



Associative Memory Models Based on Coupled Oscillators

Semester Project

24. June 2005

Author : Vlad Trifa

Supervisors : Ludovic Righetti and Jonas Buchli

Professor : Auke Jan Ijspeert

Abstract

Animals have demonstrated an impressive adaptive behavior along evolution, allowing them to survive into hostile and constantly changing environments. Such a level of adaptation is possible because the neuronal structure of the animal is not prenatally determined and it does not remain fixed throughout the entire life of the organism, but is shaped according to experience. A mechanism called *synaptic plasticity* performs such short and long-term modifications of the connections in the brain.

Our goal was to implement and perform a qualitative analysis of an interesting model of associative memory based on coupled oscillators, that is able to learn input signals by adapting the natural frequencies of the oscillators.

In the second part of this project, we describe a new model of memory we have developed by modifying the oscillator we analyzed by adding some interesting features, and we show that this new model is able theoretically to learn any kind of periodic complex signal. Finally, we show that our model is extremely robust when the presented signal is noisy, or when the oscillators are unreliable and can fail.

Our work should not be considered as an attempt to find a universal model able to perform the same tasks as the animal brain with comparable performance, but rather as an attempt to show that it is possible, at least from a mathematical point of view, to learn any periodic complex signal using a network coupled oscillating systems that mutually influence their behavior according to the relationship between their respective phases.

Acknowledgements

First, I'd like to thank very much Prof. Auke Jan Ijspeert for allowing me to make my semester project in his motivating and drilling research group.

Then, I'd like to thank my supervisors Ludovic Righetti and Jonas Buchli for being always at my disposal to help me and to propose pertinent advices when unexpected obstacles prevented the continuation of my journey through the dynamical systems wonderland, and also for showing me the path to the scientific truth, and finally for providing me as much paper, pencils, and rubbers I ever needed. Thanks to Alessandro for always having a nice software mp3ized to entertain my eardrums when I was sad in front of $n \to \infty$ bugs playing hide and seek with me. Also, two thumbs up for all the other people at the BIRG for such a pleasant environment to work in.

Also, I'd like to thanks R. Borisyuk and I. Kazanovich for answering so quickly to the questions I had concerning their article.

Finally, I'd like to thank my parents for all the sacrifices they made for allowing me to study in Switzerland.

Contents

1	Intro	Introduction					
2	Mer	emory literature review					
	2.1	Associ	ative memory models	3			
	2.2	Hopfie	ld Model	6			
		2.2.1	Mathematical formulation	7			
		2.2.2	Performance evaluation	9			
		2.2.3	High capacity associative memories	9			
		2.2.4	Discussion	10			
	2.3	Biological memory formation					
	2.4	Oscilla	tors theory	15			
		2.4.1	Synchronization	18			
	2.5	Learni	ng with oscillators	19			
		2.5.1	Central pattern generators	22			
	2.6	Discus	sion	23			
		alysis of the model 2					
3	Ana	lysis of	the model	25			
3	Ana 3.1	lysis of Descri	the model ption of the model	25 25			
3	Ana 3.1	lysis of Descri 3.1.1	the model ption of the model	25 25 26			
3	Ana 3.1	lysis of Descri 3.1.1 3.1.2	the model ption of the model Network architecture Dynamics of the network	25 25 26 27			
3	Ana 3.1	lysis of Descri 3.1.1 3.1.2 3.1.3	the model ption of the model Network architecture Dynamics of the network Implementation	25 25 26 27 29			
3	Ana 3.1 3.2	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan	the model ption of the model Network architecture Dynamics of the network Implementation nics of the network	25 25 26 27 29 29			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys	the model ption of the model Network architecture Dynamics of the network Implementation nics of the network is of the parameters	25 26 27 29 29 30			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys 3.3.1	the model ption of the model Network architecture Dynamics of the network Implementation nics of the network bis of the parameters Phase differences (ψ_{ij})	 25 26 27 29 30 31 			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys 3.3.1 3.3.2	the modelption of the modelNetwork architectureDynamics of the networkImplementationis of the networkbis of the parametersPhase differences (ψ_{ij}) Basin of attraction (v)	 25 26 27 29 30 31 34 			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys 3.3.1 3.3.2 3.3.3	the model ption of the model Network architecture Dynamics of the network Implementation nics of the network bis of the parameters Phase differences (ψ_{ij}) Basin of attraction (v) Amplitude behavior (β, γ)	 25 26 27 29 30 31 34 34 			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys 3.3.1 3.3.2 3.3.3 3.3.4	the modelption of the modelNetwork architectureDynamics of the networkImplementationincs of the networkbis of the parametersPhase differences (ψ_{ij}) Basin of attraction (v) Amplitude behavior (β, γ) Neighborhood attraction (w)	 25 26 27 29 30 31 34 34 34 			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys 3.3.1 3.3.2 3.3.3 3.3.4 3.3.5	the modelption of the modelNetwork architectureDynamics of the networkImplementationnics of the networksis of the parametersPhase differences (ψ_{ij}) Basin of attraction (v) Amplitude behavior (β, γ) Neighborhood attraction (w) Learning speed (α)	25 26 27 29 30 31 34 34 34 34			
3	Ana 3.1 3.2 3.3	lysis of Descri 3.1.1 3.1.2 3.1.3 Dynan Analys 3.3.1 3.3.2 3.3.3 3.3.4 3.3.5 3.3.6	the modelption of the modelNetwork architectureDynamics of the networkImplementationnics of the networksis of the parametersPhase differences (ψ_{ij}) Basin of attraction (v) Amplitude behavior (β, γ) Neighborhood attraction (w) Learning speed (α) Sigmoid functions (ξ_i, ρ_i)	25 25 26 27 29 30 31 34 34 34 34 34 38			

 3.5.1 Performance analysis 3.5.2 Input signal dimension 3.5.3 Distributiveness of learning 3.5.4 Threshold function 3.5.5 Positive aspects 3.5.6 Negative aspects 4.1 What was done 4.1.1 Complex signal learning 4.1.2 Improve the model 4.1.3 Memory recall 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal 4.2.3 Third experiment: complex signal with close components 4.2.4 Fourth experiment: complex signal 4.2.5 Fifth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 		3.5	Discussion						
 3.5.2 Input signal dimension 3.5.3 Distributiveness of learning 3.5.4 Threshold function 3.5.5 Positive aspects 3.5.6 Negative aspects 4.1 What was done 4.1.1 Complex signal learning 4.1.2 Improve the model 4.1.3 Memory recall 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal 4.2.3 Third experiment: complex signal with close components 4.2.4 Fourth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 			3.5.1	Performance analysis	39				
 3.5.3 Distributiveness of learning			3.5.2	Input signal dimension	42				
 3.5.4 Threshold function			3.5.3	Distributiveness of learning	43				
 3.5.5 Positive aspects			3.5.4	Threshold function	43				
 3.5.6 Negative aspects 4 Further improvements 4.1 What was done 4.1.1 Complex signal learning 4.1.2 Improve the model 4.1.3 Memory recall 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal with close components 4.2.3 Third experiment: complex signal 4.2.4 Fourth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography			3.5.5	Positive aspects	45				
 4 Further improvements What was done Complex signal learning Improve the model Improve the model Memory recall 4.2 Experiments First experiment: simple signal First experiment: simple signal A.2.3 Third experiment: complex signal A.2.4 Fourth experiment: robustness against noisy signal Fifth experiment: robustness against unreliable elements Sixth experiment: robustness against unreliable elements 5 Conclusion Appendix A Bibliography			3.5.6	Negative aspects	45				
 4.1 What was done 4.1.1 Complex signal learning 4.1.2 Improve the model 4.1.3 Memory recall 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal with close components 4.2.3 Third experiment: complex signal 4.2.4 Fourth experiment: robustness against noisy signal 4.2.5 Fifth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography	4	Further improvements 4							
 4.1.1 Complex signal learning		4.1	What	was done	47				
 4.1.2 Improve the model 4.1.3 Memory recall 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal with close components 4.2.3 Third experiment: complex signal 4.2.4 Fourth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography			4.1.1	Complex signal learning	47				
 4.1.3 Memory recall 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal with close components 4.2.3 Third experiment: complex signal 4.2.4 Fourth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements 4.2.6 Sixth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography			4.1.2	Improve the model	49				
 4.2 Experiments 4.2.1 First experiment: simple signal 4.2.2 Second experiment: simple signal with close components 4.2.3 Third experiment: complex signal 4.2.4 Fourth experiment: complex signal with close components 4.2.5 Fifth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography			4.1.3	Memory recall	53				
 4.2.1 First experiment: simple signal		4.2	Experiments						
 4.2.2 Second experiment: simple signal with close components 4.2.3 Third experiment: complex signal			4.2.1	First experiment: simple signal	54				
 4.2.3 Third experiment: complex signal			4.2.2	Second experiment: simple signal with close components .	54				
 4.2.4 Fourth experiment: complex signal with close components 4.2.5 Fifth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements . 4.3 Performance comparison			4.2.3	Third experiment: complex signal	54				
 4.2.5 Fifth experiment: robustness against noisy signal 4.2.6 Sixth experiment: robustness against unreliable elements			4.2.4	Fourth experiment: complex signal with close components	57				
 4.2.6 Sixth experiment: robustness against unreliable elements . 4.3 Performance comparison			4.2.5	Fifth experiment: robustness against noisy signal	57				
 4.3 Performance comparison 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography 			4.2.6	Sixth experiment: robustness against unreliable elements	61				
 4.4 Results and discussions 4.5 Future work 5 Conclusion Appendix A Bibliography 		4.3	Perfor	mance comparison	61				
4.5 Future work		4.4	Results	s and discussions	63				
5 Conclusion Appendix A Bibliography		4.5	Future	work	63				
Appendix A Bibliography	5	Con	clusion		65				
Bibliography	Appendix A								
	71								

Chapter

Introduction

"Thus memory belongs to the faculty of the soul to which imagination belongs; all objects which are imaginable are essentially objects of memory; all those that necessarily involve images are objects of memory incidentally." Aristotle (384-322 B.C.)

Biological systems, in particular animals have demonstrated an impressive ability to adapt to constantly changing environments by finding food, escaping predators, and by reproducing themselves. A fundamental mechanism allowing this level of flexibility is *learning*, which consists of a permanent modification of the brain structure of an organism. Such physical modifications are performed by a phenomena commonly called *synaptic plasticity*, which consists of reinforcing the connections in the brain between concepts that occur together, as it was shown by the experiments of Pavlov on animal conditioning. The idea that memory works by association of concepts is not new, as the first to propose it was Aristotle in the 3^{rd} century before our era. Nowadays, neurophysiological studies have shown that the synaptic connections between neurons that show synchronous activity tend to get strengthened, and that oscillatory activity underlies many cognitive processes in the mammalian brain.

Many formal models that could account for memory formation have been proposed for since the early 80's, but they were usually idealized in that they neglected many of the important aspects of real neurons. It is not until recently that a new family of models based on oscillating systems have been proposed, but this field is still in its infancy. In the first part of this project we analyze one of these models and we discuss about its properties and drawbacks. Then, in the second part we take inspiration of this model in order to derive a new one that is able to learn virtually any kind of complex signal reliably, and we also discuss what is interesting in this new model. Hardware implementations of these memories is a very common application in many fields ranging from hand-written pattern recognition on PDAs, face recognition for physical access to secure sites, or other high-performance computer memories. For that purpose, many research has been performed towards design of analog VSLI or optical chips, in order to implement such memories. In this direction, the models we describe in this project have even more evident and impressive implementation properties as they rely solely on oscillating systems, which is a common concept that can be found in a myriad of biological entities such as neurons, populations of animals and insects, pendulums, chemical reactions, lasers, voltage-controlled oscillators, Josephson junctions, etc.

Outline of this report

Chapter 2 gives an introduction to non-linear dynamics, especially oscillating systems and gives many examples of biological systems where such behavior has been observed. Then we review some associative memory models starting with the well known Hopfield, and then we describe more recent models of learning systems based using sets of coupled oscillators. We will try to give some performance evaluation metrics and discuss about the advantages and drawbacks of these classical models.

Chapter 3 describes the oscillatory model we implemented, and gives a detailed analysis of the behavior and properties of the model. Then a discussion concerning the important issues that could affect the performance of this model is done. Finally, a discussion about the interesting properties and about the drawbacks of this model is proposed.

Chapter 4 is an attempt towards creating a more performant model of associative memory. This model is shown to be able to learn in theory any kind of complex periodic signal, then performance of this model in comparison with the model investigated in Chapter 3 is discussed. In particular, we show that our model of oscillatory associative memory is robust to noisy input signals, and stored data is not affected when some oscillators of the system fail.

Chapter 5 concludes this work and presents future directions where to investigate and what aspects must be wisely studied for further research in neural modeling.

Chapter 2_____

Memory literature review

"He who loves practice without theory is like a seafarer who boards a ship without wheel or compass and doesn't know whether he travels." Leonardo Da Vinci (1452-1519)

We start this chapter with a short introduction to the concepts we think essential in order to understand what is an associative memory, the different models that were devised to account for biological memory formation, and what aspects we think are essential in order to devise more efficient models having a performance similar to the human brain. Then we present a short introduction to the concepts of non-linear dynamics and oscillating systems. Finally, we will discuss what makes biological neurons so efficient compared to classical neural networks, and what are the capital properties one should not neglect when devising such models.

2.1 Associative memory models

Memory in biological systems has been one of the less understood mysteries since the early days of humanity. The ability of animals to retain information in a reliable manner has intrigued the ancient philosophers, who were among the first to wonder how are we able to store information for long periods and then to recall information learned many years before with an astonishing speed. Unfortunately, their explanations were solely qualitative due to the lack of scientific knowledge of their time, and we had to wait until the first bricks of neuroscience were laid by many psychologists and physiologists and the work they performed throughout the 20^{th} century in order to have a more formal description of what happens in the brain. Experiments performed by Pavlov in 1927, showed that animals are able to learn useful information allowing them to survive by associating concepts. These experiments showed us that some neural mechanisms are able to create mappings between action and sensory information, that is animals tend to recreate actions that brings them joyful rewards, and on the contrary, they try to avoid as much as possible actions that inflicts them unpleasant sensations as pain. The neuronal mechanisms allowing such a permanent association has been studied by Donald Hebb (Gerstner, 2002), which proposed that connected neurons that are activated at the same time tend to strengthen the connections linking them. At that time, the most renowned psychological school was the $behaviorism^1$, and many scientists thought that intelligence was simply a complex form of reinforcement learning. This paradigm is effectively able to explain animal behavior, but unfortunately human behavior is not regulated by a fixed set of rules to determine which response corresponds to a given stimulus. Humans do not simply associate the same action to a given sensory input in a deterministic fashion as animals do, because we are endowed with consciousness and each of us has it's own history, feelings, education, knowledge of life, and all these experiences definitely influence the choices we make throughout our life.

It's interesting to note that Aristotle (384-322 B.C.) was the first to introduce the concepts we call the *Classical Laws of Association*. These laws state that mental items such as ideas, perceptions, sensations or feelings can be linked in memory under the following conditions :

- If they occur simultaneously ("spatial contact")
- If they occur in close succession ("temporal contact")
- If they are similar
- If they are contrary

However, such simple mathematical models are far from explaining the real processes involved into such a complex system as is the human brain, because they neglect a lot of details concerning many aspects of the plethora of interacting neural processing units bathing in a soup of hormones, neurotransmitters and other chemicals. Nonetheless, these primitive models provided to computer science a very simple and high-level abstraction of the basic cognitive abilities found in animals that can be easily implemented on a computer due to their formal mathematical modeling. Despite the fact that Hebb's conjecture on synaptic plasticity is more than 50 years old, it still remained one of the more viable and productive theoretical framework.

¹Behaviorism is a movement in psychology and philosophy, which denies importance of mind. One of the assumptions of behaviorist thought is that free will is illusory, and that all behavior is determined by the environment either through association or reinforcement.



Figure 2.1: Task of an associative memory. Noisy input (1) is presented and an unique location in the state space of the network corresponds to it. Then the dynamics of the network will move the state towards the nearest local minima (3 in this case), as when when a gradient system energy through diffusion and falls into a potential well.

In the classical memories used in computers, data is stored at a specific location designated by an unique address and thus can be accessed only if the address of the desired item is known. On the contrary, Associative Memory (AM) we will talk about throughout this project, refers to so-called *content*addressable memory in which data is retrieved directly upon presentation of sufficient partial information about data we look for. More formally, the basic task of an associative memory is to store a set of p patterns $\vec{\xi}^{\mu}$, with $1 \le \mu \le p$ the index of the pattern, in such a way that when presented with a new pattern ζ , the network responds by producing whichever one of the stored patterns most closely resembles to ζ . To be more precise, we should use the term *auto-associative* memory because the input used for retrieval is of the same nature as the stored data in the sense that the key is the input pattern. The interesting property of this kind of memory is that the data can be retrieved even when partial or noisy information is presented as input, thus enabling a certain fault-tolerance, a property found in any natural system. This retrieval methodology requires the introduction of a metric assessing the

similarity between two patterns, and this measure will depend entirely on the characteristics of the network under consideration. For example, if the patterns are binary, we usually use the *Hamming distance*, that is the number of different bits between the two patterns, formally the number of 1's in the result of the XOR operation between the patterns.

2.2 Hopfield Model

The model described in (Hopfield, 1982), was among the first formal model describing an associative memory. When this article was published in 1982, it had the effect of a bomb in the young field of neural networks. The effervescence around that model was not exactly due to its novelty, as actually many concepts it contained were already well known at that time and for this reason many scientist criticized his work (see for example (Anderson and Rosenfeld, 1988)). Nonetheless, the way these ideas were combined together to form an efficient and easily implementable memory model was totally new, and for that reason this paper deserved the noise he made. One of its major improvement was that he took inspiration from statistical mechanics and introduced the concepts used in this field of physics were well known from a mathematical point of view.

There are classes of physical systems that present interesting properties that can be used to form a model of a general, dynamic, and error-correcting content-addressable memory. Imagine a physical system whose time evolution can be interpreted in an *n*-dimensional coordinate system, where each point in this state space (either discrete or continuous) represents the instantaneous condition (state) of the whole system (see for example Figure 2.1). Motion of a particle in such a system is described by a set of dynamical equations and can be represented by a flow in the state space. The behavior desired in memory models requires a particular kind of motion where at any place in the state space there exist a single flow that is directed towards the nearest $attractor^2$, and which corresponds to a pattern in the memory. The trivial globally stable and unique attractor case becomes clearly non-deterministic under high stochasticity when multiple stable points are present, each with its own basin of attraction³. If we generalize this idea, we could say that any system governed by dynamical motion in a state space dominated by a substantial number of stable states, can be regarded as a general content-addressable

 $^{^{2}}$ A locally stable point, where the flux is equal to 0 in all directions, so the state of the networks will stabilize at that place. Formally, it corresponds to a local minima of the energy function E (see Equation 2.6).

 $^{^{3}}$ The set of initial conditions where the state will converge unambigously towards the stable state.

memory. The interesting point resides in the possibility of turning any prescribed set of information to memorize into the stable states of the system, and this exactly the role of a learning function. The processing units used in this model⁴ are simple binary units similar to those used by McCullough and Pitts (see (Gerstner, 2002; Haykin, 1999)) with two states : $x_i = -1$ (not firing) and $x_i = 1$ (firing). This can seem quite similar to the Perceptron, but there is an essential difference in that the Perceptron was designed to work in a feed-forward manner. On the contrary, the full recurrent connections used in Hopfield networks is exactly what endows this model with such interesting emergent computational properties. Moreover, the Perceptron uses a synchronous update function, which has clearly not been observed at a global level in any animal brain, as the existence of neural delays prevents such kind of perfect and global synchrony to occur. This last statement, should be handled with care as the exact properties of brain dynamics are not yet known, and many works show that sub-millisecond precision between distant neural assemblies can be attained.

2.2.1 Mathematical formulation

Any pattern one might want to store, for example an image, is divided into pixels. Each pixel *i* has a value $x_i = \pm 1$ with $1 \leq N$ where *N* is the number of pixels to store, and so we can define an image as the vector *x*. The memorized patterns are denoted $\vec{\xi}^{\mu}$ with $1 \leq \mu \leq P$ with *P* the total number of stored patterns. An associative memory is then constructed using a neuron for each pixel.

We can now define an overlap metric between the current image x (at time t) and a stored pattern μ as follows

$$m^{\mu}(t) = \frac{1}{N} \sum_{i=1}^{N} \xi_{i}^{\mu} x_{i}(t)$$
(2.1)

We can see that $\xi_i^{\mu} x_i(t)$ is equal to 1 if $\xi_i^{\mu} = x_i(t)$, else it is equal to -1. This implies that $m^{\mu}(t) = 1$ if and only if all pixels are identical, meaning that the overlap measures the degree of similarity. An algorithm able to retrieve the most similar pattern ξ^{μ} upon presentation of an input image (that is ξ^{μ} such as $m^{\mu}(t)$ is maximal) is easily implementable with a computer, but we can also resolve this task using a set of parallel processors (neurons) with a distributed memory (connections between neurons) and an implicit storage and retrieval algorithm embedded into the dynamic interactions between these neurons, and this is exactly what the Hopfield model does.

⁴We assume to discuss only about the discrete Hopfield only.

The neuron dynamics are described in discrete time and the value of a pixel x_i at the time $t + \Delta t$ is given by a sigmoid function of the potential of this neuron at the time t as follows

$$x_i(t + \Delta t) = g[h_i(t)] = \tanh[\beta \cdot h_i(t)]$$
(2.2)

where β is a positive parameter regulating the shape of the sigmoid and $h_i(t)$ is the potential at time t defined as follows

$$h_i(t) = \sum_{k=1}^{N} w_{ik} x_k(t)$$
(2.3)

where w_{ik} is the connexion strength between neuron *i* and *k* defined as follows

$$w_{ik} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu}(t) \xi_k^{\mu}(t)$$
(2.4)

It is interesting to note that this factor can be seen as the result of the learning of the network. The initial weights are set to $w_{ik} = 0$ and the following learning rule (similar to Hebb's rule) is applied incrementally for each pattern μ one wants to store :

$$\Delta w_{ik} = \frac{1}{N} \xi_i^{\mu}(t) \xi_k^{\mu}(t)$$
 (2.5)

Now we arrive at the hot spot we wanted to reach through the description of the Hopfield model. The most important aspect of the article, is that Hopfield described an *energy function* E for the his network, that is a function which returns a scalar value for each possible state S of the network

$$E = -\frac{1}{2} \sum_{i} \sum_{j} w_{ij} S_i S_j \tag{2.6}$$

As long as this function is bounded and monotonically decreasing with each update of the state, the network has a limited number of local point attractors. This is what ensures a convergence towards a stable state. This is somewhat similar to a physical system that seeks to minimize its potential energy. By ensuring that the weight matrix after learning is symmetric and was constructed using an asynchronous update function, one can guarantee (see for instance (Davey *et al.*, 2004)) that such stable states do exist, and for any initial state the nearest stable state will always be found. Although these conditions are not necessary in order to make this model practically useful, and even if these conditions were satisfied, one cannot always guarantee that all attractors in the network are fundamental memories : the inverse of each stable state is also stable. The concept of energy function is similar to the Lyapunov functions that are commonly used in control and optimization theory, which states that if one is able to find a continuously differentiable function V(x), such as $V(x_s) = 0$ for a specific x_s , and $\forall x \in D - x_s : V(x) > 0$ and $\dot{V}(x) < 0$, then x_s is an asymptotically stable point, and for any starting point into the neighborhood D^5 of x_s the network state will converge towards that stable point. Unfortunately, no methodology has yet been found in order to find such a function for any system, and in practice it is even very difficult to do so, except for very simple cases. Please note carefully that the importance of such an energy function is that it guarantees mathematically that for any starting point, the network state always converges towards an stable state (a local minima of the energy function), and this is what makes associative memory models useful.

2.2.2 Performance evaluation

As engineers, we might be interested into finding an objective criteria that can quantify the efficiency of a given model. In the case of a memory, a relevant performance metric is the *capacity* C, that is the maximal number of fundamental memories (patterns) we can store within the memory. The loading α of a network, measures the size of the training set⁶ (p), relative to the number N of processing elements in the network and is defined as $\alpha = p/N$. There is a maximal value of α above which some vectors of the training set will not be memorized correctly, and we call this value α_{max} as the maximal *loading* of the network. For the standard Hopfield model, if a small amount of error is allowed the capacity is approximately $C \approx 0.14N$, thus $\alpha_{max} \approx 0.14$. If no error is admitted, the capacity is reduced and proportional to $\frac{1}{\log N}$. This value can be different for each model, so the higher the α_{max} , the larger the amount of information that can be stored with the model under consideration. Another criterion for practical implementation is the maximal error rate one might encounter, which is commonly used to measure the reliability of the network.

2.2.3 High capacity associative memories

Hopfield neural networks are well known from a theoretical point of view, and allows one to improve the efficiency of the model. The work presented in (Davey *et al.*, 2004), is an interesting analysis attempting to evaluate quantitatively the performance of the Hopfield models. This work shows that the weakness of the standard Hopfield model performance lies in the hebbian 'one-shot'⁷ learning rule proposed, which makes resulting networks have low capacity and poor attractor performance. The conclusion is that the perfor-

⁵D in this case is the basin of attraction of x_s .

⁶This is the set of patterns that were stored in the memory.

⁷One-shot means that the learning rule is asynchronous and learns serially each pattern.

mance of the standard model can be drastically improved solely by selecting a more elaborated (as compared to the basic one-shot hebbian learning given in Equation 2.5) learning rule at the cost of an increased computational complexity.

2.2.4 Discussion

Let us first discuss the benefits and limitations of the original Hopfield model. Conceptually, this model has been an important landmark in the neural networks history, but it failed to continue on the physiological path entailed by the hebbian theory on a biologically plausible learning mechanism. The problem of the pioneering neural network models, as for example the McCullough-Pitts and Hopfield (see (Gerstner, 2002)), is that they are unable to cope with the highly complex dynamical behavior shown by real biological neurons, as for example neuronal and synaptic plasticity, structured architectures, and biological features of spatiotemporal coding as stated in (Sommer and Wennekers, 1999), whereas this is a capital aspect of the biological networks. For that reason, more extended models have been developed, as networks of *Integrate-and-fire* neurons, able to accumulate pre-synaptic activity by increasing their membrane potential in a non-linear fashion, and then generate an action potential (or more usually a *spike*) when this potential exceeds a certain threshold.

However, this is definitely not a reason to under-estimate the usefulness of associative memories derived from this model. In particular, Hopfield-like models have been widely used by many researchers from various fields, for example by neuroscientists in their attempt to explain both normal and abnormal human memory, including phenomena such as unlearning, schizophrenia and tremors caused by dementia in Alzheimer's patients (more details can be found in (Davey *et al.*, 2004)). The lesson to draw from physiological data and from bio-inspiration is that for a learning rule to be biologically plausible, weights should be adjusted by a mechanism that relies largely on purely local information, else we would assume that learning in biological systems is governed by a global supervisor ⁸, which is clearly not the case.

We can conclude by factoring out what are the important properties an efficient implementation of an associative memory should possess:

• Real neurons are not binary units, and so should be the computational units used in our model.

 $^{^{8}\}mathrm{In}$ classical literature about Artificial Intelligence, such a hero is called generally a ho-munculus.

- A regular connection topology, not all-to-all connections, but rather a high connectivity degree in local neuronal assemblies with a low connectivity level for long range connections across assemblies.
- A simple method to determine whether the network reached a stable state, and if the state is a fundamental memory.
- We must be able to add new memories without forgetting the old ones, and we should not need to retrain the whole network as new memories are added.
- A fast, local, and dynamic learning rule that must mainly rely on information that is local in time and space.
- Information can be recalled even when noisy input cues are presented to the network.
- Parameters and the time to learn new stimuli and to recall it should be linear with the number of computational units N in the network, that is the complexity of interactions should be of order $\mathcal{O}(N)$ in order to make the system scalable.

2.3 Biological memory formation

Recent investigations shows that there does not exist a specific organ dedicated to memory, but on the contrary information retainment is rather distributed throughout the whole brain. However, the hippocampus (see Figure 2.2) has been identified as a central structure playing an important role in many information processing mechanisms. As stated in (Borisyuk *et al.*, 2001), spatiotemporal coherence of activity in the hippocampus can provide basic mechanisms for the wide range of functionalities observed in the hippocampus such as short and long-term memory, attention focusing, detection and storage of new stimuli. The hippocampus is also very important in navigation as it serves as a space map. Experiments done with rats (see the work by O'Keefe and Recce (1993) referenced in (Borisyuk *et al.*, 2001)) demonstrated that animals are able to create internal representation of the environment and memorize a path in the environment. Moreover, they can plan an alternative path to the goal if the known one is blocked. Once the hippocampus is damaged, we denote that the ability to form such mental maps is lost.

Each slice of the hippocampus contains three main regions called the *den*tate gyrus, the CA_1 field and the CA_3 field. These regions contain both excitatory pyramidal neurons and inhibitory interneurons that are extensively connected in a highly organized manner along the septotemporal axis. It is stated in (Borisyuk and Hoppensteadt, 1999) that this structure is able to



Figure 2.2: *Left:* Location of the hippocampus in the human brain. *Right:* A slice of the hippocampus with the important structures it contains.

support endogenous oscillatory activity, that is without any external periodic input. The frequencies of these oscillation have been identified being in the range of *gamma rhythms (40-70 Hz*). The hippocampus receives two primary inputs at low frequency in the range of theta rhythms (3-9 Hz), that are currents injected from afferents of the entorhinal cortex, and of the medial septum. The dynamics of the network are regulated by the phase relationships between these signals. The CA_3 cells outputs to the lateral septum and complete thus a feedback loop which controls the theta rhythms generated by the medial septum. Finally, many experiments show that these inputs are propagated throughout the hippocampus during a certain amount of time $\tau = 25 - 50$ ms, due to delays introduced by synaptic transmissions, and this can lead to different phase shifts for the same input depending on its location along the hippocampus. This complex wiring and the induced interaction of theta rhythms can result in highly complex dynamics, including synchronization or chaotic behavior. The first experimental *electroencephalogram (EEG)* recordings in the 30's, have already shown some kind of rhythmic activity, but nowadays it is commonly agreed that such an oscillatory activity in the brain is a crucial component of many cognitive process such as movement generation, visual awareness, sensory perception and even consciousness. This is very important in that the oscillatory activity does not convey information upon the stimulus per se, but is rather the tool that can establish synchrony over large distances and different neurons groups.

When a new stimulus is presented to the network, the activity of the hippocampus increases and becomes stable. This activity remains and decreases slowly after the stimulus is switched off, and this reaction is said to be *tonic*. After a sufficient number of presentations, the stimulus becomes familiar, and in this case the activity is brief and decays quickly when the stimulation is halted, and we denote this reaction as *phasic*. This phasic reaction is important in that it selects what information needs to be stored, and thus avoid to treat data that is already known.

The question of how information is carried and processed into the brain has always been central in neuroscience, and scientists have been convinced during a long period that the information was encoded only by using *frequency codes*⁹, as it has been experimentally shown that the firing rate of sensory neurons encodes directly the intensity of a stimulus (see (Gerstner and Kistler, 2002)). However, now it is commonly agreed that most of information is rather encoded by *position*¹⁰ and *temporal*¹¹ codes. Position (or assembly) coding is a very plausible hypothesis that could account for the impressive performance of the visual memory. For example face recognition is extremely efficient and no computer is able to compete with a human yet. The idea that a single object is represented by a single neuron is not intuitive because

- The combinatorial complexity of images would require an immense number of neurons, one for each possible combination of color, shape, projection, even for a simple object.
- There should be a vast reservoir of unallocated and unused cells necessary in order to learn new concepts.
- The brain would not be able to cope with such a high degree of fault tolerance, in that loss of a part of the brain would result of an immediate loss of all concepts stored in the necrosed gray matter.

To give an explanation of how does the visual cortex process and store information so efficiently, the *Gestalt*¹² principle has been proposed at beginning of the century by the German psychologists. Due to the columnar organization of cortical areas, neurons more sensitive to the same features (color, depth, etc) or responding for adjacent points of the visual field are often grouped in spatially distributed cell assemblies. This is called the *retinotopic* organization of the visual cortex. A particular visual stimuli activates a large number of

⁹Frequency code: the information is contained in the firing rate (frequency) of neurons.

¹⁰Position code: each aspect of metal processing takes place into a specific cortical area. ¹¹Temporal coding: the information is contained in the exact time of spiking.

¹²From german *gestellt*=put together. The idea that a "whole" is greater than the sum of

its parts. Associative theorists broke down the "whole" and emphasized the analysis of the constituent parts of the mind.

such spatially distributed cell assemblies, each of which encode only a partial aspect of an object (a feature). They proposed that such disjoint representations of the relevant features composing an image are reassembled at lower levels in order to give us the global structure rather than the composing parts. The question raised, commonly called the *binding problem*, is how are these distributed activities reintegrated in order to generate a representation of an object as a whole? To answer this question, the second coding method we discussed above, namely the temporal coding, comes into play. This type of coding emphasizes the fact that fine temporal structure of spike trains contains relevant information. Recent works have shown that synchronization of activity can be achieved with a precision in the millisecond range, and synchronization can be established over brief intervals, leading to rapid temporal succession of synchronization in different assemblies. This precise times of spiking allows synchronization of the neural activity across different areas of the brain to occur. As hypothesized by the work of (Singer, 1993; Singer and Gray, 1995), the synchronization of oscillatory activity of distributed and feature selective cells might be a way to establish relations between the different features of a visual stimuli. In other words, all neurons that belong to the same assembly should synchronize their activity, and all groups having such internal synchrony represent the different features that compose the object looked at. The advantage is that several assemblies can be active simultaneously in the same cortical area without risks of confusion, as each assembly has its own rhythm. This concept of binding by synchrony is an interesting explanation of the processes underlying attention (von der Malsburg, 1995) or consciousness (Crick and Koch, 1990), as it provides new solutions to problems that are difficult to solve with position and rate codes only.

It's also interesting to understand how does this oscillatory activity arise and how does the coupling connections between the cell assemblies be formed. It's important to note that there exist genetically determined features of neural architecture, such as the laminar structure commonly found in the cortices, which is phylogenetically preserved and determine the functional properties of the network. Nonetheless, the architecture is not prenatally fixed, but is rather extensively modified through epigenetic modifications, which implies an experience-dependent associative learning rule. Recent investigations suggest (see (Gerstner and Kistler, 2002; Singer, 1993)) that synaptic connections are strengthened (*Long-term potentiation (LTP)*) or weakened (*Long-term depression (LTD)*) according to the relative timing between the presynaptic and the postsynanptic spike. This clearly supports the idea that the connections are formed according to the correlation between the activity of different assemblies.

The reason we summarized here these aspects of mental processing is to show the reader that synchronization of the oscillatory activity is a very efficient method able to solve complex problems, and is not only crucial in nervous signal transmission, but also for synaptic plasticity. Additionally, it is pointed out in (Buszáki and Draguhn, 2004) that oscillation-based synchrony is the most energy-efficient physical mechanism for temporal coordination. We want to show that efficient models of memory can be implemented using the inherent properties of oscillating systems at a low computational cost, and binding mechanism using temporal signal correlation is undemanding in terms of structural requirements and consequently ubiquitous and extremely flexible. Thus idea to form models where the computing units are coupled oscillators and where learning is performed through adaptation of the frequencies of the oscillators rose at the end of the 70's, and some models were proposed in the following decade. An evident advantage over Hopfield model is that memory formation in a network of N oscillators posses N parameters (the natural frequencies of each oscillator), while a Hopfield network has $\frac{1}{2}N^2$, parameters which are the connection weights.

2.4 Oscillators theory

First, we will introduce the mathematical concepts that are used for the analysis of dynamical systems (DS), and then we are going to focus on a specific subset of dynamical systems, called oscillators. The goal is to find mathematical models able to describe the evolution of a system in time, and to understand how the evolution can be affected when the parameters regulating the system are changed. This field of mathematics has recently been extensively studied and there exist a vast literature on this topic. Further details and good bibliographical references can be found in (Strogatz, 1994), (Murray, 2002), and (Pikovsky *et al.*, 2001). Many definitions and examples we give in throughout this section are taken from these books.

A dynamical system is a set of possibly interacting elements that evolves with time. As explained in any good physics book, Newton was the first to try to give a scientific explanation to these phenomena largely present in nature and laid the basis of the discipline called dynamics. There exist a subclass of these systems, the one which is of interest here, that shows a particular kind of behavior in that the dynamics show a recurrent pattern of time evolution that is repeated at regular intervals. These systems are said to be *periodic* or cyclic. Formally a value that repeats the same pattern again and again can be seen as $\forall n : f(t) = f(t + nT)$, where f(a) is the value of the variable at time a (for example the quantity of wolves in a given ecosystem), $n \in \mathbb{N}$ and T is the period of the system. One should not underestimate the importance of this class of systems, as many natural phenomenon are subject to periodic motion. To name just a few, animal populations, chemical reactions, electronic circuits, neurons, hormonal regulations, respiration and heart beats, circadian rhythms, and many other are all oscillating systems in their own rhythms. In some cases, the rhythmic behavior is produced by an internal source of energy and the system continues to oscillate autonomously even when isolated, in which case the oscillator is said to be *self-sustained*.

However, is is not enough to know the value of the system at a given time only, as we are much more interested in discover how the systems will evolve starting from a specific state, and for that we need additional variables to determine the state of the system unambiguously. Fortunately, in many cases two variables are often sufficient for this purpose. As an illustration, we present now a well-known population dynamics model dealing with predator-prey system called the *Lotka-Volterra*. We need to know the quantity of predators at time t, denoted P(t) and the quantity of preys the same time, let's say N(t). When many predators are present, they eat the preys. In the absence of preys, the predator population decays exponentially, and this allows the prey population to grow again with low predation.

The dynamics of this system evolve as follows

$$\dot{N} = aN - bNP \tag{2.7}$$

$$P = cNP - dP \tag{2.8}$$

where a, b, c and d are positive constants, a is the growth rate of preys, while d is the predators death rate. Note that predators' growth rate (cNP) is proportional to the size of both populations, as well as preys' death rate bNP. Then, we can nondimensionalise the system by setting

$$u(\tau) = \frac{cN}{d}, \qquad v(\tau) = \frac{bP}{a}, \qquad \tau = at, \qquad \alpha = \frac{d}{a}$$
 (2.9)

and the system becomes

$$\dot{u} = u(1-v)$$
 (2.10)

$$\dot{v} = \alpha v(u-1) \tag{2.11}$$

This cyclic motion can be easily visualized as a point flowing on a circle (see Figure 2.3, right), whose position can be represented as the angle between the point and the horizontal (denoted the *phase* θ) and the distance from the center of the circle (denoted the *radius* r). Such periodic systems can be seen as closed trajectories when they are plotted in the phase plane. It has been shown (Pikovsky *et al.*, 2001) that almost any periodic system can be reduced to two variables namely the *phase*, denoted θ and the *amplitude* or *radius* denoted r. Actually, it is much easier to compute and analyze a dynamical behavior using this polar coordinate system. For example, the simplest case is where the phase grows constantly:

$$\dot{\theta} = \omega \tag{2.12}$$



Figure 2.3: Dynamics of the Lotka-Volterra model. *Left:* population dynamics in the predator-prey system with initial values *top:* u, v = 0.8, *middle:* u, v = 0.5, *bottom:* u, v = 0.3. The solid line is the prey population and the dashed one is the predators population. *Right:* Limit cycles in the phase plane for the three initial conditions on the left.

Now we want to know how the phase changes along time, so after integration we get the solution of this system, that is $\theta(t) = \omega t + \theta_0$. As you can deduce represents uniform motion along a circle with a constant angular frequency ω . The periodicity can be seen when the phase is increased by 2π , that means we performed a turn along the circle. In this case the phase corresponds to the angle, but this statement is not true anymore if the limit cycle is not a perfect circle. For example, the ACPO model described in (Buchli and Ijspeert, 2004) the phase evolution is described as follows :

$$\dot{r} = -g(r - r_0) \tag{2.13}$$

We can clearly see that this equation admits a single fixed point (find by resolving $\dot{r} = 0$) being r_0 , and g is a parameter called the *damping factor* which regulates the way the amplitude returns to r_0 when a perturbation drives the r value away of the limit cycle. A *perturbation* can be described by a displacement of the point away from the limit cycle due to external influence, and we want to know the evolution of the disturbance. In this case (Equation 2.13), it is clear that the perturbation of the amplitude decays back to the limit cycle, while a change in phase induced by a external force neither grows, nor decays. The consequence of this fact is that the phase can be easily adjusted by an external influence allowing the oscillator to synchronize (see next section), and this is the reason why coupled oscillators are so interesting.

If the phase variation rate is not constant anymore, we face a much more

interesting model belonging to the category of *nonlinear oscillators*:

$$\dot{\theta} = \omega - a\sin\theta \tag{2.14}$$

The nonlinearity shown by this model arises from the $\sin \theta$ term, which makes theoretical analysis difficult to perform as it is not easy to derive a function f(t) such as $\theta(t) = f(t)$. The class of non-linear oscillators is the one all scientists prefer (or hate), as this kind of oscillator underlies the behavior of almost all natural systems we discussed so far. Many models have been devised using non-linear oscillators in order to account for dynamics of the membrane potential of neurons (Moris-Lecar, Fitzhugh-Nagumo, Hodgkin-Huxley, see (Gerstner and Kistler, 2002)), for electrical circuits (Van der Pol, which was used in early radios, see (Strogatz, 1994)), or even a simple pendulum.

2.4.1 Synchronization

A very interesting property of oscillators resides in the fact that when they interact together in a way or another, they are able to influence the activity of the others, and eventually lead to *synchronization* between their phases. Synchronization can be understood as an adjustment of rhythms of oscillating objects due to their *weak interaction*¹³. This phenomenon has been observed for the first time in 1658 by the famous Dutch scientist Huygens who noticed that two clocks hanging on the same wall synchronize the motion of their pendulae due to the weak interaction through the vibration of the wall. Nowadays, this property of oscillating objects has been observed in a myriad of natural systems: millions of fireflies are able to synchronize and act as a single homogeneous giant light bulb, girls living together in a college synchronize their menstrual cycles, applauding tends to be rhythmic, etc. This property is capital in our case as this is what allows us to memorize data with oscillators as we will explain very soon.

Frequency locking

It has been shown that two coupled oscillating systems having their own frequencies f_1 and f_2 are able to synchronize, even if their interaction is weak. That is they can adjust their rhythms and start to oscillate with a common frequency f, with $f_1 < f < f_2$. Appearance of this phenomenon, called *frequency entrainment* or *locking*, depends on two factors :

• Frequency detuning: that is the difference between their respective frequencies, *i.e.* $|f_2 - f_1|$. The smaller the difference, the easier to synchronize the oscillators, that is the weaker must be the interaction strength required synchronize them.

 $^{^{13}}$ Weak coupling in the context of interacting oscillators, means that the influence of an oscillator is low compared to its own autonomous motion.

• **Coupling strength:** this the strength of the influence of one oscillator upon the other. The greater the detuning, the stronger must be the coupling strength in order to allow oscillators to synchronize.

Phase-locking

A very simple model of phase synchronization can be found in (Strogatz, 1994), and is briefly explained here in order to give the reader a quick mathematical model of how this behavior is possible. Imagine a global stimulus, that is a periodic and uniform signal whose phase obeys the following equation

$$\dot{\Theta} = \Omega \tag{2.15}$$

Now imagine a non-uniform oscillator who tries to synchronize with this global signal, in that he accelerates if the stimulus is ahead in the cycle and slows down in the other case. This behavior is easily modeled as follows

$$\dot{\theta} = \omega + A\sin(\Theta - \theta) \tag{2.16}$$

where A is a positive parameter we call the *resetting strength*, and measures the oscillators ability to modify its instantaneous frequency. Now to see if entrainment can occur, we must look at the phase difference $\phi = \Theta - \theta$ dynamics and analyze it's stability. This is done by subtracting (2.16) from (2.15), which yields

$$\dot{\phi} = \frac{\mathrm{d}\Theta}{\mathrm{d}t} - \frac{\mathrm{d}\theta}{\mathrm{d}t} = \Omega - \omega - A\sin\phi \qquad (2.17)$$

if we divide 2.17 by A, we can derive the dimensionless equation

$$\frac{\mathrm{d}\phi}{\mathrm{d}At} = \phi' = \frac{\Omega - \omega}{A} - \sin\phi = \mu - \sin\phi \qquad (2.18)$$

This equation admits a fixed point, *i.e.* $\phi' = 0$, only in case when $-1 \le \mu \le 1$, that is synchronization can occur only if $\omega - A \le \Omega \le \omega + A$. This interval is called the *range of entrainment*, or *basin of attraction*.

2.5 Learning with oscillators

Having briefly introduced the important concepts needed to understand how do oscillators work, we are now able to describe how these systems can be used in order to store information. As the phase of a self-sustained oscillator is free and can be easily adjusted, when many of them are coupled together, the whole system will evolve towards different synchronized states with specific phase relationships among them, and these phase patterns can used to store information.



Figure 2.4: Typical phase and frequency locking. *Top:* Frequency-locking : 15 oscillators with different initial natural frequencies converge towards a single frequency. *Bottom:* Phase-locking of the oscillatory activity of these oscillators, they all will synchronize and evolve with a unique common phase.

Connections in neural networks are said to be *weak*. This assumption emanates from the fact that a presynaptic potential is rather small (approx. 1 mV) in comparison to the change in potential induced by a spike (approx. 100 mV). A very interesting model can be found in (Hoppensteadt and Izhikevich, 2001), where a model of such weakly connected neural network is proposed and described as follows

$$\dot{\mathbf{x}}_i = f(\mathbf{x}_i, \lambda_i) + \epsilon \sum_{j=1}^n g_{ij}(\mathbf{x}_i, \mathbf{x}_j)$$
(2.19)

where each vector \mathbf{x}_i describes the state of the *i*-th neuron (membrane potential, ions channels states, and other electrophysiological aspects). the function f describes dynamics depending on the bio-physical parameters (λ_i) of the neuron. The coupling is described by the function g_{ij} , and the parameter $\epsilon \ll 1$, reflects the weak coupling between them. The authors show how this system can be turned into the following canonical form

$$\dot{y}_i = r_i - y_i^3 + \sum_{j=1}^n s_{ij} y_j$$
 (2.20)

This canonical model is much simpler, as its form does not depend on the functions f, g_{ij} , and on the value of λ_{ij} . The authors also show that the Cohen-Grossberg-Hopfield Convergence Theorem states that if the connection matrix S (composed of the elements s_{ij}) is symmetric, then the canonical system is a gradient one, meaning that for each value corresponds an energy level. This result is very important because it implies that an energy function could be found, and under some conditions the network will always converge to an oscillatory phase-locked pattern. There can be many such pattern each corresponding to memorized information, meaning that we have an oscillatory associative memory. The key difference between the Hopfield and the oscillatory network is that memorized images correspond to equilibrium points (attractors) in the former and to limit cycle attractors in the latter. The authors of (Hoppensteadt and Izhikevich, 2001) suggest that as the form of the functions f and g_{ij} in Equation 2.19 were not specified, one could claim that anything that oscillates can also be used for computing, the only problem is to find how to couple the oscillators. The key is that the canonical models can provide a lot of information about the universal neuro-computational properties underlying neural systems, regardless of the details of the equation describing the models.

Most of oscillatory associative memory models rely on a renowned phaselocking model proposed by Kuramoto in 1984 (see (Nishikawa *et al.*, 2004; Seliger *et al.*, 2002)). These models usually consist of interacting oscillators, where memorized patterns are stored as phase-locked oscillations. Such a simple model for excitatory coupling between a set of oscillators describes the phase variation θ_i of the *i*-th oscillator as being

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^n \Gamma_{ij}(\theta_j - \theta_i) = \omega_i + \sum_{j=1}^n C_{ij}\sin(\theta_j - \theta_i + \psi_{ij})$$
(2.21)

where C_{ij} is the coupling coefficient and ψ_{ij} the synaptic phase delay between oscillator *i* and *j*. A careful reader should note the similarity between this generalized model and the simple one proposed in Equation 2.16. The interesting aspect is that a theorem given in (Hoppensteadt and Izhikevich, 2001) states that if a set of oscillators having equal frequencies and if the coupling function Γ_{ij} has a pairwise odd form, *i.e.* $\Gamma_{ij}(-\psi) = -\Gamma_{ji}(\psi)$, then the phase model 2.21 converges to a phase-locked pattern. This property is what allows this model to be used as an associative memory. This simple and intuitive model has been often used to design models of oscillatory associative memories (see (Nishikawa *et al.*, 2004)), but usually error-free retrieval is usually unstable, and such memories have a lower capacity compared to the Hopfield model. Recent work in (Nishikawa et al., 2004) suggest a more complex form for the term $\Gamma_{ij}(\theta_j - \theta_i)$, such as the capacity of the network becomes $2\epsilon^2/\log(n)$. Note that the capacity also scale with the number of processing units n, but it can be increased with ϵ . The problem is that when ϵ is increased too much, the solutions encoding patterns other than the stored ones can also become stable, thus making the system unable to distinguish memory patterns from non-memorized solutions. A rigorous proof is also given, and it shows that in the range $\epsilon \in [\epsilon_1; \epsilon_2]$, the optimal performance is achieved : no error occurs in the retrieval process and no solution other than the memorized patterns is stable. It is clear that in this case, the size of the basins of attraction of the encoded patterns is maximal. This article is very interesting in our work as it links the classical associative memory (e.g. Hopfield) with the oscillatory associative memory models, and it shows that with a simple modification of the classical Kuramoto model (Equation 2.21), oscillatory associative memories based on phase-locking with a hebbian connection scheme are capable of performing as well as the Hopfield network.

2.5.1 Central pattern generators

Movement production in animals has been shown as being performed by a Central Pattern Generator (CPG), a neuronal network capable of generating a rhythmic pattern of motor activity even in absence of sensory input. These kind of systems have been identified in many animals as being the basis that generate movement gaits as walking, swimming, trotting, flying, etc. The interesting part is that the oscillatory activity of CPGs can be modulated by different parameters, which are in turn modulated through modification of sensory input. This means that the same network can be used for multiple gaits, and switching from one to the other occurs when the sensory landscape changes as well, for example a salamander swimming in water is able to switch to walking as it gets out on sand. At the neuronal level, the oscillatory mechanism can be induced by alternated mutual inhibition between a pair of neurons, and other time-dependent processes such as synaptic depression and differences in time courses of synaptic activity along parallel pathways, rate recovery from inhibition (for more details, please refer to (Kandel, 2000)). This area has been widely studied by biologists from an analytical point of view by biologists, as well as by roboticists who attempted to design artificial CPGs that can be used to control locomotion in robots, as for example in (Buchli and Ijspeert, 2004; Righetti et al., 2005). The use of CPGs to generate movement in robots is a very promitting direction because they are robust against perturbations and faults, as they rely on limit-cycle oscillators. The use of CPGs could in the long run even replace well-known robot control paradigms such as behavior-based robotics or schema control as they are able to behave differently according to the sensory information from peripheral receptors (e.g. camera, laser range-finders, etc.) or other high-level control centers (eg. path planner), so a central control center is not required anymore.

Although many theoretical works about movement generation by the CPG, only a few learning models have been proposed for the acquisition of an adequate parameter set that generate a desired phase pattern. One of them is worth noting as it is able to learn a specific phase pattern according to the correlation between the effects of the signal from the CPG and the teacher signal.

Nishii et al.

An interesting model of a learning algorithm for networks of coupled neural oscillators was proposed by Nishii (Nishii, 1998, 1999), where a CPG model able to learn a phase pattern imposed by a teacher signal is proposed. Once the pattern is learned, it is able to generate it without any signal emanating from a higher control center. Even though there exist physiological evidence that CPG receive efferent signals from higher center, we do not know if they are really used as teacher signals in order to tune the CPG parameters. This models suffers from the use of this external signal, whereas many living organism might be able to learn gait patterns without any instructions. The interest to present this model, is that a learning rule modifying the intrinsic frequencies of the oscillators and their coupling weights according to the effects of the input signal on the dynamics of the oscillator is proposed. Although, even if this model is interesting for robotic implementation (e.g. learn a specific gait for example), many animals seem to learn the desired patterns without any explicit teacher signal.

Aoyagi

Another model of a neural oscillatory network was proposed by Aoyagi (Aoyagi, 1995; Aoyagi and Nomura, 1999), in which the oscillators are able to store and retrieve sparsely coded phase patterns. They model neurons as being oscillators, some of which are in nonfiring state, while others do encode information in the timing of their spikes. An interesting point is that the proposes a Lyapunov function ensuring, under suitable conditions, that this model performs a stable retrieval process. At the same time, they show that even near saturation of the memory, the basins of attraction remain sufficiently wide to allow for the recall of a memorized pattern even from a noisy input pattern.

2.6 Discussion

We briefly introduced the theory of oscillating dynamical systems, and we tried to give an idea on how important these systems are. Oscillating systems regulate everything around us, from the microscopical world (interaction between molecules) to the macroscopical one (periodic motion of planets in our solar system), and a key to understand how do all that work, lies in understanding the complex behavior of such interacting systems, and the emergence of new phenomena due to these interactions. Science of complexity is a freshly born science where only the tip of the iceberg is known, and many discoveries still remain to be done, not only in neuroscience, but also in all sciences dealing with networks of interacting elements (gene regulation networks, disease propagation due to sexual interactions, virus propagation in computer networks, self-organized behavior in animal swarms, etc.)



Analysis of the model

"The true art of memory is the art of attention." Samuel Johnson (1709-1784)

We present the model we implemented in detail and we discuss about the performance measures we used to derive its efficiency. We conclude this chapter with a discussion about the properties and drawbacks of this model, and what aspects should be improved to get a more efficient model.

3.1 Description of the model

The model we investigate in this project is described into (Borisyuk *et al.*, 2001). This paper proposes a novel kind of memory that differs from the classical models we discussed in the previously, as the idea is that many coupled oscillators are able to store multiple signal and recognize when signals are already learned. Depending on the phase relations between these signals, specific regions of the network are activated by a stimulus and memory is created through hebbian modification of the connexion strengths between oscillators. Their model is composed of an oscillatory mechanism of memory formation in that the input information is encoded into a constant frequency input signal, and storage is done through adaptation of the natural frequencies of the oscillators in the network. Recognition of stored patterns is done through resonant amplification of the activity in the network.

The new idea proposed by this model is that there is not only a modification of connection strengths between the oscillators such as is usually the case in the classical connectionist theory, but also a permanent modification of the natural frequencies of the oscillators.



Figure 3.1: Structure of the network. Each square refers to an independent group of q all-to-all connected oscillators (here q = 4), the arrows entering into each group correspond to the *i*th component of the input signal C_i (here $1 \le i \le n =$ 4). Even if each group gets all components of the input signal, each arrow has its own phase shift ψ_{ij} , where *i* refers to the group the arrow is related to, while *j* is the index of the component of the input signal, so left-most arrow has phase shift ψ_{11} , the one immediately on its right has ψ_{12} and so on and so forth.

3.1.1 Network architecture

This model consists of a network of oscillators combined into m groups $G_j(j = 1, \ldots, m)$ with q oscillators into each group, with all-to-all connections for the oscillators belonging to the same group and no connections taking place between members of different groups. It is assumed that each oscillator represents the average activity of locally coupled excitatory and inhibitory neurons, where the *local field potential*¹ is represented by the activity of the oscillator. Before the network is initialized, the natural frequency of each oscillator (ω_{ij} for oscillator i in group j) is chosen in the range [$\omega_{min}, \omega_{max}$] in a homogenous manner. An n-dimensional input signal is fed into each oscillator in the network and each dimensional component C_i ($i = 1, \ldots, n$) is a sinusoidal function

¹The local field potential is the local activity of the neuronal population.

of the form

$$C_i = \sin(2\pi\omega_0 t + \psi_{ij}) \tag{3.1}$$

where ω_0 is the frequency of the input signal that is in the range $[\omega_{min}, \omega_{max}]$ and ψ_{ij} are phase shifts that imitate different time-lags during signal transmission to a particular group of oscillators. When the network is initialized, each ψ_{ij} is chosen randomly in the interval $[-\tau, \tau]$.

This network is able to learn input signals by tuning a certain amount of oscillators to the frequency of the input signal (ω_0). The signal is presented to the network, thus turning some oscillators of the network into a resonant state, who will attract all oscillators in their group to the input frequency. An external supervisor monitors the amount of resonant oscillators in the network and counts the time needed for the amount to exceed a fixed threshold H. Of course the time needed to reach the threshold decreases at each presentation of the stimulus, as the number of tuned oscillators increase after each stimulus presentation. If the time to reach the threshold if inferior to the *critical* time Tcr, then we infer that the signal has been learned by the network and is not novel.

3.1.2 Dynamics of the network

Now we described the coupling architecture of the network, we can give the three equations regulating the dynamical behavior of the network, that is the evolution in time of the phase θ_k^j , amplitude a_k^j , and frequency ω_k^j of the oscillators.

$$\dot{\theta}_{k}^{j} = 2\pi\omega_{k}^{j} + \frac{v}{n}\sum_{i=1}^{n}\sin(2\pi\omega_{0}t + \psi_{ij} - \theta_{k}^{j}) + \frac{w}{q}\sum_{l=1}^{q}g_{1}\sin(\theta_{l}^{j} - \theta_{k}^{j}) (3.2)$$

$$\dot{a}_{k}^{j} = -\beta a_{k}^{j} + \gamma g_{2} \left(\frac{1}{n} \sum_{i=1}^{\infty} \cos^{2}_{+} (2\pi\omega_{0}t + \psi_{ij} - \theta_{k}^{j}) \right)$$
(3.3)

$$\dot{\omega}_k^j = -\alpha g_1 \left(\omega_k^j - \frac{\dot{\theta}_k^j}{2\pi} \right) \tag{3.4}$$

where $v, w, \alpha, \beta, \gamma, \xi_i, \rho_i$, are all positive parameters with values as defined in Table 3.1. Attention, the g_1 must be read as $g_1(a_l^j)$.

The $g_i(x)$ function, with i = 1, 2 is a sigmoid function defined as

$$g_i(x) = \frac{exp((x - \xi_i)/\rho_i)}{1 + exp((x - \xi_i)/\rho_i)}$$
(3.5)

The $\cos_+(x)$ is defined as

$$\cos_{+}(x) = \begin{cases} \cos(x) & \text{if } \cos(x) > 0; \\ 0 & \text{otherwise.} \end{cases}$$

Phase dynamics

Note that Equation 3.2, leads to phase-locking behavior described by the sum of three components. The first term alone, describes a simple oscillator whose phase grows uniformly in time, and growth is proportional to its natural frequency.

Then, the second term drives the oscillator to phase-lock with the input signal. For an oscillator having the same natural frequency as the input signal, this term is gradually reduced to 0, that is the phase difference between the oscillator and the input signal also converges to 0. The parameter v weighting this term influences how quickly will the oscillator locks to the phase of the input signal, or better said the strength of the influence of the input signal on the oscillator.

Finally, the rightmost term influences the phase-locking according to the phase difference between the oscillator and all other members of the same group. In one word, it makes all oscillators in the group converge to a common phase. The $g_1(a_l^j)$ factor allows only resonant oscillators (whose instantaneous amplitude is above a certain threshold determined by the parameter ξ_1) to influence efficiently the other oscillators. The w parameter denotes the coupling strength between a resonant oscillator and its neighborhood, that is the strength of attraction of the common phase onto the phase variation.

Amplitude dynamics

The differential equation governing the amplitude evolution is used to regulate the resonant response of an oscillator. The function $g_2(x)$ is as described by Equation 3.5. The second term grows exponentially as the phase difference between input signal and group phase of the oscillators decreases, and we can easily notice that the sum in Equation 3.3 will tend to γ as the phase difference between the oscillator and the input signal decreases, and will equal to γ only if the sigmoid returns 1, that is if the value of the sum is above the threshold ξ_2 . For these phase-locked oscillators, we can find a fixed point by setting the value of $\dot{a}_k^j = 0$, and we find the value of a_k^j where the amplitude stabilizes, and we denote this value $a^* = \gamma/\beta$. This is the maximal amplitude that can be reached by perfectly in-phase signal. However, we define that an oscillator with an amplitude $a_k^j \geq \frac{4}{5}a^*$ is said to be *resonant*. If the phase and frequency difference is large enough to prevent the oscillator to phase-lock to the input signal, its amplitude will remain small and thus it will not be considered as resonant.
Frequency dynamics

The frequency change of the oscillators is where the learning process really takes place, and we can see the variation is a product of three factors. First, the α parameter determines the learning speed. Then, the $g_1(a_k^j)$ term prevents non-resonant oscillators from changing their frequency, so only oscillators located in groups with an appropriate phase shift will learn the signal. Finally, the difference between internal frequency and the phase variation is what quantifies the amount and direction of the frequency change. We can see in Equation 3.2 that for phase-locked oscillators, the phase variation equals $2\pi\omega_k^j$ (as the second and third term decrease as they become in-phase with the input signal and with their resonant neighbors), thus the frequency is not changed anymore.

3.1.3 Implementation

The implementation of the model was done in C++ using the LANDS² basis created by Jonas Buchli, at the BIRG group. Moreover, we created a MAT-LAB GUI³ that allows one to choose parameters, launch a simulation, and analyze results directly, or even create scripts that allow you to run several simulations automatically and analyze the results afterwards. Finally, we also implemented the same models in MATLAB in order to perform rapid evaluation and debugging of the models. The implementation part of the simulation was done rather quickly, but we spent an enormous amount of time in order to asses the correctness of our implementation as the results we obtained using exactly the same parameters were totally different from the ones in the paper. After thorough analysis, we concluded that our implementation behaves correctly according to the mathematics underlying the proposed model, but not with the parameters as given in the paper. We noticed that the equations were mistyped in the original paper, as the 2π term was missing in two equations. Also, the value of the parameter $\tau = \pi/2$ as given in the paper was wrong, and we proved mathematically (see below) that the model cannot work with such a high value for τ . We decided thus to contact the authors of the paper who agreed with our assumptions, and indicated us that the value they used in their experiments was actually around $\tau = \pi/6$. In our simulations, we used $\tau = 0.7.$

3.2 Dynamics of the network

We are going to discuss some important aspects concerning the dynamics of this model. Let us start with the Equation 3.2 which describes the evolution

 $^{^2\}mathrm{An}$ efficient framework for easy implementation and simulation of dynamical systems. For download, check http://birg.epfl.ch.

³GUI : Graphic User Interface

of the phase for each oscillator. The third term indicates the mean phase for all the resonant oscillators in the same group. The way this term evolves is obvious, as all members will be attracted towards a common phase, which is regulated by the most in-phase oscillators having the highest amplitude due to the factor $g_1(a_1^j)$. As the standard deviation between the phase of all oscillators decreases as they learn the appropriate frequency, this term decreases as well. The evolution of the second term is much harder to visualize, due to the non-linearity introduced by the term $\sum_{i=1}^{n} \sin(2\pi\omega_0 t + \psi_{ij} - \theta_k^j)$. This term attempts to adjust the phase of the oscillator so as to phase-lock with the input signal, and has the same structure as the phase-locking behavior described by Kuramotos' model we defined in Equation 2.21 where the dynamics also seek to minimize their phase difference with the input signal. You can see the dynamics of this second term in Figure 3.2, where this term converges to 0 for any oscillator having the the same frequency as the input signal, that is $\omega_i = \omega_0$, while it fluctuates without converging when the frequencies are different. As learning changes the frequencies of the oscillators towards ω_0 , we can conclude that all oscillators in the network will sooner or later phaselock to the input signal. Even if this statement could seem appealing, we will show later that it is not true for any group of oscillators due to the different phase shifts, and this is exactly what makes the model interesting. As long as $\tau < \pi/2$, this term will always converge towards 0 for any value of n, it is only a matter of time. Then you can see the behavior of one group of oscillators having an good phase combination in Figure 3.3, whereas the behavior of a non-resonant group can be seen in 3.4.

3.3 Analysis of the parameters

Let us now describe the effects of the parameters regulating the model. First, you can have a look at the values used in the original paper in Table 3.1.

Parameter	Value	Parameter	Value	Parameter	Value
m	500	n	20	q	50
α	1	β	4	γ	4
v	0.5	w	16	ξ_1	0.7
ρ_1	0.02	ξ_2	0.86	ρ_2	0.02
T	3	T_{cr}	1.5	H	450

 Table 3.1: Numerical values of the parameters as proposed in the paper.



Figure 3.2: Dynamics of the term $\frac{v}{n}\sum_{i=1}^{n}\sin(2\pi\omega_0 t + \psi_{ij} - \theta_k^j)$ of Equation 3.2 with parameters $\alpha, \beta, \gamma = 0$ for three oscillators with different natural frequencies (resp. $\omega = 6.8$, $\omega = 7$, and $\omega = 7.2$) when exited with the same 20-dimensional input signal with frequency $\omega_0 = 7$, and identical phase shifts (chosen randomly in the range $[-\tau, \tau]$). Note the asymmetry between the oscillators with $\omega = 6.8$ and $\omega = 7.2$. Note also the phase-locking behavior of the oscillator having the same frequency as the input signal (regardless of the input dimension n), whereas this term do not converge to 0 for oscillators having a different frequency.

3.3.1 Phase differences (ψ_{ij})

The major problem encountered was the initial distribution of the values in the ψ_{ij} matrix. As these values are initialized randomly at the beginning of each experiment, the results can vary significantly even if no other parameters are changed, due to the complex (and non-linear!) interactions between the components of the model. This random initialization prevents us to deduce the real effect of a given parameter on the behavior of the network, regardless of the initial conditions. To capture the influence of a single parameter we decided to run some experiments using a single group (m = 1) with a unidimensional input signal (n = 1), and in this case the matrix ψ_{ij} becomes a single variable we set to $\psi = 0$. Unfortunately, this is what makes our model interesting, as it takes advantage of the great number of available groups, thus leading to some kind of distributed memorization of the signal in multiple groups.



Figure 3.3: Dynamics of the evolution for a group with a coherent input signal. A stimulus with a frequency $\omega_0 = 7$ has been presented during a time T = 3, to n = 20 oscillators with $\omega_{min} = 6$ and $\omega_{max} = 8$. Top left: Evolution of the second term of Equation 3.2 $\left(\frac{v}{n}\sum_{i=1}^{n}\sin(2\pi\omega_0t + \psi_{ij} - \theta_k^j)\right)$ which indicates the phase difference with the input signal. Note the way this term stabilizes towards a null value as all oscillators in the group phase-lock and their frequency converge towards ω_0 . Top right: evolution of the third term of Equation 3.2 $\left(\frac{w}{q}\sum_{l=1}^{q}g_1\sin(\theta_l^j - \theta_k^j)\right)$. Note how this term also converges to 0. Center left: second term of Equation 3.3 $\left(g_2\left(\frac{1}{n}\sum_{i=1}^{n}\cos^2_+(2\pi\omega_0t + \psi_{ij} - \theta_k^j)\right)\right)$. Center right: This is the value of $|\dot{\theta}_i - 2\pi\omega_i|$. Bottom left: Amplitudes of all oscillators in the group. Bottom right up: evolution of the θ_i of all oscillators in the group. We can clearly see the phase-locking behavior between all oscillators in the group.



Figure 3.4: Dynamics of the evolution for a group with an incoherent input signal. A stimulus with a frequency $\omega_0 = 7$ has been presented during a time T = 3 to n = 20 oscillators with $\omega_{min} = 6$ and $\omega_{max} = 8$. Top left: the evolution of the second term in of Equation 3.2 $\left(\frac{v}{n}\sum_{i=1}^{n}\sin(2\pi\omega_0t + \psi_{ij} - \theta_k^j)\right)$. Note the this term fluctuates without stabilizing. Top right: evolution of the third term of Equation $3.2\left(\frac{w}{q}\sum_{l=1}^{q}g_1\sin(\theta_l^j - \theta_k^j)\right)$. Center left: Evolution of the second term of Equation $3.3\left(g_2\left(\frac{1}{n}\sum_{i=1}^{n}\cos^2_+(2\pi\omega_0t + \psi_{ij} - \theta_k^j)\right)\right)$. Center right: This is the value of $|\dot{\theta}_i - 2\pi\omega_i|$. Bottom left: Amplitudes of all oscillators in the group. Bottom right up: Natural frequencies of the oscillators in the group. We can clearly see that no phase-locking takes place, because the deviation of the phases of the oscillators keeps increasing, and there is no convergence of the frequencies towards ω_0 .

3.3.2 Basin of attraction (v)

If one wants to be able to control the way the amplitudes evolve, we need to find a mathematical expression that yields the width of the basin of attraction of each oscillator according to the chosen parameters and to the current state of the network. With this function we can control the way the evolution of the amplitude, and we are able to decide when the learning must be stopped. Above all, we can restrict the number of oscillators that will learn an input signal. For the effects of this parameters, have a look at Figure 3.5, where the plots on the left were done using v = 0.5, while on the right we used v = 2. Note how the behavior is similar for the same value of β , γ , but how the width of the cluster composed by the resonant oscillators is increased.

3.3.3 Amplitude behavior (β, γ)

We discuss only the case where $\beta = \gamma$, as this is the required condition in order for the amplitude to be in the range [0, 1] for the amplitude. As you can see in Figure 3.5, this value to a higher value makes our system stabilize to the maximal value the amplitudes of resonant oscillators.

3.3.4 Neighborhood attraction (w)

The parameter w is important in that it asses how long it take for an oscillator to phase lock to an input signal according to the difference between its natural frequency and the input signal frequency. As can be seen in the equations, it weights the impact of the phase difference between the phases of all oscillators in the group onto the phase-locking behavior, if they are in a resonant state. A low weight will turn only oscillators with a very close frequency to phase-lock (if they are resonant of course), while a high value of w will turn all oscillators in the group to phase-lock. You can see the effect of this parameter onto the amplitude evolution in Figure 3.6, and onto the frequency adaptation of the oscillators in Figure 3.7.

3.3.5 Learning speed (α)

Ideally, we want to design a system which is able to learn as quickly as possible, but this is not always feasible without introducing other drawbacks. The learning speed is influenced by the factor α , but one should be aware of concluding that setting this factor to a higher value will directly improve the performance of the network. The problem is that if we increase the learning speed, the systems will becomes less unstable, as it it will become very responsive to small perturbation that can be considered as noise. As all oscillators will tune their frequency to this signal very quickly, the obvious drawback is that the flexibility of the network is drastically reduced in the sense that



Figure 3.5: Effects of the parameters β , γ and v on the evolution of the amplitudes. We set α and w to 0 in order to prevent the influence of frequency adaptation and of attraction between neighbors. Left top: v = 0.5, β , $\gamma = 1$, Left middle: v = 0.5, β , $\gamma = 4$, Left bottom: v = 0.5, β , $\gamma = 8$, Right top: v = 2, β , $\gamma = 0.5$, Right middle: v = 2, β , $\gamma = 4$, Right bottom: v = 2, β , $\gamma = 8$.



Figure 3.6: Influence of the resonance attraction for parameter w for 5 consecutive presentations of an input signal with $\omega_0 = 7$ during T = 3, with a single input on a signal group of 50 oscillators with a null phase shift ($\psi = 0$). We can see the dynamics of amplitudes for each oscillator when varying the factor w. The other parameters were $\alpha = 0$, to prevent influence of learning falsifying the amplitudes evolution, $\beta, \gamma = 2$ to prevent all oscillators to resonate too quickly. The feedback was blocked in order to see the full evolution of the amplitude. All other parameters are as given by Table 3.1. From left to right w = 2, w = 8, w = 10, w = 14, and w = 24.



Figure 3.7: Influence of the resonance attraction for parameter w for 5 consecutive presentations of an input signal with $w_0 = 7$ during T = 3, with a single input on a group of 40 oscillators with a null phase shift ($\psi = 0$). In this experiment the threshold is not used, so the oscillator learns the frequency of the input signal as long as it is present. We can see the influence of the factor w upon the frequency locking behavior. All other parameters are as given by Table 3.1. (a)w = 3, (b) w = 6, (c) w = 8, (d) w = 9, (e) w = 10, (f) w = 16.

all available oscillators will be used solely to encode this unique information. The duration of the stimuli presentation is simply a design issue in that it changes only the time scale. One should not concentrate on the exact time of presentation, but rather a good way to prevent under-learning.

3.3.6 Sigmoid functions(ξ_i , ρ_i)

The parameters used in the sigmoid functions depends only on the other parameters so there is no real interest to study the global behavior according to these parameters. For example, the ξ , are chosen according to the minimal value of the amplitude where an oscillator is considered as being resonant (0.8), so we will not discuss further this aspect.

3.4 Resonant oscillators

Now, let's get into the heart of this model. The fact that phase-locking occurs in some groups, that is the second term of Equation 3.3 converges to 0, does **not** imply that the second term of the amplitude dynamics (Equations 3.3) $(\sum \cos_+(t))$ will converge to 1. This is a very important remark, as it implies that only a fraction of the oscillators in the network can become resonant (reach an amplitude of 0.8) for each signal, independently of the time the signal is presented. We are interested into deriving the condition where an oscillator is considered as resonant, that is when its amplitude is superior to R = 0.8. From Equation 3.3, we want

$$a_{k}^{j} = \frac{\gamma}{\beta} g_{2} \left(\frac{1}{n} \sum_{i=1}^{n} \cos^{2}_{+} (2\pi\omega_{0}t + \psi_{ij} - \theta_{k}^{j}) \right) \ge R = 0.8$$

let's expand the $g_2(x)$ function, and isolate the exp(x) term

$$\exp\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\cos_{+}^{2}(2\pi\omega_{0}t+\psi_{ij}-\theta_{k}^{j})-\xi_{2}}{\rho_{2}}\right) \geq \frac{R\beta}{\gamma-R\beta}$$

which gives the condition to satisfy if we want our oscillator to become resonant

$$\frac{1}{n}\sum_{i=1}^{n}\cos_{+}^{2}(2\pi\omega_{0}t+\psi_{ij}-\theta_{k}^{j}) \geq \xi_{2}+\rho_{2}\ln\left(\frac{R\beta}{\gamma-R\beta}\right)$$

Additionally, if we assume that the oscillator is perfectly in phase with the input signal and with all his neighbors, $\dot{\theta}$ becomes $2\pi\omega$, whose solution is simply $\theta(t) = 2\pi\omega t$, thus the condition becomes :

$$cs = \frac{1}{n} \sum_{i=1}^{n} \cos^2_+(\psi_{ij}) \ge \xi_2 + \rho_2 \ln\left(\frac{R\beta}{\gamma - R\beta}\right) = cs_{min} \tag{3.6}$$

With the parameters given in the paper, the value of the left term is $cs_{min} = 0.8877$, which is perfectly consistent with the parameters proposed in the paper for the sigmoid $g_2(x)$ as $g_2(cs_{min}) = g_2(0.8877) = 0.8$, which is exactly the minimal value of the fixed point of the amplitude dynamics so that the oscillator is considered resonant.

In order to estimate the probability for an oscillator to become resonant, one needs to approximate the statistical distribution (also called the *Probability* Density Function (PDF) of the left term of Eq. 3.6 (cs) for different values of τ and n. We performed Monte Carlo experiments over 100'000 trials, each one with a random initialization of the phase shifts ψ , and recorded all the values of cs. In order to estimate the probability for an oscillator to be considered as resonant, we want find the probability for the value of cs to be greater than cs_{min} . We can do that in two ways : either empirically counting the number of trials where this assumption is true, or else by stating that the distribution is Gaussian. This statement is just an assumption and not a proven fact, but as we performed a huge number of trials, the Central Limit Theorem states that the distribution tends towards a normal one, with mean and variance estimates extracted from experimental results. From a statistical point of view, the probability we are looking for corresponds to the area of the dark surface under the curve in the right plot of Figure 3.9. Finally we performed the same number of runs of the complete simulation with a single oscillator having random phase shifts (ψ_{ij}) at each trial, and counted how many times the oscillator had an amplitude superior to 0.8. The numerical results are given in Table 3.2. We can see that the first and the third column are very similar thus confirming the validity of Equation 3.6, whereas the values yielded by the Gaussian approximation are more different. The reason is obvious, we cannot be sure that the distribution is Gaussian, and as you can see in the left plot of Figure 3.9, the histogram does not fit perfectly the exact distribution having the same mean and variance. Also, the more runs we perform, the better should be the results of the Gaussian approximation.

3.5 Discussion

3.5.1 Performance analysis

In order to asses the performances of the model, it could be interesting to find an analogy to the concept of basin of attraction as used in the Hopfield model we could apply for this model. We define the basin of attraction as the range of frequencies where an oscillator phase-locks to a signal, or more exactly the range of entrainment as defined in the previous chapter. This is a capital aspect in the model as it determines the amount of oscillators that will be dedicated to store each input signal. Ideally, the user should not have to tune so many params in order to use the network, but he should only choose



Figure 3.8: Value of the term $\frac{1}{n}\sum_{i=1}^{n}\cos_{+}^{2}(\psi_{i})$ of Equation 3.2. We have 5 distinct groups, each corresponding to a different value of τ . *Exp.* 1-5: $\tau = 0.3$, *Exp.* 6-10: $\tau = 0.5$, *Exp.* 11-15: $\tau = 0.7$, *Exp.* 16-20: $\tau = 1$, and *Exp.* 21-25: $\tau = \pi/2$. For each experiment we vary the dimension n of the input signal. The first column of each group corresponds to n = 5, the second to n = 10, the 3^{rd} to n = 20, the 4^{th} to n = 50, and the 5^{th} to n = 100.

au	n	Experimentally	Gaussian approximation	Simulation
au = 0.3	n = 5	1	1	1
	n = 10	1	1	1
	n = 20	1	1	1
	n = 50	1	1	1
	n = 100	1	1	1
au = 0.5	n = 5	0.8568	0.8513	0.9269
	n = 10	0.9337	0.9286	0.9602
	n = 20	0.9834	0.9807	0.9884
	n = 50	0.9996	0.9992	0.9996
	n = 100	1	1	1
au = 0.7	n = 5	0.2632	0.2795	0.4633
	n = 10	0.1851	0.1889	0.2995
	n = 20	0.1038	0.1034	0.1562
	n = 50	0.0229	0.0203	0.0283
	n = 100	0.0024	0.0016	0.0026
au = 1	n = 5	0.053	0.0484	0.14
	n = 10	0.01146	0.00708	0.0214
	n = 20	0.00065	0.00031	0.00067
	n = 50	0	0	0
	n = 100	0	0	0
$ au=\pi/2$	n = 5	0.0074	0.005	0.018
	n = 10	0.0027	0.0001	0.0003
	n = 20	0	0	0
	n = 50	0	0	0
	n = 100	0	0	0

Table 3.2: Probability for a perfectly phase-locked oscillator with the same frequency as the input signal and synchronized with its neighborhood to be considered as resonant (*i.e.* its amplitude reaches 0.8). The first column corresponds to the experiment described in the basic Monte Carlo simulation where the densities are counted empirically. The data in the middle column is obtained by approximating the results of the experiment by a Gaussian with mean and variance given by the samples, and we used the MATLAB **normspec** function in order to get the probability. The last column is the real density of oscillators that reached the resonance threshold in a complete simulation of the network.



Figure 3.9: Densities found experimentally in Experiment 15 of Figure 3.8. *Left:* the gaussian approximation of the data. *Right:* the same gaussian, but with dark area corresponding to the part of the distribution whose *cs* is superior to 0.8877, and thus will reach a resonant state.

how many oscillators he wants to dedicate for a signal to learn. The problem is that the amount of overlapping groups for the different signals is not easy to compute due to the random initialization of the phase shifts.

Error rate vs. memory filling

This is clearly one of the most annoying drawbacks of oscillatory memory models. The point is that, due to the simplicity of the mathematical models, there is no mechanism embedded in the system to prevent "overwriting", in the sense that there is no way to ensure whether previously stored data will not be affected by adding a new memory. In this model, the place where a signal is learned is define according to the phase shifts coincidence between the input and the group. As this is not an bijective function, where there exist a single group for each signal and viceversa, it may happen that the same group could be used to encode more than a single signal, thus implying that the retrieval error increases as the memory is filled.

3.5.2 Input signal dimension

As we have shown in Figure 3.8, the effect of increasing the dimensionality of the input signal, does not change the mean value of the second term of Equation 3.3, but the more dimensions we add, the lower is the variance around the mean. We know much better the probability of the sum to have a precise desired value, and thus we can control much better what will be the value of the term. This could be used in order to know exactly the ratio of oscillators that will be activated by an input signal, and how many of them will be used to store the signal.

3.5.3 Distributiveness of learning

A very interesting aspect of this model is that without the groups being actually connected, many of them are used to store the same input signal, which makes the model more robust, as the data is distributed over many groups, and loss of a group doesn't result in total loss of previously stored data. Unfortunately, recalling is done through counting of resonant oscillators in the network without autonomous feedback regulation embedded into the equations of the system. Recall of a stored signal is done through the time it takes for a certain threshold of resonant oscillator is reached, and this is also used to cut the learning activity in the network to prevent all oscillators from the network converging to the input signal. This an interesting feature as it allows a spatial distribution of an input signal onto many groups, and that can be very useful to increase the overall robustness of the system. This feature is found in biological systems, but unfortunately, a global watchdog keeping track of the number of resonant oscillators, is out of question concerning biological plausibility.

You can see in Figure 3.10 how many resonant oscillators are located in each groups, for a simulation of the experiment proposed in the paper. It's clear that some of groups have all oscillators tuned to same frequency, and some other groups have only a part of their oscillators into a resonant state. In fact, the number of oscillators in a group that codes the stimulus depends on how good the coincidence of the phase shifts is at this group. This prevents the method used in the paper to be rigorous. The problem is that a group is not totally "dedicated" for a specific signal, and this can be viewed as a huge waste of oscillators. But on the other hand, this is also the property that allows a distributed storage, thus increasing the robustness of the model.

3.5.4 Threshold function

The utility of a threshold function should be discussed. On one hand, it is a very raw technique to stop the learning process as soon as a certain amount of oscillators enter a resonant state and prevent overlearning of the network. Such a global function is clearly not biologically plausible and it can be seen rather as a hack, or artifact than an embedded property of the model. On the other hand, it's clearly an easily implementable and efficient way to ensure a control of the learning able to ensure a distributed representation of the learned signal. Unfortunately, the authors propose just a numerical value (H = 450), without giving any explanation motivating the choice of this particular value of the threshold other than the fact this value performs well according to the other parameters.



Figure 3.10: Number of resonant oscillators into each group after learning when the learned signal is presented and when the threshold is reached. Note that this is not an all-or-none scheme as there exist groups where only a few oscillators are resonant.

3.5.5 Positive aspects

First of all, the most innovative and promitting aspect is that this model is based on the nonlinear dynamical systems framework. As many natural systems have similar properties, this gives us the possibility to implement it in hardware (for example, we could use *Phase-Locked Loops (PLLs)*). This aspect gives an analogical memory compared the digital aspect of the classical memories.

Another interesting aspect of this model is that a very smart mechanism is proposed in order to find the spatial location of the storage site for a signal to memorize, according to the phase relationship between the stimuli and the groups in the network. This is very similar to what happens in the human brain. Moreover, data is stored in a distributed manner, so we do not lose all the stored information when a group is lost.

3.5.6 Negative aspects

The problem we face is that due to the relative young age of this field, there do not yet exist formal tools in order to evaluate the performance of this kind of network analytically, and we can only test it empirically under different assumptions.

Let us present what we think are the weakest points of this model

- A huge problem of this network is the number of parameters required to make it work. The problem is that many combinations of different values of the parameters can lead to apparently identical behavior.
- Due to the random initialization of the phase shifts, the efficiency of this model can vary significantly even when running exactly the same experiment several times. This prevents using the novelty detection criteria they propose, as the time needed to reach the threshold can change a lot across trials.
- Another big problem is the fact that the time is used only for learning dynamics, so it is reset at the end of each presentation of a signal. Additionally, the model is sensitive to the phase relationships between the input and the oscillators in the network, so if the phase is changed by noise the situation can become problematic. This non-absolute timing prevents to learn temporal sequences, where the order of presentation of the signals is not captured. This aspect would have been useful for robot programming by demonstration for example.
- The way the signal is given explicitly in the equations implies that we can only memorize a single frequency presented as a sum of cosine functions

with the same frequency (which is still a cosine). This drawback is clearly the bottleneck of this models as it prevents us to use this network to store more complex signals.

- The reason of using a multi-dimensional input has not been clearly stated. As we have shown in Figure 3.8, this parameter broadens or the variation of the sum. This parameter seems only to contribute to a better distribution without any efficiency and capacity improvement, nor complexity of the signal we want to store.
- Last but not the least, the amount of oscillators used to memorize a single frequency is by far too large to provide an efficient storage, and moreover, even with so many oscillators, we cannot ensure that information is stored reliably (due to overlapping between groups).



Further improvements

"The search for truth is more important than its possession." Albert Einstein (1879-1955)

In this chapter we propose how to modify the model we described in the previous chapter by combining it with another oscillator model that is able to learn complex signals.

4.1 What was done

We had two directions where search a way to improve the model. The first direction was to derive new dynamic equations and learning rules. The other was to investigate the network architecture and the coupling between oscillators.

4.1.1 Complex signal learning

Our starting point was to take inspiration from the model described in (Righetti *et al.*, 2005), where an oscillator able to learn the period of any periodic signal is proposed. The originality of this model is that it is able to learn any kind of periodic signal, without any *a priori* knowledge upon its spectral decomposition and does not require any explicit signal processing (e.g. no FFT, nor setting of a time window). This model has also a very innovative feature, unusual in the classical neural networks theory, which is the learning phase to be directly embedded in the dynamics and not as an offline optimization process.

The oscillator used in the model is based on a Hopf oscillator (perturbed

by a external stimuli F) whose dynamics are

$$\dot{x} = (\mu - r^2)x - \omega y + \epsilon F \tag{4.1}$$

$$\dot{y} = (\mu - r^2)y + \omega x \tag{4.2}$$

with radius $r = \sqrt{x^2 + y^2}$. As you can see, this oscillator is perturbed along the *x*-axis with an unknown periodic signal *F*, defined as a weighted sum of sine (or cosine) functions with respective frequencies ω_i ,

$$F(t) = \sum_{i=1}^{n} a_i \cos(\omega_i t)$$
(4.3)

If we prefer to express this system with polar coordinates $(x = r \cos(\theta), y = r \sin(\theta))$, the system becomes

$$\dot{\theta} = \omega - \frac{\epsilon}{r} F \sin(\theta)$$
 (4.4)

$$\dot{r} = (\mu - r^2)r + \epsilon F \cos(\theta) \tag{4.5}$$

It is well known that such an oscillator having a natural frequency close to the frequency of a component of the input signal will lock to the most attracting frequency (the nearest one if all components have the same coefficient a_i), this is the frequency entrainment behavior we described in Chapter 2. But if the frequency difference is too high (*high* is relative as it depends on the value of ϵ), the phase-locking cannot take place. To allow any oscillator to learn an input frequency component regardless of the detuning, a learning rule that adapts the frequency in function of the correlation between the phase of the oscillator and of the input signal has been introduced. This rule is simply described as follows

$$\dot{\omega} = -\epsilon F \sin(\theta) \tag{4.6}$$

This rule is the most important aspect of the model, as it allows a permanent form of learning to take place. Once an oscillator changes its natural frequency towards a presented input, when the detuning becomes low enough, entrainment takes place and the oscillator phase locks to the component of the input signal. This behavior is plotted in Figure 4.1 where you can see that once the entrainment takes place, the natural frequency of the learning oscillator stabilizes around the target frequency with small oscillations in amplitude, but the average value of these oscillations is the learned frequency.

This oscillator has an interesting property in that it is not sensitive to phase shifts between the input signal and the internal phases of the oscillator. Whether this property is an advantage or not can be discussed. On one hand, this insensitivity to phase shifts increases the robustness of the oscillator in that the property of phase-locking and frequency learning is independent of the



Figure 4.1: Adaptation of the natural frequencies for some perturbed oscillators with dynamics described by Equations 4.4-4.6, for different initial frequencies. We used $\mu = 1$, $\epsilon = 1$, and the perturbation is F(t) = cos(25t). Note that there are two time scales, small-amplitude oscillations are taking place very quickly compared to the time scale on which learning takes place.

phase difference between an oscillator and the input signal, so phase shifts due to noise will not affect the behavior (except in the time required by an oscillator to lock to the input signal). On the other hand, a major drawback introduced by this robustness, is that we cannot use the phase difference anymore to determine the spatial location where to store an information, and this was one of the most interesting property proposed in (Borisyuk *et al.*, 2001). Another method for storing multiple data without the risk of overwriting previously stored data is thus needed.

4.1.2 Improve the model

Our task was to take inspiration from the model we analyzed in the previous chapter, and design an oscillatory memory able to learn complex signals, as allows the oscillator we have just described. As we have to couple these oscillators in a way or another, and we want also to store more than a single signal in a network, we must be able to define precisely the range of attraction for each frequency composing the input signal, in order to prevent that all oscillators in the group will learn a unique input signal. Thus, we had to find a method that discriminates between the oscillators belonging to the basin of attraction of a frequency component of the input signal and the other oscillators, so as to limit the amount of oscillators that will be used to learn that component. The solution we propose to do that is to rely on the fact that a Hopf oscillator perturbed with a frequency close to its natural one enters into a $resonant^1$ state. So we can fix a threshold and state that only oscillators having an amplitude superior to the threshold will learn the frequency in question. This is done multiplying $\hat{\theta}$ with the factor $q(a_i)$, that is a sigmoid function of the amplitude as defined in Equation 3.5 with the threshold being ξ , which returns 1 only if $a_i > \xi$, and 0 if not. Unfortunately, the value of r keeps oscillating on a small scale of order $\mathcal{O}(\epsilon)$, and this instability is very annoying as the amplitude oscillations are greater than the difference of amplitude between the oscillators themselves, thus it was impossible to determine a relevant threshold that can discriminate efficiently and correctly. For that reason, we had to find another way to clusterize² the oscillators belonging to a basin of attraction, as well as reduce the tiny oscillations we discussed above. This was done by replacing the original amplitude in the original model by this one

$$\dot{a}_i = -\frac{a_i}{\beta} + F\cos(\theta_i) \tag{4.7}$$

This model has the advantage of amplifying the difference of the amplitude between resonant and non-resonant oscillators, and at the same time stabilizes the small oscillations of the amplitude. But, it is not (yet) a true limit-cycle oscillator, which requires a non-null fixed point of the amplitude when no perturbation is presented. This is important especially if one wants to implement this model using oscillators commonly found in nature, as they usually have a non-null fixed point.

Finally, we noticed that using this learning rule yields much better results:

$$\dot{\omega_i} = -\epsilon g(a_i) \left(\omega_i - \dot{\theta}_i \right) \tag{4.8}$$

Functionally, both equations are equivalent with the single difference that Equation 4.8, has ϵ^2 instead of ϵ . When a local neighborhood coupling is used to compute the phase variation as is the case in Equation 4.9, the difference is even stronger. You can see the behavior of both learning rules, and the effects of the ϵ^2 and of the local coupling in Figure 4.2.

¹The amplitude of oscillations increases suddenly when it's excited by a signal close in frequency.

 $^{^{2}}$ We will refer a lot to the term cluster in this chapter, and we use it to refer to all oscillators that are located in the range of entrainment of a frequency component of the input signal, thus entering into resonance.



Figure 4.2: We present a signal $F(t) = \cos(\omega t)$, with $\omega = 7$, $\mu = 0$, $\epsilon = 0.5$. *Top:* The original learning rule as defined in Equation 4.6, and amplitudes defined as in Equation 4.7. *Bottom:* The final learning rule as given by Equation 4.8, with 1-local coupling.

After many experiments and iterations, we devised the final model. We noticed that coupling each oscillator with its two nearest neighbors (upper and lower) in frequency, and let the oscillator be influenced if its neighbors and itself are resonant, yields a much quicker and precise convergence. You can also see that the oscillator became a limit-cycle one, as the amplitude now admits a non-null fixed point when there is no perturbation. This feature corresponds to the third term of the equation $\dot{\theta}_i$ of system 4.9.

$$\dot{\theta}_i = \omega_i - \epsilon F \sin(\theta_i) + vg(a_i) \sum_{j=i-1,i+1} g(a_j) \sin(\theta_i - \theta_j)$$
(4.9)

$$\dot{a}_i = \left(\mu - \frac{a_i}{\beta}\right) + \gamma F \cos(\theta_i)$$
(4.10)

$$\dot{\omega}_i = -\alpha g(a_i) \left(\omega_i - \dot{\theta}_i \right) \tag{4.11}$$

where α , β , γ , ϵ , v, μ are all positive parameters, and g(x) is a sigmoid as defined in Equation 3.5 with parameter ξ and ρ .

First, a crucial aspect of this model is the way the amplitudes of the oscillators evolve in time. The shape can be modeled by many parameters :

- ϵ defines the width of the basin of entrainment for each component. A larger value allows oscillators with a larger detuning to phase lock to the signal. We used a value of $\epsilon = 1$.
- v defines the influence of the neighboring resonant oscillators, that is how quickly will the resonant oscillators within a group converge towards a common phase. Note also that this term appears also in the frequency dynamics, which means that the frequencies will be influenced in the same way as the phase. We used a value of v = 4.
- μ that is the value of the fixed-point of the oscillator when no input signal is presented. We used a value of $\mu = 0.01$.
- β that is how does the shape of the amplitude evolve (see Figure). The larger these values, the clearer the difference between the amplitudes of neighbors oscillators. We used a value of $\beta = 25$.
- γ regulates how differently will the amplitudes of the resonant oscillators evolve depending on their detuning. The larger this parameter, the more different will be the variation of the amplitudes of the oscillators in a cluster. We used a value of $\gamma = 6$.
- α regulates how quickly will the frequency evolve. A low value should be used, as the frequencies will oscillate around the frequencies they have learned (the mean of the oscillations is that frequency), and the amplitudes of the oscillations are proportional to this parameter. We

used a value of $\alpha = 0.2$. which allows a quick learning of the frequency along with small amplitudes of oscillation in the frequency.

• ξ is the minimal amplitude for an oscillator to be considered resonant enough in order to adapt the frequency. The ρ parameter for the sigmoid g(x) was chosen to be 1, and we choose $\xi = 25$. These value must be tuned according to the task to perform.

4.1.3 Memory recall

We had to define a method assessing whether a signal has been learned or not. With the previous model, we measured the time needed for a fixed number of oscillators to increase their amplitude above a certain threshold. But with the model we propose here, as we are now able to learn complex signals having multiple frequency components, we need more flexibility as the amount of oscillators entering a resonant state can vary according to the complexity of the input signal. As we have explained, for each frequency composing the input signal, a cluster of oscillators will become resonant (see Figure 4.3). When the signal has not been learned, we can see that the amplitudes of the oscillators in the cluster have a parabolic shape, due to the phase difference induced by frequency variation among the members of the cluster. But once the signal has been learned, in which case all oscillators will have the same phase, the shape of the amplitude will be much sharper. A good method to discriminate between learned and non-learned signals is to look at the standard deviation of the amplitude between the members in each cluster. If this value is lower than a certain threshold, this means that all resonant oscillators are tuned to approximately the same frequency, so we can conclude that all these oscillators have learned the signal being presented. The question now is, how do we form these assemblies or how to infer to which group does an oscillator belong? This is another parameter we found empirically, and according to the parameters we propose, we consider that all oscillators having an amplitude greater than 10 are considered to be into a cluster.

The physical implementation of such a process can be easily done using an array of correlation detecting neurons. If all signals arrive with the same phase shifts at the neuron the activation output level should be maximal, reflecting the fact that the variance in the group is minimal.

4.2 Experiments

We tested our model in order to show empirically that it works. Our tests were done with a network of 100 oscillators with frequencies distributed between $\omega_1 = \omega_{min}$ and $\omega_{100} = \omega_{max}$, we presented a signal to the network during T = 1000, and we reset the amplitudes of all oscillators at T = 500.

Cluster :	1	2	3	4
Before learning	0.566405	0.560635	0.432205	0.845195
After learning	0.025737	0.000680	0.001751	0.000154

Table 4.1: Variance in amplitude for each cluster presented in the first experiment taken before learning at time t=50, and after learning at t=550.

Note: In order to be able to make plots of the evolution of the amplitudes as clean as possible, we plotted a time average over 10 time steps of the value of the amplitude, so one unit on the plots corresponds actually to 10t.

4.2.1 First experiment: simple signal

This is the most trivial case : we want to learn a signal where each component has the same coefficient a_i . We trained our network with the following signal

$$F(t) = \cos(3t) + \cos(5t) + \cos(7t) + \cos(9t) \tag{4.12}$$

As you can see in Figure 4.3, everything is good as the oscillators learn the frequencies they need to. You can see the variance in amplitude for the members of each cluster before and after the learning phase in Table 4.1.

4.2.2 Second experiment: simple signal with close components

Now we want to know what happens if the frequency components of the input signal are closer. We want to learn the following signal

$$F(t) = \cos(3t) + \cos(3.7t) + \cos(4.4t) + \cos(5.1t)$$
(4.13)

In this case, you can see in Figure 4.4 that the amplitudes evolve at the same speed in each cluster, but that the frequencies are much more irregular and oscillate with a bigger amplitude as compared to the previous experiment. The reason is that the influence of neighboring oscillator is much greater due to lower detuning between neighboring frequencies. In average, the value is still the learned frequency.

4.2.3 Third experiment: complex signal

We want to show that our model is able to learn any kind of signal, even when the coefficients of the components are different. We want to learn this signal

$$F(t) = \cos(3t) + 2\cos(7t) + 3\cos(11t) \tag{4.14}$$

We see clearly in Figure 4.5 that the width of the basin of attraction increases with the coefficient of the corresponding frequency component, as do the amplitudes. This is a very useful property in that we can retrieve the coefficient



Figure 4.3: Results of the first experiment. You can see how the network learns a signal composed by 4 cosine functions, each with a coefficient 1. *Top:* The amplitudes of all oscillators in the network, with 4 clusters corresponding to each frequency component. Note how the amplitudes grow identically when the signal is presented again, meaning that this component has been learned. *Bottom:* You can see the tuning of the frequencies of the concerned oscillators in the network.



Figure 4.4: Results of the second experiment. *Top:* See how the amplitudes fluctuates much more due to interferences caused by closer frequency components. *Bottom:* You can see the tuning of the frequencies of the concerned oscillators in the network, but now fluctuations are much stronger compared to the previous experiment.

 a_i of each component as we know the number of oscillators that are dedicated for a component having a coefficient $a_i = 1$. So, the more oscillators we use to cover the same range $[\omega_{min}; \omega_{max}]$, the better the resolution of the retrieval.

4.2.4 Fourth experiment: complex signal with close components

This is the same as the previous experiment, but we want to show the effects of different coefficients when the frequencies of the components are close. We learn this signal

$$F(t) = \cos(3t) + 2\cos(4t) + 3\cos(8t)$$
(4.15)

The results of this experiment, shown in Figure 4.6, clearly state the limits of our model. The stronger the coefficients of the different components, the more separated must be the corresponding frequencies, else the stronger will influence neighboring oscillators that should tune to another frequency, as the size of the basins of attraction is proportional to the ratio of energy of the corresponding frequency component to the total energy of the signal. In that case, our criteria which states if a signal has been learned, will note that the variance into each cluster is very low, so this will be a false positive, even though the frequencies were not learned carefully. Moreover, if a component has a very low coefficient compared to the others, the amplitude of the excited oscillators will be very low, so they event won't be considered as resonant and they won't learn the component they should. This is the price to pay for the gain in genericity brought by our model.

4.2.5 Fifth experiment: robustness against noisy signal

This is the same as the previous experiment, but we want to show the effects of different coefficient when the frequencies of the components are close. We learn this signal:

$$F(t) = \cos(6t) + \cos(7t) + \rho(t)$$
(4.16)

where $\rho(t)$ is a gaussian noise between $\left[\frac{-n}{100}, \frac{n}{100}\right]$, with *n* being the percentage of noise added. We tested the network with n = 0, 50, 100 and 200%, in order to get an idea of how robust is our model to external noise. The results are shown in Figure 4.7. You can see that even if we add a great amount of noise, the appropriate frequency is still learned correctly. If the noise would be different for each oscillator, the results would be worse, as the variation of the amplitudes into the same cluster would have been proportional to the amount of noise added, thus the recall procedure would fail.



Figure 4.5: Results of the third experiment. *Top:* See how the amplitudes and width of the different clusters change according to the coefficient of the different components of the signal. *Bottom:* The tuning of the frequencies of all oscillators in the network.



Figure 4.6: Results of the fourth experiment. *Top:* See how the amplitudes and width of the different clusters change according to the coefficient of different components of the signal. *Bottom:* Note how the weaker component learned the frequency of a neighboring component.



Figure 4.7: Results of the fifth experiment, where the frequency components of a noisy input signal are learned. We used in these experiments a) n = 0%, b) n = 50%, c) n = 100%, d) n = 200%. You can see that the frequencies are still correctly learned even when a random value between [-2; 2] is added to the input signal.

4.2.6 Sixth experiment: robustness against unreliable elements

In this experiment we want to show that our model is robust with faulty oscillators. We present the following signal to the network

$$F(t) = \cos(6t) + \cos(7t) \tag{4.17}$$

Then, at time t = 50, a user-specified ratio of all oscillators are disabled (in this experiment we removed 50% of them), *i.e.* their amplitudes are set to 0, and their phase and frequency are not changed any more. You can see on Figure 4.7 how does the network behave when such oscillators are disabled. Fortunately, removal of the members has no influence on the amplitudes of individual oscillators. The only noticeable effect is that we now have many small clusters of resonant oscillators, instead of the two we had before disabling the oscillators, but this does not change the results of our retrieval methodology as the variance in all these newly formed clusters is even lower than before. In this idea we have shown experimentally that our model is robust even when oscillators are prone to failure.

4.3 Performance comparison

As mentioned before, the maximal performance one can reach with our model is directly a function of the number of oscillators one wants to dedicate for each component we want to learn. This issue is clearly a tradeoff between capacity and reliability. The more oscillators are dedicated to a single component the more reliable will the retrieval be. This issue is even less trivial compared to the first model we analyzed in Chapter 3, as our model is able to learn signal composed of multiple frequencies, and each component will monopolize a full cluster. That means, the more complex our signals are (without overlapping frequencies), the less of them can be memorized within the same group. Additionally, as we have just shown, the stronger are the coefficients the more oscillators will be used per component, also reducing the capacity of the network. The advantage of our model is that we can restrict much better the quantity of oscillators that resonate when excited by the input signal, so we have a more efficient control over the number of oscillators dedicated to learn a frequency component.

However, the problem of this model is that if we want to memorize multiple frequencies with overlapping basins of attraction, there's a potential that the latest signal will attract all oscillators having a too close natural frequency, so all oscillators that were used to memorize the previous frequency will be used for the new one if this frequency is located in the basin of attraction of the new frequency to learn.



Figure 4.8: Results of the sixth experiment. *Top:* See how the amplitudes of the different clusters are identical after removal of some oscillators. *Bottom:* You can see that the correct frequencies are still memorized, even if some oscillators are removed.

4.4 Results and discussions

We have proposed here an interesting model of an associative memory able to learn any kind of complex periodic signal. Even if we do not argue that our model is biologically plausible, many features of this system are similar to the real processes involved in the mammalian brain. As for example, a local coupling between units and a learning method based on the correlation between units activity and the input stimulus. Also, our retrieval methodology states that the level of synchrony between oscillators activated by an input signal is a function of the habituation of the oscillators to the presented signal, that is the more synchronous is the oscillatory activity within a cluster, the more has the corresponding frequency of the oscillations been presented to the network.

Using this method we gained a lot of things:

- First we don't need a global coupling between all oscillators, which is of course much more computationally efficient.
- Then we don't need to use multiple dimensions to encode a single signal, which is also more computationally efficient.
- The only parameter we really need to tune is the width of the cluster of resonant oscillators activated by a standard signal. The other parameters should be adapted to shape the amplitudes evolution, which will definitely depend on the task at hand.
- The behavior of these oscillators can be more easily analyzed numerically as the model studied in the previous chapter, because it does not use a strong non linear term as is the sum regulating the phase-locking behavior, nor does it use random phase shifts that influence strongly the performance of the system. These increased simplicity makes the control of the quantity of oscillators that will be dedicated to learn a single component frequency more easy to perform.
- We don't need to reset the time after stimulus presentation, nor present the input signal with a specific phase relationship as this model is not sensitive to the phase differences.

4.5 Future work

Spatial storage location

In order to implement a fully functional version of our model, we shall need a dynamic mechanism that chooses the spatial location where to store each input signal according to its properties, so as to avoid the problem we face with our model when the frequency components are too close and the learning might be affected.

Oscillator coupling

We must define a dynamic and adaptive connectivity between oscillators, and find a way to change weights between oscillators and/or groups. For the moment, we decided to implement a fixed local neighborhood restricted to the 2 nearest oscillators.

Components links

A very interesting feature we envisioned would be to have a link between different oscillator assemblies in order to be able to have a more robust and biologically plausible model. Our idea is to create links between all the clusters activated by a single signal. This would permit to recall signals where a frequency component is weakened or missing, emulating the associative learning process. However, we must use a more complex Hebbian learning rule for these connections, in order to prevent all clusters being activated upon a single signal presentation, and also to prevent that all signals become associated altogether.

Random phase shifts

The real interest is to reduce the effects of the random initialization of the phase shifts between groups which has a strong impact on the performance of the network, and rather take advantage of this feature as randomness can significantly increase the robustness of the system.

Too many parameters

Even if this model has many parameters, many of them only affect the scale in time and amplitude of the system. Basically, one should need only to define the width of the basins of attraction and the time to learn a desired input signal, and according to these parameters, the shape of the amplitudes in a cluster and the g(x) function should be tuned automatically.

Learn action sequences

One could be interested into implementing "virtual delay lines" between clusters able to capture the timing between each step of a sequence of actions, in that the oscillators are more resonant when multiple actions are presented in the same order as they have been learned.
Chapter 5

Conclusion

"Memory is the mother of all wisdom." Aeschylus (525-456 B.C.)

Computers can do very fast computation that would require centuries for as single human, but humans can recognize patterns so much better than computers. The mechanism used by humans into face recognition for example, are much more efficient even if the raw switching speed in the brain is much slower compared to computers. Early associative memory models proposed some twenty years ago are well-suited to be implemented efficiently onto a computer, but these idealized model do not reflect properties of real neurons, which are not binary and synchronous, but rather very complex and highly dynamic computational units.

Mother Nature designed impressive "machines" able to perform highly complex tasks with very low energy requirements and processing power by using a totally different framework from the traditional computing paradigms used in computer science. Even if very accurate model of neurons, such as the Hodgkin-Huxley, were devised half a century ago, neuroscience is still in its childhood, as these models do not explain the roots of human cognition. It is clear that our intellectual abilities cannot be explained simply by understanding the behavior of individual neurons. Intelligence is an emergent property that arises due to interactions between a huge amount of "simple" units. A careful analysis of these interactions and of their emergent properties are necessary to allow the design of realistic models of cognitive processes with similar performance. Neuroantomy revealed that neurons are neither fully connected, nor statistically distributed but are rather organized in spatially distributed in clusters of sparsely connected neurons linked by long-range connections. It is also admitted that oscillatory activity of neurons is ubiquitous in the brain. In this direction, we have analyzed in this project an associative memory model based on such oscillatory activity. We discussed the properties and limitations of this model, and we try to factorize what aspects are essential into modeling of memorization processes. In the second part of this project, we propose a similar model of memory we devised by taking inspiration of a single oscillator which able to learn any kind of periodic complex signals. We have shown that our model based on properties of oscillating systems is robust against noise and units failure.

I think that the real payoff of this project is what it might provide in terms of brain research and neuronal modeling. Not because we think that the model we propose in this project can account for the real processes involved in biological memory, but rather because we have shown from a mathematical point of view that reliable storage of information encoded as a complex periodic signals can be done by using populations of limit-cycle oscillators that are able to synchronize.

Development of a good and accurate model of the brain that is able to imitate the processes involved into reasoning might help into understanding the operational basis and how it developed into the organ. This could give us a tool that can be used in order to heal mental disorders, repair broken vertebras or even lead to the development of brain-machine interfaces. In the long run, this could even allow engineers to design artificial entities with cognitive abilities that can compete with ours. However, this perspective raises ethical questions that are far beyond the scope of this project.

It can be quite astonishing to imagine that we could have one day a computer composed of biological parts replacing silicon, and this idea provides an interesting alternative to the well-known universal computing paradigm of the Turing Machine that has been dominating the computing domain. Who knows, maybe that day we would have to feed our computer before using it instead of plugging a power chord.

Appendix A

Installation of the package

Unzip the file assmem.zip in any directory, let's say ASSDIR. Then you have to recompile the source of the simulator. In order to do that modify the file ASSDIR/Makefile, and add the path of GSL (GNU mathematical library). Then type into the terminal

[user@mycomputer ASSDIR]\$ make

Once the files needed have been recompiled, start MATLAB from the ASSDIR directory and type assmem in the command window in order to launch the GUI. A general view of the user interface that appears is shown in Figure 5.1.

This interface allows user to test both the original model (described in Chapter 3) and the modified version we devised (described in Chapter 4). The upper panel contains general parameters used by both models. Please refer to the report in order as reference of the different parameters you can tune. Then change the parameters and click on Run simulation, in order to execute a simulation immediately. The Create sim is used in order to create a set of experiments (the scripts that will be fed into the simulator), you can launch in batch (serial processing) by using the test.sh shell script located into the ASSDIR directory. The directory ASSDIR/simulations will contain a folder for each experiment you create and it will contain the script to run the experiment, the log file resulting from the experiment, and the plots in the experiment.

Simulator

The simulator is run using the following command:

./assmem simname

A = = 1.4	a sea N La A	_ Simulations ge	neral parameters			
Assimeminet		#osc/or	000	wmin	5	
		Time s	50	wmax	8	
(C) June 2005, Vlac	1 TRIFA, LSL, BIRG, EPFL	rine s	0.1			
Exclusive design by Vlad TRIFA		Sim ur	100 100			
Coriginal model				-New model		
4 Crowns				al	Jpha 0.2	
# Groups	1			b	Jeta 25	
Input aimension	1			ga	amma 6	
alpha	1	Plots			v 4	
beta	4	Plot selector		ep	asilon 1	<u> </u>
gamma	4	The selector			mu 0.01	_
v	0.5				0.01	
w	16					
Tau	0			_ Panel		
Threshold(H)	450	🖌 Amplitude		xi [25 rbo	
		Amplitude 3D		Ninhin [20 1110	
				Nyno	Edit Text Minamp	10
Sigmoid parameters		Frequencies			Checktime	Edit Text
		_Groups activit	y			
xi1 0.7 rho1 0.02				Signal F		
xi2 rho2		Plot exports		F#)	*cos(**) +
0.88 0.02				r(()=	1 000 6	
		_Save plots			1 *cos(7	*t) +
					0 *cos(8	*1)
				Пи	loise	
Analyze					aults Amount (%)	50
Run simulation Cre	eate sim open			Run simi	ulation	

where *simname* is the name of a simulation created using the MATLAB interface.

Figure 5.1: The main interface of the MATLAB application. You can see on the left the panel that controls the simulation using the model analyzed in Chapter 3, and on the right the panel to control the model we described in Chapter 4.

Scripts

The signals are presented to the network by means of a script appended to the end of the configuration file (the simname.log), which contains a line for each stimulus to present. Here's an extract of such a script:

```
L 7 3
L 7 3
L 7 3
```

The syntax is very simple as each line has three tokens. The first is the action (L is for learning, no other actions were implemented), the second is the frequency to present, and the last token is the duration the stimulus is presented.

Bibliography

- Anderson, J. A. and Rosenfeld, E. (1988). Neurocomputing. MIT Press.
- Aoyagi, T. (1995). Network of neural oscillators retrieving phase information. *Phys. Rev.*, **74**, 4075–4078.
- Aoyagi, T. and Nomura, M. (1999). Oscillator neural network retrieving sparsely coded phase patterns. *Phys. Rev.*, 83, 1062–1065.
- Arbib, M. A. (1995). The handbook of brain theory and neural networks. MIT Press.
- Borisyuk, R. and Borisyuk, G. (1997). Information coding on the basis of synchronization of neuronal activity. *Biosystems*, **40**, 3–10.
- Borisyuk, R. and Hoppensteadt, F. (1998). Memorizing and recalling spatialtemporal patterns in an oscillator model of the hippocampus. *Biosystems*, 48, 3–10.
- Borisyuk, R. and Hoppensteadt, F. (1999). Oscillatory models of the hippocampus: a study of spatio-temporal patterns of neural activity. *Biol. Cybern.*, 81, 359–371.
- Borisyuk, R., Denham, M., Hoppensteadt, F., Kazanovich, Y., and Vinogradova, O. (2001). Oscillatory model of novelty detection. *Computation in neural systems*, **12**, 1–20.
- Borisyuk, R., Borisyuk, G., Kazanovich, Y., and Ivanitskii, G. R. (2002). Models of neural dynamics in brain information processing the development of the 'the decade'. *Physics Uspekhi*, **45**(10), 1073–1095.
- Buchli, J. and Ijspeert, A. J. (2004). Distributed central pattern generators for robotics application based on phase sensitivity anlysis. In Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT 2004).

- Buszáki, G. and Draguhn, A. (2004). Neuronal oscillations in cortical networks. *Science*, **304**, 1926–1929.
- Crick, F. and Koch, C. (1990). Towards a neurobiological theory of consciousness. seminars in The Neurosciences, 2, 263–275.
- Davey, N., Hunt, S., and Adams, R. (2004). High capacity recurrent associative memories. *Neurocomputing*, 62, 459–491.
- Ermentrout, B. G. and Chow, C. C. (2002). Modelling neural oscillations. *Physiology and Behavior*, 77, 629–633.
- Gerstner, W. (2002). Neural networks course. LCN, EPFL.
- Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models*. Cambridge University Press.
- Haykin, S. (1999). Neural networks, a comprehensive foundation. Prentice Hall.
- Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci., 79, 2554–2558.
- Hoppensteadt, F. and Izhikevich, E. M. (2001). Canonical neural model. Inivted chapter in Arbib MA, Brain theory and Neural Networks.
- Hoppensteadt, F. C. and Izhikevich, E. M. (1999). Oscillatory neurocomputers with dynamic connectivity. *Physical Review Letters*, 82, 2983–2986.
- Kandel, E. R. (2000). Principles of neural science, volume Thomas M. Jessel. McGraw-Hill.
- Kohonen, T. (1989). Self-organization and associative memory. Springer-Verlag.
- M.Borisyuk, R. and Kazanovich, Y. B. (2004). Oscillatory model of attentionguided object selection and novelty detection. *Neural Networks*, 17, 899– 915.
- Murray, J. (2002). *Mathematical biology*, *I. an introduction*. Springer-Verlag, Berlin.
- Nishii, J. (1998). A learning model for oscillatory networks. Neural Networks, 11, 249–257.
- Nishii, J. (1999). Learning model for coupled neural oscillators. Neural Networks, 11, 249–257.

- Nishikawa, T., Lai, Y.-C., and Hoppensteadt, F. C. (2004). Capacity of oscillatory associative-memory networks with error-free retrieval. *Phys. Rev. Lett.*, **92**.
- Pikovsky, A., Rosenblum, R., and Kurths, J. (2001). Synchronization, A universal concept in nonlinear sciences, volume 12 of Cambridge Nonlinear Science Series. Cambridge University Press, Cambridge, UK.
- Righetti, L., Buchli, J., and Ijspeert, A. J. (2005). Dynamic hebbian learning in adaptative frequency oscillators. *Physica D*, 0, 00–00.
- Seliger, P., Young, S. C., and Tsirming, L. S. (2002). Plasticity and learning in a network of coupled phase oscillators. *Phys. Rev. Lett.*, 65, 359–371.
- Singer, W. (1993). Synchronization of cortical activity and its putative role in information processing and learning. Annu. Rev. Neurosci., 55, 349–374.
- Singer, W. and Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. Annu. Rev. Neurosci., 18, 555–586.
- Sommer, F. T. and Wennekers, T. (1999). Models of distributed associative memory networks in the brain. *Physical Review Letters*, 82.
- Strogatz, S. (1994). Nonlinear dynamics and chaos. With applications to physics, biology, chemistry, and engineering. Addison Wesley Publishing Company.
- von der Malsburg, C. (1995). Binding models of perceptions and brain function. Current Opinion in Neurobiology, 5, 520–526.