nu} = 5 \cdot 10^{-3}$), the estimation is considered not satisfactory, and a new value of inertia constant $H_i^{(k)}$ is not computed at the current time step k; the previously computed value is maintained instead: $H_i^{(k)} = H_i^{(k-1)}$.

Keeping in mind that index $i = 0$ corresponds to the whole grid, the following error measures are defined to quantify the inertia estimation performance:

$$\bar{\nu}_i = \frac{\sum_{k=1}^{n_W} \nu_i^{(k)}}{n_W}, \tag{3.8}$$

$$\epsilon_i = \frac{100}{H_i^*}\left(\bar{H}_i - H_i^*\right), \tag{3.9}$$

$$\text{RMSE}_i = \sqrt{\frac{\sum_{k=1}^{n_W}\left(H_i^{(k)} - H_i^*\right)^2}{n_W}}, \tag{3.10}$$

where $\bar{H}_i = \sum_{k=1}^{n_W} H_i^{(k)}/n_W$.

Role of system order

In [211] a system of order between 1 and 10 is identified through the n4sid algorithm, which automatically determines the order that yields the lower estimation error (ν closest to 0). Actually, a better system identification does not necessarily imply a better estimation of the inertia. In order to show this, inertia estimations with $W = 10$ were performed by varying n from 1 to 8. Fig. 3.4 shows $\bar{\nu}_0$ (top panel) and ϵ_0 (bottom panel) as a function of n. It appears that $n = 4$ actually yields the best system identification ($\bar{\nu}_0 = 9.3 \cdot 10^{-6}$), but the lowest error in the inertia estimation is obtained with a first-order system ($n = 1$), with $\bar{\nu}_0 = 9.9 \cdot 10^{-4}$, which is also quite low. Tab. 3.2 lists the

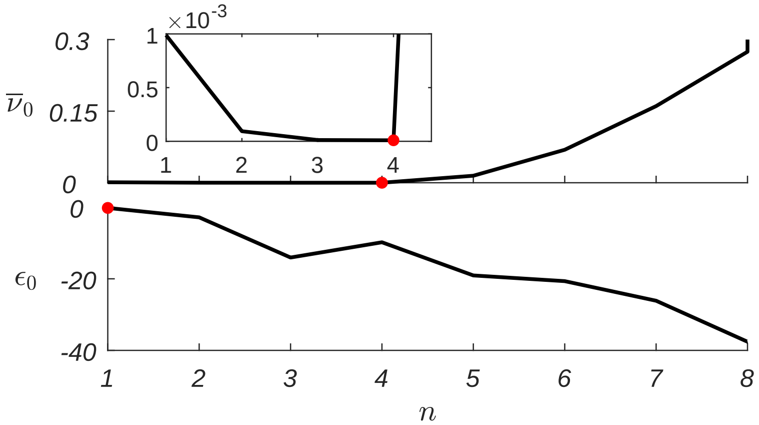


Figure 3.4: Error measures $\bar{\nu}_0$ (top panel) and ϵ_0 (bottom panel) as a function of n. The optimal values are marked in red.

results of this analysis for every generator and for the whole system, which evidences that the use of higher-order systems (as done in [211]) leads to a systematic underestimation of H_i since the initial slope of the step response is consistently steeper than that of the first-order system. Fig. 3.5 exemplifies the results of this analysis for the system as a whole and for generator G3, considering the first and fourth order systems: the initial slope (solid pink line) of the first order system step response (solid red line) is consistent with the expected slope (dashed black line), while the initial slope (light blue solid line) of the fourth order system step response (solid blue line) is clearly steeper.

If only a first-order system $\Delta f(t_{j+1}) = a\Delta f(t_j) + b\Delta P_e(t_j)$ is considered, then the initial slope of the step response (assuming $\Delta f(t_j) = 0$ and $\Delta P_e(t_j) = 1$) can be directly computed as $b/\Delta t$, where b is provided by the n4sid algorithm, without requiring to simulate the system response to a unit step and identifying the slope. The inertia can be therefore computed directly as

$$H_i = -\frac{\Delta t}{2b}, \tag{3.11}$$

which requires less computation time than the original approach [211].

Table 3.2: $\bar{\nu}_i$ and ϵ_i on generators and whole grid, for $n = 1$ and $n = 4$.

	$\bar{\nu}_i$		ϵ_i	
	$n = 1$	$n = 4$	$n = 1$	$n = 4$
$i = 0$ (whole grid)	$9.9 \cdot 10^{-4}$	$9.3 \cdot 10^{-6}$	-0.195	-9.792
$i = 1$	$5.4 \cdot 10^{-4}$	$1.4 \cdot 10^{-6}$	-0.327	-10.554
$i = 2$	$4.1 \cdot 10^{-3}$	$8.8 \cdot 10^{-6}$	0.185	-19.055
$i = 3$	$4.9 \cdot 10^{-3}$	$1.9 \cdot 10^{-5}$	0.563	-18.707
$i = 6$	$2.5 \cdot 10^{-2}$	$3.1 \cdot 10^{-4}$	4.774	-38.545
$i = 8$	$1.0 \cdot 10^{-2}$	$7.2 \cdot 10^{-5}$	-2.207	-29.889

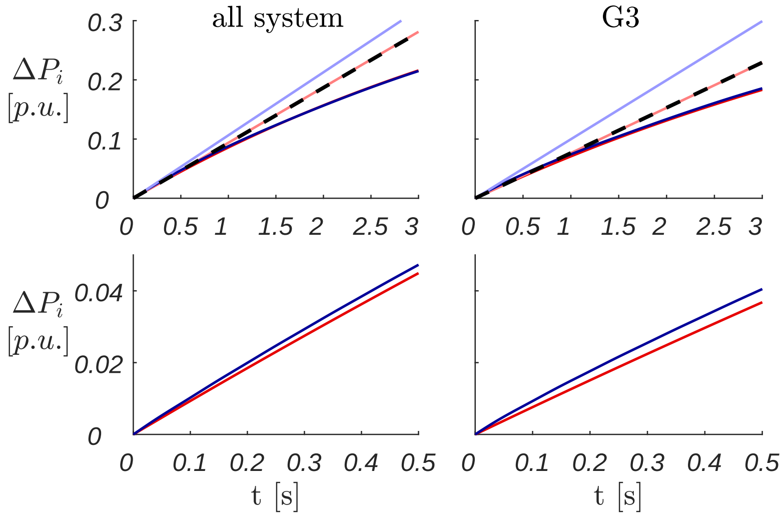


Figure 3.5: Step response of the fitted first order system (red) and fitted fourth order system (blue) and their respective initial slopes (pink and light blue) for the whole network (left panels) and generator 3 (right panels). Dashed black lines indicate the reference slopes. The first instants of the step responses are highlighted in the bottom panels.

Window length

To properly track the inertia constant in real-time, it is desirable to have the shortest delay possible in identifying significant changes. The length W of the time window, fixed at 200 s in [211], strongly influences this delay. Moreover, the longer W the higher the time necessary for identifying and validating the system. A shorter window, however, means fewer data available to identify the system, and thus a poorer accuracy of the inertia estimation. A systematic analysis of the effect of W on the quality of data fitting and subsequent inertia constant estimation, in terms of $\bar{\nu}_0$ and RMSE_0, was carried out to investigate the effect of varying W. The results with $n = 1$ and $W \in [5, 100]$ s are shown in Fig. 3.6. Fig. 3.7 shows the effect of W on the delay in detecting a change of inertia and on the accuracy of inertia constant estimation when H_1^* is artificially halved at $t = 250$ s. Notice that for $W = 5$ s the change of inertia is detected with a minimal delay (about 5 s), but large oscillations are visible in the estimated inertia, implying a higher value of RMSE$_1$. The opposite is true for $W = 50$ s, whereas $W = 10$ s represents a reasonable trade-off between delay and RMSE.

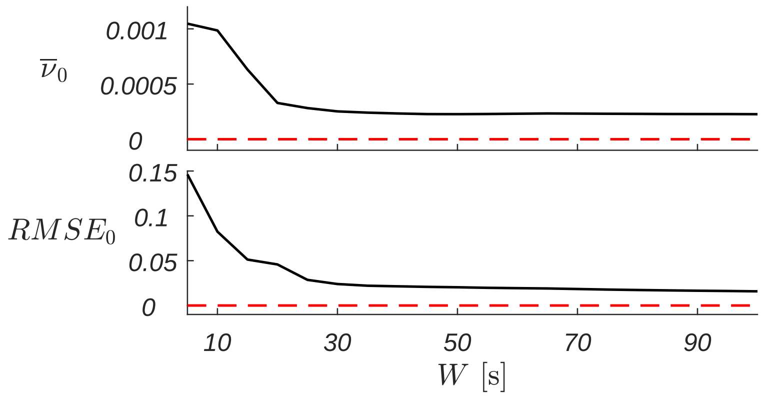


Figure 3.6: Error measures $\overline{\nu}_0$ and RMSE$_0$ versus window length W.

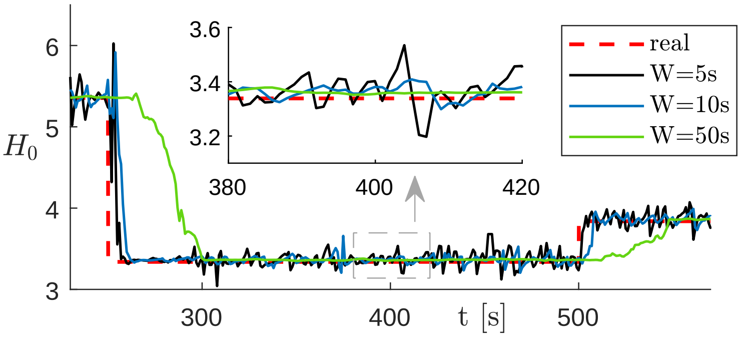


Figure 3.7: Tracked inertia constant for the whole system for different values of W. Red dashed lines mark the real values of H_0.

Computational efficiency

Assuming that the system inertia in normal conditions has small variations, it appears unnecessary to estimate a new system to fit the data for every considered time window. It is more efficient to make a new system estimation only when the previously estimated system does not fit the data of the current window, thus allowing a flexible refresh rate. A refresh condition can be defined based on the fitting ratio at iteration k: if $\nu^{(k)} > \nu^{(k-1)} + \rho$

a new estimation is carried out. The metric $r = 100\frac{n_e}{n_W}$ is introduced as the percentage ratio between the number of estimations n_e and the number of time windows. The impact of varying ρ on the quality of inertia constant estimation and on the number of new system estimations was investigated, using fixed $W = 10$ s and $n = 1$. The results are summarized in Fig. 3.8. As expected, the higher ρ the lower the number of new system estimations but also the higher the percent error on the mean value of the estimated inertia constant. Notice that r is always lower than 100%, because it may happen that $\nu_i^{(k)} < \nu_i^{(k-1)}$ and therefore the estimation is skipped also if $\rho = 0$. A value of $\rho = 10^{-5}$ was chosen as a good trade-off.

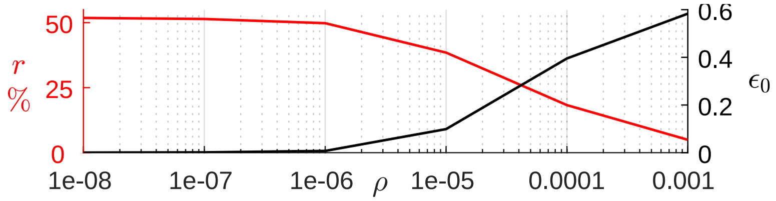


Figure 3.8: Percentage r of new system fittings and error measure ϵ_0 versus ρ, for the whole grid.

3.3.3 Results of inertia tracking

The results presented in this section are obtained by setting $n = 1$, $W = 10$ s, $T = \frac{3}{4}W$, $\tau = 1$ s, $\hat{\nu} = 5 \cdot 10^{-3}$ and $\rho = 10^{-5}$, according to the previous analysis. The IEEE-14 grid is simulated in three different scenarios.

Scenario 1

The grid is simulated under normal operating conditions for 1000 s, where all inertia values (including synthetic inertia) are kept constant. The estimated values of inertia, for all synchronous generators and for the whole grid (with the PV array), are listed in Tab. 3.3, together with ϵ_i, $\bar{\nu}_i$ and r. The percent estimation error ϵ_i is below 2% for the whole grid and generators 1-3. Higher values of ϵ_i and $\bar{\nu}_i$ are obtained for generators 6 and

8, which have a smaller rated power with respect to the others. The maximum (over all time-windows) computation time for system identification and inertia estimation is 0.15s (executed in Matlab on a laptop with an Intel core i7 2.6GHz processor) which would allow for a smaller update time τ and is about one half of the execution time (0.32 s) of the original algorithm proposed in [211]. These values do not take into consideration the acquisition of new data at each τ and the signal pre-processing.

Table 3.3: Performance of the inertia estimation in scenario 1.

	H_i^* [s]	$\overline{H}_i$ [s]	ϵ_i	$\bar{\nu}_i$	r
$i = 0$	10	9.9528	-0.4724	3.3e-4	35%
$i = 1$	5,148	5.1480	0.0010	2.6e-4	31%
$i = 2$	6,54	6.5982	0.8894	1.7e-3	43%
$i = 3$	6,54	6.6526	1.72254	2.8e-3	50%
$i = 6$	5,06	5.5569	9.8201	2.7e-2	62%
$i = 8$	5,06	5.3266	5.2688	1.7e-2	55%

Scenario 2

At time $t = 250$ s an extreme change is applied on the load connected to bus 11, causing the immediate saturation of the PV capacitor, leading to a sudden inactivity of the PV controller and to the consequent interruption of the synthetic inertia contribution, because the available energy is depleted. The load is restored at time $t = 375$ s. Fig. 3.9 shows that the algorithm correctly tracks the drop of H_0^*, from 10 s to the value only due to the synchronous machines (5.36 s).

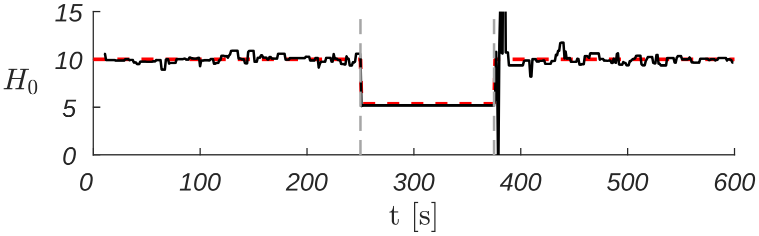


Figure 3.9: Estimated inertia constant of the whole grid H_0 (black line) in response to an extreme change load change. The real inertia constant H_0^* is shown in red. Dashed grey lines mark the time instants when the load is changed and then restored.

Scenario 3

A less extreme load change, for $t \in [250, 375]$ s, pushes the PV capacitor near the saturation point, leading to a less efficient controller response, which in turn causes a slight lowering of the synthetic inertia contribution. Fig. 3.10 shows that also the slight drop of inertia constant is detected by the algorithm. In this case, the result is only qualitative, since the real value of inertia constant during the load change is unknown, but it can reasonably be expected to be lower. A similar effect could result from a change in the solar irradiance, rather than in the load, which can also push the voltage of the PV capacitor close to the saturation point.

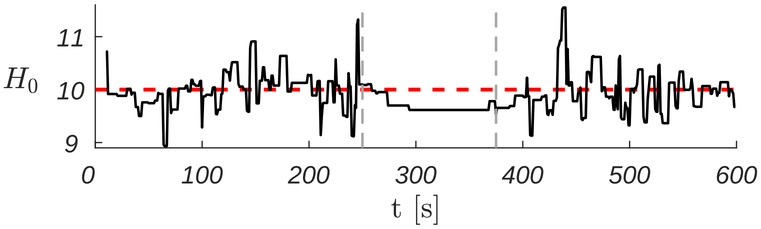


Figure 3.10: Estimated inertia constant of the whole grid H_0 (black line) in response to a less extreme change in a load. The red line marks the real inertia constant H_0^* when the controller operates in ideal conditions. Dashed grey lines mark the time instants when the load is changed and then restored.

3.4 Application 2: grid with turbine governors

Previously proposed algorithms [211, 217] that allow tracking the inertia constant in real-time under normal operating conditions do not take into consideration the action of turbine governors, which generally regulate synchronous generators and are also present in many RES control systems. These algorithms assume that, when the network is under normal operating conditions and power fluctuations are small, the mechanical power P_m never changes (i.e., its variation is 0). This is not true when a turbine governor is included in the control scheme.

This section describes an algorithm able to estimate the generator inertia also when the turbine governor is active, considering the most general scenario in which all parameters of the control scheme are assumed to be unknown.

The simulation framework is as described in section 3.3.1, with the only difference that the PV power plant is not present and generators at bus 1 and 2 are regulated by turbine governors [218, 219].

3.4.1 Detailed pipeline

Generally, the action of turbine governors can be modeled through a transfer function $T(s)$ between the input $\frac{\omega_0-\omega}{R} + P_{m0}$ and the output P_m, where ω and P_m are the rotor speed and mechanical power, respectively, whereas ω_0 and P_{m0} are their reference values, all measured in *per units* (p.u.). R is the droop coefficient (dimensionless) [45]. In this context, the governor is represented by a second-order transfer function, as a compromise between the simplicity of the model and approximation accuracy, taking into account the action of the two dominant poles:

$$T(s) = \frac{\alpha s + 1}{(\sigma s + 1)(\gamma s + 1)} \tag{3.12}$$

The complete system, including the dynamics of the turbine governor and of the swing equation, can be described by the block-chain diagram shown in Fig. 3.11. When the power grid works under normal operating conditions, the signals ω, P_e, and P_m are not perfectly constant, since they exhibit small oscillations around their nominal values due to fluctuations in the power absorbed by the loads. Under these conditions, considering the variations ($\Delta\omega$, ΔP_e, ΔP_m) around their steady-state values and knowing that $\Delta\omega$ can be approximated by the electrical frequency Δf (in p.u.) measured at the generator bus [211], the system shown in Fig. 3.11 can be written as a state-space model:

$$\begin{bmatrix} \Delta\dot{f} \\ \Delta\dot{P}_m \\ \Delta\dot{P}'_m \end{bmatrix} = \begin{bmatrix} -\frac{D}{2H} & \frac{1}{2H} & 0 \\ 0 & 0 & 1 \\ \frac{\alpha D-2H}{2HR\sigma\gamma} & -\frac{\alpha+2HR}{2HR\sigma\gamma} & -\frac{\gamma+\sigma}{\sigma\gamma} \end{bmatrix} \begin{bmatrix} \Delta f \\ \Delta P_m \\ \Delta P'_m \end{bmatrix} + \begin{bmatrix} -\frac{1}{2H} \\ 0 \\ \frac{\alpha}{2HR\sigma\gamma} \end{bmatrix} \Delta P_e \tag{3.13}$$

being $\Delta\dot{P}_m = \Delta P'_m$ the time derivative of the mechanical power variation. The electrical frequency Δf (state variable) and the electrical active power ΔP_e (considered as an input) can be measured at the generator bus through PMUs.

Since the model structure is known, the parameter vector referred to the i-th generator $\xi_i = [H_i, D_i, R_i, \alpha_i, \gamma_i, \sigma_i]$ that best fits the time evolution of the state variable Δf, given the input profile ΔP_e, can be determined through optimization. As in the previous case, the optimization process is carried out considering the signals Δf and ΔP_e within a moving time window of length W [s], with an update time τ [s], so that the inertia constant H_i is estimated at each update time; the total number of time windows is n_w.

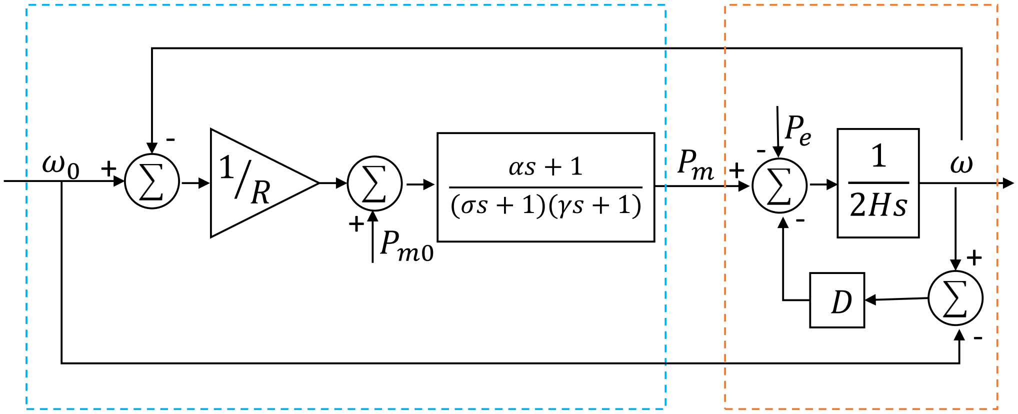


Figure 3.11: Block-chain diagram of the system, modeling the governor action (blue dashed box) and the swing equation(orange dashed box).

In this framework, the H_i value estimated at a certain instant refers to the measures acquired in the previous time window.

The optimization problem is formulated as

$$\min_{\xi_i} \Psi(\xi_i; \{\Delta P_{ej}\}, \{\Delta f_j\}) \tag{3.14}$$

where $\{\Delta P_{ej}\}$ and $\{\Delta f_j\}$ are arrays whose elements ΔP_{ej} and Δf_j represent measurements of the electrical power and frequency variations, respectively, at evenly spaced times t_j $(j = 1, \ldots, n)$ within a time window. All model parameters (vector ξ_i) are included in the optimization process since they cannot be assumed to be known a priori in realistic scenarios. The cost function Ψ_i is defined as

$$\Psi_i = g_p \frac{\sum_{j=1}^{n} (\widetilde{\Delta P_{mj}} - \widehat{\Delta P_{mj}})^2}{n} + g_f \frac{\sum_{j=1}^{n} (\Delta f_j - \widehat{\Delta f_j})^2}{n} \tag{3.15}$$

where $\widehat{\Delta P_{mj}}$ and $\widehat{\Delta f_j}$ are the mechanical power and frequency variations at time t_j, respectively, estimated by numerically integrating system (3.13). It is apparent from Eq. (3.13) that $\widehat{\Delta P_{mj}}$ and $\widehat{\Delta f_j}$ referred to the i-th generator depend on the parameter vector ξ_i. Since measurements of the mechanical power are not available through PMUs, approximated values $\widetilde{\Delta P_{mj}}$ obtained through the swing equation are used to calculate the cost function:

$$\widetilde{\Delta P_{mj}} = D\Delta f_j + 2H_i \dot{\Delta f}_j + \Delta P_{ej} \tag{3.16}$$

Here, $\dot{\Delta f}_j$ denotes the time derivative of the frequency evaluated at time t_j. The computation of the cost function is summarized in Fig. 3.12.

The above nonlinear optimization problem is solved through the Matlab function *fmincon*, which exploits an Interior-Point Algorithm [220], constraining all parameters to be positive. To improve the chances of reaching a global minimum, in each time window the optimization is carried out for 10 different initial conditions, 9 of which are chosen randomly, and one corresponds to the optimization result in the previous time window. The parameter vector ξ_i leading to the minimum cost Ψ_i^* is selected. If Ψ_i^* in a certain time window is above a fixed threshold Ψ_T, the optimization in that window is deemed unsuccessful and the corresponding H_i is discarded.

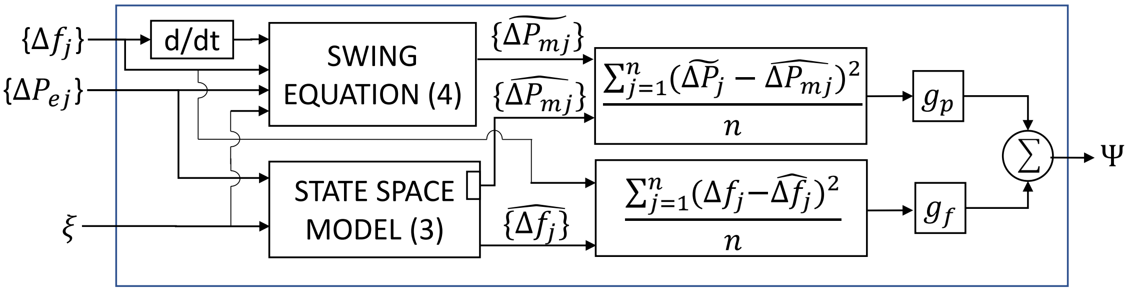

$\{\Delta f_j\}$
$\{\Delta P_{ej}\}$
ξ
d/dt
SWING EQUATION (4)
STATE SPACE MODEL (3)
$\{\widetilde{\Delta P_{mj}}\}$
$\{\widehat{\Delta P_{mj}}\}$
$\{\widehat{\Delta f_j}\}$
$\frac{\sum_{j=1}^{n}(\widetilde{\Delta P_j} - \widehat{\Delta P_{mj}})^2}{n}$
$\frac{\sum_{j=1}^{n}(\Delta f_j - \widehat{\Delta f_j})^2}{n}$
g_p
g_f
$\sum$
Ψ

Figure 3.12: Block diagram that summarizes the cost function calculation at a generic step of the optimization process.

3.4.2 Results of inertia tracking

The algorithm was tested on a 1000 s simulation of the IEEE 14-bus during normal operating conditions for different window lengths to assess the effect of W on the inertia constant tracking. The update time τ is set to 1 s in all cases and coincides with the sampling period of the signals. The cost function threshold is set heuristically as $\Psi_T = 1.5 \cdot 10^{-7}$ and the weights are set as p_m=1 and p_f=1000; p_f is chosen greater than p_m because the term based on actual measurements is considered more reliable and is associated with a much smaller error. Note that it is important to set $p_m \neq$0, otherwise, the optimization process fails. The results are summarized in Fig. 3.13: the top panel shows the relative percent error ϵ_i (see Eq. (3.9)) versus the window length W for generators 1 (black line) and 2 (blue line). The bottom panel shows how the root mean square error RMSE$_i$ (see Eq. (3.10)) decreases as the window length increases. The two metrics ϵ_i and RMSE$_i$ give information about how well estimates $\bar{H}_i$ approximate H_i^* and how much the estimated values fluctuate from window to window, respectively. The best results are obtained for longer windows, which, however, cause higher delays in the inertia constant estimation. A window length of $W = 30s$ is chosen as the best compromise between delay in detecting inertia constant changes and accuracy of the inertia constant estimation. Figures 3.14 and 3.15 show the inertia estimation results for

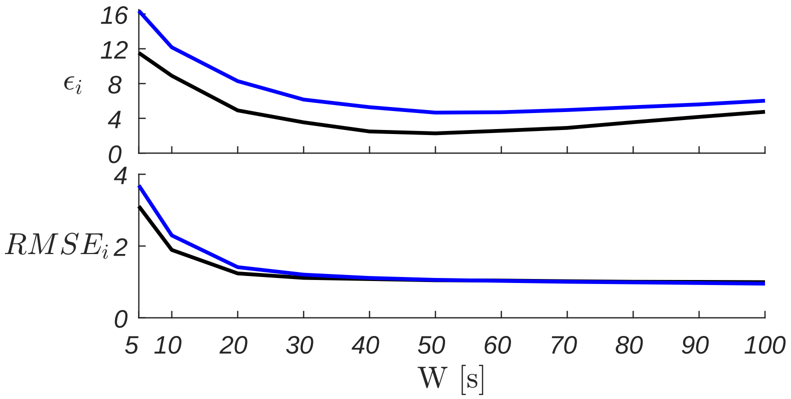


Figure 3.13: Top panel: relative percent error ϵ_i vs the window length W for generators 1 (black) and 2 (blue). Bottom panel: RMSE$_i$ vs W for generators 1 (black) and 2 (blue).

generators 1 and 2 for a window of 5 s (top panel, short delay but high RMSE$_i$), 30 s (mid panel, deemed as the best compromise) and 100 s (bottom panel, low RMSE$_i$ but long delay). Fig. 3.16 shows the comparison between the samples $\{\Delta f_j\}$ as measured by PMUs (red) and the estimated $\{\widehat{\Delta f_j}\}$ (black) in the first window of 100s. The accuracy is very high, which demonstrates that the optimization is successful in the identification of the system parameters.

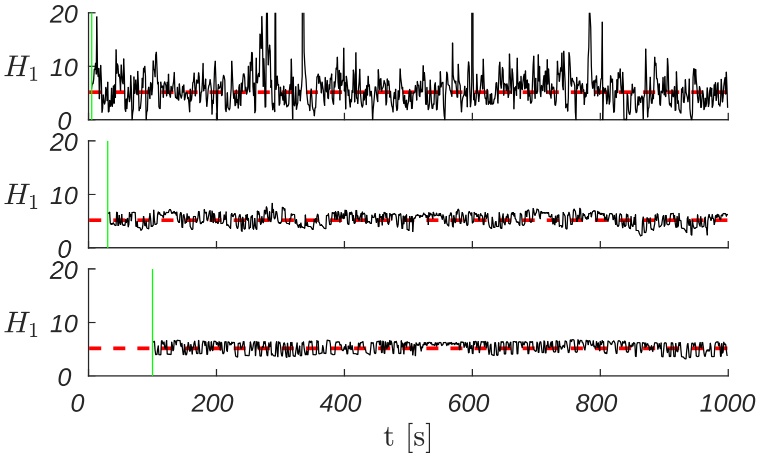


Figure 3.14: Generator 1 inertia estimation for $W = 5$ s (top panel), 30 s (mid panel) and 100 s (bottom panel). The red dashed line indicates the real value of the inertia constant and the green vertical line marks the start time of the inertia tracking (coinciding with W).

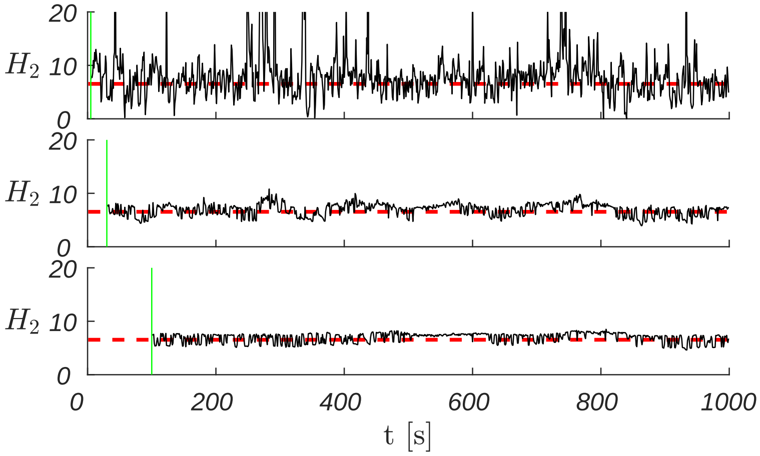


Figure 3.15: Generator 2 inertia estimation for W =5 s (top panel), 30 s (mid panel) and 100 s (bottom panel). The red dashed line indicates the real value of the inertia constant and the green vertical line evidences the start time of the inertia tracking (coinciding with W).

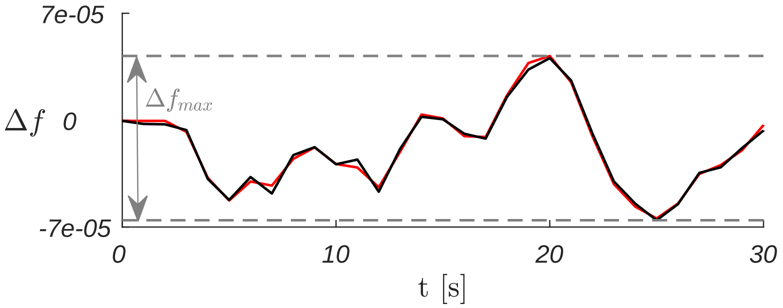


Figure 3.16: Samples $\{\Delta f_j\}$ as measured by PMUs (red) compared with the estimated $\{\widehat{\Delta f_j}\}$ (black) in the first window of 30 s for generator 1.

Table 3.4 lists the actual inertia H_i^* of all generators, the average estimated inertia $\bar{H}_i$, ϵ_i, RMSE$_i$ and ϵ_{f_i}, calculated as

$$\epsilon_{f_i} = \frac{1}{n_w} \sum_{i=1}^{n_w} \left(\frac{100}{\Delta f_{max}} \sum_{j=1}^{n} \frac{(\Delta f_j - \widehat{\Delta f_j})^2}{n} \right) \tag{3.17}$$

where Δf_{max} is the interval between the minimum and maximum values of Δf.

Table 3.4: Performance of the inertia estimation

	$\bar{H}_i$ [s]	H_i^* [s]	ϵ_i	RMSE$_i$ [s]	ϵ_{f_i} [p.u.]
i=1	4.731	5.148	3.541	1.113	$3.159 \cdot 10^{-5}$
i=2	5.9110	6.540	6.159	1.202	$3.965 \cdot 10^{-5}$
i=3	6.614	6.540	1.952	0.629	$2.546 \cdot 10^{-5}$
i=6	5.169	5.060	2.330	0.397	$4.583 \cdot 10^{-5}$
i=8	5.141	5.060	-2.162	0.832	$2.901 \cdot 10^{-5}$

Data indicates that the inertia is well estimated, with a low percent error, for all generators, both equipped and not equipped with turbine governors, accounting for the versatility of the algorithm. The values of RMSE$_i$ are slightly higher for generators with turbine governors, suggesting that the estimation may exhibit higher oscillations among the time windows. ϵ_{f_i} values are very low for all generators, confirming that the optimization is overall successful in the identification of the system parameters.

The selected window of $W = 30$ s is also used to track the inertia constant of generator 1 in a case where the generator inertia constant H_1^* changes from 5.148 s to 10 s at time instant $t = 500$ s. The result of the simulation is shown in Fig. 3.17. The band in gray highlights the time window of 30 s corresponding to the delay in detecting the inertia constant change. The percent error in the time stretch where $H_1^* = 10$ s is $\epsilon_1 = -7.715$.

The window length of 30 s is considerably shorter than the one used in [211], while the percent error on the estimated inertia constant is comparable with similar approaches [211, 217].

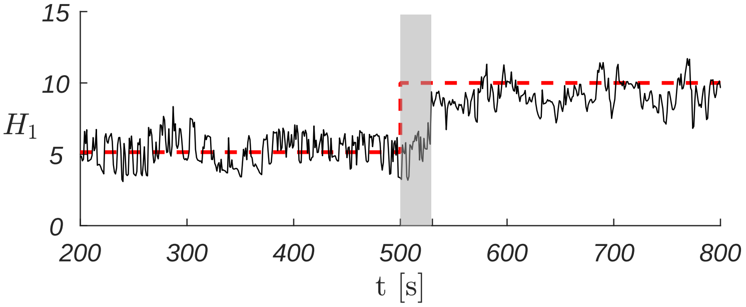


Figure 3.17: Generator 1 inertia constant tracking for $W = 30$ s in a scenario where the inertia constant is varied from 5.148 s to 10 s at $t = 500$ s. The red dashed line indicates the real value of the inertia constant.

3.5 Application 3: synthetic inertia control system

Most control systems produce synthetic inertia by imitating, ideally, the action of synchronous generators, on the basis of the swing equation. In practice, an additional power contribution, proportional to the rate of change of frequency (RoCoF), is provided by the RES inverter, to alleviate the frequency drop due to a fault or a load/production change. These controllers need to measure the electrical frequency online and estimate its time derivative. Typical devices used to measure the frequency are phased locked loops (PLLs) or transducers performing digital period measurements. Both have a dynamic response and introduce delays and noise, which prevents the evaluation of the RoCoF as a simple incremental ratio. Therefore, filtering is necessary, which raises further delay and distortions. Moreover, in real-world applications, the effects of analog filtering and quantization due to fixed-point data representation and analog-to-digital/digital-to-analog conversion are present. All delays, distortions, and noise have an impact on the inertial response of the inverter [221], making the produced synthetic inertia swerve from its desired nominal value.

Some studies have analyzed the frequency behavior under the effect of delays and/or noise, by resorting to simulations: in [222, 223, 224] the frequency stability of a low-inertia microgrid was investigated by considering communication delays, which decrease the ability to properly control the frequency. In [225, 226] the effects on synthetic inertia of frequency measurement delay and PLL dynamics are considered, by showing that the PLL dynamics could cause stability issues. The algorithm for online inertia estimation can be used as a tool to provide a quantitative evaluation of the synthetic inertia, in the presence of delays due to measurements and filtering.

This section details how the algorithm can be adapted to measure the synthetic inertia contribution provided by a controller for wind turbine generators (WTGs); both software simulations and experimental hardware-in-the-loop tests with a hardware prototype of the synthetic inertia controller [227] are exploited to consider the impact of analog filtering, data quantization, and measurement noise. In particular, the considered control system is the virtual hidden inertia emulator (VHIE) for WTGs proposed in [228]. During the frequency support modality, that will be hereafter considered, the additional output power of the controller is calculated as

$$P_{VHIE} = 2H(\omega)f\frac{df}{dt} \tag{3.18}$$

The synthetic inertia constant H depends on the WTG rotor speed ω because the frequency support action needs to be more effective during the first part of the transient, when the RoCoF is more severe, and should reduce its contribution when the rotor speed decreases, in order to avoid the stall of the wind turbine. For this reason, H is changed during the frequency support so that its maximum value is provided at the start and becomes zero for a limit value of the WTG rotor speed ω_{min}:

$$H(\omega) = H_0\frac{\omega - \omega_{min}}{\omega_0 - \omega_{min}} \tag{3.19}$$

where ω_0 is the rotor speed at the beginning of the frequency support phase.

The hardware prototype of the VHIE and the corresponding Simscape model have been developed in the context of the research project SMART WIND [229] and are described accurately in [227]. The VHIE is embedded in a Digilent Nexys Video Artix-7 FPGA board and the frequency signal is measured with a SINEAX F534 frequency transducer. The whole circuit's maximum latency is 1.12 μs

3.5.1 Synthetic inertia contribution in a network

A simple power grid is considered, including a 40 MVA wind farm, composed of 20 WTGs with a rated power of 2 MVA and initial inertia $H_0 = 17.5$ s, and two synchronous generators, G2 with a rated power of 200 MVA and an inertia constant $H_2 = 3.7$ s, G3 with a rated power of 100 MVA and an inertia constant $H_3 = 3.7$ s (see Fig. 3.18). Both the wind farm and the network are implemented and simulated in Simscape. At $t = 20$ s the load highlighted by the green circle in Fig. 3.18 (with active power 30 MW and inductive-reactive power 5 MVAr) is connected to the grid, generating a power

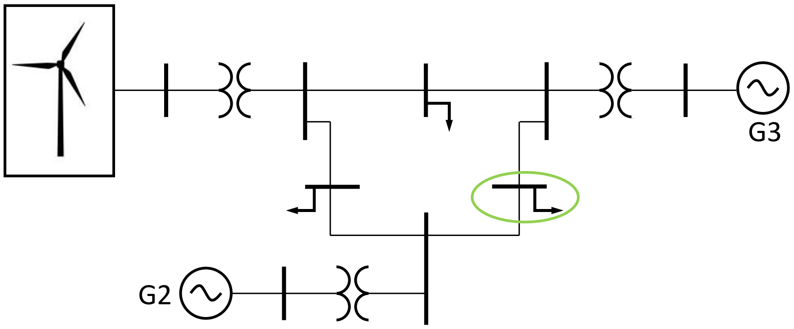


Figure 3.18: Scheme of the simple power grid including the WTG equipped with the VHIE (rated power of 40 MVA, H_0 of 17.5 s) and two synchronous generators (rated power of 200 MVA and H of 3.7 s for G2, rated power of 100 MVA and H of 3.7 s for G3).

imbalance that causes a frequency drop. During the frequency support, the active power P_e and the frequency f at the WTG bus are measured with sampling time $\Delta t = 10$ ms and the deviations Δf and ΔP_e from their nominal values are calculated. The synthetic inertia contribution to the network can be estimated assuming that Δf and ΔP behave according to the linear system $2H\dot{\Delta f} = -\Delta P_e + D\Delta f$ and minimizing the mean square error:

$$\epsilon = \frac{\sum_{j=1}^{n}(\Delta f_j - \widehat{\Delta f_j})^2}{n} \tag{3.20}$$

where Δf_j are the measurements of the frequency variation at evenly spaced times t_j ($j = 1, \ldots, n$) within a moving time window of $W = 0.2$ s and $\widehat{\Delta f_i}$ is the frequency variation obtained as the output of the linear system. In this case, the moving window can be significantly smaller than in applications 1 and 2 because the deviations Δf and ΔP_e obtained in response to the power imbalance are much steeper (they change more in the same time window) than the deviations caused by power fluctuations at steady state. The update time τ is set as short as possible, equal to the sampling time of 10 ms. The optimization is carried out with the Matlab function *fmincon*, repeating the process for 10 random initial conditions, to avoid local minima, and selecting the result of the iteration that reached the lowest value of ϵ. The result is shown in Fig. 3.19. Due to internal delays, the synthetic inertia production starts about 70 ms after the power imbalance (see the inset). For the following 0.2 seconds (the length of window W), corresponding to the gray area in the figure, the inertial contribution is present but the estimation algorithm does not provide any valid value. Notice that, besides the delay, the synthetic inertia provided by the wind farm exhibits significant oscillations. Such oscillations are due to the controller's internal delays, which cause a misalignment between the frequency and power signals. This phenomenon is better analyzed and quantified in the next section.

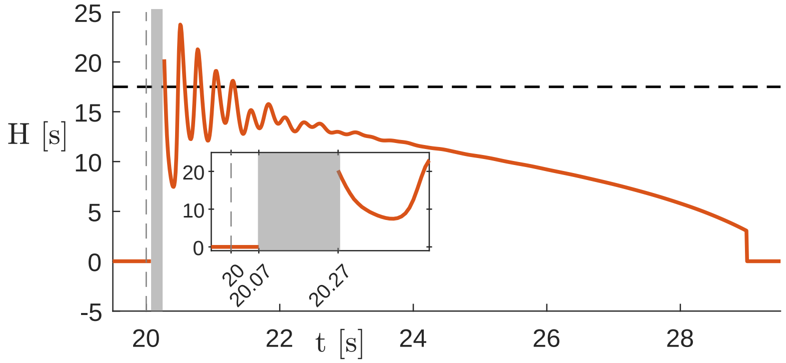


Figure 3.19: Results of the estimation of the synthetic inertia provided by the wind power plant. The horizontal dashed line marks the nominal H_0 value and the vertical dashed line marks the power imbalance onset. The gray area corresponds to the window $W = 0.2$ s needed by the algorithm to output the first estimation result.

3.5.2 Effect of delays in synthetic inertia production

To study the effect of the controller's internal delays on the synthetic inertia contribution, the WTG equipped with the VHIE is studied individually. In particular, a single WTG with a nominal power of 2 MVA is considered, connected to a 0.8 MVA grid modeled as a frequency-dependent voltage source, with an imposed frequency profile $f(t)$, shown in the top panel of Fig. 3.20. This profile has been obtained by simulating a realistic test power grid with low inertia through DIgSILENT PowerFactory®. Notice the presence of oscillations in the frequency profile, which are more evident in the RoCoF (bottom panel of Fig.

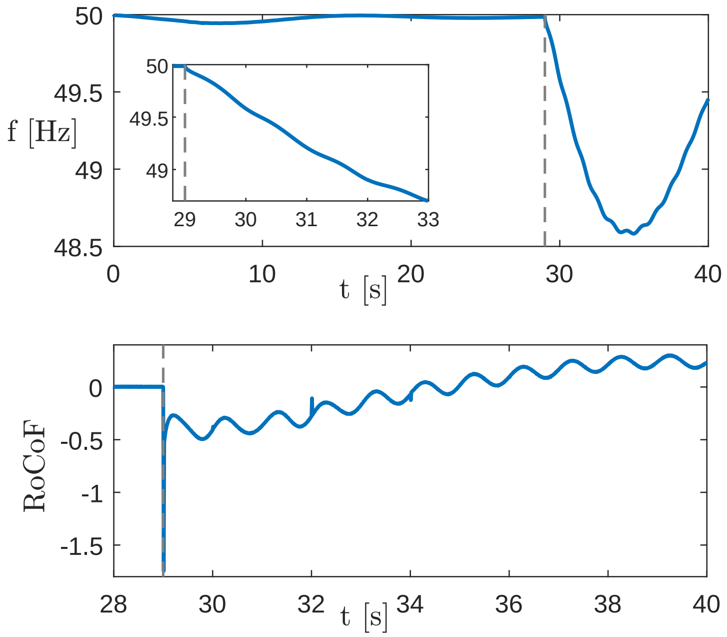


Figure 3.20: Imposed frequency profile (top panel) and corresponding RoCoF (bottom panel). The dashed vertical line marks the frequency drop at $t = 29$ s. The inset evidences the oscillations in the frequency profile, that become evident in the RoCoF.

3.20). The two primary sources of internal delays in the controller are the differentiator used to calculate the derivative df/dt and the transducer used to measure the frequency signal. The transducer needs at least two cycles to give an accurate measurement of the frequency signal, so it introduces a minimum delay of 40 ms in the case of 50 Hz systems. The circuit latency is 4 orders of magnitudes smaller than these delays, and its effect is thus negligible. The differentiator is used to effectively calculate the derivative of the noisy frequency signal; it approximates the derivative through a finite-impulse-response (FIR) filter of order N:

$$\left.\frac{df}{dt}\right|_{t=t_j} \approx \sum_{n=0}^{N-1} a_n f(t_{j-N+1+n}) \tag{3.21}$$

The higher the filter order, the smoother the signal df/dt, but also the longer the delay introduced by the operation and the stronger the signal distortion. On the contrary, low filter orders introduce little delay and signal distortion but produce a noisier df/dt signal.

The filter order should be high enough to guarantee a smooth signal df/dt, since ample oscillations in df/dt due to noise can cause improper behavior of the VHIE.

The Simscape model of the VHIE is used to simulate the controller behavior for different filter orders and different transducer delays. When considering only the VHIE, the active power deviation signal of interest is the output power reference of the controller P_{VHIE}: it represents the deviation from the power provided by the WTG without the controller. The simulated signals do not suffer from noise, so it is possible to calculate the synthetic inertia contribution algebraically as:

$$H = \frac{P_{VHIE}}{df/dt} \tag{3.22}$$

The result is shown in Fig. 3.21: the top panel refers to the synthetic inertia contribution provided when the transducer delay is null (to isolate only the differentiator effect) and the filter order is varied from 3 to 27. Note that using filter orders as low as 3 is only possible in a simulation scenario, where the frequency signal is not affected by noise. The lowest filter order introduces minimal signal distortion and delay, and thus the provided synthetic inertia (blue line) coincides almost perfectly with the target inertia imposed by the controller (black line). As the filter order increases, the oscillations amplify accordingly. The bottom panel refers to the synthetic inertia contribution provided when the filter order is fixed at 21 (standard value used in the VHIE prototype) and the transducer delay varies from 0 ms (ideal, but unattainable condition) to 160 ms. As expected, the oscillations get more pronounced as the delay increases; they are present even for null transducer delay since the delay introduced by the differentiator is still present. It can also be observed that both the filter order and the transducer delay influence the delay in deploying synthetic inertia after the frequency drop at $t = 29$ s, as evidenced in the insets. The delays go from 0 ms ($N = 3$ and transducer delay of 0 ms) to 240 ms ($N = 21$ and transducer delay of 160 ms).

Hardware-in-the-loop simulations are also carried out, where the VHIE prototype is connected (through analog interfaces) to SpeedGoat, a high-performance real-time simulator. In this case, the frequency measuring delay is fixed at 40 ms by the actual

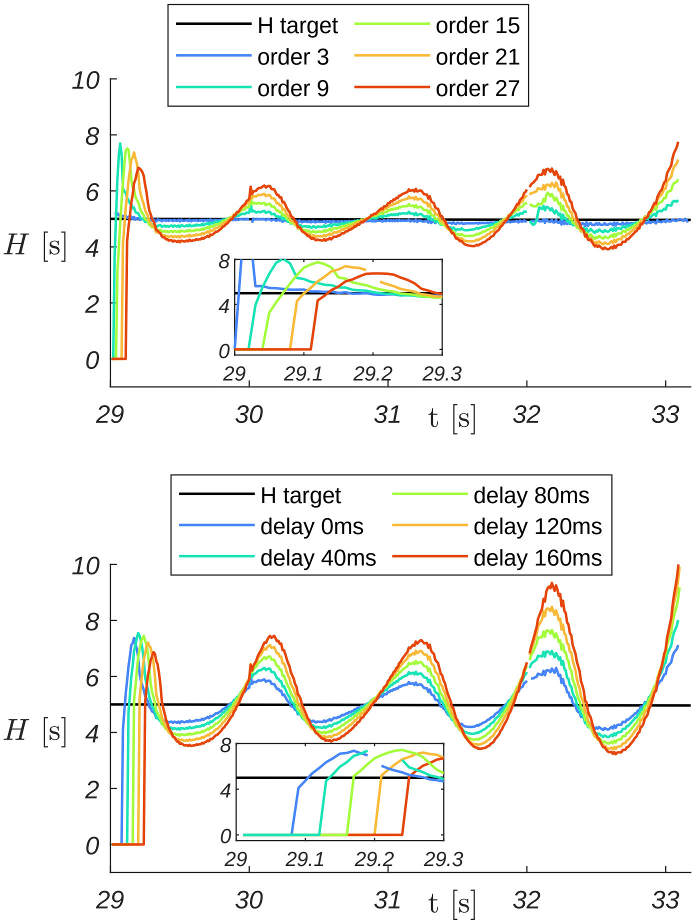


Figure 3.21: Result of estimation of the synthetic inertia provided by the controller without transducer delay and varying filter order (top panel) and with filter order fixed at 21 and varying transducer delay (bottom panel). The black line marks the target H value imposed by the controller, colored lines are the estimated inertia.

physical transducer, while the differentiator order can be changed by reprogramming the FPGA. Due to signal noise, the lowest filter order that barely allows the correct operation of the VHIE is 5. Moreover, the synthetic inertia contribution cannot be calculated algebraically and needs to be estimated as explained in section 3.5.1, with ΔP_e being P_{VHIE}. The results are shown in the bottom panel of Fig. 3.22. For comparison purposes, the top panel shows the same results obtained with the Simscape model, not affected by measurement noise. As can be seen, noise is prevalent in H traces obtained with low filter order, so much so that the advantage given by smaller oscillations in H values corresponding to smaller delays is canceled by the oscillation caused by noise. This analysis highlights that real-world effects, including measurement noise, delays, and filtering distortion, can significantly alter its synthetic inertia production in comparison with the target H value imposed by the controller. Although the delays cannot be avoided, being mindful of their effect and quantifying their impact during the controller's designing phase can improve the controller functionality and help redirect choices in terms of the controller parameters.

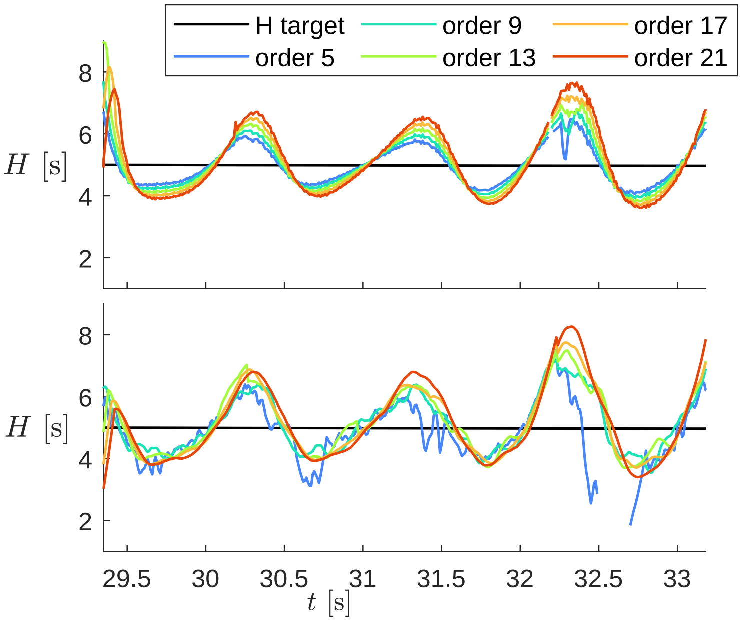


Figure 3.22: Synthetic inertia provided by the VHIE Simscape model (top panel) and the hardware VHIE prototype (bottom panel) with fixed transducer delay (40 ms) and varying filter order. The black line marks the nominal H value, colored lines are the estimated inertia. The data in the top panel (obtained algebraically) is shifted forward of $W = 0.2$ s so that it aligns with the data in the bottom panel (obtained with the inertia estimation algorithm). For $N = 5$ (blue line) the VHIE exits the frequency support regime for a short period; the resulting outlier data has been identified with Matlab's function *rmoutliers* and removed from the plot (from 32.39 s to 32.68 s).

3.6 Concluding remarks

This chapter discussed the implementation of an algorithm for the online estimation of inertia constant in power grids and how it can be adapted to be successfully applied in different scenarios. The proposed algorithm as utilized in the first two applications is not intended as a fault detection tool, nor to estimate the inertia constant in short time windows on post-fault data. Its purpose is online monitoring of the inertia of power grids when it becomes a time-varying quantity due to the presence of renewable energy sources, over a timescale compatible with weather changes.

Inertia estimation during normal operating conditions of the network has become a pivotal problem in ensuring stable energy distribution since the increasing presence of renewable energy sources in power grids. Online inertia monitoring can indeed play an important role in applying proper control methods to alleviate stability issues.

The proposed improved estimation algorithm was tested on the IEEE-14-bus power system with the addition of a PV power plant equipped with a controller for synthetic inertia generation. By exploiting the data measured online through phasor measurement units, the algorithm was able to successfully estimate the inertia constant value of each generator and of the whole network and to track its changes in different operating conditions. The window length needed to obtain a percent error on the estimated inertia constant comparable to similar approaches [211, 217], was considerably shorter than the one used in the original version of the algorithm [211]. The execution time was also halved in comparison with the original algorithm [211].

The algorithm was also extended to account for the presence of turbine governors in the power grid and was again tested on the IEEE 14-bus power system, this time with active turbine governors, obtaining equally good results: the window length is considerably shorter than the one used in [211], while the percent error on the estimated inertia constant is comparable with similar approaches [211, 217].

In the third application, the algorithm was adapted to estimate the inertia constant in a much smaller time window on transient data after a power imbalance. This case study shows how inertia estimation can be used as a tool in RES controllers design, since it allows to analyze how the synthetic inertia contribution is altered by the controller's internal delays. Although the analysis was specifically carried out for the VHIE controller, the same approach could be applied to different control strategies, since the problem of internal delays is not limited to this specific case.

Conclusions

This final section discusses how the three considered case studies align with the overarching framework of parameter identification in networks of dynamical systems. More specific considerations on the obtained results for each of the three case studies, how they are positioned within the current state of the art, and their potential applications are discussed in more detail in the *concluding remarks* sections at the end of each chapter (sections 1.6, 2.7 and 3.6).

This thesis has explored the problem of parameter identification in networks of dynamical systems, by analyzing three seemingly unrelated case studies that, fundamentally, share the same challenges: understanding how node parameters, connection features and network topology influence the emergent dynamics of the network, and tuning them accordingly. However, the individual problems posed by each scenario need to be dealt with using tailored approaches.

The CPG network represents a case with a very simple topology but complex nonlinear node dynamics, where each node is represented by an accurate burster neuron model. Moreover, the links are synaptic connections of different types, some characterized by their own dynamics. The collective behavior of the network should exhibit different features, in the terms of frequency, duty cycle and phase lags of each node, depending on the value of the control signal and according to quadruped gaits features. This complex scenario has been handled by developing a custom parameter identification strategy that takes into account the biological plausibility of the parameter values and the physical principles governing the gait transitions, using bifurcation analysis as a tool. When compared with state-of-the-art global optimization algorithms, the proposed custom strategy shows better efficiency in obtaining the desired emergent network behavior and helps to shed light on the role and effect of each network parameter.

Brain networks are characterized by a high number of nodes and a very complex topology, with thousand of weighted links; the nodes are represented by oscillators that accurately reproduce the nonlinear dynamics of the average neural activity of brain areas. In this case, the emergent behavior of the network model must fit the experimentally measured functional imaging data. To achieve this goal, the proposed parameter identification method focuses on the optimization of connection weights, mixing data-driven and model-based approaches. It is informed by both functional and structural imaging data and by network theory and nonlinear dynamics concepts, such as balanced coloring and cluster synchronization.

The considered power grids are networks of medium to small size where nodes are represented by synchronous generators, loads and renewable energy sources. The collective behavior of the network needs to allow safe operating conditions of the power system, ensuring its robustness to perturbations. In this case, the topology of the networks and the node dynamics are given, and a parameter important to the network stability, i.e., the inertia constant, needs to be inferred from measured signals. The parameter identification strategy is based on fitting the node model in its simpler form, the swing equation, to properly pre-processes power and frequency signals measured at significant network locations. The method is flexible enough to be adapted to different scenarios and applications.

Appendices

Appendix A

A.1 Thalamic Reticular Neuron Model

The thalamic reticular neuron model [112, 113] is defined by the following state equations:

$$\begin{cases} \xi \dfrac{dV}{dt} = \dfrac{-I_T - I_L - I_{Na} - I_K - I_c + I^{syn}}{C} \\ \xi \dfrac{dCa}{dt} = -\dfrac{kI_T}{2Fd} - \dfrac{K_T Ca}{Ca + K_d} \\ \xi \dfrac{dy}{dt} = \dfrac{y^\infty - y}{\tau_y}, \quad y = \{h, m, n, m_T, h_T\} \end{cases} \tag{A.1}$$

where V is the membrane potential of the neuron; the only difference with respect to the original model is the presence of the dimensionless coefficient ξ, which scales the time variable t and thus determines the burst frequency f (obtained in step 1 of section 1.4.2 by varying I_c in the burst region) so that it spans the range required to model the desired gaits. The ion currents I_T (calcium), I_{Na} (sodium), I_K (potassium), and I_L (leakage) evolve according to the following equations

$$I_T = g_{Ca} m_T^2 h_T (V - E_{Ca}), \quad I_L = g_L (V - E_L),$$
$$I_{Na} = g_{Na} m^3 h (V - E_{Na}), \quad I_K = g_k n^4 (V - E_k),$$

which depend on V, on the intracellular calcium concentration Ca and on a set of further state variables (called *gating variables*) h, m, n, m_T, h_T. The differential equations governing these gating variables have the common structure written above (for the generic gating variable y), where:

$$
\begin{aligned}
&y^{\infty} = a_y/(a_y + b_y), \quad \tau_y = 1/(a_y + b_y) \quad (y = \{h, m, n\}) \\
&a_h = 0.128 e^{\frac{17-V}{18}}, \quad b_h = \frac{4}{e^{-0.2(V-40)} + 1}, \\
&a_m = \frac{0.32(13-V)}{e^{0.25(13-V)} - 1}, \quad b_m = \frac{0.28(V-40)}{e^{0.2(V-40)} - 1} \\
&a_n = \frac{0.032(15-V)}{e^{0.2(15-V)} - 1}, \quad b_n = 0.5 e^{\frac{10-V}{40}} \\
&m_T^{\infty} = \frac{1}{1 + e^{-\frac{V+52}{7.4}}}, \quad \tau_{mT} = 0.44 + \frac{0.15}{e^{\frac{V+27}{10}} + e^{-\frac{V+102}{15}}}, \\
&h_T^{\infty} = \frac{1}{1 + e^{\frac{V+80}{5}}}, \quad \tau_{hT} = 62.7 + \frac{0.27}{e^{\frac{V+48}{4}} + e^{-\frac{V+407}{50}}}.
\end{aligned}
$$

In the above equations, h and m are the inactivation and activation variables of the Na^+ current; n is the activation variable of the K^+ current; m_T and h_T are the activation and inactivation variables of the low-threshold Ca^{2+} current; the leakage current I_L has conductance $g_L = 0.05\ \frac{mS}{cm^2}$ and reversal potential $E_L = -78$ mV; I_{Na} and I_K are the fast Na^+ and K^+ currents responsible for the generation of action potentials, with conductances $g_{Na} = 100\ \frac{mS}{cm^2}$ and $g_k = 10\ \frac{mS}{cm^2}$ and reversal potentials $E_{Na} = 50$ mV and $E_k = -95$ mV; I_T is the low-threshold Ca^{2+} current that mediates the rebound burst response, with conductance $g_{Ca} = 1.75\ \frac{mS}{cm^2}$ and reversal potential $E_{Ca} = k_0 \frac{RT}{2F} \log(\frac{Ca_0}{Ca})$; I^{syn} is the synaptic current (see Eq. (1.3) and Eq. (1.6)).

When the control current I_c is in the range $[-0.43, 0.13]\ \frac{\mu A}{cm^2}$ the neuron exhibits bursting behavior. The other parameters are set as follows: $C = 1\,\frac{\mu F}{cm^2}, \quad Ca_0 = 2\,mM, \quad d = 1\,\mu m,\ K_T = 0.0001\,mM \cdot ms, \quad K_d = 0.0001\,mM.$ $F = 96.489\ \frac{C}{mol}$ is the Faraday constant, $R = 8.31441\ \frac{J}{mol \cdot K}$ is the universal gas constant and the temperature T is set at 309.15 K.

A.2 Exponential Integrate and Fire Neuron Model

The exponential integrate and fire (eIF) neuron model [124] is defined by the following state equations:

$$\begin{cases} \dfrac{dV}{dt} = \dfrac{-g_L(V - E_L) + g_e e^{\frac{V - V_T}{\Delta_T}} - u + I_{ext} + I^{syn}}{C} \\ \\ \dfrac{du}{dt} = \dfrac{a(V - E_L) - u}{\tau_w} \end{cases} \tag{A.2}$$

where V is the membrane potential of the neuron; u is the adaptation variable; $g_L = 30\,[nS]$ is the leakage conductance and $E_L = -70.6\,[mV]$ is the leakage reversal potential; I^{syn} is the synaptic current (see Eq. (1.3)).

When the conductance g_e is set at 110 nS, the external current I_{ext} is varied in the range $[690, 1110]$ pA (Fig. 1.5). When the external current I_{ext} is set at 800 pA, the conductance g_e is varied in the range $[20, 160]$ nS (Fig. 1.6). For this range of parameter values, the neuron exhibits bursting behavior. The other parameters are set as follows: $C = 2007.4\,pF, \quad V_T = -50.4\,mV, \quad \Delta_T = 2\,mV, \quad \tau_w = 285.7\,ms, \quad a = 4\,nS.$

A.3 Plant Neuron Model

The Plant neuron model [123, 122] is defined by the following state equations:

$$\begin{cases} \dfrac{dV}{dt} = \dfrac{-I_T - I_L - I_{Na} - I_K - I_{KCa} + I_{ext} + I^{syn}}{C} \\ \dfrac{dCa}{dt} = \rho(K_c x(V_{Ca} - V) - Ca) \\ \\ \dfrac{dy}{dt} = \dfrac{y^{\infty} - y}{\tau_y} \quad y = \{h, n, x\} \end{cases} \tag{A.3}$$

where

$$
\begin{aligned}
& I_T = g_T x(V - E_I), \quad I_L = g_L(V - E_L), \\
& I_{Na} = g_I m_\infty^3 h(V - E_I), I_K = g_K n^4 (V - E_K), \\
& I_{KCa} = g_{KCa} \frac{Ca}{Ca + 0.5}(V - E_K), \\
& m_\infty = \frac{\frac{0.1(50-V_s)}{e^{\frac{50-V_s}{10}}}}{\frac{0.1(50-V_s)}{e^{\frac{50-V_s}{10}}} + 4e^{\frac{25-V_s}{18}}}, \\
& h^\infty = \frac{0.07e^{\frac{25-V_s}{20}}}{0.07e^{\frac{25-V_s}{20}} + \frac{1}{1+e^{\frac{55-V_s}{10}}}}, \\
& \tau_h = \frac{12.5}{0.07e^{\frac{25-V_s}{20}} + \frac{1}{1+e^{\frac{55-V_s}{10}}}}, \\
& n^\infty = \frac{\frac{0.01(55-V_s)}{e^{\frac{55-V_s}{10}}} - 1}{\frac{0.01(55-V_s)}{e^{\frac{55-V_s}{10}}} - 1 + 0.125e^{\frac{45-V_s}{80}}}, \\
& \tau_n = \frac{12.5}{\frac{0.01(55-V_s)}{e^{\frac{55-V_s}{10}}} - 1 + 0.125e^{\frac{45-V_s}{80}}}, \\
& x^\infty = \frac{1}{e^{0.15(-V-50)} + 1}, \\
& V_s = \frac{127V}{105} + \frac{8265}{105}
\end{aligned}
$$

V is the membrane potential of the neuron; Ca is the intracellular calcium concentration; x is the activation variable of the slow inward Ca^{2+} current; h is the inactivation variable of the Na^+ current; n is the activation variable of the K^+ current; I_L is the leakage current, with conductance $g_L = 0.003$ nS and reversal potential $E_L = -40$ mV; I_{Na} and I_K are the fast inward Na^+ and outward K^+ currents, respectively, with conductances $g_I = 8$ nS and $g_K = 1.3$ nS (these values ensure undershoot, see paper) and reversal potentials $E_I = 30$ mV and $E_K = -75$ mV; I_T is the slow inward tetrodotoxin-resistant Ca^{2+} current, with conductance $g_T = 0.01$ nS and reversal potential $E_T = 30$ mV; I_{KCa} is the outward Ca^{2+} sensitive K^+ current, with conductance $g_{KCa} = 0.03$ nS and reversal potential E_K; I^{syn} is the synaptic current (see Eq. (1.3)).

The external current I_{ext} is set to 0.028 μA. The other parameters are set as follows: $C = 1\,\frac{\mu F}{cm^2}$, $\quad \rho = 0.00015\,mV^{-1}$, $\quad K_c = 0.0085\,mV^{-1}$, $\quad V_{Ca} = 140\,mV$, $\quad \tau_x = 235\,ms$.

A.4 Parameter Values

In table A.1, column A lists the parameter values used for the gHCO with the thalamic reticular neuron model, first-order dynamic excitatory synapses and static inhibitory synapses (Figs. 1.2 and 1.3). Column B lists the parameter values used for the gHCO with the eIF neuron model when varying I_{ext}, first-order dynamic excitatory synapses and static inhibitory synapses (Fig. 1.5). Column C lists the parameter values used for the gHCO with the eIF neuron model when varying g_e, first-order dynamic excitatory synapses and static inhibitory synapses (Fig. 1.6). Column D lists the parameter values used for the gHCO with the Plant neuron model, first-order dynamic excitatory synapses and static inhibitory synapses (Fig. 1.4).

Table A.2 lists all parameter values determined in each step of section 1.4.2 referring to the results presented in section 1.4.3 and obtained with the first-order dynamic synapse.

Table A.3 lists all parameter values determined in each step of section 1.4.2 referring to the results presented in section 1.4.4 and obtained with the alternative synapse model.

Table A.1: Parameter values for the gHCO. Notice that for columns B and C the synaptic conductances are expressed in nS instead of nS/cm^2

	A	B	C	D
α^{ex} [kHz]	0.1556	10	10	0.5
β^{ex} [kHz]	0.005	0.26	26	0.0005
θ^{ex} [mV]	25	-40	-40	-42
g^{ex} [nS/cm^2]	0.0005	1	0.4	0.0001
E^{ex} [mV]	60	20	20	50
θ^{in} [mV]	-30	-48.5	-48.5	-53
g^{in} [nS/cm^2]	0.0005	0.6	0.1	0.0001
E^{in} [mV]	-80	-110	-110	-80
ν [mV^{-1}]	10	10	10	10

Table A.2: Parameter values set in each step of section 1.4.2 to obtain the results presented in section 1.4.3.

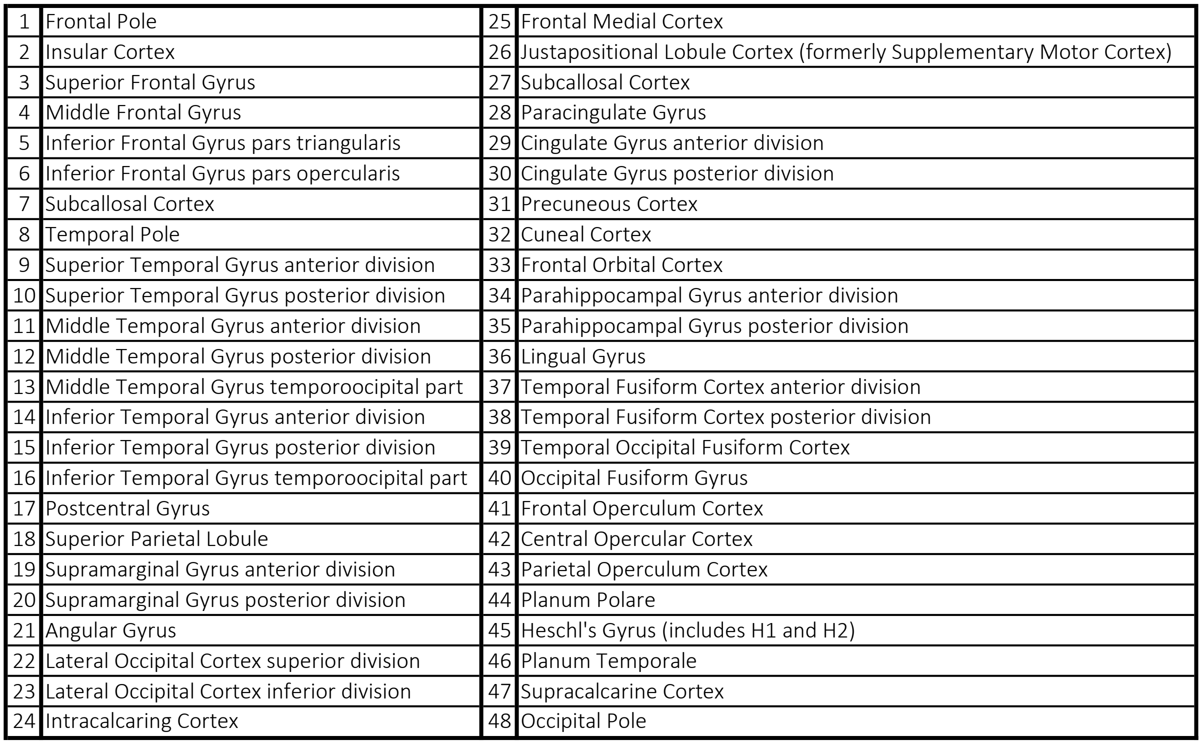
Parameter	Value	Parameter	Value
Step 0			
θ^E	-50 mV	g_{ij}^D	$0.0119 g_{ij}^S$
θ^D	-50 mV	g_{41}^F/g_{14}^F	2.3327
θ^F	-50 mV	g_{32}^F/g_{23}^F	2.3327
ν	10 mV^{-1}	δ	half of burst period
Step 1			
ξ	3.0303	V_t	-50 mV
Step 2			
α	0.7543 kHz	θ^S	17.5 mV
β	0.0391 kHz		
Step 3			
g_{41}^F	0.0048 nS/cm^2	g_{32}^F	0.0048 nS/cm^2
g_{14}^F	0.0111 nS/cm^2	g_{23}^F	0.0111 nS/cm^2
Step 4			
g_{12}^S	0.1207 nS/cm^2	g_{12}^E	0.0052 nS/cm^2
g_{21}^S	0.1207 nS/cm^2	g_{21}^E	0.0052 nS/cm^2
g_{34}^S	0.1207 nS/cm^2	g_{34}^E	0.0052 nS/cm^2
g_{43}^S	0.1207 nS/cm^2	g_{43}^E	0.0052 nS/cm^2

Table A.3: Parameter values set in each step of section 1.4.2 to obtain the results presented in section 1.4.4.

Parameter	Value	Parameter	Value
		Step 0	
θ^E	-50 mV	g_{ij}^D	$0.0119 g_{ij}^S$
θ^D	-50 mV	g_{41}^F/g_{14}^F	2.3327
θ^F	-50 mV	g_{32}^F/g_{23}^F	2.3327
ν	10 mV^{-1}	δ	half of burst period
		Step 1	
ξ	3.0303	V_t	-50 mV
		Step 2	
α	0.5000 kHz	θ^S	5 mV
β	0.0250 kHz		
		Step 3	
g_{41}^F	0.0106 nS/cm^2	g_{32}^F	0.0106 nS/cm^2
g_{14}^F	0.0248 nS/cm^2	g_{23}^F	0.0248 nS/cm^2
		Step 4	
g_{12}^S	0.0200 nS/cm^2	g_{12}^E	0.0001 nS/cm^2
g_{21}^S	0.0200 nS/cm^2	g_{21}^E	0.0001 nS/cm^2
g_{34}^S	0.0200 nS/cm^2	g_{34}^E	0.0001 nS/cm^2
g_{43}^S	0.0200 nS/cm^2	g_{43}^E	0.0001 nS/cm^2

Appendix B

B.1 Harvard-Oxford Cortical Structural Atlas



1	Frontal Pole	25	Frontal Medial Cortex
2	Insular Cortex	26	Justapositional Lobule Cortex (formerly Supplementary Motor Cortex)
3	Superior Frontal Gyrus	27	Subcallosal Cortex
4	Middle Frontal Gyrus	28	Paracingulate Gyrus
5	Inferior Frontal Gyrus pars triangularis	29	Cingulate Gyrus anterior division
6	Inferior Frontal Gyrus pars opercularis	30	Cingulate Gyrus posterior division
7	Subcallosal Cortex	31	Precuneous Cortex
8	Temporal Pole	32	Cuneal Cortex
9	Superior Temporal Gyrus anterior division	33	Frontal Orbital Cortex
10	Superior Temporal Gyrus posterior division	34	Parahippocampal Gyrus anterior division
11	Middle Temporal Gyrus anterior division	35	Parahippocampal Gyrus posterior division
12	Middle Temporal Gyrus posterior division	36	Lingual Gyrus
13	Middle Temporal Gyrus temporooccipital part	37	Temporal Fusiform Cortex anterior division
14	Inferior Temporal Gyrus anterior division	38	Temporal Fusiform Cortex posterior division
15	Inferior Temporal Gyrus posterior division	39	Temporal Occipital Fusiform Cortex
16	Inferior Temporal Gyrus temporooccipital part	40	Occipital Fusiform Gyrus
17	Postcentral Gyrus	41	Frontal Operculum Cortex
18	Superior Parietal Lobule	42	Central Opercular Cortex
19	Supramarginal Gyrus anterior division	43	Parietal Operculum Cortex
20	Supramarginal Gyrus posterior division	44	Planum Polare
21	Angular Gyrus	45	Heschl's Gyrus (includes H1 and H2)
22	Lateral Occipital Cortex superior division	46	Planum Temporale
23	Lateral Occipital Cortex inferior division	47	Supracalcarine Cortex
24	Intracalcaring Cortex	48	Occipital Pole

Figure B.1: Numbering of cortical areas according to the Harvard-Oxford Cortical Structural Atlas (RRID:SCR_001476), as they appear in NeuroData's MRI Graphs pipeline [230].

B.2 Master stability function

The cluster synchronization state is defined as:

$$\dot{\boldsymbol{s}}_p(t) = \boldsymbol{F}(\boldsymbol{s}_p(t)) + \boldsymbol{\Gamma}\left(\boldsymbol{s}_p(t), \sigma \sum_q r_{pq}\boldsymbol{G}(\boldsymbol{s}_q(t))\right), \tag{B.1}$$

where the $k-$dimensional matrix $R = \{r_{pq}\}$ is the quotient matrix, such that $r_{pq} = \sum_{j\in\mathcal{C}_q} a_{ij}$ $(i \in \mathcal{C}_p, \quad p, q = 1, 2, \ldots, k)$.
The following definitions are preliminarily given

$$D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t)) = D\boldsymbol{F}(\boldsymbol{s}_p(t)) + D\boldsymbol{\Gamma}_1\left(\boldsymbol{s}_p(t), \sigma\sum_q r_{pq}\boldsymbol{G}(\boldsymbol{s}_q(t))\right) \tag{B.2}$$

and

$$D\tilde{\boldsymbol{\Gamma}}_p = D\boldsymbol{\Gamma}_2\left(\boldsymbol{s}_p(t), \sigma\sum_{q=1}^{k} r_{pq}\boldsymbol{G}(\boldsymbol{s}_q(t))\right), \tag{B.3}$$

where $D\boldsymbol{F}$ is the $m \times m$ Jacobian of the nodes' vector field and the m-dimensional matrix $D\boldsymbol{\Gamma}_1$ $(D\boldsymbol{\Gamma}_2)$ is the derivative of $\boldsymbol{\Gamma}$ with respect to its first (second) argument. The linearized equations governing the dynamics of the perturbations about the synchronous solution $\boldsymbol{s}_p(t)$ can be written as:

$$\dot{\boldsymbol{w}}_i(t) = D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t))\boldsymbol{w}_i + D\tilde{\boldsymbol{\Gamma}}_p\sigma\left[\sum_{q=1}^{k} D\boldsymbol{G}(\boldsymbol{s}_q(t))\sum_{j\in\mathcal{C}_q} a_{ij}\boldsymbol{w}_j(t)\right] \tag{B.4}$$

Eq. (B.4) can be rewritten in vector form by stacking all the state perturbation vectors together in one vector $\boldsymbol{W}(t)$. Moreover, the $N \times N$ diagonal matrix E_p is introduced, which is the cluster indicator matrix: E_p has entries $E_{p,ii} = 1$, if node $i \in \mathcal{C}_p$, 0 otherwise, i.e., this matrix identifies all the nodes i's that belong to cluster $\mathcal{C}_p$. Therefore:

$$\begin{aligned}\dot{\boldsymbol{W}}(t) = &\left[\sum_{p=1}^{k} E_p \otimes D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t))\right]\boldsymbol{W}(t) \\ &+ \left(\sum_{p=1}^{k} E_p \otimes D\tilde{\boldsymbol{\Gamma}}_p\right)\sigma\left[A \otimes I_n \textstyle\sum_{q=1}^{k} E_q \otimes \boldsymbol{G}(\boldsymbol{s}_q(t))\right]\boldsymbol{W}(t)\end{aligned} \tag{B.5}$$

The canonical transformation matrix T [175] is computed as the orthogonal matrix that simultaneously block-diagonalizes the matrices A, E_1, E_2,..., E_C into D diagonal blocks, $T = \begin{pmatrix} T_\parallel \\ T_\perp \end{pmatrix} = \mathcal{SBD}(A, E_1, E_2, \ldots, E_C)$. The process to construct the canonical transformation matrix T is described in [175]. Application of the matrix T yields $TAT^{-1} = \hat{A}$, where $\hat{A} = \hat{A}_\parallel \oplus \hat{A}_\perp = \oplus_{j=1}^{D} \hat{A}_j$. The symbol $\oplus$ denotes the direct sum of matrices and the blocks $\hat{A}_j$ $j = 1, 2, \ldots, D$ have the same dimension. Note that $\hat{A}_\parallel = \hat{A}_1$ and $\hat{A}_\perp = \oplus_{j=2}^{D} \hat{A}_j$. Moreover, we have that $TE_pT^{-1} = E_p$.

By using matrix T, the variational equation for the transverse perturbations becomes

$$\begin{aligned} \dot{\boldsymbol{\eta}}_\perp(t) = \underbrace{\left[\sum_{p=1}^{k} E_{p\perp} \otimes D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t)) \right]}_{\rho_1(\{\boldsymbol{s}_p(t)\})} \boldsymbol{\eta}_\perp(t) + \\ \underbrace{\left(\sum_{p=1}^{k} E_{p\perp} \otimes D\tilde{\boldsymbol{\Gamma}}_p \right) \left[\sigma \left(\hat{A}_\perp \otimes I_n \sum_{q=1}^{k} E_{q\perp} \otimes D\boldsymbol{G}(\boldsymbol{s}_q(t)) \right) \right]}_{\rho_2(\{\boldsymbol{s}_p(t)\})} \boldsymbol{\eta}_\perp(t) \end{aligned} \tag{B.6}$$

where the block-diagonal matrix marked with $\perp$ is a minor of the complete matrix $\hat{A} = TAT^{-1}$, containing only the blocks related to the transverse perturbations.

B.3 Master stability function for a network with heterogeneous delays

The cluster synchronization state is defined as:

$$\dot{\boldsymbol{s}}_p(t) = \boldsymbol{F}(\boldsymbol{s}_p(t)) + \boldsymbol{\Gamma}\left(\boldsymbol{s}_p(t), \sigma \sum_l \sum_q r^l_{pq} \boldsymbol{G}(\boldsymbol{s}_q(t-\tau^l))\right), \tag{B.7}$$

where the $k-$dimensional matrix $R^l = \{r^l_{pq}\}$ is the quotient matrix, such that $r^l_{pq} = \sum_{j\in\mathcal{C}_q} a^l_{ij}$ $(i \in \mathcal{C}_p, \quad p, q = 1, 2, \ldots, k)$.
Eqs. (B.2) and (B.3) become

$$D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t)) = D\boldsymbol{F}(\boldsymbol{s}_p(t)) + D\boldsymbol{\Gamma}_1\left(\boldsymbol{s}_p(t), \sigma\sum_l \sum_q r^l_{pq} \boldsymbol{G}(\boldsymbol{s}_q(t-\tau^l))\right) \tag{B.8}$$

and

$$D\tilde{\boldsymbol{\Gamma}}_p = D\boldsymbol{\Gamma}_2\left(\boldsymbol{s}_p(t), \sigma\sum_l \sum_q r^l_{pq} \boldsymbol{G}(\boldsymbol{s}_q(t-\tau^l))\right), \tag{B.9}$$

The linearized equations governing the dynamics of the perturbations about the synchronous solution $\boldsymbol{s}_p(t)$ can be written as:

$$\dot{\boldsymbol{w}}_i(t) = D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t))\boldsymbol{w}_i(t) + D\tilde{\boldsymbol{\Gamma}}_p\sigma\sum_l \left[\sum_{q=1}^{k} D\boldsymbol{G}(\boldsymbol{s}_q(t-\tau^l))\sum_{j\in\mathcal{C}_q} a^l_{ij}\boldsymbol{w}_j(t-\tau^l)\right] \tag{B.10}$$

and Eq. (B.10) can then be rewritten in vector form by stacking all the state perturbation vectors together in one vector $\boldsymbol{W}(t)$:

$$\begin{aligned}\dot{\boldsymbol{W}}(t) &= \left[\sum_{p=1}^{k} E_p \otimes D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t))\right]\boldsymbol{W}(t) \\ &+ \left(\sum_{p=1}^{k} E_p \otimes D\tilde{\boldsymbol{\Gamma}}_p\right)\sigma\sum_l \left[\left(A^l \otimes I_n \sum_{q=1}^{k} E_q \otimes \boldsymbol{G}(\boldsymbol{s}_q(t-\tau^l))\right)\boldsymbol{W}(t-\tau^l)\right]\end{aligned} \tag{B.11}$$

The canonical transformation matrix T is computed as the orthogonal matrix that simultaneously block-diagonalizes the matrices A^1, A^2,. . ., A^L, E_1, E_2,. . ., E_C into D diagonal blocks, $T = \begin{pmatrix} T_{\parallel} \\ T_{\perp} \end{pmatrix} = \mathcal{SBD}(A^1, A^2, \ldots, A^L, E_1, E_2, \ldots, E_C)$. Application of the matrix T yields $TA^lT^{-1} = \hat{A}^l$, where $\hat{A}^l = \hat{A}^l_{\parallel} \oplus \hat{A}^l_{\perp} = \oplus_{j=1}^{D} \hat{A}^l_j$. By using matrix T, the variational equation for the transverse perturbations becomes

$$\begin{gathered}
\dot{\boldsymbol{\eta}}_{\perp}(t) = \underbrace{\left[\sum_{p=1}^{k} E_{p\perp} \otimes D\tilde{\boldsymbol{F}}(\boldsymbol{s}_p(t))\right]}_{\rho_1(\{\boldsymbol{s}_p(t)\})} \boldsymbol{\eta}_{\perp}(t) + \\
\sum_{l} \Bigg[\underbrace{\left(\sum_{p=1}^{k} E_{p\perp} \otimes D\tilde{\boldsymbol{\Gamma}}_p\right) \sigma \left(\hat{A}^l_{\perp} \otimes I_n \sum_{q=1}^{k} E_{q\perp} \otimes D\boldsymbol{G}(\boldsymbol{s}_q(t - \tau^l))\right)}_{\rho_2^l(\{\boldsymbol{s}_p(t)\})} \boldsymbol{\eta}_{\perp}(t - \tau^l) \Bigg]
\end{gathered} \tag{B.12}$$

where the block-diagonal matrix marked with $\perp$ is a minor of the complete matrix $\hat{A}^l = TA^lT^{-1}$, containing only the blocks related to the transverse perturbations.

Bibliography

[1] Leon Chua. Device modeling via nonlinear circuit elements. *IEEE Transactions on Circuits and Systems*, 27(11):1014–1044, 1980.

[2] Oliver Nelles. *Nonlinear system identification: from classical approaches to neural networks, fuzzy models, and gaussian processes*. Springer Nature, 2020.

[3] Michael Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.

[4] Adam Thelen, Xiaoge Zhang, Olga Fink, Yan Lu, Sayan Ghosh, Byeng D Youn, Michael D Todd, Sankaran Mahadevan, Chao Hu, and Zhen Hu. A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies. *Structural and Multidisciplinary Optimization*, 65(12):1–55, 2022.

[5] Panagiotis Stavropoulos and Dimitris Mourtzis. Digital twins in industry 4.0. In *Design and Operation of Production Networks for Mass Personalization in the Era of Cloud Technology*, pages 277–316. Elsevier, 2022.

[6] Roland Molontay and Marcell Nagy. Two decades of network science: as seen through the co-authorship network of network scientists. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 578–583, 2019.

[7] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440–442, 1998.

[8] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[9] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[10] Alessandro Vespignani. Twenty years of network science. *Nature*, 558(7711):528–529, 2018.

[11] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288(5):60–69, 2003.

[12] Mark Newman. *Networks: An Introduction*. Oxford University Press, 03 2010.

[13] O Ravaka Andriamihaja, Florence Metz, Julie G Zaehringer, Manuel Fischer, and Peter Messerli. Land competition under telecoupling: Distant actors' environmental versus economic claims on land in north-eastern madagascar. *Sustainability*, 11(3):851, 2019.

[14] Adilson E Motter, Seth A Myers, Marian Anghel, and Takashi Nishikawa. Spontaneous synchrony in power-grid networks. *Nature Physics*, 9(3):191–197, 2013.

[15] Andreas Bruder, André Frainer, Thibaut Rota, and Raul Primicerio. The importance of ecological networks in multiple-stressor research and management. *Frontiers in Environmental Science*, 7:59, 2019.

[16] Esther Landhuis. Probing fine-scale connections in the brain. *Nature*, 586:631, 2020.

[17] C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary Huang, Kazunori Shinomiya, Jeremy Maitin-Shepard, David Ackerman, Stuart Berg, et al. A connectome of the adult drosophila central brain. *BioRxiv*, 2020.

[18] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* CRC press, 2018.

[19] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[20] Clive WJ Granger. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980.

[21] George Sugihara, Robert May, Hao Ye, Chih-hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. Detecting causality in complex ecosystems. *science*, 338(6106):496–500, 2012.

[22] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

[23] Bethany Lusch, Pedro D Maia, and J Nathan Kutz. Inferring connectivity in networked dynamical systems: Challenges using granger causality. *Physical Review E*, 94(3):032220, 2016.

[24] Dan Mønster, Riccardo Fusaroli, Kristian Tylén, Andreas Roepstorff, and Jacob F Sherson. Causal inference from noisy time-series data—testing the convergent cross-mapping algorithm in the presence of noise and external influence. *Future Generation Computer Systems*, 73:52–62, 2017.

[25] Shizhe Chen, Ali Shojaie, and Daniela M Witten. Network reconstruction from high-dimensional ordinary differential equations. *Journal of the American Statistical Association*, 112(520):1697–1707, 2017.

[26] Anna Krakovská, Jozef Jakubík, Martina Chvosteková, David Coufal, Nikola Jajcay, and Milan Paluš. Comparison of six methods for the detection of causality in a bivariate time series. *Physical Review E*, 97(4):042207, 2018.

[27] Alex Eric Yuan and Wenying Shou. Data-driven causal analysis of observational biological time series. *Elife*, 11:e72518, 2022.

[28] George Stepaniants, Bingni W Brunton, and J Nathan Kutz. Inferring causal networks of dynamical systems through transient dynamics and perturbation. *Physical Review E*, 102(4):042309, 2020.

[29] Dongchuan Yu, Marco Righero, and Ljupco Kocarev. Estimating topology of networks. *Physical Review Letters*, 97(18):188701, 2006.

[30] Liang Chen, Jun-an Lu, and K Tse Chi. Synchronization: an obstacle to identification of network topology. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(4):310–314, 2009.

[31] Junchan Zhao, Qin Li, Jun-An Lu, and Zhong-Ping Jiang. Topology identification of complex dynamical networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(2):023119, 2010.

[32] Marc Timme. Revealing network connectivity from response dynamics. *Physical review letters*, 98(22):224101, 2007.

[33] Dongchuan Yu and Ulrich Parlitz. Inferring local dynamics and connectivity of spatially extended systems with long-range links based on steady-state stabilization. *Physical Review E*, 82(2):026108, 2010.

[34] Longkun Tang, Jun-an Lu, Xiaoqun Wu, and Jinhu Lü. Impact of node dynamics parameters on topology identification of complex dynamical networks. *Nonlinear Dynamics*, 73(1):1081–1097, 2013.

[35] Marc Timme and Jose Casadiego. Revealing networks from dynamics: an introduction. *Journal of Physics A: Mathematical and Theoretical*, 47(34):343001, 2014.

[36] Rico Berner, Serhiy Yanchuk, and Eckehard Schöll. What adaptive neuronal networks teach us about power grids. *Physical Review E*, 103(4):042315, 2021.

[37] Ronald M. Harris-Warrick and Jan-Marino Ramirez. Neural networks for the generation of rhythmic motor behaviors. In *Neurobiology of Motor Control*, chapter 8, pages 225–262. 2017.

[38] Valentina Baruzzi, Matteo Lodi, Marco Storace, and Andrey Shilnikov. Generalized half-center oscillators with short-term synaptic plasticity. *Physical Review E*, 102(3):032406, 2020.

[39] Valentina Baruzzi, Matteo Lodi, Marco Storace, and Andrey Shilnikov. Towards more biologically plausible central-pattern-generator models. *Physical Review E*, 104(6):064405, 2021.

[40] The Brain Initiative. www.braininitiative.org. Accessed: 2022-10-12.

[41] The Human Brain Project. www.humanbrainproject.eu. Accessed: 2022-10-12.

[42] The Virtual Brain. www.thevirtualbrain.org. Accessed: 2022-10-12.

[43] Rosalyn Moran, Dimitris A Pinotsis, and Karl Friston. Neural masses and fields in dynamic causal modeling. *Frontiers in computational neuroscience*, 7:57, 2013.

[44] Valentina Baruzzi, Matteo Lodi, Francesco Sorrentino, and Marco Storace. Bridging functional and anatomical neural connectivity through nonlinear dynamics. Submitted January 17th, 2023.

[45] Prabha Kundur, Neal J Balu, and Mark G Lauby. *Power system stability and control*, volume 7. McGraw-hill New York, 1994.

[46] Valentina Baruzzi, Matteo Lodi, Alberto Oliveri, and Marco Storace. Analysis and improvement of an algorithm for the online inertia estimation in power grids with RES. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2021.

[47] Valentina Baruzzi, Matteo Lodi, Alberto Oliveri, and Marco Storace. Estimation of inertia in power grids with turbine governors. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2022.

[48] Alessandro Ravera, Valentina Baruzzi, Matteo Lodi, Alberto Oliveri, and Marco Storace. Embedded implementation of an algorithm for online inertia estimation in power grids. In *International Conference on System-Integrated Intelligence*, pages 84–94. Springer, 2023.

[49] Ole Kiehn and Kimberly Dougherty. Locomotion: circuits and physiology. In *Neuroscience in the 21st Century*, pages 1337–1365. Springer, 2016.

[50] Sten Grillner. Biological pattern generation: the cellular and computational logic of networks in motion. *Neuron*, 52(5):751–766, 2006.

[51] Kaoru Takakusaki. Neurophysiology of gait: from the spinal cord to the frontal lobe. *Mov. Disord.*, 28(11):1483–1491, 2013.

[52] Vittorio Caggiano, Roberto Leiras, Haizea Goñi-Erro, Debora Masini, Carmelo Bellardita, Julien Bouvier, V Caldeira, Gilberto Fisone, and Ole Kiehn. Midbrain circuits that set locomotor speed and gait selection. *Nature*, 553(7689):455–460, 2018.

[53] Theodore Holmes Bullock. The origins of patterned nervous discharge. *Behaviour*, 17(1):48–58, 1961.

[54] DM Wilson and RJ Wyman. Motor output patterns during random and rhythmic stimulation of locust thoracic ganglia. *Biophysical Journal*, 5(2):121–143, 1965.

[55] T Graham Brown. On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *J. Physiol.*, 48(1):18–46, 1914.

[56] E Jankowska, MGM Jukes, S Lund, and A Lundberg. The effect of dopa on the spinal cord 5. reciprocal organization of pathways transmitting excitatory action to alpha motoneurones of flexors and extensors. *Acta physiologica Scandinavica*, 70(3-4):369–388, 1967.

[57] E Jankowska, MGM Jukes, S Lund, and A Lundberg. The effect of dopa on the spinal cord 6. half-centre organization of interneurones transmitting effects from the flexor reflex afferents. *Acta Physiologica Scandinavica*, 70(3-4):389–402, 1967.

[58] Sten Grillner. Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiological reviews*, 55(2):247–304, 1975.

[59] Ole Kiehn and Ole Kjaerulff. Distribution of central pattern generators for rhythmic motor outputs in the spinal cord of limbed vertebrates a. *Annals of the New York Academy of Sciences*, 860(1):110–129, 1998.

[60] Ole Kiehn and Simon JB Butt. Physiological, anatomical and genetic identification of cpg neurons in the developing mammalian spinal cord. *Progress in neurobiology*, 70(4):347–361, 2003.

[61] Avis H Cohen, G Bard Ermentrout, Tim Kiemel, Nancy Kopell, Karen A Sigvardt, and Thelma L Williams. Modelling of intersegmental coordination in the lamprey central pattern generator for locomotion. *Trends in neurosciences*, 15(11):434–438, 1992.

[62] Sten Grillner, David Parker, and Abdel˙ Jabbar El Manira. Vertebrate locomotion-a lamprey perspective a. *Annals of the New York Academy of Sciences*, 860(1):1–18, 1998.

[63] Auke Jan Ijspeert and Jérôme Kodjabachian. Evolution and development of a central pattern generator for the swimming of a lamprey. *Artificial life*, 5(3):247–269, 1999.

[64] Scott L Hooper and Ralph A DiCaprio. Crustacean motor pattern generator networks. *Neurosignals*, 13(1-2):50–69, 2004.

[65] W Otto Friesen and William B Kristan. Leech locomotion: swimming, crawling, and decisions. *Current opinion in neurobiology*, 17(6):704–711, 2007.

[66] Ansgar Büschges, Turgay Akay, Jens P Gabriel, and Joachim Schmidt. Organizing network action for locomotion: insights from studying insect walking. *Brain research reviews*, 57(1):162–171, 2008.

[67] Pierre A Guertin. The mammalian central pattern generator for locomotion. *Brain research reviews*, 62(1):45–56, 2009.

[68] Taryn Klarner and E Paul Zehr. Sherlock holmes and the curious case of the human locomotor central pattern generator. *Journal of neurophysiology*, 120(1):53–77, 2018.

[69] Paolo Arena, Luigi Fortuna, Mattia Frasca, and Giovanni Sicurella. An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1823–1837, 2004.

[70] Alessandro Crespi and Auke Jan Ijspeert. Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on Robotics*, 24(1):75–87, 2008.

[71] Zhenli Lu, Shugen Ma, Bin Li, and Yuechao Wang. 3d locomotion of a snake-like robot controlled by cyclic inhibitory cpg model. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3897–3902. IEEE, 2006.

[72] Hiroshi Kimura, Yasuhiro Fukuoka, and Avis H Cohen. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *The International Journal of Robotics Research*, 26(5):475–490, 2007.

[73] Rodolphe Héliot and Bernard Espiau. Multisensor input for cpg-based sensory—motor coordination. *IEEE Transactions on Robotics*, 24(1):191–195, 2008.

[74] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neur. Netw.*, 21(4):642–653, 2008.

[75] Junzhi Yu, Min Tan, Jian Chen, and Jianwei Zhang. A survey on cpg-inspired control models and system implementation. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(3):441–456, 2013.

[76] HG Traven, L Brodin, A Lansner, O Ekeberg, P Wallén, and S Grillner. Computer simulations of nmda and non-nmda receptor-mediated synaptic drive: sensory and supraspinal modulation of neurons and small networks. *Journal of neurophysiology*, 70(2):695–709, 1993.

[77] Jeanette Hellgren, Sten Grillner, and Anders Lansner. Computer simulation of the segmental neural network generating locomotion in lamprey by using populations of network interneurons. *Biological cybernetics*, 68(1):1–13, 1992.

[78] James T Buchanan. Neural network simulations of coupled locomotor oscillators in the lamprey spinal cord. *Biological cybernetics*, 66(4):367–374, 1992.

[79] Thelma L Williams. Phase coupling in simulated chains of coupled oscillators representing the lamprey spinal cord. *Neural Computation*, 4(4):546–558, 1992.

[80] Örjan Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological cybernetics*, 69(5):363–374, 1993.

[81] Martin Golubitsky and Ian Stewart. *The symmetry perspective: from equilibrium to chaos in phase space and physical space*, volume 200. Springer Science & Business Media, 2003.

[82] Yoshiki Kuramoto. Collective behavior of coupled phase oscillators. *The handbook of brain theory and neural networks*, pages 223–226, 2003.

[83] Avis H Cohen, Philip J Holmes, and Richard H Rand. The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: A mathematical model. *Journal of mathematical biology*, 13(3):345–369, 1982.

[84] Kiyotoshi Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological cybernetics*, 56(5):345–353, 1987.

[85] Gregor Schöner, Wenying Y Jiang, and JA Scott Kelso. A synergetic theory of quadrupedal gaits and gait transitions. *Journal of theoretical Biology*, 142(3):359–391, 1990.

[86] N Kopell, GB Ermentrout, and TL Williams. On chains of oscillators forced at one end. *SIAM Journal on Applied Mathematics*, 51(5):1397–1417, 1991.

[87] James J Collins and Sally A Richmond. Hard-wired central pattern generators for quadrupedal locomotion. *Biological cybernetics*, 71(5):375–385, 1994.

[88] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *science*, 315(5817):1416–1420, 2007.

[89] Kiyotoshi Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological cybernetics*, 52(6):367–376, 1985.

[90] Gentaro Taga, Yoko Yamaguchi, and Hiroshi Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological cybernetics*, 65(3):147–159, 1991.

[91] Hiroshi Kimura, Seiichi Akiyama, and Kazuaki Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous robots*, 7(3):247–258, 1999.

[92] Jonas Buchli and Auke Jan Ijspeert. Distributed central pattern generator model for robotics application based on phase sensitivity analysis. In *International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pages 333–349. Springer, 2004.

[93] Matteo Lodi, Andrey Shilnikov, and Marco Storace. Design of minimal synthetic circuits with sensory feedback for quadruped locomotion. In *2018 IEEE Int. Symp. Circ. Syst. (ISCAS)*, pages 1–5. IEEE, 2018.

[94] Simon M Danner, Natalia A Shevtsova, Alain Frigon, and Ilya A Rybak. Computational modeling of spinal circuits controlling limb coordination and gaits in quadrupeds. *Elife*, 6:e31050, 2017.

[95] Pietro-Luciano Buono and Martin Golubitsky. Models of central pattern generators for quadruped locomotion I. Primary gaits. *J. Math. Biol.*, 42(4):291–326, 2001.

[96] Carla MA Pinto and Martin Golubitsky. Central pattern generators for bipedal locomotion. *J. Math. Biol.*, 53(3):474–489, 2006.

[97] Yaroslav I Molkov, Bartholomew J Bacak, Adolfo E Talpalar, and Ilya A Rybak. Mechanisms of left-right coordination in mammalian locomotor pattern generation circuits: a mathematical modeling view. *PLoS Comp. Biol.*, 11(5), 2015.

[98] Matteo Lodi, Andrey Shilnikov, and Marco Storace. Design of synthetic central pattern generators producing desired quadruped gaits. *IEEE Trans. Circ. Syst. I*, 65(3):1028–1039, 2017.

[99] Jessica Ausborn, Abigail C Snyder, Natalia A Shevtsova, Ilya A Rybak, and Jonathan E Rubin. State-dependent rhythmogenesis and frequency control in a half-center locomotor cpg. *J. Neurophysiol.*, 119(1):96–117, 2018.

[100] Matteo Lodi, Andrey L Shilnikov, and Marco Storace. Design principles for central pattern generators with preset rhythms. *IEEE Trans. Neural Netw. Learn. Syst.*, pages 1–12, 2020.

[101] Ronald L Calabrese. Half-center oscillators underlying rhythmic movements. *Nature*, 261:146–148, 1995.

[102] John A White, Carson C Chow, Jason Rit, Cristina Soto-Treviño, and Nancy Kopell. Synchronization and oscillatory dynamics in heterogeneous, mutually inhibited neurons. *Journal of computational neuroscience*, 5(1):5–16, 1998.

[103] Robert C Elson, Allen I Selverston, Henry DI Abarbanel, and Mikhail I Rabinovich. Inhibitory synchronization of bursting in biological neurons: dependence on synaptic time constant. *Journal of Neurophysiology*, 88(3):1166–1176, 2002.

[104] Timothy J Lewis and John Rinzel. Dynamics of spiking neurons connected by both inhibitory and electrical coupling. *Journal of computational neuroscience*, 14(3):283–309, 2003.

[105] Tiaza Bem and John Rinzel. Short duty cycle destabilizes a half-center oscillator, but gap junctions can restabilize the anti-phase pattern. *J. Neurophysiol.*, 91(2):693–703, 2004.

[106] Anca Doloc-Mihu and Ronald L Calabrese. A database of computational models of a half-center oscillator for analyzing how neuronal parameters influence network activity. *J. Biol. Phys.*, 37(3):263–283, 2011.

[107] N Dale and A Roberts. Dual-component amino-acid-mediated synaptic potentials: excitatory drive for swimming in xenopus embryos. *J. Physiol.*, 363(1):35–59, 1985.

[108] M Pinco and A Lev-Tov. Synaptic transmission between ventrolateral funiculus axons and lumbar motoneurons in the isolated spinal cord of the neonatal rat. *J. Neurophysiol.*, 72(5):2406–2419, 1994.

[109] Charuni Gunaratne, Akira Sakurai, and Paul S. Katz. Variations on a theme: species differences in synaptic connectivity do not predict central pattern generator activity. *J. Neurophysiol.*, 118:1123–1132, 2017.

[110] Akira Sakurai and Paul S. Katz. Command or obey? homologous neurons differ in hierarchical position for the generation of homologous behaviors. *J. Neurosc.*, 39(33):6460–6471, 2019.

[111] Eugene M Izhikevich. Neural excitability, spiking and bursting. *Int. J. Bifurcat. Chaos*, 10(06):1171–1266, 2000.

[112] Alain Destexhe, Diego Contreras, Terrence J Sejnowski, and Mircea Steriade. A model of spindle rhythmicity in the isolated thalamic reticular nucleus. *J. Neurophysiol.*, 72(2):803–818, 1994.

[113] Roman Nagornov, Grigory Osipov, Maxim Komarov, Arkady Pikovsky, and Andrey Shilnikov. Mixed-mode synchronization between two inhibitory neurons with post-inhibitory rebound. *Commun. Nonlinear Sci.*, 36:175–191, 2016.

[114] X-J Wang. Fast burst firing and short-term synaptic plasticity: a model of neocortical chattering neurons. *Neurosc.*, 89(2):347–362, 1999.

[115] Dean V Buonomano. Decoding temporal information: a model based on short-term synaptic plasticity. *J. Neurosc.*, 20(3):1129–1141, 2000.

[116] Sajiya Jalil, Igor Belykh, and Andrey Shilnikov. Spikes matter for phase-locked bursting in inhibitory neurons. *Phys. Rev. E*, 85(3):036214, 2012.

[117] David Somers and Nancy Kopell. Rapid synchronization through fast threshold modulation. *Biol. Cybern.*, 68(5):393–407, 1993.

[118] Sajiya Jalil, Dane Allen, Joseph Youker, and Andrey Shilnikov. Toward robust phase-locking in melibe swim central pattern generator models. *Chaos*, 23(4):046105, 2013.

[119] Jeremy Wojcik, Justus Schwabedal, Robert Clewley, and Andrey L Shilnikov. Key bifurcations of bursting polyrhythms in 3-cell central pattern generators. *PloS One*, 9(4), 2014.

[120] Le Zhao and Alain Nogaret. Experimental observation of multistability and dynamic attractors in silicon central pattern generators. *Phys. Rev. E*, 92(5):052910, 2015.

[121] Matteo Lodi, Andrey Shilnikov, and Marco Storace. CEPAGEs: a toolbox for central pattern generator analysis. In *Proc. IEEE ISCAS*, pages 1–4, 2017.

[122] Deniz Alaçam and Andrey Shilnikov. Making a swim central pattern generator out of latent parabolic bursters. *Int. J. Bifurcat. Chaos*, 25(07):1540003, 2015.

[123] Richard E Plant. Bifurcation and resonance in a model for bursting nerve cells. *J. Math. Biol.*, 11(1):15–32, 1981.

[124] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.*, 94(5):3637–3642, 2005.

[125] Carmelo Bellardita and Ole Kiehn. Phenotypic characterization of speed-associated gait changes in mice reveals modular organization of locomotor networks. *Curr. Biol.*, 25(11):1426–1436, 2015.

[126] Maxime Lemieux, Nicolas Josset, Marie Roussel, Sébastien Couraud, and Frédéric Bretzner. Speed-dependent modulation of the locomotor behavior in adult mice reveals attractor and transitional gaits. *Front. Neurosci.*, 10:42, 2016.

[127] Sten Grillner and Alexander Kozlov. The CPGs for limbed locomotion–facts and fiction. *International Journal of Molecular Sciences*, 22(11):5882, 2021.

[128] Simon M Danner, Simon D Wilshin, Natalia A Shevtsova, and Ilya A Rybak. Central control of interlimb coordination and speed-dependent gait expression in quadrupeds. *J. Physiol.*, 594(23):6947–6967, 2016.

[129] Xiao-Jing Wang and John Rinzel. Alternating and synchronous rhythms in reciprocally inhibitory model neurons. *Neural computation*, 4(1):84–97, 1992.

[130] Norman C Heglund and C Richard Taylor. Speed, stride frequency and energy cost per stride: how do they change with body size and gait? *Journal of Experimental Biology*, 138(1):301–318, 1988.

[131] Timothy M Griffin, Rodger Kram, Steven J Wickler, and Donald F Hoyt. Biomechanical and energetic determinants of the walk–trot transition in horses. *Journal of Experimental Biology*, 207(24):4215–4223, 2004.

[132] J. Scully and A.L. Shilnikov. Personal communication, 2019.

[133] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[134] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[135] H-M Gutmann. A radial basis function method for global optimization. *Journal of global optimization*, 19(3):201–227, 2001.

[136] Efrén Mezura-Montes and Carlos A Coello Coello. Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm and Evolutionary Computation*, 1(4):173–194, 2011.

[137] Magnus Erik Hvass Pedersen. Good parameters for particle swarm optimization. *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001*, pages 1551–3203, 2010.

[138] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1989.

[139] Danielle S Bassett, Kathleen E Cullen, Simon B Eickhoff, Martha J Farah, Yukiko Goda, Patrick Haggard, Hailan Hu, Yasmin L Hurd, Sheena A Josselyn, Baljit S Khakh, et al. Reflections on the past two decades of neuroscience. *Nature Reviews Neuroscience*, 21(10):524–534, 2020.

[140] Raphael Liégeois, Augusto Santos, Vincenzo Matta, Dimitri Van De Ville, and Ali H Sayed. Revisiting correlation-based functional connectivity and its relationship with structural connectivity. *Network Neuroscience*, 4(4):1235–1251, 2020.

[141] Gustavo Deco, Viktor Jirsa, Anthony R McIntosh, Olaf Sporns, and Rolf Kötter. Key role of coupling, delay, and noise in resting brain fluctuations. *Proceedings of the National Academy of Sciences*, 106(25):10302–10307, 2009.

[142] Olaf Sporns. The human connectome: a complex network. *Annals of the new York Academy of Sciences*, 1224(1):109–125, 2011.

[143] Aaron Alexander-Bloch, Renaud Lambiotte, Ben Roberts, Jay Giedd, Nitin Gogtay, and Ed Bullmore. The discovery of population differences in network community structure: new methods and applications to brain functional networks in schizophrenia. *Neuroimage*, 59(4):3889–3900, 2012.

[144] Danielle S Bassett and Olaf Sporns. Network neuroscience. *Nature neuroscience*, 20(3):353–364, 2017.

[145] Olaf Sporns and Danielle S Bassett. New trends in connectomics, 2018.

[146] Christopher W Lynn and Danielle S Bassett. The physics of brain network structure, function and control. *Nature Reviews Physics*, 1(5):318–332, 2019.

[147] Gerald Hahn, Gorka Zamora-López, Lynn Uhrig, Enzo Tagliazucchi, Helmut Laufs, Dante Mantini, Morten L Kringelbach, Bechir Jarraya, and Gustavo Deco. Signature of consciousness in brain-wide synchronization patterns of monkey and human fmri signals. *NeuroImage*, 226:117470, 2021.

[148] Ashwini Oswal, Chunyan Cao, Chien-Hung Yeh, Wolf-Julian Neumann, James Gratwicke, Harith Akram, Andreas Horn, Dianyou Li, Shikun Zhan, Chao Zhang, et al. Neural signatures of hyperdirect pathway activity in parkinson's disease. *Nature communications*, 12(1):1–14, 2021.

[149] Andrzej K Nowak, Robin R Vallacher, Ryszard Praszkier, Agnieszka Rychwalska, and Michal Zochowski. *In sync: The emergence of function in minds, groups and societies*, chapter 2, 3, 4. Springer Nature, 2020.

[150] Xiao-Jing Wang. Neurophysiological and computational principles of cortical rhythms in cognition. *Physiological reviews*, 90(3):1195–1268, 2010.

[151] Hugh R Wilson and Jack D Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal*, 12(1):1–24, 1972.

[152] Hugh R Wilson and Jack D Cowan. Evolution of the wilson–cowan equations. *Biological cybernetics*, 115(6):643–653, 2021.

[153] Karl J Friston, Lee Harrison, and Will Penny. Dynamic causal modelling. *Neuroimage*, 19(4):1273–1302, 2003.

[154] Louis M Pecora and Thomas L Carroll. Master stability functions for synchronized coupled systems. *Physical Review Letters*, 80(10):2109, 1998.

[155] Louis M Pecora, Francesco Sorrentino, Aaron M Hagerstrom, Thomas E Murphy, and Rajarshi Roy. Cluster synchronization and isolated desynchronization in complex networks with symmetries. *Nature communications*, 5(1):1–8, 2014.

[156] Michael T Schaub, Neave O'Clery, Yazan N Billeh, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona. Graph partitions and cluster synchronization in networks of oscillators. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(9):094821, 2016.

[157] Abu Bakar Siddique, Louis Pecora, Joseph D Hart, and Francesco Sorrentino. Symmetry-and input-cluster synchronization in networks. *Physical Review E*, 97(4):042217, 2018.

[158] MRI Cloud. https://neurodata.io/mri/. Accessed: 21-07-2022.

[159] Olaf Sporns. *Networks of the Brain*. MIT press, 2016.

[160] Joana Cabral, Morten L Kringelbach, and Gustavo Deco. Functional connectivity dynamically evolves on multiple time-scales over a static structural connectome: Models and mechanisms. *NeuroImage*, 160:84–96, 2017.

[161] Xiao Liu, Xiao-Hong Zhu, Peihua Qiu, and Wei Chen. A correlation-matrix-based hierarchical clustering method for functional connectivity analysis. *Journal of neuroscience methods*, 211(1):94–102, 2012.

[162] Yanlu Wang and Tie-Qiang Li. Analysis of whole-brain resting-state fmri data using hierarchical clustering approach. *PloS one*, 8(10):e76315, 2013.

[163] Kristen N Warren, Molly S Hermiller, Aneesha S Nilakantan, Jonathan O'Neil, Robert T Palumbo, and Joel L Voss. Increased fmri activity correlations in autobiographical memory versus resting states. *Human brain mapping*, 39(11):4312–4321, 2018.

[164] Stephen J Gotts, Adrian W Gilmore, and Alex Martin. Brain networks, dimensionality, and global signal averaging in resting-state fmri: Hierarchical network structure results in low-dimensional spatiotemporal dynamics. *NeuroImage*, 205:116289, 2020.

[165] Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.

[166] Romesh G Abeysuriya, Jonathan Hadida, Stamatios N Sotiropoulos, Saad Jbabdi, Robert Becker, Benjamin AE Hunt, Matthew J Brookes, and Mark W Woolrich. A biophysical model of dynamic balancing of excitation and inhibition in fast oscillatory large-scale networks. *PLoS computational biology*, 14(2):e1006007, 2018.

[167] Sean R Bittner, Ryan C Williamson, Adam C Snyder, Ashok Litwin-Kumar, Brent Doiron, Steven M Chase, Matthew A Smith, and Byron M Yu. Population activity structure of excitatory and inhibitory neurons. *PloS one*, 12(8):e0181773, 2017.

[168] Robert L Grubb Jr, Marcus E Raichle, John O Eichling, and Michel M Ter-Pogossian. The effects of changes in PaCO2 cerebral blood volume, blood flow, and vascular mean transit time. *Stroke*, 5(5):630–639, 1974.

[169] Ben Jeurissen, Maxime Descoteaux, Susumu Mori, and Alexander Leemans. Diffusion mri fiber tractography of the brain. *NMR in Biomedicine*, 32(4):e3785, 2019.

[170] Martin Golubitsky and Ian Stewart. Nonlinear dynamics of networks: the groupoid formalism. *Bulletin of the american mathematical society*, 43(3):305–364, 2006.

[171] Francesco Sorrentino and Louis Pecora. Approximate cluster synchronization in networks with symmetries and parameter mismatches. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(9):094823, 2016.

[172] Frank Uhlig. Simultaneous block diagonalization of two real symmetric matrices. *Linear Algebra and Its Applications*, 7(4):281–289, 1973.

[173] Takanori Maehara and Kazuo Murota. A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with general irreducible components. *Japan journal of industrial and applied mathematics*, 27(2):263–293, 2010.

[174] Kazuo Murota, Yoshihiro Kanno, Masakazu Kojima, and Sadayoshi Kojima. A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with application to semidefinite programming. *Japan Journal of Industrial and Applied Mathematics*, 27(1):125–160, 2010.

[175] Shirin Panahi, Isaac Klickstein, and Francesco Sorrentino. Cluster synchronization of networks via a canonical transformation for simultaneous block diagonalization of matrices. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(11):111102, 2021.

[176] Masanori Shimono and Naomichi Hatano. Efficient communication dynamics on macro-connectome, and the propagation speed. *Scientific reports*, 8(1):1–15, 2018.

[177] Matteo Lodi, Francesco Sorrentino, and Marco Storace. One-way dependent clusters and stability of cluster synchronization in directed networks. *Nature communications*, 12(1):1–13, 2021.

[178] Anastasiia Panchuk, David P Rosin, Philipp Hövel, and Eckehard Schöll. Synchronization of coupled neural oscillators with heterogeneous delays. *International Journal of Bifurcation and Chaos*, 23(12):1330039, 2013.

[179] Joana Cabral, Francesca Castaldo, Jakub Vohryzek, Vladimir Litvak, Christian Bick, Renaud Lambiotte, Karl Friston, Morten L Kringelbach, and Gustavo Deco. Metastable oscillatory modes emerge from synchronization in the brain spacetime connectome. *Communications Physics*, 5(1):1–13, 2022.

[180] David Meunier, Renaud Lambiotte, Alex Fornito, Karen Ersche, and Edward T Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3:37, 2009.

[181] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[182] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[183] Tanushree B Luke, Ernest Barreto, and Paul So. Complete classification of the macroscopic behavior of a heterogeneous network of theta neurons. *Neural computation*, 25(12):3207–3234, 2013.

[184] Carlo R Laing. Derivation of a neural field model from a network of theta neurons. *Physical Review E*, 90(1):010901, 2014.

[185] Ernest Montbrió, Diego Pazó, and Alex Roxin. Macroscopic description for networks of spiking neurons. *Physical Review X*, 5(2):021028, 2015.

[186] Stephen Coombes and Áine Byrne. Next generation neural mass models. In *Nonlinear dynamics in computational neuroscience*, pages 1–16. Springer, 2019.

[187] Halgurd Taher, Alessandro Torcini, and Simona Olmi. Exact neural mass model for synaptic-based working memory. *PLoS Computational Biology*, 16(12):e1008533, 2020.

[188] Gustavo Deco, Morten L Kringelbach, Viktor K Jirsa, and Petra Ritter. The dynamics of resting fluctuations in the brain: metastability and its dynamical cortical core. *Scientific reports*, 7(1):1–14, 2017.

[189] Anton V Chizhov, Artyom V Zefirov, Dmitry V Amakhin, Elena Yu Smirnova, and Aleksey V Zaitsev. Minimal model of interictal and ictal discharges “epileptor-2”. *PLoS computational biology*, 14(5):e1006186, 2018.

[190] Fei Liu, Jiang Wang, Chen Liu, Huiyan Li, Bin Deng, Chris Fietkiewicz, and Kenneth A Loparo. A neural mass model of basal ganglia nuclei simulates pathological beta rhythm in parkinson’s disease. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(12):123113, 2016.

[191] Anton Filipchuk, Joanna Schwenkgrub, Alain Destexhe, and Brice Bathellier. Awake perception is associated with dedicated neuronal assemblies in the cerebral cortex. *Nature Neuroscience*, pages 1–12, 2022.

[192] Sachin Ranade. A common voice for neural data. *Nature Neuroscience*, 25(12):1583, 2022.

[193] Maria Giulia Preti, Thomas AW Bolton, and Dimitri Van De Ville. The dynamic functional connectome: State-of-the-art and perspectives. *Neuroimage*, 160:41–54, 2017.

[194] A. Pepiciello, A. Vaccaro, D. Villacci, and F. Milano. A method to evaluate the inertial response of frequency controlled converter-interfaced generation. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, pages 1–6, 2020.

[195] Pieter Tielens and Dirk Van Hertem. The relevance of inertia in power systems. *Renewable and Sustainable Energy Reviews*, 55:999–1009, 2016.

[196] D del Giudice, A Brambilla, S Grillo, and F Bizzarri. Analysis of the impact of synthetic inertia on frequency variations and on turbine governor dead-bands. In *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), The Hague, Netherlands, 26-28 Oct. 2020*, pages 564–568. IEEE, 2020.

[197] Ana Fernández-Guillamón, Emilio Gómez-Lázaro, Eduard Muljadi, and Ángel Molina-Garcia. A review of virtual inertia techniques for renewable energy-based generators. In *Power Systems*. IntechOpen, 2020. DOI: 10.5772/intechopen.92651.

[198] Ujjwol Tamrakar, Dipesh Shrestha, Manisha Maharjan, Bishnu P Bhattarai, Timothy M Hansen, and Reinaldo Tonkoski. Virtual inertia: Current trends and future directions. *Applied Sciences*, 7(7):1–29, 2017.

[199] Ana Fernández-Guillamón, Emilio Gómez-Lázaro, Eduard Muljadi, and Ángel Molina-García. Power systems with high renewable energy sources: A review of inertia and frequency control strategies over time. *Renewable and Sustainable Energy Reviews*, 115:109369, 2019.

[200] Peter Wall and Vladimir Terzija. Simultaneous estimation of the time of disturbance and inertia in power systems. *IEEE Transactions on Power Delivery*, 29(4):2018–2031, 2014.

[201] Rakesh Kumar Panda, Abheejeet Mohapatra, and Suresh Chandra Srivastava. Online estimation of system inertia in a power network utilizing synchrophasor measurements. *IEEE Transactions on Power Systems*, 35(4):3122–3132, 2019.

[202] Guowei Cai, Bo Wang, Deyou Yang, Zhenglong Sun, and Lixin Wang. Inertia estimation based on observed electromechanical oscillation response for power systems. *IEEE Transactions on Power Systems*, 34(6):4291–4299, 2019.

[203] GR Moraes, A Berizzi, V Ilea, and G D'Antona. Inertia estimation of equivalent areas by a pmu-based approach following perturbations. In *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pages 1–6, 2018.

[204] Guido Rossetto Moraes, Fabio Pozzi, Valentin Ilea, Alberto Berizzi, Enrico Maria Carlini, Giorgio Giannuzzi, and Roberto Zaottini. Measurement-based inertia estimation method considering system reduction strategies and dynamic equivalents. In *2019 IEEE Milan PowerTech*, pages 1–6, 2019.

[205] Davide del Giudice and Samuele Grillo. Analysis of the sensitivity of extended kalman filter-based inertia estimation method to the assumed time of disturbance. *Energies*, 12(3):483, 2019.

[206] Junbo Zhang and Hanchen Xu. Online identification of power system equivalent inertia constant. *IEEE Transactions on Industrial Electronics*, 64(10):8098–8107, 2017.

[207] Xue Cao, Bruce Stephen, Ibrahim F Abdulhadi, Campbell D Booth, and Graeme M Burt. Switching markov gaussian models for dynamic power system inertia estimation. *IEEE Transactions on Power Systems*, 31(5):3394–3403, 2015.

[208] Flavio Allella, Elio Chiodo, Giorgio Maria Giannuzzi, Davide Lauria, and Fabio Mottola. On-line estimation assessment of power systems inertia with high penetration of renewable generation. *IEEE Access*, 8:62689–62697, 2020.

[209] Reynaldo F Nuqui and Arun G Phadke. Phasor measurement unit placement techniques for complete and incomplete observability. *IEEE Transactions on Power Delivery*, 20(4):2381–2388, 2005.

[210] Federico Bizzarri, Davide del Giudice, Samuele Grillo, Daniele Linaro, Angelo Brambilla, and Federico Milano. Inertia estimation through covariance matrix. *IEEE Transactions on Power Systems*, 2023.

[211] Fanhong Zeng, Junbo Zhang, Ge Chen, Zikun Wu, Siwei Huang, and Yingqi Liang. Online estimation of power system inertia constant under normal operating conditions. *IEEE Access*, 8:101426–101436, 2020.

[212] John J Grainger and William D Stevenson. *Power system analysis*. McGraw-Hill, 1994.

[213] Federico Milano and Rafael Zárate-Miñano. A systematic method to model power systems as stochastic differential algebraic equations. *IEEE Transactions on Power Systems*, 28(4):4537–4544, 2013.

[214] Federico Bizzarri and Angelo Brambilla. PAN and MPanSuite: Simulation vehicles towards the analysis and design of heterogeneous mixed electrical systems. In *2017*

New Generation of CAS (NGCAS), Genoa (Italy), 6-9 Sept. 2017, pages 1–4. IEEE, 2017.

[215] Daniele Linaro, Davide del Giudice, Federico Bizzarri, and Angelo Brambilla. Pansuite: A free simulation environment for the analysis of hybrid electrical power systems. *Electric Power Systems Research*, 212:108354, 2022.

[216] Ning Zhou, John W Pierre, and John F Hauer. Initial results in power system identification from injected probing signals using a subspace method. *IEEE Transactions on power systems*, 21(3):1296–1302, 2006.

[217] Peter Makolo, Ramon Zamora, and T Lie. Online inertia estimation for power systems with high penetration of RES using recursive parameters estimation. *IET Renewable Power Generation*, 2021.

[218] Federico Bizzarri and Angelo Brambilla. Generalized power flow analysis of electrical power systems modeled as mixed single-phase/three-phase sub-systems. *IEEE Transactions on Power Systems*, 35(2):1284–1293, 2020.

[219] Federico Bizzarri, Davide del Giudice, Daniele Linaro, and Angelo Brambilla. Numerical approach to compute the power flow solution of hybrid generation, transmission and distribution systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(5):936–940, 2020.

[220] Richard H Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical programming*, 89(1):149–185, 2000.

[221] Yan Jiang, Richard Pates, and Enrique Mallada. Performance tradeoffs of dynamically controlled grid-connected inverters in low inertia power systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5098–5105. IEEE, 2017.

[222] Ricky J Concepcion, Felipe Wilches-Bernal, and Raymond H Byrne. Effects of communication latency and availability on synthetic inertia. In *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2017.

[223] Gholam Ali Alizadeh, Tohid Rahimi, Mohsen Hasan Babayi Nozadian, Sanjeevikumar Padmanaban, and Zbigniew Leonowicz. Improving microgrid frequency regulation based on the virtual inertia concept while considering communication system delay. *Energies*, 12(10):2016, 2019.

[224] Siqi Fu, Yao Sun, Zhangjie Liu, Xiaochao Hou, Hua Han, and Mei Su. Power oscillation suppression in multi-vsg grid with adaptive virtual inertia. *International Journal of Electrical Power & Energy Systems*, 135:107472, 2022.

[225] Elyas Rakhshani and Pedro Rodriguez. Inertia emulation in ac/dc interconnected power systems using derivative technique considering frequency measurement effects. *IEEE Transactions on Power Systems*, 32(5):3338–3351, 2016.

[226] Thongchart Kerdphol, Fathin Saifur Rahman, Masayuki Watanabe, and Yasunori Mitani. Robust virtual inertia control of a low inertia microgrid considering frequency measurement effects. *IEEE Access*, 7:57550–57560, 2019.

[227] Andrea Bonfiglio, Matteo Lodi, Alessandro Rosini, Alberto Oliveri, and Renato Procopio. Design, realization and testing of a synthetic inertia controller for wind turbine power generators. *submitted to the International Journal of Electrical Power & Energy Systems*, 2023.

[228] Andrea Bonfiglio, Marco Invernizzi, Alessandro Labella, and Renato Procopio. Design and implementation of a variable synthetic inertia controller for wind turbine generators. *IEEE Transactions on Power Systems*, 34(1):754–764, 2018.

[229] Andrea Bonfiglio. SMART WIND: sviluppo prototipale di controllore ausiliario per supporto di frequenza per generatori eolici non inerziali.

https://unige.it/ricerca/smart-wind-sviluppo-prototipale-controllore-ausiliario-supporto-frequenza-generatori-eolici. Accessed: 11.01.2023.

[230] Gregory Kiar, Eric W Bridgeford, William R Gray Roncal, Vikram Chandrashekhar, Disa Mhembere, Sephira Ryman, Xi-Nian Zuo, Daniel S Margulies, R Cameron Craddock, Carey E Priebe, et al. A high-throughput pipeline identifies robust connectomes but troublesome variability. *bioRxiv*, page 188706, 2018.