

Universal Computation and Memory by Neural Switching

Dissertation
for the award of the degree
“Doctor of Philosophy” (Ph.D.)

Division of Mathematics and Natural Sciences
Georg-August-University, Göttingen

submitted by

Fábio Schittler Neves

from Santa Rosa, Brazil

Göttingen 2010

Advisor : Prof. Dr. Marc Timme (reviewer)
Co-advisors : Prof. Dr. Florentine Wörgötter (reviewer)
: Prof. Dr. Theo Geisel

Day of the oral examination : 28.10.2010

To my parents for their support and love

Contents

1	Introduction	9
2	Fundamentals	13
2.1	Concepts of orbits and saddles	13
2.1.1	Real-time dynamical systems	13
2.1.2	Discrete-time dynamical systems	14
3	Symmetrically pulse-coupled oscillators	15
3.1	Network of pulse-coupled oscillators	15
3.2	Synchronization mechanisms	17
3.3	Integrate-and-Fire oscillators	18
4	Periodic orbit dynamics and clustered states	19
5	Nonlinear dynamics of controlled switching	23
5.1	Clustered state symmetry $\mathbf{S}_3 \times \mathbf{S}_2$	24
5.2	Clustered state symmetry $\mathbf{S}_4 \times \mathbf{S}_1$	27
5.3	Clustered state symmetry $\mathbf{S}_2 \times \mathbf{S}_2 \times \mathbf{S}_1$	30
5.4	Conclusion	32
6	Noise-induced switching and complex periodic orbits	33
6.1	Noise-induced switching	33
6.1.1	$\mathbf{S}_3 \times \mathbf{S}_2$ and $\mathbf{S}_4 \times \mathbf{S}_1$ cluster symmetries	34
6.1.2	$\mathbf{S}_2 \times \mathbf{S}_2 \times \mathbf{S}_1$ cluster symmetry	35
6.2	Complex periodic orbits in asymmetric networks	39
6.2.1	Computational aspects: general remarks	41
6.3	Conclusion	43
7	Computing by switching dynamics	45
7.1	Bitwise computation	45
7.2	Autonomous navigation: field gradients	51
7.3	Conclusion	54
8	Memory by switching dynamics	57
8.1	Short-term memory	57
8.2	Visual memory: 6-bit RGB color palette	60
8.3	Conclusion	62

9 Discussion	63
Appendix	68
A Periodic orbits: non-perturbed system	68
B Partial derivatives	70
C Unperturbed orbits: return maps	72
D Perturbed orbit dynamics: return maps.	75
Bibliography	79
Acknowledgments	82

List of Figures

3.1	Dynamics of two pulse-coupled oscillators	16
4.1	Diagrams of clustered states	19
4.2	Clustered states in different descriptions: V , ϕ and pulse sequences	21
5.1	Switchings due to negative perturbations starting from a $S_3 \times S_2$ state	26
5.2	Switchings due to positive perturbations starting from a $S_3 \times S_2$ state	27
5.3	Switchings due to positive perturbations starting from a $S_4 \times S_1$ state	28
5.4	Example of symmetry change by perturbation-induced switching due to negative perturbations starting from a $S_4 \times S_1$ state	29
5.5	Switchings due perturbations starting from a $S_2 \times S_2 \times S_1$ state . .	31
5.6	Five steps state transition diagram for the symmetry $S_2 \times S_2 \times S_1$.	31
6.1	Noise-induced switching examples.	35
6.2	Noise-induced switching process: $S_2 \times S_2 \times S_1$ states.	36
6.3	Random switching sequence induced by noise.	36
6.4	Noise-generated paths in a state space hyperplane	37
6.5	Noise-induced switching: residence times distribution diagram. . . .	38
6.6	Mean transition time as a function of noise intensity	38
6.7	Superposition of induced complex periodic orbit and all possible transitions in state space	40
6.8	Asymmetry-induced switching: residence time histogram	41
6.9	Relation between symmetry-breaking inputs and complex periodic orbits	42
6.10	Relation between percept classes and complex periodic orbits	43
7.1	Bitwise computation realized by switching	47
7.2	Computation representation as a set transformation in two steps. . .	49
7.3	Three bitwise operations: superposition of currents.	50
7.4	Error fraction vanishes with time after function switched.	50
7.5	Diagram of autonomous agent.	52
7.6	Relative positions of agent and radial source field.	53
7.7	Agent converges towards moving radial source field	54
8.1	Threshold shift for four different set of parameters as a function of time.	58
8.2	Example of dynamical memory by threshold shift.	59
8.3	Color detection: network diagram.	61
8.4	Visual memory example.	62

A.1	Periodic orbit for cluster symmetry $S_3 \times S_2$	68
A.2	Periodic orbit for cluster symmetry $S_4 \times S_1$	69
A.3	Periodic orbit for cluster symmetry $S_2 \times S_2 \times S_1$	69

Chapter 1

Introduction

Computation defines the way we interact with the world around us and constitutes an important field of contemporary research. Although computation has different definitions, one of its general aspects is the capability to interpret and process information. Systems with such capabilities can be implemented in hardware as different as digital computers, mechanical machines and biological systems. Depending on the nature of the implementation, different computational paradigms are expected to arise. Whereas digital processors are designed to implement sequential bitwise operations, complex systems such as biological or artificial neural networks present a variety of non-trivial collective dynamics related to memory and cognition [1–6].

Complex systems constitute a broad class of systems particularly promising for the discovery and design of new computational paradigms [7–10]. A complex system can be defined in general as a collection of interactive components restricted to a certain topology. Such systems may present unpredictable collective dynamics even if the dynamics of each single component is well characterized. Thus, such systems have computational features of two different levels, one regarding its components (that could be complex systems by themselves) and another regarding its collective dynamics. Often the focus of interest lies on the collective dynamics rather than in modeling its components. In these cases different levels of abstraction may be implemented in its components and interactions in order to simplify the problem mathematically, while isolating the features responsible for the dynamical behavior of interest.

Networks of pulse-coupled elements are complex systems handling information in the most powerful computer known to date—the human brain. Neuronal circuits are networks of non-linear elements (neurons) that interact via electrical signals. A neuron receives input pulses and, depending only on its internal potential (local information), may itself release a pulse depending on well described non-linear processes. Furthermore, it is believed that complex behavior arising in these networks should not depend on the specific form of the generic non-linearity but on the very presence of non-linearities and on the network topology. This is one of the reasons why the study of idealized neuronal models and/or small networks are powerful tools to better understand complex behaviors present in a broader class of complex systems.

Pulse-coupled oscillators are neuronal models that, even though highly simplified, keep characteristic features of biological neurons [11]. A biological neuron when constantly charged by an external current exhibits a slow monotonic increase of its membrane potential until it reaches a threshold value. When this happens, there is a sudden and fast increase of the potential, followed by a relaxation to its rest value (action-potential). Pulsed-coupled oscillator models consist of threshold elements that are driven monotonically from one initial phase (rest potential) to a threshold and then instantaneously reset to an initial phase. Note that while the slow processes in those elements are similar, the fast dynamics related to the action-potential is replaced by a simple reset in the oscillator model. This is motivated by the difference of magnitude in the time taken by these two main processes involved in the neuronal dynamics. Furthermore, pulses that are dynamically generated in real neurons are postulated to be released at the reset times in the oscillator model. A further simplification may be made, when assuming that the width of the postsynaptic response to an incoming pulse is much smaller than the oscillator's intrinsic period, idealizing the pulses to delta-pulses. Thus one pulse delivers an instantaneous phase jump to connected oscillators. This last simplification yields an analytical event-based computation of the system dynamics [12], avoiding numerical error and increasing the speed of simulations.

Networks of pulse-coupled oscillators have been successfully used to investigate mechanisms underlying dynamical phenomena in biological systems such as synchronization and desynchronization of network components that constitute the basis for a variety of collective dynamics with different complexities, conceptually responsible for encoding and processing information in those systems. The complexity of synchronous states can be characterized by its cyclic order. The simplest class is given by period-one states, where all elements must elicit one pulse before a second pulse is elicited by any component of the network. Its most well-known examples are fully synchronized [12, 13] and phase-locked states [14]. More complex states are achieved for cycles of higher order. Other examples of complex collective phenomena are asynchronous states and partial synchronization [15, 16]. Simplified models such as pulse-coupled oscillators, thus, may constitute important tools for the study of new collective dynamics.

It was shown in recent works that providing delayed interactions to systems with symmetry [17, 18] may support heteroclinically connected saddle periodic orbits (partially synchronized states). A heteroclinic connection in a dynamical system is a distinguished trajectory that links two saddles in state space. It occurs if unstable directions of one saddle are contained in the stable manifold of a second saddle. A sequence of such connections linking several saddles cyclically is called a heteroclinic cycle. Their relations to symmetry make heteroclinic cycles of high current interest mathematically [17–23]. Simultaneously, their specific dynamical feature, supporting repetitive switching close to the saddles, poses a promising challenge for the study of information encoding and computation, in particular in bio-inspired systems [5, 6, 10, 15, 16, 24–26].

For instance, it becomes more and more clear how information may be encoded by systems with heteroclinic cycles [3, 6, 27]. Recent studies have even provided insights about how external perturbations may be processed [8, 10, 28], providing

hints on how such systems may actually compute using switching among saddles. Nevertheless, it is still unclear whether and how biologically inspired neural systems may actually perform generic computations using switching trajectories.

In this thesis we investigate this new computational paradigm, i.e. how controllable switching trajectories may be exploited for computation. Our results yield a theoretical basis for universal computation in biologically inspired systems and a novel concept of artificial short-term memory, consistent with current views on working memory in the brain.

This work has three main parts divided into nine concise chapters: the first part is introductory. It is composed of this brief introduction, Chapter 2 and Chapter 3. Together they provide the context (Chapter 1), all fundamental definitions necessary for a better understanding of the following chapters (Chapter 2) and introduce the model class of pulse-coupled neurons on which we focus this study (Chapter 3). In the second part we derive the theoretical basis for computation by persistent switching processes (Chapters 4-6). In Chapter 4 we present our approach to find and to characterize partially synchronized states. In Chapter 5 we present a complete analysis of the stability properties for these states. In Chapter 6 we show how information may be encoded by inducing limit cycles close to a heteroclinic network. The third part is devoted to two different applications (Chapters 7 and 8). In Chapter 7 we show how universal computation can be performed by exploiting heteroclinic cycles. Two examples of computation are provided. The first consists of a two bits bitwise computer composed of five oscillators capable of performing universal computation by encoding all basic logical operations (AND, OR, XOR); as a second example we present an autonomous agent exhibiting spatially distributed sensors that is capable of detecting and following a moving source field by calculating the field gradient. In Chapter 8 we present a novel model for artificial short-term memory that exploits perturbed heteroclinic connections in the presence of network asymmetries. As an actual example of short-term memory, we provide a network architecture to memorize pictures with six bits colored pixels. In Chapter 9 we summarize our results and discuss their implications.

Chapter 2

Fundamentals

This chapter provides the fundamentals necessary to better understand the following chapters. Here, we introduce few definitions of orbits and saddles relevant to this work.

2.1 Concepts of orbits and saddles

This work is focused in collective oscillatory dynamics arising in networks of oscillators. Thus, it is convenient to briefly explain the adequate terminology in order to avoid confusion with non-oscillatory systems. We first introduced the definitions in the context of real-time dynamical systems, and second for discrete-time dynamical systems.

2.1.1 Real-time dynamical systems

Consider a dynamical system with time evolution given by a ordinary differential equation

$$\dot{x} = f(x), \tag{2.1}$$

and its Taylor expansion around a point x^* ,

$$\dot{x} = f(x^*) + J\delta x + O(\delta x^2), \tag{2.2}$$

where J is a Jacobian matrix at $x = x^*$ and $O(\delta x^2)$ denotes higher order terms.

Saddle Point. A saddle point is a fixed point x^* satisfying $f(x^*) = 0$ that in local stability analysis the Jacobian matrix J of the linear approximation $\dot{x} = J\delta x$ exhibits no zero eigenvalue, at least one eigenvalue with positive real part and at least one eigenvalue with negative real part.

Periodic Orbit. If $\dot{x} = f(x)$ has a periodic solution $\tilde{x}(t + T) = \tilde{x}(t)$ for all t , the set $Po = \{\tilde{x}(t) | t \in \mathfrak{R}^+\}$ is called the periodic orbit representing such solution. The minimum T satisfying this solution is called the period of the orbit.

Saddle Periodic Orbit. A saddle periodic orbit is an orbit given by a periodic solution $\tilde{x}(t+T) = \tilde{x}(t)$ for all t that in Floquet analysis exhibits no zero Floquet multiplier, at least one Floquet multiplier with real positive part and at least one Floquet multiplier with negative real part.

Heteroclinic Connection. Given two saddle fixed points x' and x'' . We say that x' is heteroclinically connected to x'' if at least part of its unstable manifold $\omega_{x'}^u$ is contained in the stable manifold $\omega_{x''}^s$ of x'' .

Heteroclinic Cycle. A heteroclinic cycle is a collection of saddles and the heteroclinic orbits that cyclically connect them.

Heteroclinic Network. A heteroclinic network is a collection of heteroclinic cycles. It can be represented by a graph in which all vertices are saddles and all arrows are heteroclinic orbits.

2.1.2 Discrete-time dynamical systems

Consider a dynamical system with time evolution given by a map

$$x_{n+1} = g(x_n), \quad (2.3)$$

and its Taylor expansion around a point x^*

$$g(x^* + \delta x) = g(x^*) + J\delta x + O(\delta x^2), \quad (2.4)$$

where J is a Jacobian matrix at $x_n = x^*$. This equation can be rearranged as

$$g(x^* + \delta x) - g(x^*) = J\delta x + O(\delta x^2) \quad (2.5)$$

$$\delta g = J\delta x + O(\delta x^2). \quad (2.6)$$

Thus, δg is the difference after one iteration $x_{n+1} = g(x_n)$ between two states with close initial conditions $x_0 = x^*$ and $x'_0 = x^* + \delta x$.

Saddle Point. A saddle point is a fixed point x^* satisfying $g(x^*) = x^*$ that in local stability the Jacobian matrix J of the linear approximation $\delta g = J\delta x$ exhibits no eigenvalue equal one, at least one eigenvalue with real part larger than one and at least one eigenvalue with real part smaller than one.

Periodic Orbit. If $x_{n+1} = g(x_n)$ has a periodic solution $g^n(\tilde{x}_0) = \tilde{x}_0$, where $g^n(\tilde{x}_0)$ represents the map g iterated consecutively n times with an initial state \tilde{x}_0 , the set $Po = \{\tilde{x}_m | m \in \{0, \dots, n-1\}\}$ is called the periodic orbit representing such solution.

Saddle Periodic Orbit. A saddle periodic orbit is an orbit given by a periodic solution $g^n(\tilde{x}_0) = \tilde{x}_0$ where n is the number of iterations that in Floquet analysis exhibits no Floquet multiplier equal to one, at least one Floquet multiplier with real part larger than one and at least one Floquet multiplier with real part smaller than one.

Chapter 3

Symmetrically pulse-coupled oscillators

In this work we study a class of systems known as pulse-coupled threshold units in the special case where the units are constantly driven towards their thresholds (exhibiting a periodic dynamics when not coupled). As shown by Mirolo and Strogatz [12], given certain conditions their dynamics may be described in two distinct, but equivalent, representations: either through numerical integration of a potential-like variable or through analytical event-based maps. Throughout this work we will use both approaches depending on the aspect to be studied. Whereas stability properties and transition rules are studied in the event description, because it yields an analytical approach (see Chapter 5), the analysis of induced cycles (Chapter 6) and computation examples (Chapters 8 and 7) are evaluated in the potential picture.

This chapter is divided into three sections. The first describes in a general form both representations and discusses conditions of their equivalence. The second describes mechanisms underlying synchronization in networks of pulse-coupled oscillators. The third introduces the special case of Integrate-and-Fire oscillators and defines a small network, in which we explain the phenomenon of persistent switching and its implications for computation.

3.1 Network of pulse-coupled oscillators

We consider a network of N oscillators that are connected homogeneously all-to-all without self-connections through delayed pulse-couplings. The state of each oscillator $i \in \{1, \dots, N\}$ at time t is specified by a single phase-like variable $\phi_i(t)$ [12, 15]. In the absence of interactions, its dynamics is given by

$$\frac{d\phi_i}{dt} = 1, \quad 0 \leq \phi_i \leq 1. \quad (3.1)$$

When oscillator i reaches its threshold $\theta = 1$, $\phi_i(t^-) = 1$, its phase is reset to zero, $\phi_i(t) = 0$, and the oscillator is said to send a pulse (see Figure 3.1). Such a pulse is sent to all other oscillators which receive this signal after a delay time τ . The

incoming signal induces a phase jump

$$\phi_i(t) = H_\epsilon(\phi_i(t^-)) = U^{-1} [U(\phi_i(t^-)) + \epsilon], \quad (3.2)$$

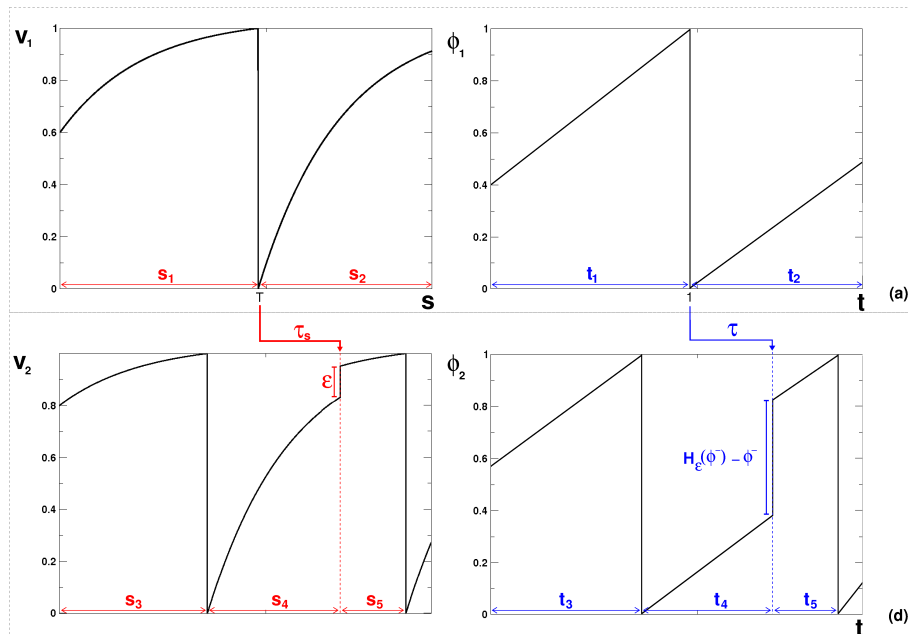


Figure 3.1: **Dynamics of two pulse-coupled oscillators.** Sketch of the time evolution of dynamic variables in both representations showing the one-to-one relation between them, it is $t_{1,2,3,4,5} = s_{1,2,3,4,5}/T_{\text{IF}}$ and $\tau = \tau_s/T_{\text{IF}}$; a) an oscillator is continuously charged and sends one pulse after reaching its threshold; b) a second identical oscillator, apart from a phase/potential lag, receives the pulse and performs an instantaneous phase/potential jump. While the coupling strength in the potential picture is fixed (ϵ), the effective coupling $H_\epsilon(\phi^-) - \phi^-$ in the phase picture adapts to maintain the equivalence between the event sequences.

which depends on the instantaneous phase $\phi_i(t^-)$ of the post-synaptic oscillator and the excitatory coupling strength $\epsilon > 0$. The phase dependence is determined by a twice continuously differentiable potential function $U(\phi)$ that is assumed to be strictly increasing ($U'(\phi) > 0$), concave down ($U''(\phi) < 0$), and normalized such that $U(0) = 0$, $U(1) = 1$. As shown in [25, 26], this phase dynamics is equivalent to the ordinary differential equations

$$\frac{dV_i}{ds} = f(V_i) + S_i(s), \quad (3.3)$$

where

$$S_i(s) = \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{k \in \mathbb{Z}} \epsilon \delta(s - \tau_s - s_{jk}), \quad (3.4)$$

is a sum of delayed δ -currents induced by presynaptic oscillators. Oscillator j sends its k th pulse at time s_{jk} whenever its phase variable crosses threshold, $V_j(s_{jk}) \geq 1$;

thereafter, it is instantaneously reset, $V_j(s_{jk}) \rightarrow 0$. The k th pulse of oscillator j is received by i after a delay τ_s . The positive function $f(V) > 0$ yields a free ($S_i(s) \equiv 0$) solution $V_i(s) := V(s) = V(s + T_0)$ of intrinsic period T_0 . The above function $U(\phi)$ is related to this solution via

$$U(\phi) = V(\phi T_0), \quad (3.5)$$

defining a natural phase ϕ by rescaling the time axis, $t = s/T_0$ and $\tau = \tau_s/T_0$.

Throughout this work we will represent the system state in both equivalent forms, depending on the aspect to be studied. Whereas the stability properties are analytically derived in the phase representation (equations (3.1) and (3.2)), the study of induced cycles and computation examples are described in the potential picture (equations (3.3) and (3.4)). Figure 3.1 shows that the timing between all relevant events are indeed equivalent between the representations (one-to-one relation).

3.2 Synchronization mechanisms

In this work, we consider networks of generic pulse-coupled oscillators, biologically inspired neuronal models that present a monotonical increase of a phase-like variable $\phi(t)$ towards a threshold θ , with a subsequent reset to an initial phase $\phi(t^+) \rightarrow 0$, where t is the time variable and t^+ is the time immediately after an event occurring at t (infinitely close to t , from the right). In addition to this smooth dynamics and reset condition, a second strong non-linearity takes place: the reception of pulses that deliver infinitely fast phase jumps. These three characteristics combined provide four types of local dynamical events related to (non-local) synchronization and desynchronization, which we define now.

Sub-threshold input signal event. An arriving pulse ρ_i from oscillator i at time t is said to be a sub-threshold event to oscillator j if the resulting phase $\phi_j(t^+)$ is smaller than a threshold θ_j of oscillator j .

Supra-threshold input signal event. An arriving pulse ρ_i from oscillator i at time t is said to be a supra-threshold event to oscillator j if the resulting phase $\phi_j(t^+)$ is equal or larger than a threshold θ_j of oscillator j .

Reset event. A reset event is a compulsory instantaneous phase change $\phi_i(t^+) \rightarrow 0$, that occurs each time $\phi_i(t) \geq \theta_i$.

Synchronization by supra-threshold event. A synchronization by supra-threshold event is said to occur for a pair of oscillators $\{i, j\}$ if at any time t a supra-threshold input signal occur simultaneously at i and j , such that initially different phases $\phi_i(t) \neq \phi_j(t)$ become equal after the event: That is, $\phi_i(t^+) \rightarrow 0$ and $\phi_j(t^+) \rightarrow 0$, and thus $\phi_i(t^+) \equiv \phi_j(t^+)$.

3.3 Integrate-and-Fire oscillators

In this work we focus in a neuronal oscillator model known as the leaky integrate-and-fire oscillator [11]. It is a one-dimensional threshold model defined as

$$\frac{dV_i}{ds} = I - \gamma V_i + S_i(s), \quad (3.6)$$

where I is a driving current and γ is a dissipative factor. The interaction $S_i(s)$ is given by (3.4).

From equation (3.6) we recognize

$$f(V) = I - \gamma V_i, \quad (3.7)$$

of the general form (3.1) and calculate the oscillator's intrinsic period

$$T_{\text{IF}} = \frac{1}{\gamma} \log \left(1 - \frac{\gamma}{I} \right)^{-1}. \quad (3.8)$$

With equations (3.7), (3.8) and (3.5) we find the specific form of $U(\phi)$

$$U(\phi_i) = U_{\text{IF}}(\phi_i) = \frac{I_i}{\gamma} (1 - e^{-\phi_i T_{\text{IF}}}), \quad (3.9)$$

that together with equation (3.1) completes our set of equation for both representations.

For the sake of clarity, we here focus on a small network already exhibiting a rich switching dynamics, a globally coupled network of $N = 5$ oscillators. This choice makes the system mathematically simpler, while it still exhibits a rich variety of non-trivial dynamics as a complex network of heteroclinically connected saddle periodic orbits (see Chapters 4 and 5). Although in principle this choice of network size and model may seem restrictive, the results that follow are general with respect to those two aspects. It was shown in [25] that unstable attractors are prevalent in larger networks, enhancing the likelihood of finding heteroclinic cycles. As a concrete example, it was shown for $N = 100$ a persistent switching among states with cluster symmetry $S_{21} \times S_{21} \times S_{21} \times S_{21} \times S_{16}$. Regarding the oscillator model, any $U(\phi)$ sufficiently close to $U_{\text{IF}}(\phi)$ brings qualitatively the same results because, as will be shown in the following chapters, the phenomenon of controllable switching will depend rather on the properties of heteroclinic cycles than on the exact slope of V .

Chapter 4

Periodic orbit dynamics and clustered states

This work is focused on the study of spatiotemporal patterns arising in networks of pulse-coupled oscillators. As shown in the previous chapter, when a constant external input I_i to a single oscillator i is sufficiently strong to drive the membrane potential to cross a predefined threshold ($U' > 0$), the potential dynamics, and thus the phase dynamics, becomes periodic with period T_0 . Therefore, networks of such oscillators exhibit different invariant sets in the form of periodic orbits [15, 21, 25, 29]. It was known before that, provided delayed connections, such networks exhibit partially synchronized states, periodic orbits where groups of oscillators are identically synchronized into clusters, and may support a persistent switching between unstable yet attractive orbits through heteroclinic cycles. In this work we will often refer to the clustered states by its symmetry, where each cluster contributes to the total symmetry with a permutation group S_n , where n is the number of elements in the cluster. In this section, we identify and characterize three distinct partially synchronized states that robustly occur for a network with $N = 5$ oscillators.

To explore the possible unstable periodic orbits we systematically varied the

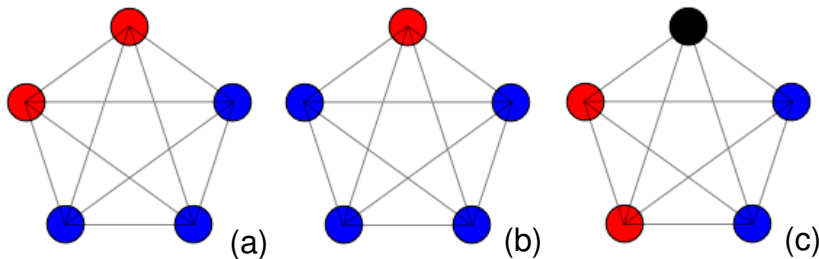


Figure 4.1: **Diagrams of three main clustered states.** Sketch of three globally coupled networks with $N = 5$ presenting three distinct clustered states. Each color label one clusters. Equivalent states can be achieved by node permutations. a) Displays one of the 10 possible $S_3 \times S_2$ states; b) displays one of the 5 possible $S_4 \times S_1$ states; c) displays one of the 30 possible $S_2 \times S_2 \times S_1$ states.

network parameters and initial conditions, and found numerically that three clustered states present the most persistent state symmetries to perturbations (seems to preserve the total symmetry). Two of these states are composed of two clusters, with permutation symmetries $S_3 \times S_2$ and $S_4 \times S_1$, respectively; another one is composed of two clusters and one single element, with permutation symmetry $S_2 \times S_2 \times S_1$. A graphical representation of those three states is displayed in Figure 4.1.

After defining numerically the most promising orbits, we derived analytical event sequences (tables C.1, C.3, C.4 and C.6) that fully characterize each orbit. For the symmetry $S_3 \times S_2$, the resulting orbit is such that the event sequence is

$$\xi_1 = (\sigma_1, \sigma_2, \sigma_3)(\rho_1, \rho_2, \rho_3, \sigma_4, \sigma_5)(\rho_4, \rho_5)(\sigma_1, \sigma_2, \sigma_3) \dots, \quad (4.1)$$

where each parentheses defines one given instant, σ_i represents a elicited pulse by oscillator i and ρ_i represents the reception of a pulse elicited by oscillator i . In the same way, the sequences defining the orbits with symmetries $S_4 \times S_1$ and $S_2 \times S_2 \times S_1$ are respectively given by

$$\xi_2 = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)(\rho_1, \rho_2, \rho_3, \rho_4, \sigma_5)(\rho_5)(\sigma_1, \sigma_2, \sigma_3, \sigma_4) \dots, \quad (4.2)$$

and

$$\xi_3 = (\sigma_1, \sigma_2)(\rho_3^*, \rho_4^*, \sigma_5)(\rho_1, \rho_2, \sigma_3, \sigma_4)(\rho_5)(\sigma_1, \sigma_2) \dots \quad (4.3)$$

where the stars indicate that the event results from a pulse coming from the previous cycle. For each of these three cluster periodic orbits, the event sequence of sending and reception of pulses fully defines the type of periodic orbit such that the analytical conditions for existence of a family of such orbits can be directly read from the detailed event sequences (Appendix C). In particular, these three families of periodic orbits exist for an open set of parameters close to the three examples numerically specified in tables C.2, C.5 and C.7. The existence conditions for each periodic orbit naturally imply that the phases of all oscillators return exactly to the same value after a fixed period; at the same time, the predefined event sequence must be kept. Figure 4.2 displays a graphical representation of three limit cycles as a sequence of pulses and as a Poincaré section. For a visual representation of the complete orbits in state space, please refer to Appendix A.

As shown in the event tables, the clusters persist synchronized at all times after one periodic orbit is established (in the absence of perturbations) and each oscillator elicit only one spike during one cycle. Thus, these orbits are uniquely defined by a cluster-state vector that samples one single point over these orbits (Poincaré section). Here we choose the point s' such that $V_1(s') = 0$, that yields a vector

$$\mathbf{V}(s') = (0, V_2(s'), V_3(s'), V_4(s'), V_5(s')), \quad (4.4)$$

unique during one cycle. Equivalent vectors are yield by the periodic condition, $V(s') \equiv V(s' + nT)$ with $n = 1, 2, 3, \dots$ where T is the period of the orbit. In order to simplify the notation, we now substituted the exact potential of each oscillator

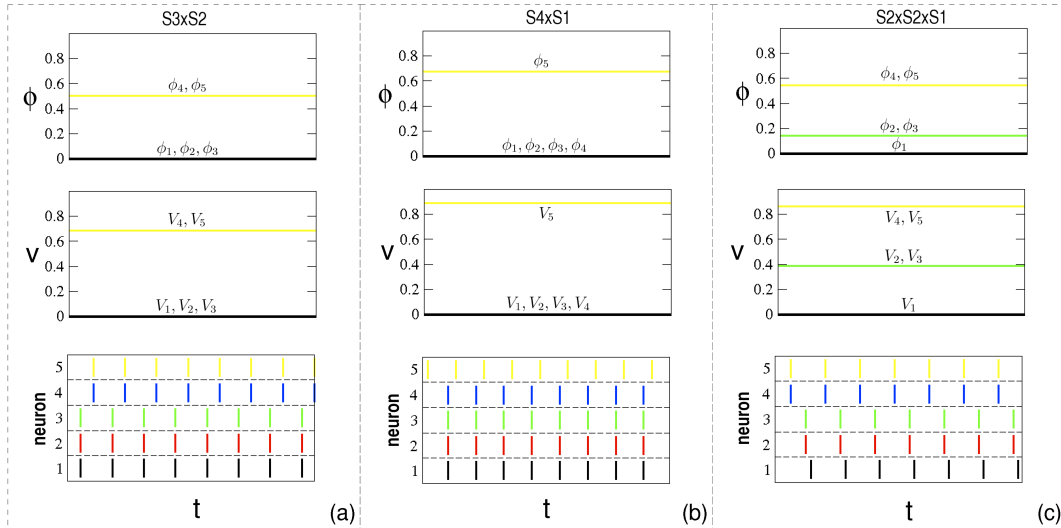


Figure 4.2: **Clustered states in different descriptions: V , ϕ and pulse sequences.** Three distinct limit cycles for a globally coupled networks with $N = 5$ displayed as Poincareh sections, for both representations, and as a sequence of pulses. Each color correspond to one oscillator. a) Displays one of the 10 possible $S_3 \times S_2$ states, oscillators $\{1, 2, 3\}$ and $\{4, 5\}$ are synchronized ($\tau = 0.31$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$); b) displays one of the 5 possible $S_4 \times S_1$ states, oscillators $\{1, 2, 3, 4\}$ are synchronized ($\tau = 0.27$, $\epsilon = 0.0015$, $I = 1.1$, $\gamma = 1$); c) displays one of the 30 possible $S_2 \times S_2 \times S_1$ states, oscillators $\{2, 3\}$ and $\{4, 5\}$ are synchronized ($\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$).

at the moment oscillator number one is reset (σ_1 event times) by a cluster label $V_i \in \{a, b, c\}$, where a , b and c are cluster labels for the network elements (maximum of three labels since we consider symmetries with two and three clusters); a denotes an unstable cluster, b a stable cluster and c an isolated unit at a stable position. In an equivalent phase representation, taking t' such $\phi_1(t') = 0$ yields

$$\phi(t') = (0, \phi_2(t'), \phi_3(t'), \phi_4(t'), \phi_5(t')), \quad (4.5)$$

with the same label representation $\phi_i \in \{a, b, c\}$. The actual values of $\{a, b, c\}$ are determined by the specific cluster symmetry and the representation chosen. Note that there is no loss of generality by choosing to sample the orbit at $V_1(s') = 0$. Any other given point s'' over the orbit ($V_1(s'')$) or other choice of oscillator (V_i with $i \neq 1$) yields an equivalent vector with different values of $\{a, b, c\}$. In particular, all states with cluster symmetries $S_3 \times S_2$, $S_4 \times S_1$ and $S_2 \times S_2 \times S_1$ are represented by the set of all permutations of the vectors:

$$\mathbf{V}(s') = (a, a, a, b, b) \Leftrightarrow \phi(t') = (a', a', a', b', b'), \quad (4.6)$$

$$\mathbf{V}(s') = (a, a, a, a, b) \Leftrightarrow \phi(t') = (a', a', a', a', b'), \quad (4.7)$$

$$\mathbf{V}(s') = (a, a, b, b, c) \Leftrightarrow \phi(t') = (a', a', b', b', c'), \quad (4.8)$$

where each vector element on the right hand side of each equality represents one oscillator and the letters indicate to which cluster it belongs. Here $\{a', b', c'\}$ were used instead of $\{a, b, c\}$ to emphasize that the values denoted by the labels are different for each representation. As mentioned above, the actual values of $\{a, b, c\}$ depend also on the symmetry of the state. The number of permutation-equivalent

configurations for each state symmetry is given by the number of permutations of the vectors presented above, which results in

$$\binom{5}{3} = 10, \quad (4.9)$$

equivalent states for the symmetry $S_3 \times S_2$,

$$\binom{5}{4} = 5, \quad (4.10)$$

states for the symmetry $S_4 \times S_1$, and

$$\binom{5}{2,2} = 30, \quad (4.11)$$

states for the symmetry $S_2 \times S_2 \times S_1$.

In summary, these results suggest that for $N = 5$ three distinct cluster periodic orbits exhibits the most promising features to sustain a persistent switching dynamics. Analytical descriptions of each orbit are given by event sequences and a cluster-state vector has been introduced. In the next chapter, we study the stability of these cluster states and the possible state transitions when subject to small perturbations.

Chapter 5

Nonlinear dynamics of controlled switching

In this chapter we study, case by case, the stability of the periodic orbits presented in the last section and show that they present unstable and stable manifolds. We first show that the orbits are unstable by a local stability analysis and later, by studying the long term effect of perturbations, we show that these orbits are attractive, such that the basin of attraction of one orbit lies in the vicinity of other orbits. Furthermore, we provide general transition rules between clustered states with symmetry $S_2 \times S_2 \times S_1$. The content of this chapter was published in the "J. Phys. A: Math. Theor.", vol. 42, 2009."

To study the local stability of these attractors, we introduce a perturbation vector,

$$\boldsymbol{\delta}(n) = (\delta_2(n), \delta_3(n), \delta_4(n), \delta_5(n)), \quad (5.1)$$

that has four components, since only the relative phases among the oscillators are relevant. The analysis presented here consists of a study of the temporal evolution of this perturbation vector at each cycle. Thus,

$$\delta_i(n) := \phi_i(t_{1,n}) - \phi_i^*(t_{1,n}) \quad (5.2)$$

are the perturbations to phases on the periodic orbit just after oscillator one has sent its n th pulse and been reset, i.e. $\delta_1 \equiv 0$.

After a small enough initial perturbation that is added to the phase vector at $t = t_{1,1}$, the temporal evolution of the perturbation vector is defined as the difference between this perturbed vector after one cycle of the system dynamic and the unperturbed phase vector at the same time. Analytically tracking the periodic orbit dynamics (cf. tables D.1, D.2, and D.3) yields the perturbation vector after one cycle as a function of the perturbation in the previous cycle,

$$\boldsymbol{\delta}(n+1) = \mathbf{F}(\boldsymbol{\delta}(n)), \quad (5.3)$$

which can be linearly approximated by

$$\boldsymbol{\delta}(n+1) \doteq \mathbf{J}\boldsymbol{\delta}(n), \quad (5.4)$$

where \mathbf{J} is the Jacobian matrix at $\boldsymbol{\delta}(n) = 0$, describing the local dynamics.

After analyzing the local stability properties, we study non-local effects in response to single oscillator perturbations in order to reveal the attractive aspects of the periodic orbits. The procedure consists of perturbing only one oscillator at a time. We consider negative perturbations, instantaneous decrements on the phase, and positive ones, instantaneous increments on the phase. When possible, we include state-transition diagrams displaying all possible transitions.

5.1 Clustered state symmetry $\mathbf{S}_3 \times \mathbf{S}_2$

For the clustered state $S_3 \times S_2$, assuming $\delta_2 < \delta_3$, and $\delta_4 < \delta_5$, the temporal evolution for the perturbation vector $\boldsymbol{\delta}$ in one cycle is given by the difference between the perturbed and the unperturbed orbits after one cycle (last row of table D.1 and C.1, respectively), resulting in

$$\boldsymbol{\delta}(n+1) = \mathbf{F}(\boldsymbol{\delta}(n)) = (\delta_2(n+1), \delta_3(n+1), \delta_4(n+1), \delta_5(n+1)), \quad (5.5)$$

with

$$\delta_2(n+1) = H_{2\epsilon}(H_\epsilon(H_\epsilon(\delta_2 + \tau - \delta_3) + \delta_3) + \tau - \delta_2) - p_{1,2} \quad (5.6)$$

$$\delta_3(n+1) = H_{2\epsilon}(H_\epsilon(H_\epsilon(\tau + \delta_3 - \delta_2) + \delta_2) + \tau - \delta_2) - p_{1,2} \quad (5.7)$$

$$\delta_4(n+1) = H_{2\epsilon}(H_\epsilon(\delta_2) + \tau - \delta_2) + 1 - p_{1,2} \quad (5.8)$$

$$\delta_5(n+1) = H_{2\epsilon}(H_\epsilon(\delta_2) + \tau - \delta_2) + 1 - p_{1,2} \quad (5.9)$$

where

$$p_{1,2} = H_{2\epsilon}(H_\epsilon(H_\epsilon(\tau - \delta_3) + \delta_3 - \delta_2) + \tau) \quad (5.10)$$

Remember that H is the transfer function defined in (3.2).

These equations fully characterize the time evolution in one cycle. Now, to study its local dynamics we perform a linear approximation of (5.5),

$$\boldsymbol{\delta}(n+1) \doteq \mathbf{J}\boldsymbol{\delta}(n), \quad (5.11)$$

where the dynamics is dictated by the following Jacobian matrix:

$$\left. \frac{\partial \boldsymbol{\delta}(n+1)}{\partial \boldsymbol{\delta}(n)} \right|_{\boldsymbol{\delta}(n)=0} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ j_{21} & \beta & 0 & 0 \\ \gamma & j_{32} & 0 & 0 \\ \gamma & j_{42} & 0 & 0 \end{pmatrix}, \quad (5.12)$$

with,

$$\alpha = \left[-1 + \left[1 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) \right] H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (5.13)$$

$$\beta = \left[-1 + 2H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (5.14)$$

$$\gamma = \left[-1 + H'_\epsilon(0) \right] H'_{2\epsilon}(\tau + 0_\epsilon) + H'_\epsilon(\tau_\epsilon) H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (5.15)$$

$$j_{21} = - \left[1 + \left[-2 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) \right] H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (5.16)$$

$$j_{32} = j_{42} = \left[-1 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (5.17)$$

with the short notation

$$x_y = H_y(x), \quad (5.18)$$

for $x \in \{0, \tau, (\tau - \tau'), (\tau - \tau' + \tau'_y)\}$ and $y \in \{\epsilon, 2\epsilon, 3\epsilon\}$.

Here α , β , and γ are positive reals larger than one, j_{32} and j_{42} are positive, and j_{21} is much smaller than the other elements. This matrix has two zero eigenvalues with eigenvectors that correspond to the directions of δ_4 and δ_5 ; and two non-zero eigenvalues given by,

$$\lambda_1 = \alpha = \left[-1 + \left[1 + H'_\epsilon(\tau) \right] H'_\epsilon(H_\epsilon(\tau)) \right] H'_{2\epsilon}(\tau + H_{2\epsilon}(\tau)) \quad (5.19)$$

$$\lambda_2 = \beta = \left[-1 + 2H'_\epsilon(\tau) \right] H'_\epsilon(H_\epsilon(\tau)) H'_{2\epsilon}(\tau + H_{2\epsilon}(\tau)), \quad (5.20)$$

where

$$H'_\epsilon(\phi) = \frac{\partial}{\partial \phi} U^{-1}(U(\phi + \epsilon)).$$

Here λ_1 and λ_2 are larger than one (see Lemma 1), noticing that all terms are due to sub-threshold events. Thus a perturbation can effectively disturb the system in two different possible directions, showing that the cluster S_2 is stable and the S_3 is unstable.

Lemma 1 *If $H_\epsilon(\phi)$ given by (3.2) mediates a sub-threshold reception event and $\epsilon > 0$, $U'(\phi) > 0$, and $U''(\phi) < 0$, then $H'_\epsilon(\phi) > 1$.*

Proof Assume $\epsilon > 0$. By definition

$$H'_\epsilon(\phi) = \frac{\partial}{\partial \phi} U^{-1}(U(\phi + \epsilon)) = \frac{U'(\phi)}{U'(U^{-1}(U(\phi) + \epsilon))} = \frac{U'(\phi)}{U'(H_\epsilon(\phi))},$$

Since $U'(\phi)$ is a monotonic decreasing function and $H_\epsilon(\phi) > \phi$ we have $U'(\phi) > U'(H_\epsilon(\phi))$ for any $H_\epsilon(\phi)$, and consequently $H'_\epsilon(\phi) > 1$.

Now we describe the long-term effect of a single oscillator perturbation to the unstable cluster S_3 . A negative perturbation to one of the elements on the unstable cluster ($\phi^+ = (a, a, a - \delta_3, b, b)$) puts one of its elements phases slightly behind; then the initial stable cluster S_2 begins to receive an additional pulse just after it is reset, increasing its relative phase in each cycle, and thus approaches the phase of the elements in the originally stable cluster. After some cycles it finally joins that cluster by a simultaneous reset, forming a new $S_3 \times S_2$ clustered state. This switching process is illustrated just after the second and fourth perturbations in Figure 5.1. The final state has the same symmetry as the initial state but this time, while the clusters themselves are stable, the orbit described by the new cluster S_2 is unstable (see table C.3). A further perturbation to the cluster S_2 does not change the elements of each cluster but rather returns the system to the initial phase difference, as shown in the first, third and fifth perturbations in Figure 5.1.

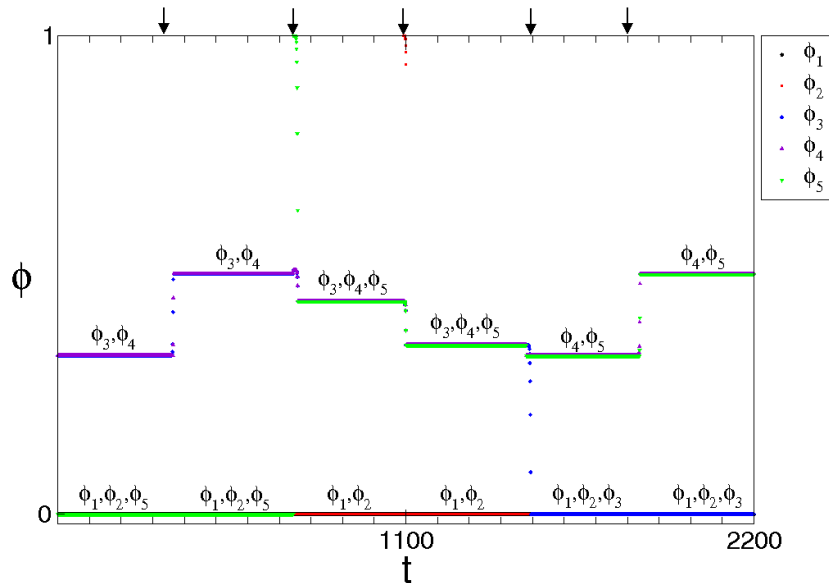


Figure 5.1: **Switchings due to negative perturbations.** Example of perturbation-induced switching starting from a $S_3 \times S_2$ state. The response of the system to a sequence of five negative single oscillator perturbations preserving a $S_3 \times S_2$ clustered symmetry ($\tau = 0.31$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$). The phases of all oscillators are plotted at the moment when oscillator one is reset, each color representing the phase of one oscillator. There are transitions through two steps, where in a first moment the cluster S_3 is unstable, after one perturbation (as shown by the second and fourth perturbation) it reaches a new configuration with the cluster S_2 in an unstable phase position, a second perturbation (as shown in the first, third and fifth perturbations) is needed to put the system in the initial phase difference again, maintaining the cluster components but changing its stability. The symmetry of the unstable attractors are preserved. The sequence of states given by the plateaus are $(a, a, b, b, a)^* \rightarrow (a, a, b, b, a) \rightarrow (b, b, a, a, a)^* \rightarrow (b, b, a, a, a) \rightarrow (a, a, a, b, b)^* \rightarrow (a, a, a, b, b)$, where the star indicate the states where S_2 is in an unstable phase.

Intriguingly, positive perturbations ($\phi^+ = (a, a, a + \delta_3, b, b)$) result in a completely different dynamic, as can be seen in Figure 5.2 which presents a sequence of three negative and two positive perturbations. A positive perturbation puts just one oscillator from the unstable cluster ahead, increasing its phase in relation to its original cluster in each cycle until it begins to be reset by pulses coming from the originally stable cluster. The original S_2 cluster changes its phase to conform with this new pulse configuration, but still has been reset by pulses coming from the two elements left on the unstable cluster. Thus the S_3 cluster splits into two clusters, and the new configuration becomes $S_2 \times S_2 \times S_1$. A further perturbation puts the system in a stable cyclic state.

Hence the symmetry $S_3 \times S_2$ is not preserved upon a general perturbation. however, simulations suggest that if we only consider negative single oscillator perturbations, or one negative perturbation with a larger magnitude than the others, the symmetry is preserved, and it is possible to write a transition rule that permits us to know which will be the next state after one perturbation.

Transition rule for cluster symmetry $S_3 \times S_2$:

$$(a - \delta_1, a, a, b, b) \rightarrow (a, b, b, a, a). \quad (5.21)$$

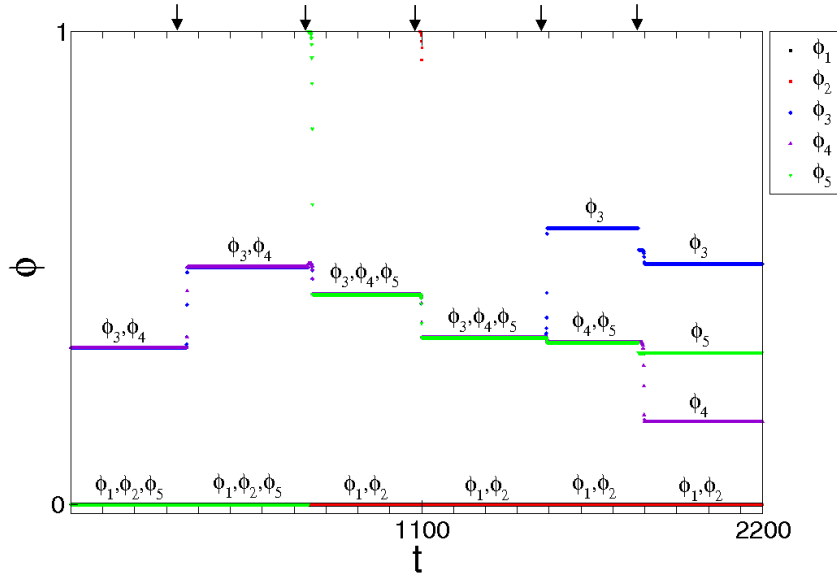


Figure 5.2: **Switchings due to positive perturbations.** Example of symmetry change by perturbation-induced switching. A sequence of three negative and two positive single oscillator perturbations (same parameters as in Figure 5.1). Showing that the positive perturbations splits the unstable cluster until the system reaches some periodic orbit. The symmetry is not preserved.

Although the symmetry is not always preserved after perturbations, this result shows that orbits with cluster symmetry $S_3 \times S_2$ are indeed attractors with at least part of their attraction basin close to another symmetrically related orbit.

5.2 Clustered state symmetry $S_4 \times S_1$

Consider now the $S_4 \times S_1$ symmetry. Assuming $\delta_2 < \delta_3 < \delta_4$ and $\delta_5 > 0$, in an analogous procedure as in the last section we obtain the following Jacobian matrix (see last row of tables D.2 and C.4):

$$\left. \frac{\partial \delta(n+1)}{\partial \delta(n)} \right|_{\delta(n)=0} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ j_{21} & \beta & 0 & 0 \\ j_{31} & j_{32} & \gamma & 0 \\ j_{41} & j_{42} & \theta & 0 \end{pmatrix}. \quad (5.22)$$

Here α , β , γ , and θ are larger than one, j_{41} and j_{42} are positive, and j_{21} , j_{31} , and j_{32} are much smaller than the other elements (see Appendix B). This matrix has one zero eigenvalue $\lambda_0 = 0$, with the eigenvector in the direction of δ_5 . It means that any perturbation in this direction is erased after one cycle. Furthermore, it presents three non-zero eigenvalues given by

$$\lambda_1 = \alpha = \left[-1 + \left[1 + H'_\epsilon(\tau) H'_\epsilon(H_\epsilon(\tau)) \right] H'_\epsilon(H_{2\epsilon}(\tau)) \right] H'_\epsilon(\tau + H_{3\epsilon}(\tau)), \quad (5.23)$$

$$\lambda_2 = \beta = \left[-1 + \left[1 + H'_\epsilon(\tau) \right] H'_\epsilon(H_\epsilon(\tau)) \right] H'_\epsilon(H_{2\epsilon}(\tau)) H'_\epsilon(\tau + H_{3\epsilon}(\tau)), \quad (5.24)$$

$$\lambda_3 = \gamma = \left[-1 + 2H'_\epsilon(\tau) \right] H'_\epsilon(H_\epsilon(\tau)) H'_\epsilon(H_{2\epsilon}(\tau)) H'_\epsilon(\tau + H_{3\epsilon}(\tau)). \quad (5.25)$$

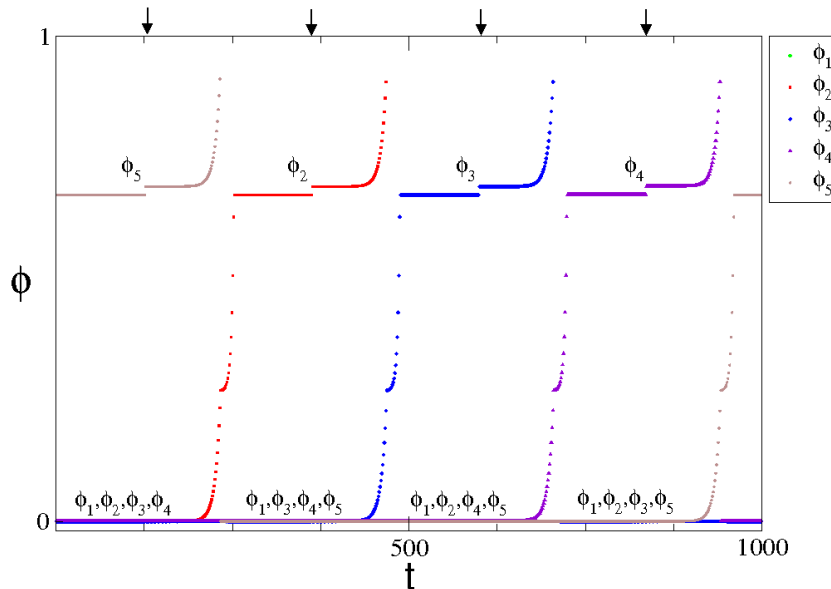


Figure 5.3: **Switchings due to positive perturbations.** Example of perturbation-induced switching starting from a $S_4 \times S_1$ state. A sequence of four positive single oscillator perturbations preserving a $S_4 \times S_1$ clustered symmetry ($\tau = 0.27$, $\epsilon = 0.0015$, $I = 1.1$, $\gamma = 1$). The phases of all oscillators are plotted each time oscillator one is reset. Each color represents the phase of one oscillator. The perturbed oscillator leaves the cluster S_4 and replaces the S_1 oscillator that join the cluster S_4 , preserving the symmetry. The sequence of states corresponding to the plateaus are $(a, a, a, a, b) \rightarrow (a, b, a, a, a) \rightarrow (a, a, b, a, a) \rightarrow (a, a, a, b, a)$.

Using $H'_\epsilon(\phi) > 0$ for all $\epsilon > 0$ and all ϕ (see lemma 1), these three eigenvalues are necessarily larger than one ($\lambda_1, \lambda_2, \lambda_3 > 1$), showing the instability of cluster S_4 .

Again we study the effect of single oscillator perturbations. As can be seen in Figure 5.3, positive perturbations to an element of the unstable cluster S_4 put one element from the unstable group ahead. This difference in phase will increase each cycle, since pulses received at larger phases more strongly shift the oscillator's phase; at the same time the S_1 element is not only reset by the pulses arriving from other oscillators, but also receives additional pulses that makes its phase to approach the unstable cluster. After some cycles, the element S_1 joins the original unstable cluster, forming a new S_4 cluster while the perturbed oscillator forms the new S_1 , returning the system to its original phase difference and symmetry, but with different elements composing the clusters.

When a negative perturbation is applied (see Figure 5.4) the perturbed element is put backwards, and, as before, the elements ahead increase the difference in phase in relation to the perturbed element; at each cycle, after being reset, the original S_1 element receives an additional pulse coming from the perturbed element, increasing its phase. After some cycles this increase makes the S_1 element join the perturbed element, forming a new cluster S_2 . The new configuration becomes $S_3 \times S_2$, where S_3 is unstable. A second perturbation to the S_3 cluster moves the perturbed element to the cluster S_2 . A last perturbation either puts the system towards its initial configuration or splits the cluster into two, depending on the position on the periodic orbit to which it is applied. The symmetry of this state

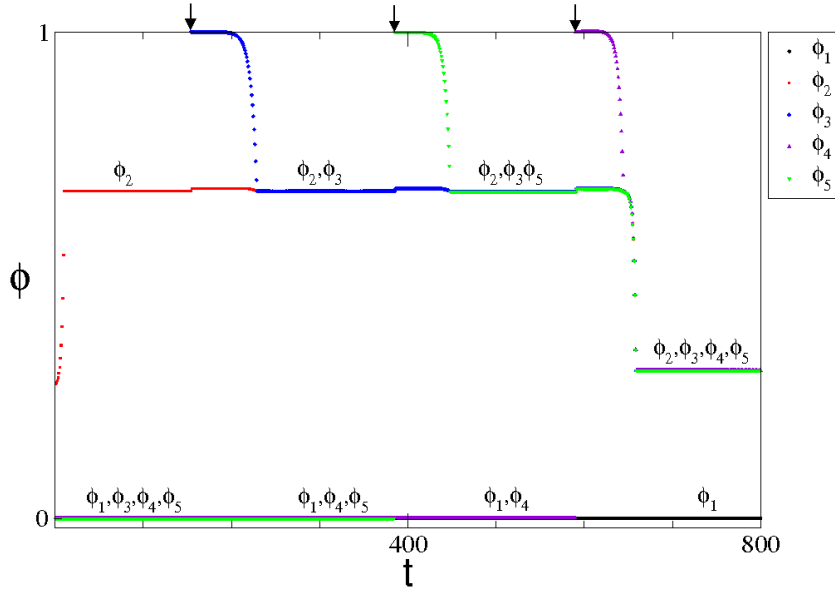


Figure 5.4: **Example of symmetry change by perturbation-induced switching due to negative perturbations starting from a $S_4 \times S_1$ state.** A sequence of three negative single oscillator perturbations to the initially S_4 cluster for the same parameter used on Figure 5.3. The perturbed oscillator joins the S_1 oscillator, forming a S_2 cluster, a second perturbations (to the S_3 cluster) yields an new $S_3 \times S_2$ configuration where S_2 is unstable, and a last perturbation (to the S_2 cluster) puts the system in the initial symmetry. The symmetry is not preserved.

is obviously not preserved for negative perturbations, and computer simulations indicate that more general perturbations bring an even more complicated switching dynamic due the large number of elements in the unstable cluster.

Considering only single positive perturbations yields a transition rule between two states when subject to a single positive perturbation.

Transition rule for cluster symmetry $S_4 \times S_1$:

$$(a + \delta_1, a, a, a, b) \rightarrow (b, a, a, a, a). \quad (5.26)$$

The resulting transition diagram is a fully connected one, and it is possible to jump from one equivalent permutation state to any other one applying only one perturbation.

Notice that, as in the last section, this transition rule implies that this cluster periodic orbit is an attractor with at least part of its basin of attraction close to an orbit with the same cluster symmetry.

5.3 Clustered state symmetry $\mathbf{S}_2 \times \mathbf{S}_2 \times \mathbf{S}_1$

For the symmetry $S_2 \times S_2 \times S_1$, assuming $\delta_3 < \delta_4$ and $\{\delta_2, \delta_5\} > 0$, we have the following Jacobian matrix:

$$\left. \frac{\partial \boldsymbol{\delta}(n+1)}{\partial \boldsymbol{\delta}(n)} \right|_{\boldsymbol{\delta}(n)=0} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 \\ \gamma & 0 & 0 & 0 \end{pmatrix}, \quad (5.27)$$

where $\alpha > \beta > \gamma > 0$ (see Appendix B), in accordance with tables D.3 and C.4. This matrix has three zero eigenvalues $\lambda_0 = \lambda_1 = \lambda_2 = 0$, with eigenvector in the direction of δ_3 , δ_4 and δ_5 . This means that the system is super-stable in those directions, and any perturbation is erased within one cycle. The largest and only non-zero eigenvalue is given by

$$\begin{aligned} \lambda_3 = \alpha = & H'_\epsilon \left(\tau' + H_\epsilon \left(\tau - \tau' + H_{2\epsilon}(\tau') \right) \right) \times \\ & \times \left[-1 + H'_\epsilon \left(\tau - \tau' + H_{2\epsilon}(\tau') \right) \left(1 + H'_{2\epsilon}(\tau') \right) \right], \end{aligned} \quad (5.28)$$

which is larger than one accordingly to Lemma 1 with a eigenvector in the direction of δ_2 . The fact that there is only one eigenvalue larger than one and the eigenvector align to one of the perturbation dimensions not only shows that there is only one unstable cluster, but also that perturbations change only the difference in phase between the two elements on this cluster. As a result, any general perturbation can be mapped to a single oscillator perturbation.

Different from the last two considered symmetries ($S_4 \times S_1$ and $S_4 \times S_1$), we here have one single element, one stable S_2 clusters, and one symmetric unstable S_2 cluster. When perturbed, the initial unstable cluster S_2 splits into two, the additional pulse received now by the initial single S_1 element just after its reset makes it approach the element that was put behind on the unstable S_2 cluster, forming a new stable S_2 cluster. This occurs because it is reset by supra-threshold pulses. Moreover the element ahead begins to be reset by pulses and stops increasing its phase, becoming stable and the original stable S_2 cluster after changing its phase, is not reset by pulses anymore, becoming unstable. The final state has the same symmetry and stability properties as the former state.

The preservation of the symmetry implies a closed transition diagram among all the possible $S_2 \times S_2 \times S_1$ states (see Figure 5.6). We state two simple and equivalent switching rules.

Transition rule for cluster symmetry $\mathbf{S}_2 \times \mathbf{S}_2 \times \mathbf{S}_1$:

Considering first a positive representation we have

$$(a, a + \delta_2, b, b, c) \rightarrow (b, c, a, a, b) \quad (5.29)$$

that can be rewritten for negative perturbations simply as

$$(a - \delta_1, a, b, b, c) \rightarrow (b, c, a, a, b). \quad (5.30)$$

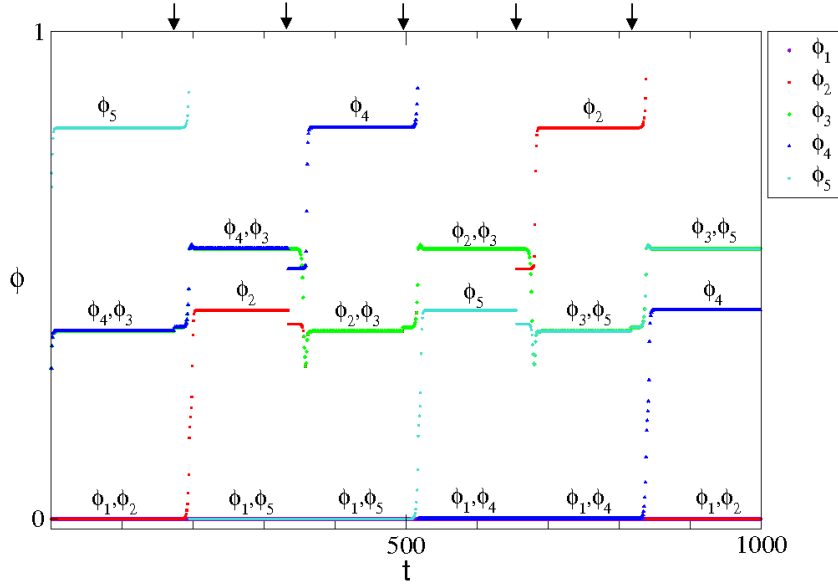


Figure 5.5: **Example of perturbation-induced switching starting from a $S_2 \times S_2 \times S_1$ state.** A sequence of five perturbations driving the system through different states with symmetry $S_2 \times S_2 \times S_1$ ($\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$). The phase of all oscillators are plotted at the moment when oscillator number one is reset, each color representing the phase of one oscillator. The apparent change of the phase differences among the clusters just after the perturbations depends on the cluster to which the reference oscillator belongs. The symmetry of the unstable attractors is preserved. The sequence of states corresponding to the plateaus is $(a, a, b, b, c) \rightarrow (b, c, a, a, b) \rightarrow (a, b, b, c, a) \rightarrow (b, a, a, b, c) \rightarrow (a, c, b, a, b) \rightarrow (b, b, a, c, a)$.

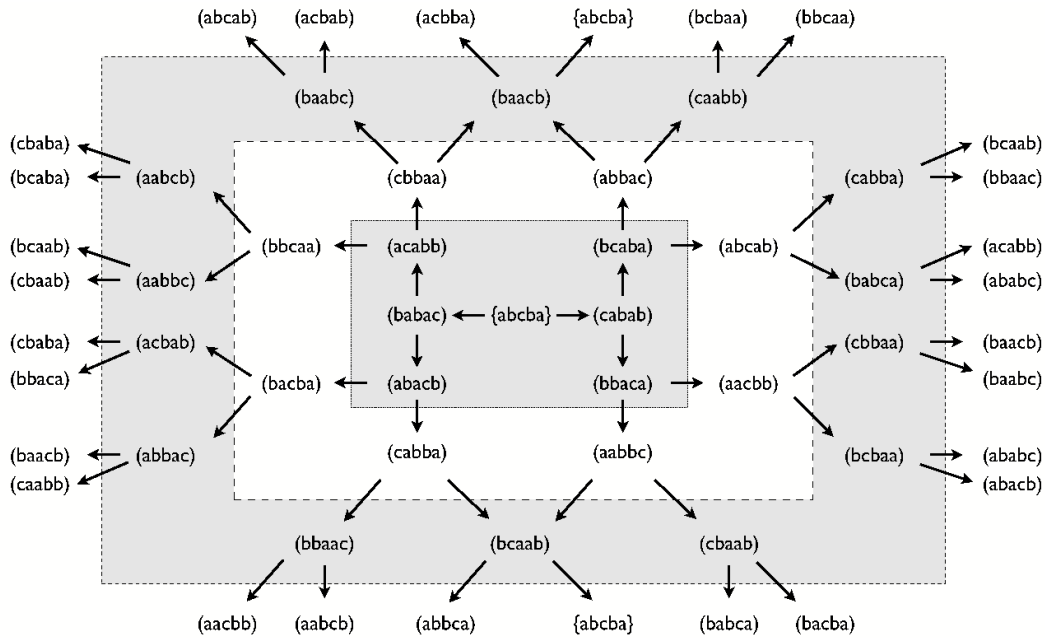


Figure 5.6: **Five steps state transition diagram for the symmetry $S_2 \times S_2 \times S_1$.** This diagram shows all the possible 5 steps paths, sequence of states, beginning on the state $\{a, b, c, b, a\}$. Each arrow corresponds to one of the two possible perturbations, subject to (5.29) and (5.30). It is necessary to have at least 5 perturbations to reach the initial state again.

We conclude that for this symmetry the unstable attractors are linked to form a heteroclinic network, characterized by (5.29) and (5.30), forming a closed set of saddle periodic orbits among which the systems switches in a controlled way upon small external perturbations. We remark that in the absence of noise this dynamic does not exhibit spontaneous transitions between nearby saddle states [5, 22], but instead displays convergence to unstable attractors. The free dynamic of each element evolves continuously up to reset; still, the collective dynamic of the entire network (network state) evolves continuously almost always, only interrupted by discrete jumps due to the infinitely fast phase response of the interaction. As (i) the transitions are fully controlled by external perturbations and thus predictable, and (ii) the symmetry is preserved when the network is subject to sufficiently small, general perturbations, arbitrarily small external noise would trigger a persistent switching dynamic in which the network states are constrained to the closed set of periodic orbits with initial $S_2 \times S_2 \times S_1$ symmetry (see Chapter 6).

5.4 Conclusion

In this chapter we have shown that the three cluster periodic orbits introduced in the last chapter are in fact unstable attractors, such that their basins of attraction are far away, located in the vicinity of a distinct but specific attractor with the same symmetry. In particular, for the cluster symmetry $S_2 \times S_2 \times S_1$, the attractors are heteroclinically connected to form a closed network, where the symmetry is always preserved after a switching, yielding precise transition rules in this close set of attractors for general perturbations. This controllable switching process [10] underlies all computational properties presented in the following chapters. In the next chapter we study how symmetry breaking input signals support stable periodic orbits and show that these orbits are well-characterized by specific sequences of unstable periodic orbits studied in this chapter.

Chapter 6

Noise-induced switching and complex periodic orbits

In the previous chapter we defined the transition properties between saddle periodic orbits arising in a symmetric network with $N = 5$. We first studied the local stability of these periodic orbits and later the long term effect (convergence to another saddle state) of directed perturbations. Here we study the effect of noise and asymmetric driving forces on its dynamics, revealing its basic computational properties. Although we address all three cluster symmetries, special attention is paid to the $S_2 \times S_2 \times S_1$ cluster symmetry, since it presents the most promising computational features as the formation of a heteroclinic network and a transition rule for general perturbations.

We argue here that the nature of small external signals added to an oscillator (perturbations or asymmetric currents) is irrelevant for the transition rules during one oscillator cycle, since any added quantity to the right side of the defining differential equations of dynamical variables (ϕ, V) cannot be distinguished from others after temporal integration. Furthermore, the pulse-like interaction discussed until now results in a non-interactive system for most of the time where the difference of phases (or potentials) in a cluster is computed at discrete times $\{t_{i,k}\}$ (in the phase picture), or equivalently at $\{s_{i,k}\}$ (in the potential picture), where $s_{i,k} = t_{i,k}T_{IF}$. Thus, these extra currents can be mapped into an effective perturbation at those specific times.

We first study the effect of noise, highlighting the relation between the arising transitions and the transition rules defined in the last chapter; second we show how network asymmetries generate and control limit cycles and discuss their implications to neural computation.

6.1 Noise-induced switching

Here we show under which conditions and how noise triggers a persistent switching process, whether the symmetries are maintained in such a process and how different noise intensities affect such switchings. To be specific, we assume a uniformly

distributed noise given by

$$\nu = (\nu_1, \nu_2, \nu_3, \nu_4, \nu_5), \quad (6.1)$$

where the components are independent of each other.

Our analysis proceeds in three main steps. We first check for the existence of a switching process by observing the time evolution of V when subject to noise; second, if such a process is present, the resulting paths generated by the potentials in the state space and the fraction of time spent close to different regions in this space are studied. These paths are depicted in a state space hyperplane as a three-dimensional plot of the potentials $V_{2,3,4}(s_{1,k})$ of oscillators $\{2, 3, 4\}$ in relation to each other at all times $\{s_{1,k}\}$ (reset times of oscillator $i = 1$). The data set is generated by noise-induced switchings during a period long enough to cover all possible transitions. Due to the system symmetry, any group of three elements from $\{1, 2, 3, 4, 5\}$ or any other choice of referential oscillator, $\{s_{i,k}\}$ for oscillator i , yields an equivalent structure. Regarding the time factor, we first define as points of interest the saddle periodic orbits Po_n described in Chapter 3, where n is the saddle index. Then, we calculate the fraction of time spent at different distances from those attractors. Specifically, we measure at each time step the minimum distance to a point L_i contained in the set of all attractors (saddle periodic orbits)

$$d_{min}(s) = \min\left\{\min_{L_i \in Po_1}\{|V(s) - L_i|\}, \dots, \min_{L_i \in Po_n}\{|V(s) - L_i|\}\right\} \quad (6.2)$$

where n is the number of saddles. Since the attractors themselves are periodic orbits, the distance is first calculated as the minimum distance to a point over each orbit, $\min_{L_i \in Po_n}\{|V(s) - L_i|\}$, to then calculate the minimum of all results. Here we approximate the periodic orbits to 283 chosen points over each orbit, equally spaced in time during a noise-free dynamics.

6.1.1 $S_3 \times S_2$ and $S_4 \times S_1$ cluster symmetries

The simplest switching process is given by the $S_3 \times S_2$ symmetry. As shown in Figure 6.1 (a), after noise is introduced the unstable cluster S_3 splits into three and the final state becomes stable. Thus, no switching dynamics is achieved. This result was expected, since single neuron perturbations show that a sequence of positive single oscillator perturbations drive the system to the same stable state.

Interestingly, starting from a cluster state with $S_4 \times S_1$ symmetry, a persistent switching is achieved, even though single oscillator perturbations can lead to a stable attractor depending on the moment during the periodic orbit at which it is applied. Although no mathematical proof is given, long term computations suggest that the multi-oscillator nature of the effective perturbation and the continuous interaction with the noise prevents the convergence to the given stable state (see Figure 5.2).

These results show how unpredictable the dynamics of similar system can be when standard conditions are changed. One could expect that noise-induced switchings would lead to a stable attractor for both symmetries, since single neuron perturbations can lead to such states in both cases. However, on the contrary, a

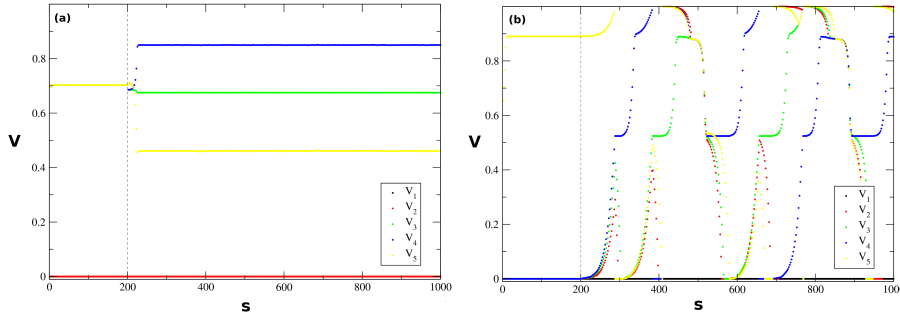


Figure 6.1: Noise-induced switching leading to a stable (a) and unstable attractors (b) attractor. a) Example of noise-induced switching in a initial $S_3 \times S_2$ state ($\tau = 0.31$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$). The potential V of all oscillators are plotted at the times $\{s_{1,k}\}$, oscillator number one reset times. Each color represent one oscillator. Shortly after noise is introduced ($s = 200$), the system switches to a stable attractor with symmetry $S_2 \times S_1 \times S_1 \times S_1$. b) Example of noise-induced switching in an initial $S_4 \times S_1$ state ($\tau = 0.27$, $\epsilon = 0.015$, $I = 1.1$, $\gamma = 1$). The potential V of all oscillators are plotted at the times $\{s_{1,k}\}$, oscillator number one reset times. Each color represents one oscillator. After noise is introduced ($s = 200$), a persistent switching takes place, preserving the initial state symmetry. The dynamics do not converge to the stable periodic orbit predicted by single-oscillator perturbations.

more general perturbation seems to prevent such transitions for the $S_4 \times S_1$ symmetry, at least for a considerable period of time. The mathematical mechanism that explains such an effect are still to be explained.

6.1.2 $S_2 \times S_2 \times S_1$ cluster symmetry

In contrast to the other two cluster symmetries, for the symmetry $S_2 \times S_2 \times S_1$ the system exhibits a closed set of unstable attractors, saddle periodic orbits that are linked by heteroclinic connections forming a network. Thus, due to its intrinsic instability, any sufficiently small noise is expected to trigger a persistent switching process as specified by equations (5.29) and (5.30) (see Figure 6.2) that always approaches the previously defined heteroclinic framework (see Figure 6.4). If the noise is too large, the system ceases any predictable switching features. We study here the first case, demonstrate the effects of different noise intensities on the network dynamics and calculate the upper bound of noise intensity where predictable switching breaks down.

The noise-induced switching process exhibits dynamical features that can be viewed on two different levels. On the network level, due the heteroclinic connections, the noise strongly influences the system when close to one of the saddles, forcing the dynamics to follow one unstable direction, converging to the next saddle. From the unit perspective, the phenomenon of persistent switching is a result of two events: when close to one saddle,

- i. the presence of a constant noise ν adds a different extra current to each oscillator, inducing the elements in the unstable cluster to reset at different times, what is essentially independent of the noise intensity or distribution, since the states are mathematically unstable;

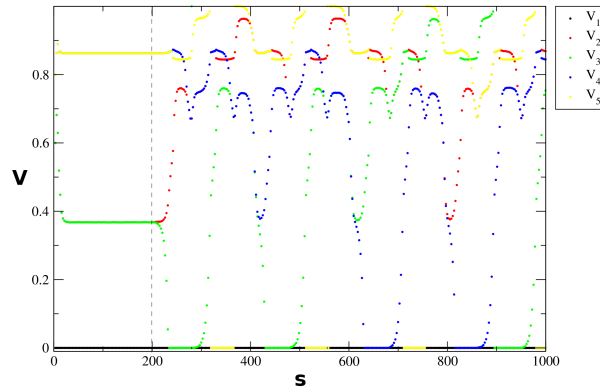


Figure 6.2: **Noise-induced switching process for the $S_2 \times S_2 \times S_1$ cluster symmetry.** Example of noise-induced switching from an initial $S_2 \times S_2 \times S_1$ state ($\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$). The potential V of all oscillators are plotted at the times $\{s_{1,k}\}$, oscillator number one reset times. Each color represents one oscillator. After noise is introduced ($s = 200$), a persistent switching takes place preserving the initial state symmetry.

- ii. any difference in the potentials V between the elements in the stable cluster caused by ν is erased each cycle by simultaneous supra-threshold input signals.

Therefore, for this symmetry, only the components of ν over the unstable cluster strongly influence the system dynamics, and the effective noise influence can be mapped into a single neuron perturbation at each cycle of the oscillators. In other words, for small enough noise intensities, the resulting dynamics resembles a sequence of single oscillator perturbations that obey the transition rules (5.29) and (5.30). Notice that this sequence is unpredictable, since the desynchronization directions (one in two possible directions at a time) depends only on the temporal realization of the random noise (see Figure 6.3).

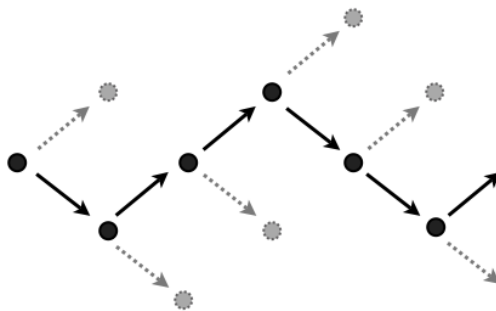


Figure 6.3: **Random switching sequence induced by noise.** Each time the system dynamics approach what have been a saddle periodic orbit, the presence of noise induces a switching to one of two directions, converging in the vicinity of another former saddle. Dashed lines represent possible transition choices not taken (gray); continuous lines represent the transitions taken (black).

As shown above, the dynamics of this system is ruled by deterministic transitions (the reiterated convergence to the vicinity of distinct saddles). But how does

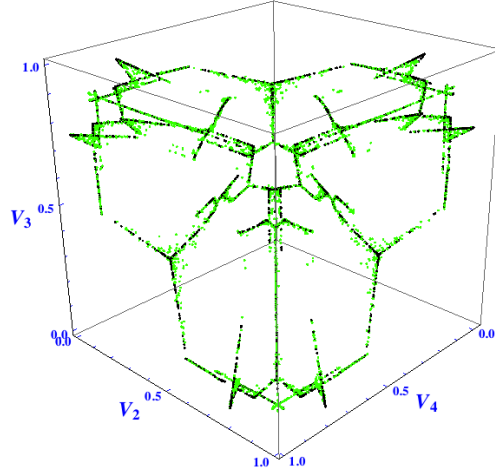


Figure 6.4: **All possible transitions in a state space hyperplane.** Two superimposed paths generated by two different noise intensities in the subspace containing $V_{2,3,4}$, 10^{-5} (black) and 10^{-3} (green), ($\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$). The value of $V_{2,3,4}$ is registered at times $\{t_{1,k}\}$, oscillator one reset times. The data set is generated by noise-induced switchings during a period long enough to cover many times all possible transitions. The depicted structures are qualitatively the same, stronger noise brings a fuzzier structure. Some apparently distant points are actually close by the reset condition ($1 \rightarrow 0$).

the dynamics when subject to different noise intensities differ in state space and in time? Let us first study the structure generated by noise in state space. Figure 6.4 shows all possible transitions in a state space hyperplane for two different noise intensities, two orders of magnitude apart (10^{-5} and 10^{-3}). Both are qualitatively equivalent, the only perceptible difference is that stronger noise makes the plot fuzzier due to fluctuations of the order of magnitude determined by the noise. No major structural changes are caused by noise. This result is expected, since both noise intensities are not strong enough to overcome the predicted transition rules, and thus the switching is restricted to the same regions in state space, among the same saddle states.

Considering the temporal aspect, while noise does not change the set of possible transitions, it does change the typical time needed to perform each transition and does change the distribution of times at different distances from the attractors. Figure 6.5 shows the probability density of finding the state V at a distance d_{min} from the closest saddle periodic orbit (see Equation 6.2); the black curve represents the noise free case, and deviates from a δ function due to the initial condition that implies a short convergence time, and due the approximation of the orbit to 283 equidistant (in time) points over the orbit. Based on this noise-free distribution, we define a criterion for a clear bounding to one attractor as $d_{min} < 0.02$. The other distributions in Figure 6.5 (colored curves) represent different noise intensities; up to a noise intensity of the order of 10^{-3} , a relevant fraction of time is spent close to one of the saddles; for noise intensities of the order of 10^{-2} , the distribution of times changes qualitatively, and no clear bounding to any attractor can be inferred.

The distributions presented in Figure 6.5 are as expected, since a strong noise yields larger perturbations to the trajectory in the state space, such that being

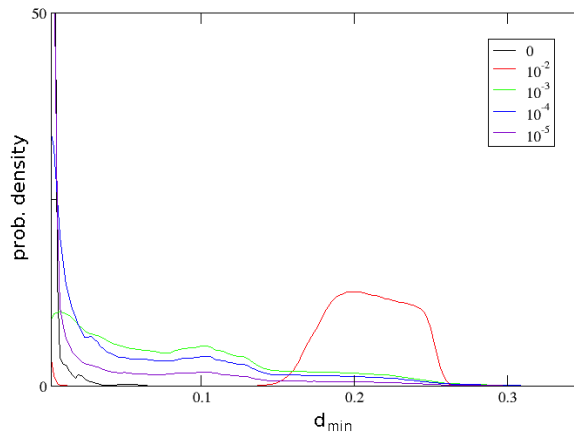


Figure 6.5: **Time distribution under different noise intensities.** The density of probability of finding a dynamical state V at different distances to the closest attractor is depicted ($\tau = 0.49, \epsilon = 0.025, I = 1.04, \gamma = 1$). Each curve corresponds to a different noise intensity. When the intensity is increased, the time spent close to the attractors ($d_{min} < 0.02$) is decreased in relation to the time spent over transients ($d_{min} > 0.02$). Noise intensities of the order of 10^{-2} (red curve) qualitatively change the dynamics and no clear bound to any attractor can be inferred by this plot.

close to one of the saddles forces an "early departure" towards the vicinity of the next saddle. Notice that the fast dynamics intrinsic to the transients is not dramatically altered by the noise, and the shape of the curves are rather shifted than altered, suggesting that the time spent in the transient is not dramatically altered. In the other hand, the relative spent times between the fast dynamics and the slow dynamic close to the saddles are changed (the number of transitions is increased in the same period of time). Figure 6.6 displays the average transition period for different noise intensities; it shows that the transition times, besides a small increase of the mean value, are remarkably stable for a large range of noise intensities ($2 \times 10^{-5} - 1 \times 10^{-3}$).

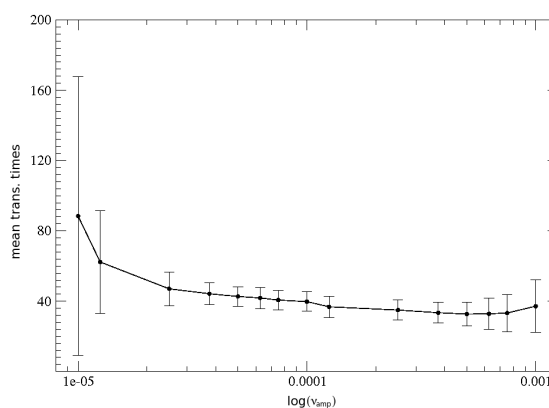


Figure 6.6: **Mean transition time as a function of noise intensity.** The dots represent the mean transition times calculated during 20000 time units (in the order of few hundred transitions); the bars represent the standard deviation. ($\tau = 0.49, \epsilon = 0.025, I = 1.04, \gamma = 1$).

Thus, for a noise intensity of the order of 10^{-3} or below, a persistent switching process preserving the initial cluster symmetry $S_2 \times S_2 \times S_1$ is achieved and the mean switching times decrease very slowly, reflecting the reduced time spent close to the saddles when subject to increasing noise intensities (maximum noise amplitudes).

6.2 Complex periodic orbits in asymmetric networks

In the last section we have shown that, for initial clustered states with symmetry $S_2 \times S_2 \times S_1$, a persistent switching process is indeed triggered by noise and that, although unpredictable, the resulting path is well-characterized by a sequence of unstable periodic orbits subject to the transition rules (5.29) and (5.30). Here we show how symmetry-breaking input signals yield specific limit cycles in a controllable way, the exact relation between the resulting cycles and the inputs, and discuss its potential for neuronal-like computation.

As in the last section, we study the regime where asymmetric currents (noise or external currents) do not overcome the switching properties defined in Chapter 5. Here, we assume as a symmetry-breaking input a constant external current (real value) added independently to each oscillator, such that each component differs in value, given by

$$\Delta = (\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5), \quad (6.3)$$

with $\Delta_i \neq \Delta_j$, for all i and j .

This additional current yields an effective driving current variable when added to the right side of equation (3.6), and the oscillator dynamics is now given by

$$\frac{dV_i}{ds} = (I + \Delta_i) - \gamma V_i + S_i(s). \quad (6.4)$$

Notice that contrary to the noise-induced switchings, this asymmetric input increases the charging rate for specific oscillators in a reiterated way, what again sequentially desynchronizes any unstable cluster each time it is formed, this time forcing the trajectory in state space towards a stable limit cycle (complex periodic orbit) close to a specific sub-set of the heteroclinic network (see Figure 6.7). It happens ultimately because the desynchronization direction in the unstable manifold of a given saddle is already determined by the symmetry-breaking currents.

Until now we have shown that asymmetries generate complex periodic orbits but, do different asymmetries yield the same cycle? As shown in the last section, one way to characterize these cycles is through the former saddle states the cycle approaches (see Figure 6.8). Consider an initial state (c, b, a, a, b) and an external current Δ such that

$$\Delta_1 > \Delta_2 > \dots > \Delta_5. \quad (6.5)$$

The resulting cycle is characterized by repetitively applying perturbations (5.29) with $\delta_i > \delta_{i+1}$ for all i , resulting in

$$\begin{aligned} (\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{a}, \mathbf{b}) &\rightarrow (b, a, c, b, a) \rightarrow (a, c, b, a, b) \rightarrow (c, b, a, b, a) \rightarrow \\ &(b, a, c, a, b) \rightarrow (a, c, b, b, a) \rightarrow (\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{a}, \mathbf{b}) \rightarrow \dots \end{aligned} \quad (6.6)$$

this cycle thus consists of a sequential comparison of the potential values in the unstable cluster a generating this specific cyclical sequence.

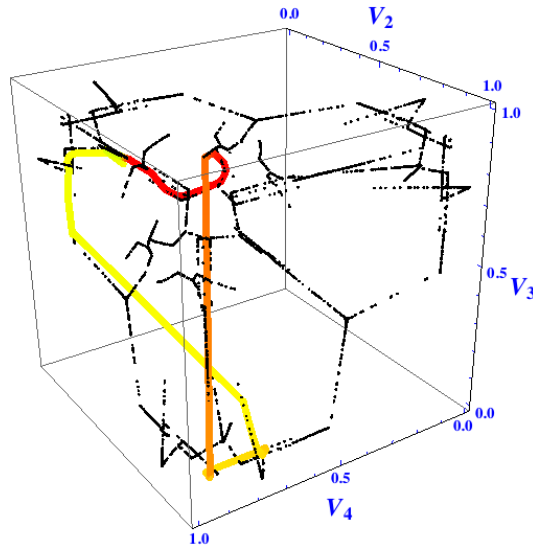


Figure 6.7: **Superposition of induced complex periodic orbit and all possible transitions in state space.** In black a state space structure generated by a persistent noise-induced switching is depicted (see Figure 6.4); superimposed to the noise-induced structure, the colored line represents a four dimensional trajectory in the state space describing a cycle that approaches what have been six saddle periodic orbits. At times $\{s_{1,k}\}$, the potential of oscillators two to four are plotted as a three dimensional line and the potential of oscillator number five is given by a color gradient from zero (yellow) to one (red). The large overlap between the two confirms that the cycles are indeed well characterized by an ordered sub-set of the original state set.

The sorting process (6.6) exhibits three distinct features: Given $\Delta_{i+1} < \Delta_i$,

- i. V_5 can only switch from $a \rightarrow b \rightarrow a$ recurrently. The transition $a \rightarrow c$ is forbidden because $\Delta_5 < \Delta_{1,2,3,4}$. (If the initial condition is such that $V_5 \approx c$, the sequence is $c \rightarrow b \rightarrow a$, the system leaves such states forever and $V_5 \approx c$ is transient);
- ii. V_4 can only switch from $b \rightarrow a \rightarrow b$ recurrently, alternating a and b with V_5 . The transition $a \rightarrow c$ is forbidden because $\Delta_4 < \Delta_{1,2,3}$. (The transition $a \rightarrow c$ is only possible if $V_{4,5} = (a, a)$ that switches to $V_{4,5} = (c, b)$ that switches to $V_{4,5} = (b, a)$, and is thus transient);
- iii. if after a transition the labels $\{a, b, c\}$ are all present in $V_{1,2,3}$, the labels $\{a, b, c\}$ were all present in $V_{1,2,3}$ before the transition; thus $a \rightarrow c$, $b \rightarrow a$ and $c \rightarrow b$. The transition $a \rightarrow b$ is forbidden because $\Delta_{1,2,3} > \Delta_{4,5}$.

Due the initial symmetry of the system (for $\Delta = 0$), any other ordering of the Δ_i implies a sequence of transitions equivalent to (6.6) among saddles with the units' values permuted.

According to features (i) and (ii) during a cycle the values of units receiving the two weaker currents cannot assume the state c (after a possible transient).

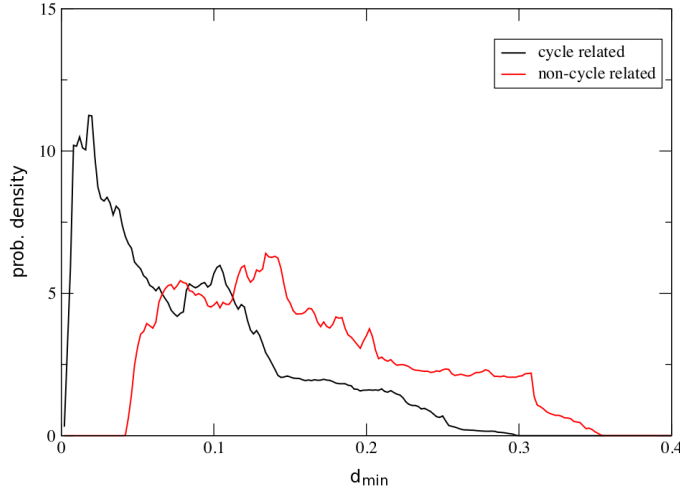


Figure 6.8: **Residence time distribution at different distances to an induced periodic cycle.** The probability density of finding a dynamical state V at different distances from an attractor belonging to an induced cycle (black) and from attractors that do not belong to the induced cycle (red) ($\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$). The distances are calculated as the minimum distance to a point contained in each of these two groups. The graph shows that only states belonging to the cycle are approached (by criteria defined in the previous section $d_{min} < 0.02$).

This feature thus constitutes a dynamical marker to separate those three elements receiving the three strongest inputs from those two receiving the two weakest. Furthermore, permuting the elements subject to the two weakest input currents generates a new valid cycle. Feature (iii) implies that neither the $V_{1,2,3}$ nor the $V_{4,5}$ are compared among themselves. Therefore, the switching dynamics reveals no discrete information about the relative magnitude of asymmetries within these two groups and it is only possible to differentiate the three stronger from the two weaker inputs observing only the cyclic sequence.

6.2.1 Computational aspects: general remarks

This subsection is devoted to defining which aspects will be studied as basic elements of computation and establishing a common terminology for the next two chapters. In what concerns computational aspects, although a precise evaluation of the symmetry-breaking input vector may constitute a feasible mathematical problem, for robustness' sake we exploit only the switching feature and the information regarding the sequence of approached saddles. By making this choice, we direct our study to features that rather depend on a sequence of key states than a long and precise sequence of spikes.

As shown in the first part of this section, by dynamically generating a periodic orbit resembling a sequence of saddle states, the system classifies the five continuous asymmetric inputs into two different groups, the three stronger inputs and the two weaker inputs. This process constitutes thus a mapping from continuous inputs to cycles providing a dynamical classification of real-valued inputs into ten discrete classes, since for five inputs the number of possible combinations (three in five) is

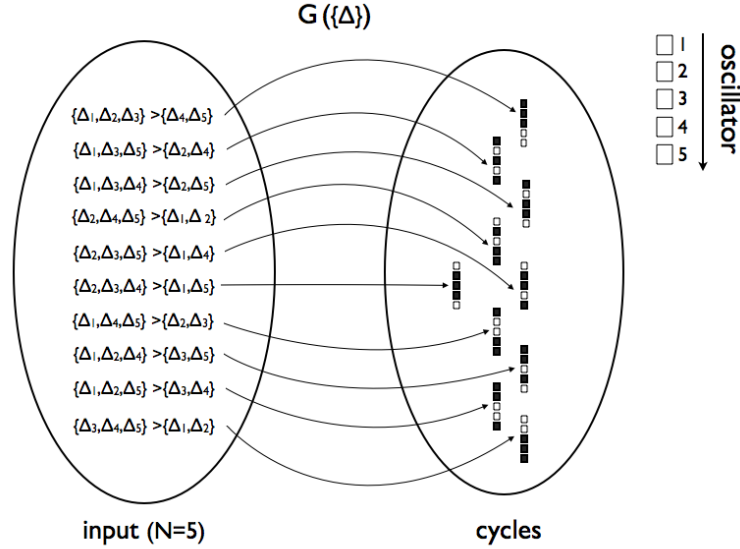


Figure 6.9: **Relation between symmetry-breaking inputs and complex periodic orbits** The set in the left represents the possible asymmetric inputs, the comparison among the different elements in the asymmetric vector means that all elements in the first group are larger than all elements in the second group; The set in the right presents the possible resulting cycles in an abstract representation, black for stronger input classification and white for weaker input classification, the site index runs vertically (five squares). The function G connects any asymmetric vector fulfilling the condition in the left side to one of the possible ten percepts (for reasonable asymmetry intensities).

given by,

$$\binom{5}{3} = 10. \quad (6.7)$$

In other words, the continuous-valued symmetry-breaking input is internally perceived as a binary vector that may be represented as a vector $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5)$ where $r_i \in \{0, 1\}$ and $\sum_i r_i = 3$. Notice that, although convenient, in r the values $\{0, 1\}$ are not relevant, they just represent, respectively, the elements subject to the two weaker inputs and to the three stronger inputs during one oscillator cycle. A more general (abstract) representation of those vectors is given by a color code, where black represents higher oscillatory frequencies (stronger inputs) and white lower oscillatory frequencies (weaker inputs), see Figure 6.9. During this work both representations are used, the color code is specially used to emphasize the difference between input currents, where real values have a precise meaning, from cycles, where the information per site concerns only differences in the oscillators' frequencies and not absolute values. Furthermore, for each percept \mathbf{r} there are two possible cycles (specific to that \mathbf{r}), such that the initial condition determines which one is selected. These classes, thus, constitute a basic form of computation which relates inputs to an internal representations (a set of two cycles, see Figure 6.10), independent of the initial state.

There are two main features to be exploited in the following chapters. First, the characterization of cycles by the approached saddles, yielding robustness regarding

(r color code)		cycles	
\uparrow	r		
	(1,1,1,0,0)	(c,a,b,b,a) → (b,c,a,a,b) → (a,b,c,b,a) → (c,a,b,a,b) → (b,c,a,b,a) → (a,b,c,a,b) → (c,a,b,b,a)	(b,a,c,a,b) → (a,c,b,b,a) → (c,b,a,a,b) → (b,a,c,b,a) → (a,c,b,a,b) → (c,b,a,b,a) → (b,a,c,a,b)
	(1,1,0,1,0)	(c,a,a,b,b) → (b,c,b,a,a) → (a,b,a,c,b) → (c,a,b,b,a) → (b,c,a,a,b) → (a,b,b,c,a) → (c,a,a,b,b)	(b,a,b,c,a) → (a,c,a,b,b) → (c,b,b,a,a) → (b,a,a,c,b) → (a,c,b,b,a) → (c,b,a,a,b) → (b,a,b,c,a)
	(1,1,0,0,1)	(c,a,a,b,b) → (b,c,b,a,a) → (a,b,a,b,c) → (c,a,b,a,b) → (b,c,a,b,a) → (a,b,b,a,c) → (c,a,a,b,b)	(b,a,b,a,c) → (a,c,a,b,b) → (c,b,b,a,a) → (b,a,a,b,c) → (a,c,b,a,b) → (c,b,a,b,a) → (b,a,b,a,c)
	(1,0,1,1,0)	(c,a,a,b,b) → (b,b,c,a,a) → (a,a,b,c,b) → (c,b,a,b,a) → (b,a,c,a,b) → (a,b,b,c,a) → (c,a,a,b,b)	(b,b,a,c,a) → (a,a,c,b,b) → (c,b,b,a,a) → (b,a,a,c,b) → (a,b,c,b,a) → (c,a,b,a,b) → (b,b,a,c,a)
	(1,0,1,0,1)	(c,a,a,b,b) → (b,b,c,a,a) → (a,a,b,b,c) → (c,b,a,a,b) → (b,a,c,b,a) → (a,b,b,a,c) → (c,a,a,b,b)	(b,b,a,a,c) → (a,a,c,b,b) → (c,b,b,a,a) → (b,a,a,b,c) → (a,b,c,a,b) → (c,a,b,b,a) → (b,b,a,a,c)
	(0,1,1,1,0)	(b,c,b,a,a) → (a,b,a,c,b) → (b,a,c,b,a) → (a,c,b,a,b) → (b,b,a,c,a) → (a,a,c,b,b) → (b,c,b,a,a)	(b,a,b,c,a) → (a,c,a,b,b) → (b,b,c,a,a) → (a,a,b,c,b) → (b,c,a,b,a) → (a,b,c,a,b) → (b,a,b,c,a)
	(0,1,1,0,1)	(b,c,b,a,a) → (a,b,a,b,c) → (b,a,c,a,b) → (a,c,b,b,a) → (b,b,a,a,c) → (a,a,c,b,b) → (b,c,b,a,a)	(b,a,b,a,c) → (a,c,a,b,b) → (b,b,c,a,a) → (a,a,b,b,c) → (b,c,a,a,b) → (a,b,c,b,a) → (b,a,b,a,c)
	(1,0,0,1,1)	(b,a,b,a,c) → (a,b,a,c,b) → (c,a,b,b,a) → (b,b,a,a,c) → (a,a,b,c,b) → (c,b,a,b,a) → (b,a,b,a,c)	(a,b,a,b,c) → (c,a,b,a,b) → (b,b,a,c,a) → (a,a,b,b,c) → (c,b,a,a,b) → (b,a,b,c,a) → (a,b,a,b,c)
	(0,1,0,1,1)	(a,c,b,a,b) → (b,b,a,c,a) → (a,a,b,b,c) → (b,c,a,a,b) → (a,b,b,c,a) → (b,a,a,b,c) → (a,c,b,a,b)	(b,b,a,a,c) → (a,a,b,c,b) → (b,c,a,b,a) → (a,b,b,a,c) → (b,a,c,b,b) → (a,c,b,b,a) → (b,b,a,a,c)
	(0,0,1,1,1)	(a,b,b,c,a) → (b,a,a,b,c) → (a,b,c,a,b) → (b,a,b,c,a) → (a,b,a,b,c) → (b,a,c,a,b) → (a,b,b,c,a)	(b,a,c,b,a) → (a,b,b,a,c) → (b,a,a,c,b) → (a,b,c,b,a) → (b,a,b,a,c) → (a,b,a,c,b) → (b,a,c,b,a)

Figure 6.10: **Relation between percept classes and complex periodic orbits.** At left the percept classes r are labeled as a vector and a color code: black/1 represents higher oscillatory frequencies; white/0 represents lower oscillatory frequencies. At right, the two possible cycles exclusively related to the class at left.

noise and differences in the magnitude of the asymmetric input current, that may generate extremely different spike patterns despite converging to the same saddles. Second, the superposition of different sources of asymmetry, since only the oscillators frequencies are relevant and thus only the total/effective asymmetry plays a role. This approach is in fact a trade of information about the input for reliability regarding the acquired information, since it yields a computation principle that does not rely on a precise realization of a long sequence of spikes, but rather a spike activity that resembles a sequence of quasi-synchronized states (clustered states).

6.3 Conclusion

We have shown that inducing and controlling complex stable periodic orbits can reliably generate an internal spatiotemporal representation of the inputs, classifying analogical inputs into discrete classes. The process is robust regarding noise due to the nature of its elements and proved to persist for a broad range of input magnitudes and small differences in the input vector. This is due to the nature of the comparison (direct comparison between values detected by saddle periodic orbit dynamics). Both are interesting features for processing information. We

should remark that any asymmetry that increases or decreases the frequency for a certain element should yield the same behavior as adding external asymmetric currents. In the next two chapters we will present a memory model based on the principles studied here (Chapter 8), and introduce a new concept of neuron-like computation based on induced complex limit cycles (Chapter 7).

Chapter 7

Computing by switching dynamics

In Chapter 6 we have shown how providing symmetry-breaking input signals to the otherwise symmetric system defined in Chapter 3 results in the generation of limit cycles in the state space. These limit cycles consist of stable complex periodic orbits that always predictably approach what have been sets of six distinct saddle states in the symmetric system. Analyzing the relation between input signals and the periodic orbits revealed that the inputs are reliably classified into one of ten binary classes with activity $3/5$. Each input is uniquely classified by one of these classes, and for each class the system converges to one out of two exclusive periodic orbits. Which of the two is determined by the initial condition. We now show how this classification process forms the basis for a new type of universal computation in pulse-coupled systems.

The basic idea is to combine different sources of asymmetric currents (linear superposition of currents) to form an effective asymmetry and after that, to map the resulting cycles into different output states. To demonstrate this idea, we present two practical examples: first, a bitwise computer that performs all fundamental forms of binary logic computations, in particular the XOR operation which provides a NOT operator and guarantees that the system may universally compute any desired function; and as a second example of application, we show how to make use of the switching process as a guide system for an autonomous agent, navigating in relation to a field source when solving a gradient problem.

7.1 Bitwise computation

Bitwise operations ultimately underly all processes in digital computations. Their logic is based on the comparison of two or more bits, that may assume one out of two states $\{0, 1\}$, implemented in a clockwise cycle (well defined frequency). Thus, by constituting the standard computational logic, it becomes a reasonable benchmark for computation. We know from Section 6.2 that the network presented in Section 3.3,

$$\frac{dV_i}{ds} = I - \gamma V_i + S_i(s) + \Delta_{\text{eff},i}, \quad (7.1)$$

is able to classify continuous inputs, here given by an effective asymmetric vector Δ_{eff} , into one out of ten possible binary classes \mathbf{r} with activity 3/5. By this activity we mean that the three stronger inputs are distinguishing from the two weaker. Here we study how to design meaningful relations between symmetry-breaking input signals, that are given by a vector \mathbf{R} , and a pre-defined output, the desired result of the computation. Notice that, although all of the information is already contained in the cycles as spatiotemporal patterns, a second classification of percepts \mathbf{r} by an output layer into a spatial pattern yields a clear visualization of the output.

Here we choose two-bits bitwise operations as our illustrative example. It not only reduces the input/output sets to a binary set, $\mathbf{R}_i \in \{0, 1\}$ (as the percepts classes \mathbf{r}), but also yields an intuitive understanding of the process. Once the basis is explained through this example, we will discuss the general aspects of such a process.

Coming to the actual computer, it is composed of three layers (see Figure 7.1):

1. **the input**, a five-bit binary vector \mathbf{R} , determines both a logical operation (first three bits \mathbf{R}_{123} ; we choose OR, AND, and XOR as examples) and also define the two input bits for computation (\mathbf{R}_{45});
2. **the process layer**, that consists of a network of five neuronal units (3.3); inputs combine to form an effective asymmetric vector $\Delta_{\text{eff}} = \Delta + \omega \circ \mathbf{R}$, where Δ is a fixed asymmetric input current, ω is a fixed weight vector between the inputs \mathbf{R} and the single neuron each R_i targets in the process layer, and \circ is an element-wise multiplication; this asymmetry induces a specific cyclical pulse pattern classified by a vector \mathbf{r} ;
3. **the output**, where the cycles are classified by temporally fixed neural pattern recognition (see Table 7.1) into two different groups (“0” and “1”) according to the desired output activity.

Now that we stated the basic lines of our approach, we should define the constants of the problem (base current and connection weights), which by itself presents a non-trivial problem. How should these parameters be set in order to maximize the number of implemented operations? The answer lies in mapping as many inputs as possible to the same percept \mathbf{r} .

In the actual configuration, from the possible eight combinations in \mathbf{R}_{123} we defined the three basic operations as orthogonal vectors with the same module: $(1, 0, 0)$ to represent the OR; $(0, 0, 1)$ for AND; and $(0, 1, 0)$ for XOR. Now to setup the constants of the system we consider features that simplify the mapping. First, we know that all of these operators should yield the same value when operating over the pair $(0, 0)$. Therefore in order to minimize the number of mapped percepts they should be represented by the same percept \mathbf{r} . One possible way to do so is to choose a base current such that

$$\Delta_1 > \Delta_2 > \Delta_3 > \Delta_4 > \Delta_5, \quad (7.2)$$

and thus by

$$\Delta_{\text{eff}} = \Delta + \omega \circ \mathbf{R} \quad (7.3)$$

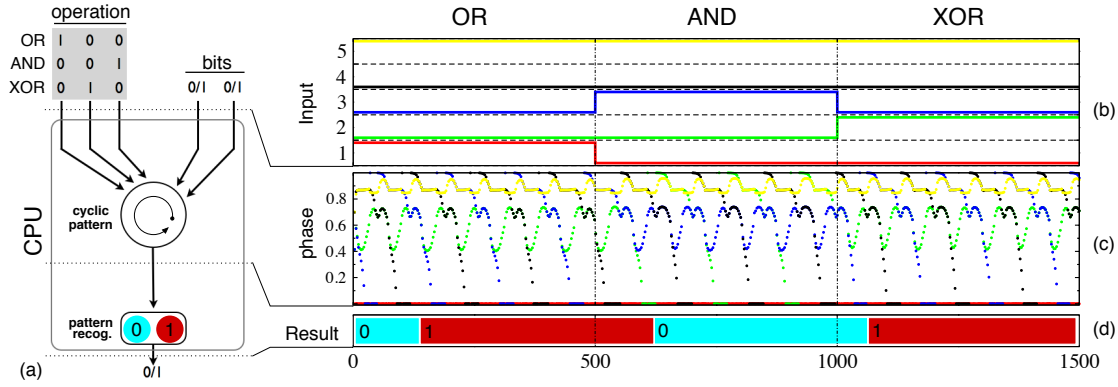


Figure 7.1: **Bitwise computation realized by switching.** Computer diagram and a simultaneous illustration of all three layers dynamics ($\omega = (2.5, 2.5, 2.5, 5, 5) \times 10^{-4}$, $\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$, $\Delta = (4, 3, 2, 1, 0) \times 10^{-4}$). a) Diagram of bitwise computer. Asymmetric external currents ($R_i \in \{0, 1\}$) generate specific limit cycles as internal representations. Conform previously defined categories the spike sequences are associated to a zero or one pattern by the output layer. b) Three consecutive sets of external inputs as constant currents ($t = 0$, $t = 500$, $t = 1000$); c) Poincaré section showing the resulting limit cycles, all potentials plotted in the moment oscillator number one is reset; d) visual representation of the current output.

any of the three operations over $\mathbf{R}_{43} = (0, 0)$ results in the cycle $(1, 1, 1, 0, 0)$, thus

$$\mathbf{R} = (1, 0, 0, 0, 0) \vee \mathbf{R} = (0, 1, 0, 0, 0) \vee \mathbf{R} = (0, 1, 0, 0, 0) \Leftrightarrow \mathbf{r} = (1, 1, 1, 0, 0). \quad (7.4)$$

As a second step, the functions OR and XOR should yield the same cycles for all operands but $(1, 1)$. It is achieved directly by imposing that the connection weights ω between the inputs and the network are such that

$$\{\Delta_{\text{eff},i} > \Delta_{\text{eff},j} \quad \forall \quad i, j \quad \text{iff} \quad R_i = 1 \wedge R_j = 0\}, \quad (7.5)$$

with $i = \{1, 2, 3, 4, 5\}$ and $j = \{1, 2, 3, 4, 5\}$. This last restriction also yields a reduced number or percepts, such that,

$$\mathbf{R} = (1, 0, 0, 0, 1) \vee \mathbf{R} = (0, 1, 0, 0, 1) \Leftrightarrow \mathbf{r} = (1, 1, 0, 0, 1), \quad (7.6)$$

$$\mathbf{R} = (1, 0, 0, 1, 0) \vee \mathbf{R} = (0, 1, 0, 1, 0) \Leftrightarrow \mathbf{r} = (1, 1, 0, 1, 0), \quad (7.7)$$

$$\mathbf{R} = (1, 0, 0, 1, 1) \Leftrightarrow \mathbf{r} = (1, 0, 0, 1, 1), \quad (7.8)$$

$$\mathbf{R} = (0, 1, 0, 1, 1) \Leftrightarrow \mathbf{r} = (0, 1, 0, 1, 1). \quad (7.9)$$

The third and last step show a straight forward mapping of the AND operation, since the last three operations have no bounds to the other two operators to be fulfilled. Thus,

$$\mathbf{R} = (0, 0, 1, 0, 1) \Leftrightarrow \mathbf{r} = (1, 0, 1, 0, 1), \quad (7.10)$$

$$\mathbf{R} = (0, 0, 1, 1, 0) \Leftrightarrow \mathbf{r} = (1, 0, 1, 1, 0), \quad (7.11)$$

$$\mathbf{R} = (0, 0, 1, 1, 1) \Leftrightarrow \mathbf{r} = (0, 0, 1, 1, 1). \quad (7.12)$$

The combined results of the three steps are summarized on Table 7.1. This choice of parameters seems to be a natural choice, since it yields an almost identical set

Table 7.1: **Logic function definitions.** The input vectors are categorized into activity and non-activity according to each function. The given input is weighted and combined to the base asymmetric current to define an effective asymmetry related to one of the 10 possible internal representations \mathbf{r} . ($\boldsymbol{\omega} = (2.5, 2.5, 2.5, 5, 5) \times 10^{-4}$, $\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$, $\boldsymbol{\Delta} = (4, 3, 2, 1, 0) \times 10^{-4}$)

binary logic function	input (\mathbf{R})	cycle (\mathbf{r})	output
AND	(0,0,1,0,0)	(1,1,1,0,0)	0
	(0,0,1,0,1)	(1,0,1,0,1)	0
	(0,0,1,1,0)	(1,0,1,1,0)	0
	(0,0,1,1,1)	(0,0,1,1,1)	1
OR	(1,0,0,0,0)	(1,1,1,0,0)	0
	(1,0,0,0,1)	(1,1,0,0,1)	1
	(1,0,0,1,0)	(1,1,0,1,0)	1
	(1,0,0,1,1)	(1,0,0,1,1)	1
XOR	(0,1,0,0,0)	(1,1,1,0,0)	0
	(0,1,0,0,1)	(1,1,0,0,1)	1
	(0,1,0,1,0)	(1,1,0,1,0)	1
	(0,1,0,1,1)	(0,1,0,1,1)	0

of cycles \mathbf{r} for the functions OR and XOR, except for the pair $\mathbf{R}_{45} = (1, 1)$ (as it should be), and maps the results of all functions over $\mathbf{R}_{45} = (0, 0)$ to the same \mathbf{r} . (see table 7.1) Thus, it minimizes the necessary number of representations \mathbf{r} while maintaining the input \mathbf{R} in a maximum activity of $\sum_i R_i = 3$ as in \mathbf{r} . Figure 7.2 combines all information present in Figure 7.1(a) and Table 7.1 in a single picture, and displays all possible computations for the same set of parameters as in Table 7.1, as a set transformation, showing the relation of each allowed input \mathbf{R} (pre-defined) to the percepts \mathbf{r} and between \mathbf{r} and the output as well.

In what concerns the output layer there is only a short remark to be made. As can be seen in Table 7.1, all information about the operation's results are already contained in \mathbf{r} , thus the output layer merely changes how the results are presented. Different methods can be implemented to classify spatiotemporal patterns [30, 31], yielding equivalent results. In the data presented here spatiotemporal patterns are decoded by simultaneous events detection.

As an actual example of computation, Figures 7.1(b,c,d) display the three computing layers for the three operators (OR, AND and XOR) given a binary operand (0, 1) (see Figure 7.3 for summarized process). Operationally speaking, the result emerges only some time after changing the operation (at times $t = 500$ and $t = 1000$): besides a short dynamic transient, this is due to the time required for detecting the cyclic patterns. For the specified chosen parameters, it takes about 300 time steps to guarantee the right answer (see Figure 7.4).

We should now stress three aspects of this new computation paradigm: first, by concatenating these three functions an arbitrary logic function can be computed so that the system serves as an universal computational device; second, that the choice of binary operations is merely illustrative, all properties discussed here rely

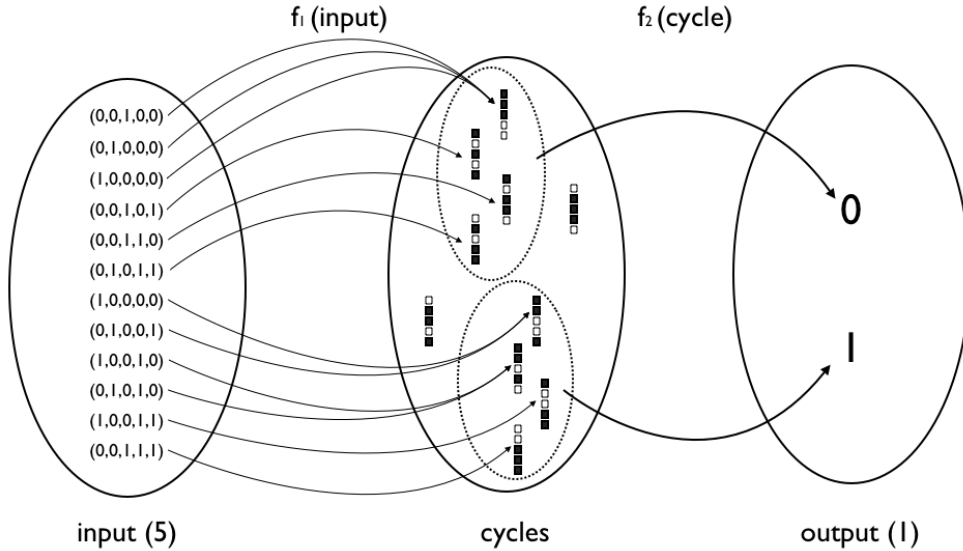


Figure 7.2: **Computation representation as a set transformation in two steps.** The bitwise computer depicted in Figure 7.1, in accordance with table 7.1. The function f_1 is defined by the connection weights between the input and the process layer plus the base asymmetry. The function f_2 is defined by the weights between the process layer and the output layer.

only on the mapping of input signals to a set of orbits. In a network with $N = 5$, in principal, we could compare up to 4 inputs and still have one operation discriminator, although the number of feasible functions is decreased. Furthermore, in principle, for larger networks the number of inputs increases linearly with the network, whereas the number of distinct stable periodic orbits increases exponentially, and thus may yield a much larger number of possible functions involving different number of inputs.

This shows that switching dynamics in neural networks provides a natural way of universal computation in dynamical systems. In practical applications, such dynamics can be easily implemented as networks of different oscillator types [3, 25, 27] because only the feature of appropriate input classification at the saddles is required; transition rules sufficiently resembling (5.29) are also present in larger networks of coupled oscillator cf. [25, 26] and may equivalently be exploited. As an example, for $N = 100$ a persistent switching process among unstable states with cluster symmetry $S_{21} \times S_{21} \times S_{21} \times S_{21} \times S_{16}$ robustly takes place. In previous works on coding and computational properties of oscillatory circuits, e.g. [3, 8, 27], individual saddles serve as conceptual coding elements. The fundamental neural computer presented above now represents inputs as cyclic sequences among a number of saddle states (cf. [8]) and exploits the advantages coming with this re-interpretation, e.g. the independence of the dynamical coding of the initial condition.

The main dynamical property we exploit is that symmetry-breaking input signals guide the system to an input-specific cycle in state space. As many very similar such cycles already exist in small networks, our novel concept ideally exploits features characteristic of generic heteroclinic networks in (almost) symmetric

	OR(0,1)	AND(0,1)	XOR(0,1)
Input	1 0 0 0 1	0 0 1 0 1	0 1 0 0 1
connect. (ϵ)	\times 2.5 2.5 2.5 5 5	\times 2.5 2.5 2.5 5 5	\times 2.5 2.5 2.5 5 5
result input	2.5 0 0 0 5	0 0 2.5 0 5	0 2.5 0 0 5
base asym.	$+$ 4 3 2 1 0	$+$ 4 3 2 1 0	$+$ 4 3 2 1 0
total asym.	6.5 3 2 1 5	4 3 4.5 1 5	4 5.5 2 1 5
result cycle 'r'	1 1 0 0 1	1 0 1 0 1	1 1 0 0 1

Figure 7.3: **Three bitwise operations: superposition of currents.** The three operations depicted on Figure 7.1. It shows step by step how different sources of currents adds to form a effective asymmetry in accordance to Table 7.1 and Figure 6.10. Each input is multiplied by a connection weight and added to a base asymmetry, resulting in an effective asymmetry. Red values represent constants of the network, black values are variables resulting from the operations, and blue values represent the associated cycles. Each operation is done independently in each column.

systems. The system is also intrinsically robust to noise due to the continuous-time integration of the input signals and with increasing noise mainly the average period of the cycles decreases but not its overall shape in state space. Additionally, other naturally emerging features are of practical interest. One of them is redundancy that occurs because the internal representation to a continuous input is cyclic; A second is that no learning rule is required after the initial asymmetry is imposed and all of the cyclic dynamics, representing the results of the computation, then emerges spontaneously.

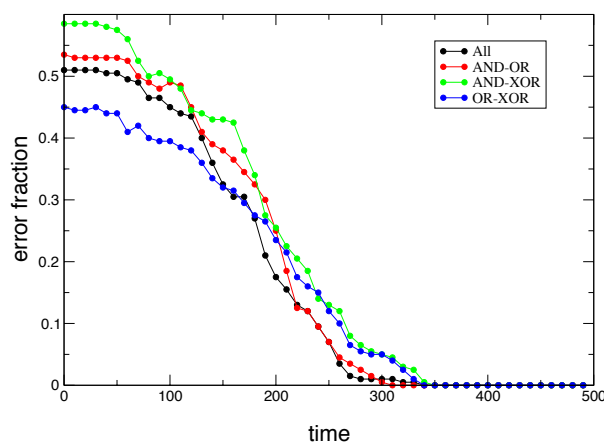


Figure 7.4: **Error fraction vanishes with time after function switched.** The fraction of error expected after an operation change as a function of time. Each color brings the expected fraction of error after a switch among specific operations (average over 200 switches). A new operation out of AND, OR or XOR is chosen randomly every five hundred time steps.

7.2 Autonomous navigation: field gradients

Autonomous navigation constitutes an important subject of current research in fields as different as avionics, robotics and neuroscience. A minimal example of navigation is given by an autonomous agent that navigates (converges) in relation to a source field. Depending on the area of research, different strategies are investigated in order to solve this sort of problem, from standard procedural computation to a mix of neural network and procedural computation. Here, we introduce a new method of navigation realized only by bio-inspired neuron-like units, where the neural activity yields spatiotemporal spike patterns characterized in state space by a persistent switching among saddle states.

The idea is to design a movable agent that exhibits field detectors with a well-defined spatial distribution; each of these detectors provides input to one neuron of a neural network in which it induces specific periodic spatiotemporal pulse patterns, limit cycles in the state space (see Chapter 6), conforms its position in relation to a field source; based on the achieved periodic pattern, a turning decision is made to yield a convergent movement towards the source of the field. Notice that in this picture, after detecting the field, the agent generates an abstract internal representation of the environment (field gradient) to then make a turning decision.

For the sake of simplicity, here we present the minimal example of such an agent. It is composed of three layers:

1. **the detectors:** five field detectors spatially distributed over a circle of radius ρ and equally spaced from each other by an angle σ (see Figure 7.5a); each detector unidirectionally targets only one neuron in the processing layer (see Figure 7.5b) and delivers an input that is a function $f(\psi)$ of the field ψ in its position;
2. **the processing layer** consists of a network of five neuronal units (3.3); asymmetric inputs $\Delta_i = f(\psi_i)$ delivered by the detectors induce specific cyclic pulse patterns classified by a vector \mathbf{r} (see Table 6.10), that consists of an internal representation of the inputs;
3. **the output:** by calculating the minimal distance to different saddle periodic orbits, the induced limit cycle in the process layer is detected, and proper turning decisions are made in order to direct the agent towards the field source.

Concerning the movement of the agent, we assume again the minimal (non-trivial) model where the agent advances at a fixed step size and there is only one source field. We here consider a two-dimensional environment with coordinates (x, y) , such that the agent's position and orientation is completely defined by the position (x_c, y_c) of its center and one of the coordinates of one of its detectors (x_i) (since the radius is fixed, ρ) or, in an equivalent way, the position of its center and the angle θ drawn in between the Cartesian x-axis and the line defined by the center of the agent and the central detector.

Consider a convergent movement to the source. The movement is composed of only two actions. In a first moment the decision of turning is made and in

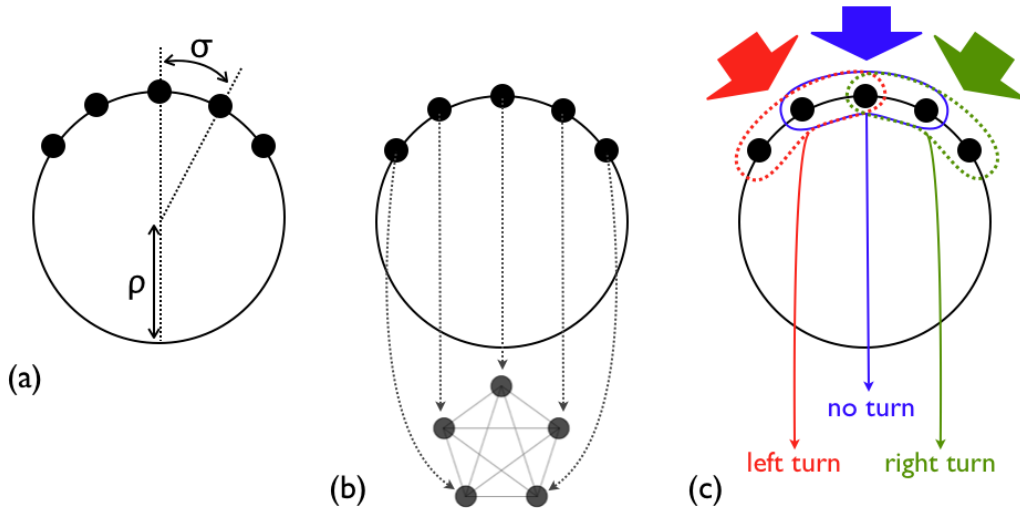


Figure 7.5: **Diagram of autonomous agent.** a) Five field detectors are distributed equally every σ degrees over a segment of a circle (total angle equaling 4σ) of radius ρ ; b) Each sensor is connected to only one element in the neural network, unidirectionally; c) For a convergent to the source behavior, the desired turn decision for the three possible situations: stronger inputs from the left (Red) to turn left; stronger inputs from the front (Blue) to no turning; stronger inputs from the right (Green) to turn right.

the second the actual step takes place. The turning decision begins by evaluating the field intensity over each of the detectors. Due to the large variation of field intensities at different distances from the source, the detectors implement a function $f(\psi)$ of ψ in order to normalize the field to reasonable values ($10^{-5} - 10^{-3}$, as shown in the last chapter), the exact form of the function is not relevant. The numerical simulation presented here is given by,

$$f(\psi_i) = \frac{\psi_i}{\max(\psi_j : j = \{1, \dots, 5\})} \times C, \quad (7.13)$$

where C is a constant. After this normalization, the resulting signals are streamed down to the neural network causing one out of three situations (see Figure 7.5c): first, the field is stronger on the left side of the agent, i.e. $\Delta_{1,2,3} > \Delta_{4,5}$ yielding the percept $\mathbf{r} = (1, 1, 1, 0, 0)$ and triggering a turn-left decision,

$$\theta = \theta + \eta, \quad (7.14)$$

where η is a fixed amount; second, the field is stronger on the right side, $\Delta_{3,4,5} > \Delta_{1,2}$ yielding the percept $\mathbf{r} = (0, 0, 1, 1, 1)$, triggering in the output layer a turn-right decision,

$$\theta = \theta - \eta; \quad (7.15)$$

and third, the field is stronger in the front, $\Delta_{2,3,4} > \Delta_{1,5}$ yielding the percept $\mathbf{r} = (0, 1, 1, 1, 0)$, that does not trigger any turning. Notice that, as in the last section, the output layer should be set up to respond in accordance with the desired agent behavior. The second action composing the movement is a simple fixed step

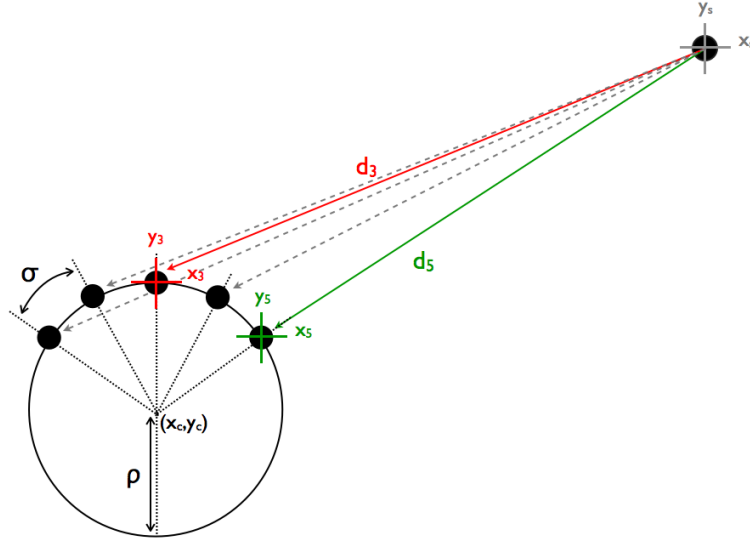


Figure 7.6: **Relative positions of agent and radial source field.** Diagram highlighting the distances d_3 and d_5 of detectors three and five with respect to the source field. The position of each detector is given by a cartesian pair (x_i, y_i) , where i is the detector index; (x_c, y_c) is the position of the center of the agent; and (x_s, y_s) is the position of the source.

in the actual direction of the central detector. After each step, the neural activity is updated in accordance with the new asymmetric input and the two steps of the movement are repeated again.

As an actual example of navigation, we implement an agent that pursues a single moving source field (see Figure 7.7). The field considered here is radial, defined by

$$\psi(d) = \frac{1}{d^2}, \quad (7.16)$$

where r is the distance between a point in the plane (x, y) and the source. This represents, e.g., a physical light source in two dimensions. While the source advances at a constant pace in a fixed direction, the agent adjust its trajectory in order to chase the source. The resulting dynamics is such that, at each time step, first the distance between the different detector and the source are calculated,

$$d_i = \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2}, \quad (7.17)$$

where (x_s, y_s) are the source coordinates and (x_i, y_i) are the coordinates of detector i (see Figure 7.6); second, the fields are calculated through 7.19,

$$\psi_i = \psi(d_i), \quad (7.18)$$

and third, the effective asymmetry is calculated according to 7.13,

$$\Delta_{\text{eff},i} = f(\psi_i). \quad (7.19)$$

And, by calculating the larger three $\Delta_{\text{eff},i}$ the adequate turning decision is made. Notice that due to the few detectors (low resolution) the agent does not align

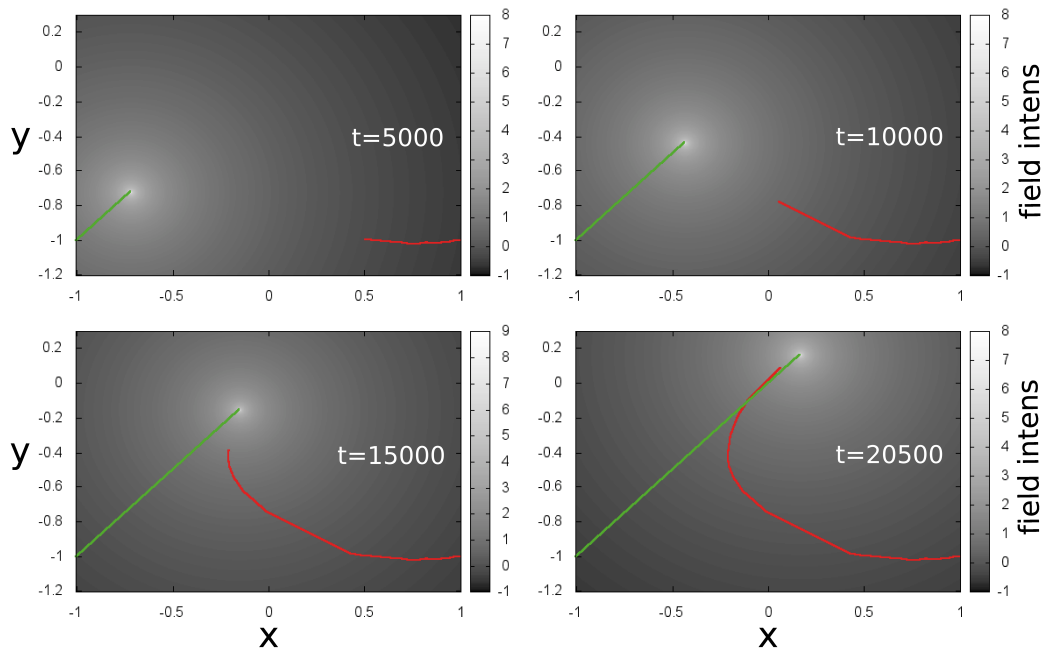


Figure 7.7: **Agent converges towards moving radial source field.** Four panels sampling different times (5000,10000,15000,20500). The gray gradient in the background depict the intensity of the field ($1/d^2$) in a logarithmic scale. The green line represents the filed source trajectory. The red line represents the agent trajectory. The simulation is ended as soon as the agent (its center) approaches a distance equal to two times the agent radius in relation to the source.

precisely with the source from large distances, because the spatial distribution of detectors imposes an effective angle (between the source and the agent orientation) to trigger each turning decision and, thus, an effective displacement proportional to r . Therefore, smaller values of σ and ρ yield a better resolution when close to the source field, whereas larger values provides a better resolution at larger distances. Furthermore, when the alignment tends to stay in the frontier between the non-turning situation and one of the two turning situations, the transient resulting from the constant change between these two percepts may cause momentary wrong turning decisions and the resulting path is therefore not a sequence of coherent corrections towards the desired path. Apart from this variation, on average the turning is in the direction of the source. Thus, if the agent's speed is high enough, the agent eventually overtakes the source.

7.3 Conclusion

In summary, we presented and analyzed how switching dynamics may be used for a new type of universal computation in pulse-coupled systems and applied this to steer an autonomous agent. Here, a simple five neuron network provides an internal representation of sensory input and at the same time perform all basic information processing. Providing small asymmetries to a previously symmetric

network of dynamically connected saddles triggers specific limit cycles close to the predefined heteroclinic network. By entering into cycles in a controllable way [10], the asymmetries can be associated with cycles, which in turn become abstract representations of the input asymmetry. Two distinct examples of computation were provided: First, we use part of the symmetry-breaking asymmetry to define a logic function and another part to define the input to this function. The resulting cycles themselves code the solution of a computation. As a second example, we provided an autonomous agent capable of pursuing a source field by implementing a similar computation as in the first example, i.e., by generating an abstract internal representation of the environment to then perform the required action. Understanding the computational capabilities of persistent switching processes in more detail will help to efficiently implement more complex computations; for instance, the cyclical spatiotemporal pattern itself (instead of a binary output) could serve as an input to a downstream neural device. Moreover, it might be possible to implement more complex functions such as multiplications, pattern recognition etc., in a single but larger oscillatory system exhibiting controllable switching.

Chapter 8

Memory by switching dynamics

In the previous chapters, we have studied how external inputs control a switching process that yields specific limit cycles. It was shown that, provided symmetry-breaking input signals, the system generates an internal representation of such inputs that are maintained as long as the asymmetry is present. In this chapter we show how to implement memory in this model by sustaining the internal representation even though the external asymmetric signal is removed after a short time. As a practical example, we show how to use this dynamics to memorize pictures of pixels colored via a six bits color palette.

In contrast to standard memory models, that require a structural change (synaptic plasticity) in order to store information, we here propose a dynamical memory mechanism based on perturbations of heteroclinic networks. It consists of a time dependent modification of an internal parameter of each oscillator, proportional to an external asymmetry, increasing in different proportions the intrinsic frequency of each oscillator, i.e., we altered an internal property of the oscillators in order to mimic the external asymmetric signal without changing the connection between two oscillators.

8.1 Short-term memory

In neuroscience, short-term memory is a process by which information is stored transiently. It is essential to many levels of animal behavior such as stimulus detection and processing, long-term memory formation, and to realizing working memory as well. In theory, the role of short-term memory in neuronal systems is analogous to those in digital computation, they serve either as information buffers (RAM) or as workspaces where information can be handled (cache/processor). This makes artificial neuronal systems exhibiting those features of great relevance for current research.

In this work, the relevant network states are complex periodic orbits that arise in the presence of network asymmetries. Thus, in principle, the implementation of memory could be realized by a sustained asymmetry. As shown in Chapters 6 and 7, the nature of the asymmetries is not relevant, but only the differences in the intrinsic frequency of each oscillator, which results in an effective asymmetry.

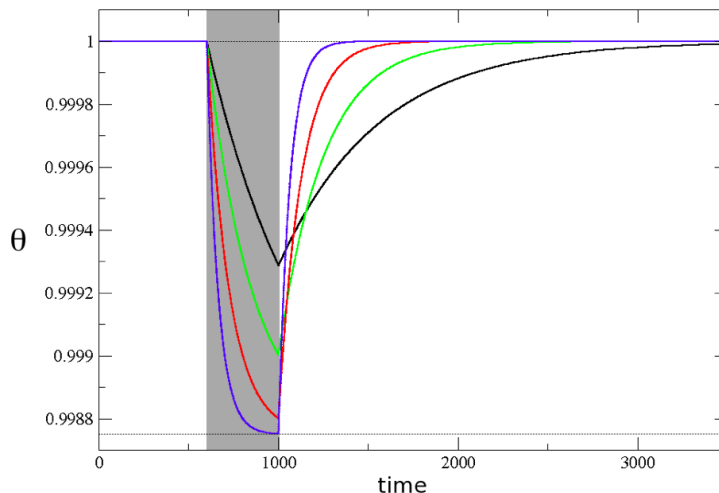


Figure 8.1: **Threshold shift for four different sets of parameters as a function of time.** The same external signal ($\Delta_i = 5 \times 10^{-4}$) is applied for the same period of time, 400 time steps (gray background), for four different parameter sets: $b = 0.005$ and $c = 0.002$ (black curve); $b = 0.01$ and $c = 0.004$ (green curve); $b = 0.02$ and $c = 0.04$ (red curve); $b = 0.04$ and $c = 0.016$ (blue curve). The shaded lines represents the two fixed points (8.2) and `erefe3`, common to all three sets of parameters: upper $\theta = 0$ for $\Delta = 0$; lower $\theta = 0.99875$ for $\Delta_i = 5 \times 10^{-4}$.

Therefore, the relevant question to be answered in this section is: is there a way to sustain the differences in frequency among the oscillators after the external input is turned off, by local mechanisms only? The answer is yes, and as we shall see, the mechanism is not only local to the network, but to each oscillator.

To approach this question, we extended the Integrate-and-Fire model introduced in Chapter 3 ((3.3) and (3.5)) to include spike facilitation: oscillators that receive stronger stimulation exhibit larger firing rates for a certain period after the stimulus. Here, it is implemented through a second differential equation that slightly shifts down the potential threshold θ_i of oscillator i as a function of external currents Δ_i (adaptive spiking threshold),

$$\frac{d\theta_i}{ds} = -b\Delta_i + c(\theta_0 - \theta_i), \quad (8.1)$$

where θ_0 is the initial, non-altered, threshold, s is the time variable and b and c are constants that regulate the speeds of the threshold shifts, decrement and increment respectively. Furthermore, the right side of equation (8.1) imposes a saturation point to θ_i (fixed point solution),

$$\frac{d\theta_i}{ds} = 0 \Leftrightarrow \theta_i = \theta_0 - \frac{b\Delta_i}{c}, \quad (8.2)$$

that guarantees, through a reasonable choice of b and c (by limiting the maximum threshold shift), that the basic switching dynamic is preserved. In the absence of asymmetries ($\Delta_i = 0$), the fixed point is shifted back to the original value,

$$\frac{d\theta_i}{ds} = 0 \Leftrightarrow \theta_i = \theta_0. \quad (8.3)$$

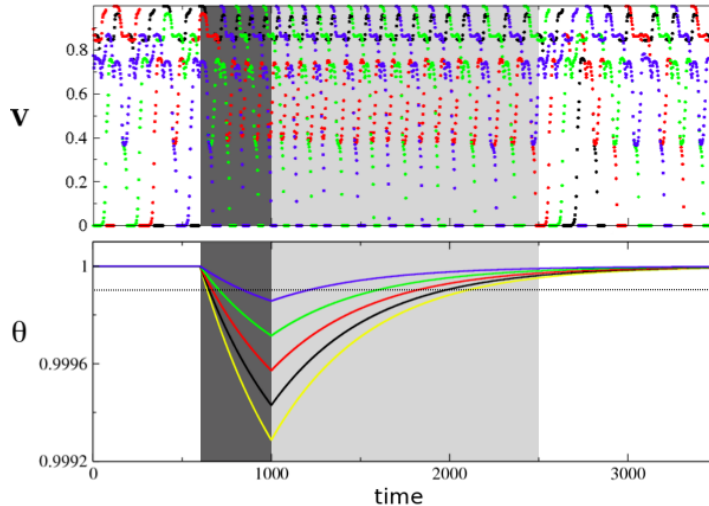


Figure 8.2: **Example of dynamical memory by threshold shift.** An external signal $\Delta = (5, 4, 3, 2, 1) \times 10^{-4}$ induces different shifts in the thresholds of distinct neurons. The resulting asymmetry sustains the specific complex limit cycles associated to the input signal after it is turned off. Before and after the memory process (white background), switchings are generated by noise (maximum amplitude of the order of 10^{-4}); After an imprinting period (dark gray background, time steps 600 to 1000), the asymmetry is sustained by the threshold shift (light gray background).

Notice that here, for the sake of simplicity and clarity, we choose the simplest possible implementation of spike facilitation, an adaptive spiking threshold, assuming a linear relation between input signals and the threshold shifts. More complex nonlinear dependencies will not qualitatively change the overall dynamics if the thresholds are shifted weakly. Thus, the simple linear version provides insights to generic aspects of this type of short-term memory.

In this model, the memory is realized in two phases: first, there is an imprinting period, where the external asymmetry Δ , in (8.1), induces a different shift down in the threshold of each oscillator; and second, the actual memory that takes place after the input is turned off; it consists of the period between the end of the input signal and the moment where the frequency differences generated by the imprinted asymmetry are overcome by the background noise (see Figure 8.2). Thus, from that moment on, any information about the input is erased, the periodic cycle is lost. Notice that, in the absence of noise, the memory will never be erased, but the induced cycle becomes longer and longer as the asymmetry becomes smaller.

Even though equation (8.1) is independent of (3.6), there are two constraints to be considered: first, the saturation point should be close enough to the initial value (θ_0) to maintain the initial switching properties, as discussed in chapter 6; second, the memory phase should be long enough to perform at least one complete cycle in order to guarantee an unambiguous cycle identification.

This simple modification to a neural oscillator model provides a straightforward memory device that is based on local information only. It provides an information buffer where the memory period is set by b and c , and a workspace where signals from different sources can be handled, since the internal network state can be manipulated by a process as simple as a linear superposition of currents.

8.2 Visual memory: 6-bit RGB color palette

One of the most fascinating human capabilities is the reliable way we identify and memorize new images and, as a second aspect, how fast we erase such information when it becomes not relevant anymore at a given moment. As humans, we are constantly bombarded with visual information, mostly discarded as irrelevant almost instantaneously. Therefore, the study of visual short-term memory models may yield the discovery of new aspects in visual information processes not only in artificial but also in biological systems.

In this section we implement a network capable of performing visual short-term memory based on the memory mechanism presented in Section 8.1. We consider an architecture such that small independent networks handle independent image pixels. Each pixel is composed of three basic colors, in the RGB color system, red, green and blue. In this system different color are defined by different intensities of these three basic colors, from no color (0% of available variant color) to full color (100% of available variant color). Here, we consider a six bits color resolution, two bits per basic color, resulting in a total of 64 colors. Thus, the color of each pixel $p_{x,y}$, with Cartesian coordinates (x, y) is defined by a binary vector

$$p_{x,y} = (p_{x,y}^1, p_{x,y}^2, p_{x,y}^3, p_{x,y}^4, p_{x,y}^5, p_{x,y}^6), \quad (8.4)$$

such that

$$i \in \{1, \dots, 6\} \quad \text{for all} \quad p_{x,y}^i \in \{0, 1\}. \quad (8.5)$$

The first two bits are reserved for the four red variants (no red to bright red), the third and fourth are reserved for the four green variants, and the fifth and sixth for the four blue variants. The intensity of each variant color increases from no color

$$(p_{x,y}^i, p_{x,y}^{i+1}) = (0, 0) \quad (8.6)$$

to full color

$$(p_{x,y}^i, p_{x,y}^{i+1}) = (1, 1), \quad (8.7)$$

with $i \in \{1, 3, 5\}$. The two intermediate intensities are labeled as $(0, 1)$ and $(1, 0)$, with the first being less intense than the second. Thus, $p_{x,y} = (0, 0, 0, 0, 0, 0)$ represents the black color (0% of red, 0% of green and 0% of blue) and $p_{x,y} = (1, 1, 1, 1, 1, 1)$ represents the white color (100% of red, 100% of green and 100% of blue). Apart from these two values, the exact values of the other colors in the RGB palette are not relevant, since internally the input colors assume a abstract representation (percepts). For illustrative purpose only, here the intermediate pair $(0, 1)$ represents a color intensity of 25% and the pair $(1, 0)$ a color intensity of 50%. On the other hand, the number of different abstract representations is relevant, and as mentioned before, here fixed at $2^6 = 64$.

In order to process those 64 different inputs, we here make use of two identical networks, composed of five oscillators each, in accordance with the network introduced in Chapter 3. As showed in Chapter 5, each of these networks provide ten

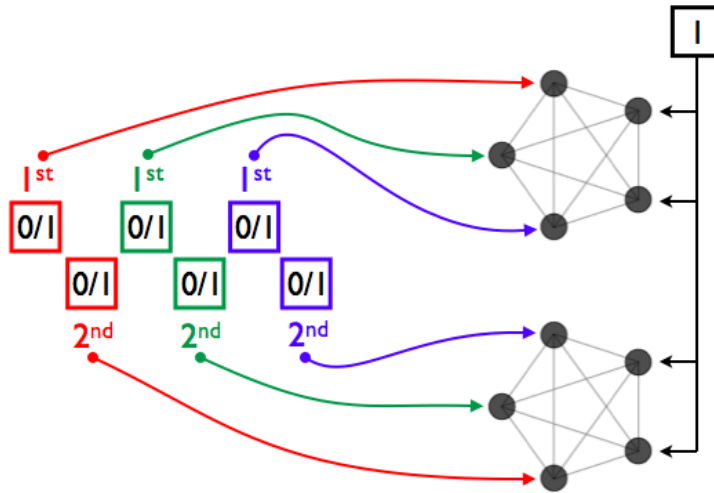


Figure 8.3: **Color detection: network diagram.** A six bits binary vector defines one out of 64 colors for a pixel, two bits per basic color (red-green-blue, RGB). The bits representing each color are separated into pairs of two, and used independently as inputs for two neural networks. Each bit is weight and added as an additional external current for a single oscillator ($\omega_i \Delta_i$). The lasting two oscillators of each network receive an constant input equal to one ($(\Delta_4, \Delta_5) = (1, 1)$)

percepts. Thus, the combination of two independent networks (total of $N = 10$) provides 100 distinct percepts, more than enough to represent 64 colors. The information about each color is divided between the two networks, such that the first network receives the first bit of every color, $p_{x,y}^i$ with $i \in \{1, 3, 5\}$, and the second network the second bit, $p_{x,y}^i$ with $i \in \{2, 4, 6\}$ (see Figure 8.3). The connections between these inputs and the networks are such that each input targets only one neuron in one of the networks. Therefore, for each pixel (x, y) , in the first network,

$$\frac{dV_i}{ds} = I - \gamma V_i + S_i(s) + \omega_i \Delta_i \quad (8.8)$$

with

$$\frac{d\theta_i}{ds} = -b\omega_i \Delta_i + c(\theta_0 - \theta_i), \quad (8.9)$$

and

$$\Delta = (p^1, p^3, p^5, 1, 1), \quad (8.10)$$

where ω_i is the connection weight between the input Δ_i and oscillator i . Notice that the subindex (x, y) is contracted, since it would appear in all constants and in all elements of every vector; notice also that the input component $(\Delta_4, \Delta_5) = (1, 1)$ is fixed in order to guarantee an effective asymmetry. In an equivalent way, the second network receives an input vector,

$$\Delta = (p^2, p^4, p^6, 1, 1). \quad (8.11)$$

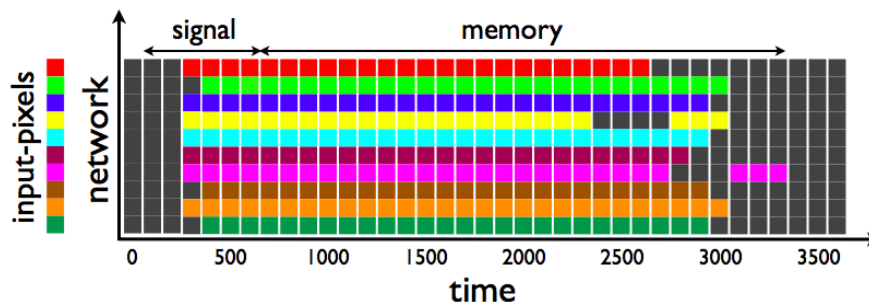


Figure 8.4: **Visual memory example.** A visual input is presented to the network as a set of colored pixels (columns in the left side of the picture). The network internal state is sampled every 100 time steps, depicting the dynamics. This input induce, after a transient, a change in the internal network state from no-information (black color in every pixel) to the same information presented in the input. The input is maintained from time 100 until 600; after time 2300, the information begins to degrade until 3300, when all information is lost.

This choice of distributing the bits is arbitrary, since the bits are independent. Here it is motivated only by symmetry reasons, since each color is defined by two bits, this separation maintains the equivalence between the two networks concerning the colors. That is, each network receives exactly half of the information about each color.

As shown in the previous section, the asymmetric input generates complex periodic orbits in both networks, that are sustained by the threshold shift until the information is erased by the background noise. Thus, the desired internal representation of a given color is achieved as soon as both complex periodic orbit are stabilized by the asymmetry, and lost as soon as one of the orbits is changed. Figure 8.4 show such dynamics for a string of pixels with different colors.

This approach brings two new interesting features related to short-term memory: first, since there is no synaptic plasticity, the noisy dynamics by itself erases all information after a given time; and second, a wrong color detection becomes unlikely, since it would require a valid sequence of six saddles when subject to an asymmetry that induces a different cycle.

8.3 Conclusion

In this chapter we introduced a novel conceptual model for short-term memory in systems of pulse-coupled neural oscillators. It exploits induced switchings between saddle periodic orbits that result in specific stable limit cycles, close to a heteroclinic framework. As a practical example, we implemented a visual short-term memory in color, by processing in parallel the color of each pixel. This model yields a reliable and flexible new kind of short term memory, since the phenomenon is robust to noise, due the nature of the network elements, and no learning rule is required to memorize new items, because the local dynamics of each element dynamically induces a limit cycle as an internal representation. Moreover, since there is no synaptic plasticity, any traces of a memory are dynamically erased due to noise.

Chapter 9

Discussion

In this thesis, we studied a new paradigm of natural computation by persistent switching dynamics among saddle periodic orbits. Such switching processes emerge in a broad range of network dynamical systems with symmetry. We focused on two main questions: how to control this switching processes by applying symmetry-breaking input signals and how to translate the relation between inputs and the resulting orbits into meaningful computations.

As specified in the first part (Chapters 1-3), we restricted this work to networks of generic pulse-coupled oscillators. This choice had two strong motivations: such models keep characteristic aspects of biological neuronal networks (as pulse-interactions and strong non-linearities), whereas they simultaneously yield analytic access to stability and switching properties [10, 25]. Therefore, our results may help to understand new aspects in computation not only in abstract model networks, but also in biological as well artificial systems.

In the second part (Chapters 4-6), theoretical aspects of robust heteroclinic cycles in symmetric networks of pulse-coupled oscillators were studied analytically. One prominent aspect of symmetric model networks is the reduced number of parameters especially when considering large networks. Thus, by choosing to study networks that are symmetric in all respects we simplified the initial task of identifying potential saddle periodic orbits. This was even further simplified by the existence of symmetrically related states. In such systems, by finding one orbit, a family of orbits is always identified due to the full permutation symmetry of the system.

This symmetry is one of the three main differences between encoding via switching dynamics and standard memory models exhibiting stable attractor dynamics [1, 32]. Whereas the first requires symmetry assumptions to identify a large number of symmetrically related states by varying few parameters, the second requires learning rules in order to adjust a large number of parameters proportional to the number of connections. A second contrasting aspect concerns the way information is encoded in time. As shown in Chapter 6, signals are encoded via switching dynamics by breaking the system symmetry, thus inducing heteroclinic cycles according to precise transition rules. Therefore, signals are encoded as spatiotemporal patterns, that in principle may yield an exponential encoding capacity, since the information is spread in time when combining cyclically different saddle

orbits. As a third aspect, we showed that only the rank order of the values of the asymmetric signals are relevant, not their absolute values. Based on these three features, we argue that encoding information in sequences of saddle states provides the basis for a new kind of neural computation that is independent of precise spatiotemporal pulse patterns, but only dependent on sequences of certain partially synchronous states.

In the third part of this work (Chapters 7-8), we present two applications that make use of the encoding process mentioned above: universal computation and short-term memory. In Chapter 7, we showed that a small network with $N = 5$ is capable of performing all basic bitwise operations (AND, OR and XOR) over a two bits operand by defining a mapping between input signals and complex periodic orbits. By construction, any computer capable of performing all basic operations is capable of universal computation. This argument in fact, is valid for any neural network model capable of performing such a mapping.

Therefore, the relevant questions regarding computation not only concerns the basic operations, but also the reliability and scalability of the neural computer architecture. Concerning reliability, the computational model introduced here exhibits two interesting unusual features. It yields a fast switching between different complex periodic orbits due to the intrinsic instability of the saddle periodic orbits and simultaneously provides, in the presence of input signals, an immediate internal representations intrinsically robust to noise due to the nature of the network elements. Concerning scalability, for the class of systems studied here, large networks exhibit prevalence of unstable attractors over stable ones [25], therefore increasing the likelihood of establishing a persistent switching process. Moreover, for these systems, the number of percepts does not increase linearly with the number of network elements as for stable attractor dynamical systems, but exponentially, because the number of percepts is proportional to the number of allowed saddle sequences and not the number of saddles.

Other approaches of natural computation based on distributed spatiotemporal dynamics of nonlinear neuron-like systems are markedly different from the one presented here. For instance, sparse networks operating in the "echo state" [34] or "liquid state" [33] are based on chaotic dynamics and represent (memorize) inputs for some time due to small positive Lyapunov exponents. The represented information is read out by learning output connections. These kinds of computation, generically known as reservoir computing, are similar to our computation by persistent switching processes in that information is represented as spatiotemporal dynamics and the system does not learn internally, only output connections are adapted. The main differences are that our computations are based on non-chaotic, heteroclinic switching, are therefore intrinsically robust to noise, are analytically well understood and, most importantly, fully controllable by the input. In addition, only one pattern of given temporal length is required to be learned in the output layer, because all induced complex periodic orbits are symmetrically related to each other.

Concerning memory, the original model as introduced in Chapter 3 does not exhibit memory features in the strict sense, because the system is only capable of representing an input as long as the input still present. In order to properly imple-

ment memory we extended the model to include spike facilitation, that sustains the asymmetry generated by an input signal through a threshold shift. This approach resulted in a novel implementation of short-term memory, intrinsically robust to noise and versatile with respect to state changes, due to its instability properties and the absence of synaptic adaptations. Moreover, contrary to stable attractor dynamics, in this memory process there is no error corrections or convergence to any state other than the percept related to the given input. Thus, while on one hand the system misses some properties of associative memory for categorization, it yields an almost instantaneous representation and storage of new information in the short-term even though the new item is not correlated to any previously established memory.

This study was motivated by recent works especially in two fields, biology and mathematics. On one hand, an ever growing body of evidence shows that biological systems may use spatiotemporal spike patterns to encode and process information [3, 35]. On the other hand, new techniques on how to study the interaction of pulse coupled oscillators have been developed. Of particular interest were the mathematical analysis of delayed pulse interactions and unstable yet attractive orbits [17]. Therefore, spatiotemporal patterns generated by a controllable switching in networks of pulse-coupled oscillators became an interesting subject. In this thesis our aim was to develop new natural forms of artificial computation.

When developing any scientific theory, two important points should be taken into account: clarity and generality. Here, these points were addressed by the choice of the network model. In this work, all mechanisms leading to a persistent switching process were mathematically derived based on the properties of heteroclinic cycles, and thus, are not restricted to any specific model. On the other hand, the pulse-interactions create an important link to biological systems. For instance, recent works [8, 23, 28] showed that controllable switching also occurs for distinct models, such as phase-coupled oscillators, that are conceptually different from our system due to its temporally continuous interactions. Concerning clarity, the size of the network played an important role. By assuming $N = 5$, we reduced the number of events for a given orbit, yielding a straightforward analysis. Notice that, as shown in [25], increasing the network size leads to a prevalence of unstable attractors and may increase the probability of finding heteroclinic connections. As an example, it was shown [25] that switchings with similar cluster splitting robustly occur for $N = 100$. An analysis analogous to the one presented in this thesis yield that the system is capable to distinguish the five strongest inputs from the other 95 after five switches.

Our results show how non-trivial switching dynamics underlies reliable computation in network dynamical systems. Future studies on different computation strategies in complex systems and the study of the interactions of their mechanisms may lead to new ideas on how more complex computations are performed by collective dynamics. In particular, as a possible direction of future research we consider the interaction of short-term and long term-memory in bio-inspired systems (pulse-coupled oscillator networks) to generate complex dynamics with different aspects than the ones presented by each network alone. As an initial step, we could extend the autonomous agent model to implement both, short- and long-term memory

in different scenarios in order to study nontrivial emergent behavior. Whereas the short-term memory could hold working memory processes exploiting saddle-induced switchings, the long-term memory could make use of stable attractors in order to recover, correct and store new items.

The most interesting example of a complex system acting as a computer is the brain. Many animal functions are dependent in the collective dynamics of neuronal units. It is reasonable to assume that highly complex behavior, for example reasoning, arises from the interactions of different neuronal circuits, possibly presenting distinct dynamical features. Thus, the study of new computational properties of bio-inspired networks are not only interesting in itself but also from this broader perspective of circuitry interactions. We hope that by presenting complex switching dynamics as a new computation paradigm, we contribute with a small step towards a better understanding of complex behavior in biological and artificial systems.

Appendix A

Periodic orbits: non-perturbed system

In Chapter 4 we introduced three unstable periodic orbits, clustered states with symmetries $S_3 \times S_2$, $S_4 \times S_1$ and $S_2 \times S_2 \times S_1$. Here we illustrate those orbits in the potential representation for all coordinates, $\mathbf{V} = (V_1, V_2, V_3, V_4, V_5)$ in state space. Each figure presents two superposed plots: the first displays real data point from the simulations as small consecutive circles (Red); the second is an interpolation of such data points (Green), included in order to explicitly show the relation between points far apart due to the discontinuous reset and due to the instantaneous jumps in the potential of the oscillators each time a pulse is received. In all figures the potential of the unstable cluster is used as the horizontal coordinate.

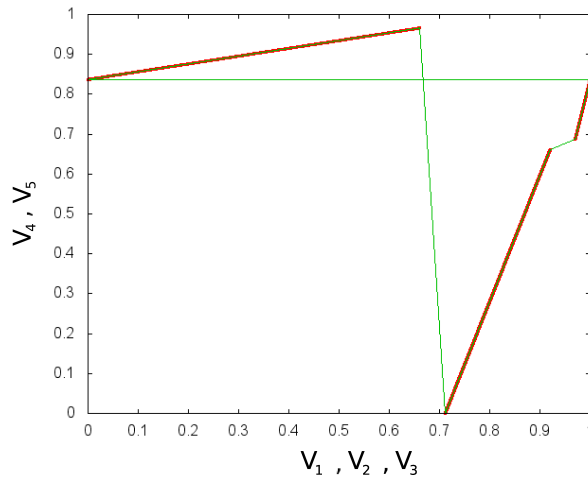


Figure A.1: **Periodic orbit for cluster symmetry $S_3 \times S_2$.** Red circles represent data points from the simulations; Green line is the interpolation of the data points, showing the relation between points far apart due to the discontinuous reset of the oscillators and phase jumps. ($\tau_s = 1.01002$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$).

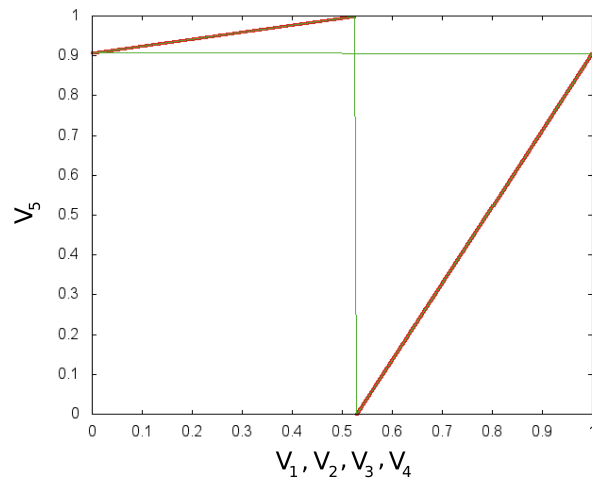


Figure A.2: **Periodic orbit for cluster symmetry $S_4 \times S_1$.** Red circles represent data points from the simulations; Green line is the interpolation of the data points, showing the relation between points far apart due to the discontinuous reset of the oscillators and phase jumps. ($\tau_s = 0.64743$, $\epsilon = 0.0015$, $I = 1.1$, $\gamma = 1$).

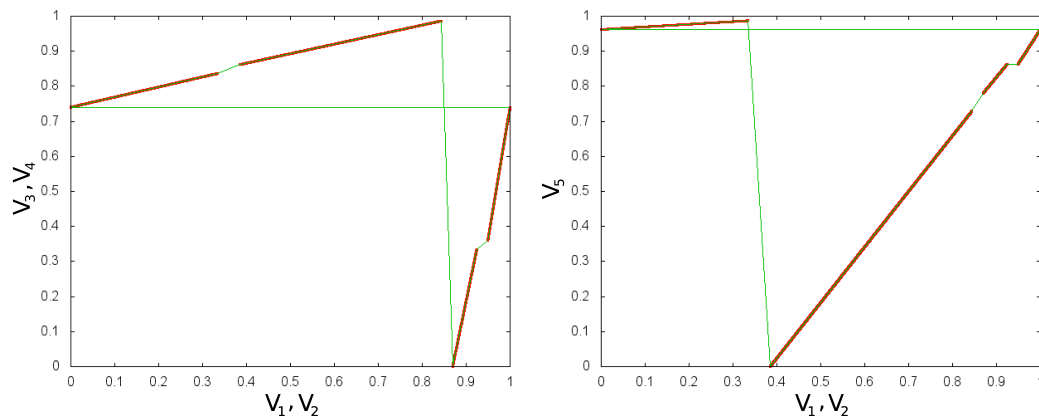


Figure A.3: **Periodic orbit for cluster symmetry $S_2 \times S_2 \times S_1$.** Red circles represent data points from the simulations; Green line is the interpolation of the data points, showing the relation between points far apart due to the discontinuous reset of the oscillators and phase jumps. ($\tau_s = 1.59646$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$).

Appendix B

Partial derivatives

Here we present the analytical expressions for the non-zero elements of the Jacobian matrices introduced on Chapter 5. We here use the short notation

$$x_y = H_y(x), \quad (\text{B.1})$$

with $x \in \{0, \tau, (\tau - \tau'), (\tau - \tau' + \tau'_y)\}$ and $y \in \{\epsilon, 2\epsilon, 3\epsilon\}$.

S3 × S2: non-zero elements for the Jacobian matrix

$$\alpha = \left[-1 + \left[1 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) \right] H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (\text{B.2})$$

$$\beta = \left[-1 + 2H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (\text{B.3})$$

$$\gamma = \left[-1 + H'_\epsilon(0) \right] H'_{2\epsilon}(\tau + 0_\epsilon) + H'_\epsilon(\tau_\epsilon) H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (\text{B.4})$$

$$j_{21} = - \left[1 + \left[-2 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) \right] H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (\text{B.5})$$

$$j_{32} = j_{42} = \left[-1 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) H'_{2\epsilon}(\tau + \tau_{2\epsilon}) \quad (\text{B.6})$$

S4 × S1: non-zero elements for the Jacobian matrix

$$\alpha = \left[-1 + \left[1 + H'_\epsilon(\tau) H'_\epsilon(\tau_\epsilon) \right] H'_\epsilon(\tau_{2\epsilon}) \right] H'_\epsilon(\tau + \tau_{3\epsilon}) \quad (\text{B.7})$$

$$\beta = \left[-1 + \left[1 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) \right] H'_\epsilon(\tau_{2\epsilon}) H'_\epsilon(\tau + \tau_{3\epsilon}) \quad (\text{B.8})$$

$$\gamma = \left[-1 + 2H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) H'_\epsilon(\tau_{2\epsilon}) H'_\epsilon(\tau + \tau_{3\epsilon}) \quad (\text{B.9})$$

$$j_{31} = j_{21} = - \left[1 + \left[-2 + H'_\epsilon(\tau_\epsilon) \right] H'_\epsilon(\tau_{2\epsilon}) \right] H'_\epsilon(\tau + \tau_{3\epsilon}) \quad (\text{B.10})$$

$$j_{41} = - \left[-1 + H'_\epsilon(0_\epsilon) \right] H'_\epsilon(0_{2\epsilon}) + \left[-1 + H'_\epsilon(\tau_\epsilon) \right] H'_\epsilon(\tau + \tau_\epsilon) \quad (\text{B.11})$$

$$j_{42} = - \left[-1 + H'_\epsilon(0) \right] H'_\epsilon(0_\epsilon) H'_\epsilon(0_{2\epsilon}) + \left[-1 + H'_\epsilon(\tau_\epsilon) \right] H'_\epsilon(\tau_{2\epsilon}) H'_\epsilon(\tau + \tau_{3\epsilon}) \quad (\text{B.12})$$

$$\theta = -1 + H'_\epsilon(0) H'_\epsilon(0_\epsilon) H'_\epsilon(0_{2\epsilon}) + \left[1 + \left[-1 + H'_\epsilon(\tau) \right] H'_\epsilon(\tau_\epsilon) H'_\epsilon(\tau_{2\epsilon}) \right] H'_\epsilon(\tau + \tau_{3\epsilon}) \quad (\text{B.13})$$

S2 × S2 × S1: non-zero elements for the Jacobian matrix

$$\lambda_1 = \alpha = H'_\epsilon \left(\tau' + \left(\tau - \tau' + \tau'_{2\epsilon} \right)_\epsilon \right) \left[-1 + H'_\epsilon \left(\tau - \tau' + \tau'_{2\epsilon} \right) \left[1 + H'_{2\epsilon}(\tau') \right] \right], \quad (\text{B.14})$$

$$\beta = H'_\epsilon(0)H'_\epsilon(\tau' + 0_\epsilon) + H'_\epsilon \left(\tau' + \left(\tau - \tau' + \tau_{2\epsilon} \right)_\epsilon \right) \left[-1 + H'_\epsilon(\tau - \tau' + \tau'_{2\epsilon}) \right] \quad (\text{B.15})$$

$$\gamma = - \left[-1 + H'_\epsilon(\tau - \tau') \right] H'_\epsilon \left(\left(\tau - \tau' \right)_\epsilon \right) + H'_\epsilon \left(\tau' + \left(\tau - \tau' + \tau'_\epsilon \right)_\epsilon \right) \left[-1 + H'_\epsilon(\tau - \tau' + \tau'_{2\epsilon}) \right] \quad (\text{B.16})$$

Appendix C

Unperturbed orbits: return maps

Here we explain step by step the periodic orbit dynamics introduced in Chapter 4. We introduce and describe the return maps that define the three families of orbits. The event sequences are displayed in Tables C.1, C.3, C.4 and C.6. The event notation is: σ_i indicates that oscillator i elicited a pulse; ρ_i indicates that a pulse coming from the oscillators i is received by the other oscillators; capital letters indicate constants; $H_\epsilon(\phi)$ is the transfer function introduced in Chapter 3; and $p_{i,j}$ is a short notation for the phase of oscillator i at event number j .

Specific realizations of the general descriptions of the orbits dynamics are presented in tables C.2, C.5 and C.7. As there are no approximations to the corresponding analytical event sequences, the specific values completely agree with the iterated simulation in Figures 5.1, 5.3 and 5.5.

Unperturbed $S_3 \times S_2$ orbit

The initial condition is such that no pulse was sent before time zero. At time zero, the first event takes place, that is, oscillators one, two and three elicit one pulse each (σ_{123}); the second event is the reception of these pulses at τ time later (ρ_{123}), that simultaneously is an supra-threshold event to oscillators four and five. This supra-threshold event induce oscillators four and five to elicit a pulse each (σ_{45}) in the same instant they are reset. The third event is the reception of pulses coming from oscillators four and five (ρ_{45}). And the last event is the reset of oscillators one, two and three (σ_{123}) by reaching the threshold.

Table C.1: Analytic table of condition for an unperturbed $S_3 \times S_2$ orbit, S_3 unstable.

event	time	ϕ_1, ϕ_2, ϕ_3	ϕ_4, ϕ_5	event num.
$\sigma_{1,2,3}$	0	0	A	0
$\rho_{1,2,3}; \sigma_{4,5}$	τ	$H_{2\epsilon}(\tau) = p_{1,1}$	$H_{3\epsilon}(A + \tau) > 1 \rightarrow 0$	1
$\rho_{4,5}$	2τ	$H_{2\epsilon}(p_{1,1} + \tau) = p_{1,2}$	$H_\epsilon(\tau) = p_{4,2}$	2
$\sigma_{1,2,3}$	$2\tau + 1 - p_{1,2}$	$1 \rightarrow 0$	$p_{4,2} + 1 - p_{1,2}$	3

For any choice of the parameters resulting in $A = p_{4,2} + 1 - p_{1,2}$ (while preserving the event sequence), we have a period-one attractor, since the initial state is

obtained after one pulse of each oscillator. A numerical example of such a structure is presented in table C.2. From this map, we can conclude that the cluster S_2 is stable, since any small variation will be restored when its elements are reset together by the incoming pulse.

Table C.2: Analytic prediction of phase dynamics for parameters $\tau = 0.31$, $\epsilon = 0.025$, $I = 1.04$ and $\gamma = 1$, realizing a $S_3 \times S_2$ cycle.

event	time	ϕ_1, ϕ_2, ϕ_3	ϕ_4, ϕ_5	event num.
$\sigma_{1,2,3}$	0.000000	0.000000	0.501612	0
$\rho_{1,2,3}; \sigma_{4,5}$	0.310000	0.353450	0.000000	1
$\rho_{4,5}$	0.620000	0.829344	0.330956	2
$\sigma_{1,2,3}$	0.790655	0.000000	0.501612	3

Unperturbed $S_3 \times S_2$ orbit, S_2 unstable

This map describe the partner orbit of table C.1, since they coexist for the same range of the parameter, but for different initial conditions. The initial condition here is such that pulses coming from oscillators four and five are received ($\rho_{(4,5)}$) at time equal zero, forcing oscillators one, two and three to elicit one pulse each ($\sigma_{1,2,3}$), already defining the first event. The second event is the reception of these pulses ($\rho_{1,2,3}$) τ time later. The third event is the reset of oscillators number four and five upon reaching the threshold ($\sigma_{4,5}$), and the consequent generation of two new pulses. The last event is the reception of these new pulses ($\rho_{4,5}$), causing oscillators one, two and three to elicit one pulse each ($\sigma_{1,2,3}$). In this case the S_3 cluster is stable, since any small variation on the phase of its components will disappear in the next cycle when all its elements are reset at the same time by the incoming pulses.

Table C.3: Analytic table of condition for an unperturbed $S_3 \times S_2$ orbit, S_2 unstable.

event	time	ϕ_1, ϕ_2, ϕ_3	ϕ_4, ϕ_5	event num.
$\rho_{(4,5)}; \sigma_{1,2,3}$	0	0	C	0
$\rho_{1,2,3}$	τ	$H_{2\epsilon}(\tau) = p_{1,1}$	$H_{3\epsilon}(C + \tau) = p_{4,1}$	1
$\sigma_{4,5}$	$1 - p_{4,1}$	$1 + p_{1,1} - p_{4,1} = p_{1,2}$	$1 \rightarrow 0$	2
$\rho_{4,5}; \sigma_{1,2,3}$	$\tau + 1 - p_{4,1}$	$H_{2\epsilon}(p_{1,2} + \tau) > 1 \rightarrow 0$	$H_{\epsilon}(\tau)$	3

Unperturbed $S_4 \times S_1$ orbit

This map describes another period-one attractor, where no pulse was sent before time zero. The first event is set as elicited pulses by oscillators one, two, three and four ($\sigma_{1,2,3,4}$). The second event is the reception of these pulses after τ time ($\rho_{1,2,3,4}$), which makes oscillator number five to elicit one pulse due to this supra-threshold input (σ_5). The third event is the reception of this last pulse at time 2τ (ρ_5). The last event is the pulse generation from oscillators one, two, three, and four upon reaching the threshold. A numerical example of this orbit structure is presented in table C.5.

Table C.4: Analytic table of condition for an unperturbed $S_4 \times S_1$ orbit.

event	time	$\phi_1, \phi_2, \phi_3, \phi_4$	ϕ_5	event num.
$\sigma_{1,2,3,4}$	0	0	B	0
$\rho_{1,2,3,4}; \sigma_5$	τ	$H_{3\epsilon}(\tau) = p_{1,1}$	$H_{4\epsilon}(B + \tau) > 1 \rightarrow 0$	1
ρ_5	2τ	$H_\epsilon(p_{1,1} + \tau) = p_{1,2}$	τ	2
$\sigma_{1,2,3,4}$	$2\tau + 1 - p_{1,2}$	$1 \rightarrow 0$	$\tau + 1 - p_{1,2}$	3

Table C.5: Analytic prediction of phase dynamics for parameters $\tau = 0.27$, $\epsilon = 0.0015$, $I = 1.1$ and $\gamma = 1$, realizing a $S_4 \times S_1$ cycle.

event	time	$\phi_1, \phi_2, \phi_3, \phi_4$	ϕ_5	event num.
$\sigma_{1,2,3,4}$	0.000000	0.000000	0.672908	0
$\rho_{1,2,3,4}; \sigma_5$	0.270000	0.303940	0.000000	1
ρ_5	0.540000	0.597091	0.270000	2
$\sigma_{1,2,3,4}$	0.942909	0.000000	0.672908	3

Unperturbed $S_2 \times S_2 \times S_1$ orbit

For this map, the initial conditions are such that pulses from oscillators three and four will be received at time τ' after time zero. The first event is the pulse generation from oscillators one and two upon reaching the threshold ($\sigma_{1,2}$). The second event is the reception of pulses from oscillators three and four ($\rho_{(3,4)}$) at time τ' and the pulse generation by oscillator five (σ_5) caused by this supra-threshold input. The third event is the reception of pulses coming from oscillators one and two ($\rho_{1,2}$) at time τ that forces oscillators three and four to elicit a pulse each ($\sigma_{3,4}$). The fourth event is the reception of the pulse coming from oscillator five (ρ_5). The last event is the pulse generation by oscillators one and two ($\sigma_{1,2}$) upon reaching the threshold.

Table C.6: Analytic table of condition for an unperturbed $S_2 \times S_2 \times S_1$ orbit.

event	time	ϕ_1, ϕ_2	ϕ_3, ϕ_4	ϕ_5	event num.
$\sigma_{1,2}$	0	0	D	E	0
$\rho_{(3,4)}; \sigma_5$	τ'	$H_{2\epsilon}(\tau) = p_{1,1}$	$H_\epsilon(D + \tau') = p_{3,1}$	$H_{2\epsilon}(E + \tau') > 1 \rightarrow 0$	1
$\rho_{1,2}; \sigma_{3,4}$	τ	$H_\epsilon(p_{1,1} + \tau - \tau') = p_{1,2}$	$H_{2\epsilon}(p_{3,1} + \tau - \tau') > 1 \rightarrow 0$	$H_{2\epsilon}(\tau - \tau')$	2
ρ_5	$\tau + \tau'$	$H_\epsilon(p_{1,2} + \tau') = p_{1,3}$	$H_\epsilon(p_{3,2} + \tau') = p_{3,3}$	$p_{5,2} + \tau' = p_{5,3}$	3
$\sigma_{1,2}$	$\tau + \tau' + 1 - p_{1,3}$	$1 \rightarrow 0$	$p_{3,3} + 1 - p_{1,3}$	$p_{5,3} + 1 - p_{1,3}$	4

Table C.7: Analytic prediction of phase dynamics for parameters $\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$ and $\gamma = 1$, realizing a $S_2 \times S_2 \times S_1$ cycle.

event	time	ϕ_1, ϕ_2	ϕ_3, ϕ_4	ϕ_5	event num.
$\sigma_{1,2}$	0.000000	0.000000	0.381978	0.795680	0
$\rho_{(3,4)}; \sigma_5$	0.119095	0.141656	0.541358	0.000000	1
$\rho_{1,2}; \sigma_{3,4}$	0.490000	0.554491	0.000000	0.424775	2
ρ_5	0.609095	0.748191	0.130168	0.543870	3
$\sigma_{1,2}$	0.860904	0.000000	0.381978	0.795680	4

This map implies three periodic conditions to describe a period-one attractor: $D = p_{3,3} + 1 - p_{1,3}$, $E = p_{5,3} + 1 - p_{1,3}$, and $\tau = 2\tau' + 1 - p_{1,3}$. An example of this structure is presented in table C.7.

Appendix D

Perturbed orbit dynamics: return maps.

Here we present three tables that show the changes to the periodic orbit dynamics shown in tables C.1, C.4 and C.6 after an incremental perturbation $\delta = (0, \delta_2, \delta_3, \delta_4, \delta_5)$, where $0 < \delta_2 < \delta_3 < \delta_4 < \delta_5 \ll 1$. For all three orbits, without loss of generality, oscillator one was taken as the referential phase to define the new cycle, since this doesn't affect the perturbed orbit dynamics itself but only the point of reference. The notation used here is the same as in C.

Table D.1: Perturbed $S_3 \times S_2$ orbit.

event	time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	event num.
$\sigma_{1,(2,3)}$	0	0	δ_2	δ_3	$A + \delta_4$	$A + \delta_5$	0
ρ_3	$\tau - \delta_3$	$H_\epsilon(\tau - \delta_3) = p_{1,1a}$	$H_\epsilon(\delta_2 + \tau - \delta_3) = p_{2,1a}$	τ	$H_\epsilon(A + \delta_4 + \tau - \delta_3) = p_{4,1a}$	$H_\epsilon(A + \delta_5 + \tau - \delta_3) = p_{5,1a}$	1a
$\rho_2, \sigma_{4,5}$	$\tau - \delta_2$	$H_\epsilon(p_{1,1a} + \delta_2 - \delta_3) = p_{1,1b}$	$p_{2,1a} + \delta_3 - \delta_2 = p_{2,1b}$	$H_\epsilon(\tau + \delta_3 - \delta_2) = p_{3,1b}$	$H_\epsilon(p_{4,1a} + \delta_3 - \delta_2) = 1 \rightarrow 0$	$H_\epsilon(p_{5,1a} + \delta_3 - \delta_2) = 1 \rightarrow 0$	1b
ρ_1	τ	$p_{1,1b} + \delta_2 = p_{1,1}$	$H_\epsilon(p_{2,1b} + \delta_2) = p_{2,1}$	$H_\epsilon(p_{3,1b} + \delta_2) = p_{3,1}$	$H_\epsilon(\delta_2) = p_{4,1}$	$p_{4,1}$	1
$\rho_{4,5}$	$2\tau - \delta_2$	$H_{2\epsilon}(p_{1,1} + \tau - \delta_2) = p_{1,2}$	$H_{2\epsilon}(p_{2,1} + \tau - \delta_2) = p_{2,2}$	$H_{2\epsilon}(p_{3,1} + \tau - \delta_2) = p_{3,2}$	$H_{2\epsilon}(p_{4,1} + \tau - \delta_2) = p_{4,2}$	$p_{4,2}$	2
σ_3	$2\tau - \delta_2 + 1 - p_{3,2}$	$p_{1,2} + 1 - p_{3,2}$	$p_{2,2} + 1 - p_{3,2}$	$1 \rightarrow 0$	$p_{4,2} + 1 - p_{3,2}$	$p_{4,2} + 1 - p_{3,2}$	3a
σ_2	$2\tau - \delta_2 + 1 - p_{2,2}$	$p_{1,2} + 1 - p_{2,2}$	$1 \rightarrow 0$	$p_{3,2} - p_{2,2}$	$p_{4,2} + 1 - p_{2,2}$	$p_{4,2} + 1 - p_{2,2}$	3b
σ_1	$2\tau - \delta_2 + 1 - p_{1,2}$	$1 \rightarrow 0$	$p_{2,2} - p_{1,2}$	$p_{3,2} - p_{1,2}$	$p_{4,2} + 1 - p_{1,2}$	$p_{4,2} + 1 - p_{1,2}$	3

Table D.2: Perturbed $S_4 \times S_1$ orbit.

event	time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	event num.
$\sigma_{1,(2,3,4)}$	0	0	δ_2	δ_3	δ_4	$B + \delta_5$	0
$\rho_4; \sigma_5$	$\tau - \delta_4$	$H_\epsilon(\tau - \delta_4) = p_{1,1a}$ $p_{1,1c}$	$H_\epsilon(\tau + \delta_2 - \delta_4)$ $p_{2,1a}$	$H_\epsilon(\tau + \delta_3 - \delta_4)$ $p_{3,1a}$	τ	$H_\epsilon(B + \delta_5 + \tau - \delta_4)$ $1 \rightarrow 0$	1a
ρ_3	$\tau - \delta_3$	$H_\epsilon(p_{1,1a} + \delta_4 - \delta_3)$ $p_{1,1b}$	$H_\epsilon(p_{2,1a} + \delta_4 - \delta_3)$ $p_{2,1b}$	$p_{3,1a} + \delta_4 - \delta_3 = p_{3,1b}$	$H_\epsilon(\tau + \delta_4 - \delta_3)$ $p_{4,1b}$	$H_\epsilon(\delta_4 - \delta_3) = p_{5,1b}$	1b
ρ_2	$\tau - \delta_2$	$H_\epsilon(p_{1,1b} + \delta_3 - \delta_2)$ $p_{1,1c}$	$p_{2,1b} + \delta_3 - \delta_2 = p_{2,1c}$	$H_\epsilon(p_{3,1b} + \delta_3 - \delta_2)$ $p_{3,1c}$	$H_\epsilon(p_{4,1b} + \delta_3 - \delta_2)$ $p_{4,1c}$	$H_\epsilon(p_{5,1b} + \delta_3 - \delta_2)$ $p_{5,1c}$	1c
ρ_1	τ	$p_{1,1c} + \delta_2 = p_{1,1}$	$H_\epsilon(p_{2,1c} + \delta_2) = p_{2,1}$	$H_\epsilon(p_{3,1c} + \delta_2) = p_{3,1}$	$H_\epsilon(p_{4,1c} + \delta_2) = p_{4,1}$	$H_\epsilon(p_{5,1c} + \delta_2) = p_{5,1}$	1
ρ_5	$2\tau - \delta_4$	$H_\epsilon(p_{1,1} + \tau - \delta_4)$ $p_{1,2}$	$H_\epsilon(p_{2,1} + \tau - \delta_4)$ $p_{2,2}$	$H_\epsilon(p_{3,1} + \tau - \delta_4)$ $p_{3,2}$	$H_\epsilon(p_{4,1} + \tau - \delta_4)$ $p_{4,2}$	$p_{5,1} + \tau - \delta_4 = p_{5,2}$	2
σ_4	$2\tau - \delta_4 + 1 - p_{4,2}$	$p_{1,2} + 1 - p_{4,2}$	$p_{2,2} + 1 - p_{4,2}$	$p_{3,2} + 1 - p_{4,2}$	$1 \rightarrow 0$	$p_{5,2} + 1 - p_{4,2}$	3a
σ_3	$2\tau - \delta_4 + 1 - p_{3,2}$	$p_{1,2} + 1 - p_{3,2}$	$p_{2,2} + 1 - p_{3,2}$	$1 \rightarrow 0$	$p_{4,2} - p_{3,2}$	$p_{5,1} + \tau - \delta_4 + 1 - p_{3,2} = p_{5,3b}$	3b
σ_2	$2\tau - \delta_4 + 1 - p_{2,2}$	$p_{1,2} + 1 - p_{2,2}$	$1 \rightarrow 0$	$p_{3,2} - p_{2,2} = p_{3,3c}$	$p_{4,2} - p_{2,2}$	$p_{5,3b} + p_{3,2} - p_{2,2} = p_{5,3c}$	3c
σ_1	$2\tau - \delta_4 + 1 - p_{1,2}$	$p_{1,2} + 1 - p_{1,2}$	$p_{2,2} - p_{1,2}$	$p_{3,2} - p_{1,2}$	$p_{4,2} - p_{1,2}$	$p_{5,3c} + p_{2,2} - p_{1,2}$	3

Table D.3: Perturbed $S_2 \times S_2 \times S_1$ orbit.

event	time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	event num.
$(\sigma_{3,4}; \sigma_{1,(2)})$	0	0	δ_2	$D + \delta_3$	$D + \delta_4$	$E + \delta_5$	0
$\rho_{3,4}; \sigma_5$	τ'	$H_{2\epsilon}(\tau') = p_{1,1}$	$H_{2\epsilon}(\tau' + \delta_2) = p_{2,1}$	$H_{\epsilon}(D + \tau' + \delta_3) = p_{3,1}$	$H_{\epsilon}(D + \tau' + \delta_4) = p_{4,1}$	$H_{2\epsilon}(E + \tau' + \delta_5) > 1$ $1 \rightarrow 0$	1
$\rho_2; \sigma_{3,4}$	$\tau - \delta_2$	$H_{\epsilon}(p_{1,1} + \tau - \tau' - \delta_2) = p_{1,2a}$	$p_{2,1} + \tau - \tau' - \delta_2 = p_{2,2a}$	$H_{\epsilon}(p_{3,1} + \tau - \tau' - \delta_2) > 1 \rightarrow 0$	$H_{\epsilon}(p_{4,1} + \tau - \tau' - \delta_2) > 1 \rightarrow 0$	$H_{\epsilon}(\tau - \tau' - \delta_2) = p_{5,2a}$	2a
ρ_1	τ	$p_{1,2a} + \delta_2 = p_{1,2}$	$H_{\epsilon}(p_{2,2a} + \delta_2) = p_{2,2}$	$H_{\epsilon}(\delta_2) = p_{3,2}$	$H_{\epsilon}(\delta_2) = p_{4,2}$	$H_{\epsilon}(p_{5,2a} + \delta_2) = p_{5,2}$	2
ρ_5	$\tau - \tau'$	$H_{\epsilon}(p_{1,2} + \tau') = p_{1,3}$	$H_{\epsilon}(p_{2,2} + \tau') = p_{2,3}$	$H_{\epsilon}(p_{3,2} + \tau') = p_{3,3}$	$p_{3,3}$	$p_{5,2} + \tau'$	3
σ_2	$\tau + \tau' + 1 - p_{2,3}$	$p_{1,3} + 1 - p_{2,3}$	$1 \rightarrow 0$	$p_{3,3} + 1 - p_{2,3} = p_{3,4a}$	$p_{3,4}$	$p_{5,2} + \tau' + 1 - p_{2,3}$	4a
σ_1	$\tau + \tau' + 1 - p_{1,3}$	$1 \rightarrow 0$	$p_{2,3} - p_{1,3}$	$p_{3,3} + 1 - p_{1,3}$	$p_{3,3} + 1 - p_{1,3}$	$p_{5,2} + \tau' + 1 - p_{1,3}$	4

Bibliography

- [1] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554–2558, 1982.
- [2] B. Widrow and M. A. Lehr. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proc. of the IEEE*, 78:1415–1442, 1990.
- [3] M. Rabinovich, A. Volkovskii, P. Lecanda, R. Huerta, H.D.I. Abarbanel, and G. Laurent. Dynamical encoding by networks of competing neuron groups: Winnerless competition. *Phys. Rev. Lett.*, 87, 2001.
- [4] J. Niessing and R. W. Friedrich. Olfactory pattern classification by discrete neuronal network states. *Nature*, 465:47–52, 2010.
- [5] C. Kirst and M. Timme. From networks of unstable attractors to heteroclinic switching. *Phys. Rev. E*, 78, 2008.
- [6] C. Bick and M. I. Rabinovich. Dynamical origin of the effective storage capacity in the brain’s working memory. *Phys. Rev. Lett.*, 103:218101, 2009.
- [7] T. Munakata, S. Sinha, and W. L. Ditto. Chaos computing: Implementation of fundamental logical gates by chaotic elements. *IEEE Trans. Circ. Syst. Fund. Theor. Appl.*, 49:1629–1633, 2002.
- [8] P. Ashwin and J. Borresen. Discrete computation using a perturbed heteroclinic network. *Physics Letters A*, 347, 2005.
- [9] O. Feinerman, A. Rotem, and E. Moses. Reliable neuronal logic devices from patterned hippocampal cultures. *Nature Physics*, 4:967–973, 2008.
- [10] F. S. Neves and M. Timme. Controlled perturbation-induced switching in pulse-coupled oscillator networks. *J. Phys. A: Math. Theor.*, 42:345103, 2009.
- [11] W. Maass and C. M. Bishop. Pulsed neural networks. *MIT Press, Cambridge*, 1999.
- [12] R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM J. Appl. Math.*, 50, 1990.
- [13] C. Chen. Threshold effects on synchronization of pulse-coupled oscillators. *Phys. Rev. E*, 49:2668–2672, 1994.

- [14] W. Gerstner. Rapid phase locking in systems of pulse-coupled oscillators with delays. *Phys. Rev. Lett.*, 76(10):1755–1758, 1996.
- [15] U. Ernst, K. Pawelzik, and T. Geisel. Synchronization induced by temporal delays in pulse-coupled oscillators. *Phys. Rev. Lett.*, 74, 1995.
- [16] U. Ernst, K. Pawelzik, and T. Geisel. Delay-induced multistable synchronization of biological oscillators. *Phys. Rev. E*, 57:2150–2162, 1998.
- [17] P. Ashwini and M. Timme. Unstable attractors: existence and robustness in networks of oscillators with delayed pulse coupling. *Nonlinearity*, 18, 2005.
- [18] H. Broer, K. Efstathiou, and E. Subramanian. Robustness of unstable attractors in arbitrarily sized pulse-coupled networks with delay. *Nonlinearity*, 21:13–49, 2008.
- [19] J. Guckenheimer and P. Holmes. Structurally stable heteroclinic cycles. *Proc. Camb. Phil. Soc.*, 103:189–192, 1988.
- [20] C. M. Postlethwaite and J. H. P. Dawes. Regular and irregular cycling near a heteroclinic network. *Nonlinearity*, 18:1477–1509, July 2005.
- [21] J. H. P. Dawes and T. L. Tsai. Frequency locking and complex dynamics near a periodically forced robust heteroclinic cycle. *Phys. Rev. E*, 74, 2006.
- [22] H. Broer, K. Efstathiou, and E. Subramanian. Heteroclinic cycles between unstable attractors. *Nonlinearity*, 21:1385–1410, 2008.
- [23] P. Ashwin, G. Orosz, and S. Townley. Dynamics on networks of cluster states for globally coupled phase oscillators. *SIAM Journal on Applied Dynamical Systems*, 6:728–758, 2007.
- [24] D. Hansel, G. Mato, and C. Meunier. Clustering and slow switching in globally coupled phase oscillators. *J. Phys. E*, 48:3470–3477, 1993.
- [25] M. Timme, F. Wolf, and T. Geisel. Prevalence of unstable attractors in networks of pulse-coupled oscillators. *Physical Review Letters*, 89, 2002.
- [26] M. Timme, F. Wolf, and T. Geisel. Unstable attractors induce perpetual synchronization and desynchronization. *Chaos*, 13, 2003.
- [27] P. Ashwin and J. Borresen. Encoding via conjugate symmetries of slow oscillations for globally coupled oscillators. *Phys. Rev. E*, 70, 2004.
- [28] G. Orosz, P. Ashwin, J. Wordsworth, and S. Townley. Cluster synchronization, switching and spatiotemporal coding in a phase oscillator network. *Proceedings in Applied Mathematics and Mechanics*, 7:1030703–1030704, 2007.
- [29] K. Kaneko. Dominance of milnor attractors and noise-induced selection in a multiattractor system. *Physical Review Letters*, 78, 1997.

- [30] R. Gütig and H. Sompolinsky. The tempotron: a neuron that learns spike timing-based decisions. *Nature Neuroscience*, 9:420–428, 2006.
- [31] C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton. Dynamics of self-organized delay adaptation. *Phys. Rev. Lett.*, 82:1594–1597, 1999.
- [32] D. J. Amit. *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge University Press, 1992.
- [33] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.
- [34] J. Herbert. The “echo state” approach to analysing and training recurrent neural networks - with an erratum note. 2001.
- [35] G. Laurent. Olfactory network dynamics and the coding of multidimensional signals. *Nature Rev. Neuroscience*, 3:884–895, 2002.

Acknowledgments

First, I would like to thank my advisor Marc Timme for all his support and advice during the last years, without whom this work wouldn't be possible.

Special thanks for Theo Geisel for the fantastic working environment in the MPIDS and for his continuous support of all students in this department.

I would like to thank Lishma Anand, Katja Fiedler, Holger Hennig and Frank Van Bussel for their support and friendship, both invaluable during my work.

I would like to thank Christian Bick and Juan Daniel Flórez Weidinger for fruitful discussions.

Special thanks to my brother Eduardo Schittler Neves and his wife Adriana Turqueti Neves for their support since I arrived in this country, and Jessica Marieluise Poschen for her support since we met each other.

Finally I would like to thank my parents for their unconditional support.

