UNIVERSITÁ DEGLI STUDI DI GENOVA

DOCTORATE SCHOOL Humanoid Technologies - XXI Cycle

Thesis for the degree of Doctor of Philosophy

Nonlinear Dynamical Systems and applications to Robotics



Boris A. Durán

Supervisors: prof. Giulio Sandini prof. Giorgio Metta ii

Contents

Summary				
A	ckno	wledgments	vii	
In	trod	uction	ix	
Ι	$\mathbf{T}^{\mathbf{I}}$	ne Theory behind Dynamical Systems	1	
1	Inti	roduction	3	
	1.1	Dynamical Systems	3	
	1.2	Time	4	
2	Nonlinear Dynamical Systems			
	2.1	Linear Dynamical Systems	8	
	2.2	Fixed points	11	
	2.3	Periodicity	15	
	2.4	Chaos	19	
	2.5	Summary	24	
3	Neı	ıral Networks	25	
	3.1	Biological Neural Networks	26	
	3.2	Feedforward Networks	28	
	3.3	Recurrent Networks	29	
	3.4	Summary	36	

II	Α	pplications in Robotics	37			
4	Visual-motor behaviors					
	4.1	Introduction	39			
	4.2	Coupled Chaotic Systems	40			
	4.3	Methodology	42			
	4.4	Results	45			
	4.5	Summary	53			
5	Dynamic Field Theory					
	5.1	Introduction	57			
	5.2	Dynamic Field Theory	58			
	5.3	The "A-not-B" paradigm	61			
	5.4	Methodology	62			
	5.5	Results	68			
	5.6	Summary	70			
6	Con	aclusions	73			
\mathbf{A}	Published Articles 7					
	A.1	Paper I	. 77			
	A.2	Paper II	82			
	A.3	Paper III	. 89			
Bibliography						

Summary

The increasing complexity of humanoid robots and their expected performance in real dynamic environments demands an equally complex, autonomous and dynamic solution. The main goal of this thesis is to propose the first steps toward the implementation of a dynamic and flexible cognitive architecture founded on the *enactive* paradigm of cognition. The approach is based on the strategic use of the tools provided by nonlinear dynamical systems theory and will propose a novel way of controlling the behavior of artificial agents. Coupled chaotic systems are also studied as a part of nonlinear dynamical systems theory due to the richness of their dynamics. The thesis is grouped in two main parts (besides the Introduction): Part one introduces the theoretical framework of our approach including a review of neural networks with a special focus on recurrent configurations; Part two describes two projects that implemented the theory studied in Part one.

Acknowledgments

I would like to express my gratitude to Prof. Giulio Sandini for his support and guidance during these years. I really cannot thank him enough for his motivating words and continuous faith in myself. I am also thankful to Prof. Giorgio Metta for his support, advice and friendship. Further I acknowledge my gratitude to the people from the Robotics, Brain ans Cognitive Sciences department at the Italian Institute of Technology for their help and support. To all my colleagues and friends for all those, not always scientific but always fruitful social interactions, especially to Luca and Vishu.

I must acknowledge the wonderful work of Prof. Gregor Schöner and his students. Even though I came across his model late in my research, I managed to get a good grip on his theories. I owe a special note of gratitude to Christian Faubel for the hand he gave me with the understanding and implementation of the DFT model, without him I could not have been able to complete my work.

I want to thank my parents Zita and Boris for their faith and encouragement throughout my life. Their support and love put the foundation for building the person I became, a fighter for my dreams, myself and the people I care. I want to thank my sisters Vanessa and Karina for being there whenever I needed them, and in particular for coming to me whenever they were in need. Above all I am grateful to my loving girlfriend Lisa for her patience and support during these years; I wouldn't have been able to accomplish any of this if it wasn't because of you sharing this adventure with me.

The work presented in this thesis has been partially supported by the RobotCub project (IST-2004-004370), funded by the European Commission through the Unit E5 -Cognitive Systems.

Introduction

Research in humanoid robotics dates back approximately 30 years and was founded on a strong dualist view of human nature. On one hand, the body of an agent¹ has been controlled by using a 50-year-old tradition of control theory that started with industrial automation at the beginning of the 1960s. And on the other, the mind of an has been treated independently of its body by using *cognitivist* approaches, which was a very promising area of research that gave birth to what it is known as *artificial intelligence*.

Cognitivism is one of the two main paradigms of cognition and has been relatively successful in solving task specific problems. This approach sees cognition as a set of computations defined over symbols or representations of the world; these representations are pre-designed by a human programmer [62]. The main problems this approach has been and is still facing are due to its dependency on the programmer's knowledge of the world: the symbol grounding problem, the frame problem, and the combinatorial problem [61].

The focus of the present thesis is on the *enactive* paradigm of cognition. This paradigm sees cognition as a process of co-determination between the environment and the agent. In other words, the agent constructs its reality through the interaction with the world [61]. Therefore, in contrast with the *cognitivist* approach, the mind can not be independent of the body and the knowledge about the world is not predefined by the designer².

Related work

Based on experimental results, several groups have suggested the presence of nonlinear (chaotic) dynamics in the brain [16, 28, 7]. With little differences

 $^{^1{\}rm The \ term \ }agent$ will represent in this book an artificial entity used for the simulation of human behavior

²See [62] for a thorough study on cognitive systems.

on the exact role of chaos, these results show a dynamic link between perception and memory built within a nonlinear dynamic space. Even though the different models derived from these studies have reported great improvements when compared to conventional neural networks, a final mechanism of actuation in the same dynamic terms is missing in all of them. In other words, only perception and memory are being addressed in these models, but they do not consider motor outputs as part of these dynamic systems; in some cases they discretize the output space [19], thus loosing the efficiency gained before.

On the other hand, several groups have focused their research in the reflexive part of their agents [43, 33, 32, 49, 15]. In these models actuation is performed without considering the history of the agent. These and other groups have developed remarkable advances in adaptive behavior but, in contrast with the previous group, only perception and actuation is being addressed; thus leaving memory out of sensori-motor loop.

In the same way, nonlinear dynamical systems has found several followers within Developmental Psychology [58, 2, 56]. Esther Thelen and Linda Smith support, among others, a theory of ontogenetic development based on the concepts of dynamic adaptation and flexibility found in dynamical systems. According to them, cognitive agents can not be modeled as simple causeand-effect systems but as systems with a history that changes them over time [58].

Current models of cognition make little or none use of nonlinear dynamical systems theory. Most cognitive architectures belonging to the *enactive* paradigm of cognition use modular versions of feed-forward neural networks for classification and regression [9, 53, 10, 30]. This type of networks are nonlinear static networks, i.e., a given input is associated with a given output and remains in a steady state as long as the input remains the same. This leads to a reduced memory capacity and a great dependence to the quality of the training datasets. Other architectures like the Self-aware and Self-effective (SASE) cognitive architecture [34], use statistics in the form of Markov Decision Processes. Here again it is necessary to find other ways for saving, addressing and retrieving memories which results in expensive computation processes.

Motivation

Modern and classical control theories have been the two frameworks applied for controlling any kind of autonomous system. They have proved to be very accurate and efficient inside industrial environments where machine and environment can be modeled precisely since they work within fixed spaces. Even though mobile and specially humanoid robotics are pushing us to reconsider the usefulness of this approach in dynamic an unpredictable environments, several state-of-the-art platforms keep using the traditional tools of control theory. Albeit the exponential growth of computational power has helped to deal with the expensive treatment of inverse kinematics/dynamics and environment modeling of these systems, they fail every time they find a situation that demands fast reactions or motions that were not coded by the programmer.

A dynamic, flexible and autonomous cognitive architecture is needed today more than ever to get a better understanding of human nature. A system based on nonlinear dynamical systems theory would be of great importance for the study of areas such as epilepsy [21, 54], developmental psychology [58, 2, 56], psychotherapy [46, 45], motor control [18, 12, 20], neurosciences [17, 47, 59], and many others. At the same time, a cognitive architecture like the one proposed here will be of great use for, but it will mainly try to integrate several areas of scientific research like human-robot interaction, imitation, motor control, computer vision, and machine learning. Each one of them has demonstrated to be a complex subject and integrating them will be a challenge on its own.

Objectives

The main goal of our research is to propose the first steps toward the creation of a cognitive architecture that, using the mathematical tools from nonlinear dynamical systems theory, integrates the information coming from sensors with the information coming from an internal space (memory), and that finally modulates the motor outputs inside a reflexive physical layer. The major contribution of a system like the one proposed here is the possibility of having a dynamic sensori-motor loop that includes history as a modulator in the decision making step for the final motor behavior.

Among other advantages of working in specific chaotic regimes like *chaotic*

itinerancy [60], with respect to non-recurrent neural networks, it is possible to mention the dynamic retention of information, increased learning capabilities, improved patter recognition, efficient search of memories, memories can be represented by dynamic processes and not only as static patterns, simultaneous process of learning and recalling [59].

The design and implementation of a completely autonomous dynamic cognitive architecture will have its starting point in carrying out the proposed thesis by achieving the following goals:

- Design and implementation of a reactive physical layer following the approach of Pasemann and colleagues. Motor behaviors in their platforms are created by either evolving recurrent neural networks [43] or, whenever the input-output relationships are simple enough, by manually tuning excitatory and inhibitory connections [35].
- Depending on the experimental setup, a reflexive system will be designed first by studying optimal configurations of recurrent neuromodules. The properties of hysteresis found in recurrent neurons are of special interest in the pre-processing of sensory signals and will be studied and used carefully in the platforms.
- Once having an optimal reactive physical layer working, it will be time for focusing in a higher level of processing incoming information. The most important component in the 'mind' of our system is a dynamic memory and it will be necessary to study advantages and disadvantages of different approaches.
- Long-term memories, life-long memory systems and on-line learning has been mentioned as some of the main characteristics of all approaches studying chaotic dynamics for saving information, however there has been no methodology reporting the implementation of these ideas. From our point of view, a way of implementing these features should be studied in parallel when designing an autonomous cognitive architecture. After implementing a dynamic memory, we will focus on the methodology for creating on-line and life-long memory processing.
- This same approach will be studied for different kinds of sensor information; i.e., sound, visual, tactile, proprioceptive, etc. All of them will

be integrated in the dynamics of the network, in other words, a memory of an event or a sequence of events will be created as an attractor in this nonlinear dynamic space fed by different sources.

• Finally, it will be necessary to couple the higher level of information processing with the physical layer. A methodology for modifying a simple reactive behavior with the information gathered through time and saved in the 'mind' network will be designed and implemented.

Preprocessing of information has probed to be a difficult task for many areas such as vision, sound, and proprioception; however it is expected that the use of nonlinear dynamical systems theory will facilitate their integration. This is on its own, another very important goal derived from the proposed methodology.

Part I

The Theory behind Dynamical Systems

Chapter 1 Introduction

In this part of the book we will describe the basic theory behind the different tools used in our attempt to create a more dynamic cognitive architecture. The following section gives a very brief introduction to cognitive systems. The rest of this introduction contains some basic definitions and notations related to Nonlinear Dynamical Systems that will be studied in more detail in Chapter 2. Neural Networks in their different configurations will be described in Chapter 3 since they also form part of the powerful tools used in robotics research.

1.1 Dynamical Systems

Anything that changes its *state* over time is consider a dynamic system. To describe a dynamical system it is necessary to have access to its *state* and to the *function* that defines how the system behaves as time flows.

State of a system

The *state* of system is defined as the value or values representing one or multiple characteristics of the studied system e.g., position, velocity, temperature. Throughout this book the following notation will be used for representing the state of a dynamical system:

- A single number will be represented by a lowercase letter as in x_1
- A vector will be represented by a bold lowercase letter as in $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Transfer function

Dynamical systems evolve through time thanks to some specific function of one or more variables that define the *dimension* of the system. As we can imagine the more variables a phenomenon has the more complex dynamics it will show. The transfer function, sometimes called *evolution rule* is defined by differential equations, Eq.(1.1), when working with continuous time or by difference equations, Eq.(1.2), when working with discrete time and in this case the function is usually called a *map* or *iterated map*.

$$\dot{\mathbf{x}} = f(\alpha, \mathbf{x}, t); \qquad \alpha \in \Re, \mathbf{x} \in \Re^n, t \in \Re^+.$$
 (1.1)

$$\mathbf{x}(k+1) = f(\alpha, \mathbf{x}(k), k); \qquad \alpha \in \Re, \mathbf{x} \in \Re^n, k \in \aleph.$$
(1.2)

A dynamical system is called *deterministic* when the function governing its internal dynamics does not include random or noisy inputs or parameters.

1.2 Time

Time is the main motor in any dynamical system, it helps us to move toward our future state starting from our present state. This motion through different states can be analyze through two different glasses: *continuous*, a smooth progress in time, or *discrete*, through periodic intervals of time.



Figure 1.1: Typical plots of one-dimensional functions: continuous time (left) and discrete time (right).

Continuous

Differential equations are the mathematical form to describe any dynamical system moving continuously in time. In this type of description, our variables depend on t which is a non-negative real number since we start our analysis at time t = 0. Since t is changing continuously, differential equations can not tell us what our state will be in the 'next' instant, however they can help us to know how our system is changing at a given instant, Fig. 1.1. Examples of dynamical systems moving continuously in time are:

- An object thrown up into the air with gravity pulling it downward continuously.
- The dynamics of pendulum problems in general.
- Electrical circuits.
- Chemical reactions.
- Electromagnetic flows.

Discrete

Discrete dynamical systems make use of difference equations for predicting the state of a system at any given instant given the initial conditions (state and parameters) and the rule or map that governs its dynamics. Analyzing a system in discrete time has several advantages over differential equations, among others: difference equations reduce significantly the expensive computational cost of differential equations especially for systems with large dimensions, the mathematics of difference equations are much easier to analyze and follow, real-time autonomous agents work also in discrete time with an integer number of sensors and actuators. Throughout this book we will focus on discrete dynamical systems mainly. If it is true that most systems in nature change continuously over time, in some cases it has much more sense to analyze their behavior in periodic intervals of time. Some examples of discrete dynamical systems are:

- Micro/Macro economics: financial markets, currency rates, savings, etc.
- Climate change.
- Population growth and decay.

Chapter 2 Nonlinear Dynamical Systems

The theory of nonlinear dynamical systems tries to study and model all physical phenomena that go through spatial and temporal evolution. The range of phenomena studied with this theory goes from simple pendulum analysis to the motion of planets and celestial bodies, from chemical reactions to lasers and telecommunications. During the last decades the spectrum of scientific communities interested in nonlinear dynamical systems theory has become wider, e.g. neurosciences, economics, sociology, biology, psychology; and with them the theory has also grown significantly. The amount of knowledge accumulated during the years is enormous and would require several books to describe the whole scope of this theory. In this chapter will try to describe the basic concepts and tools of the theory with a special focus on chaotic dynamics since it is the angular stone where this thesis is based.

A dynamical system is said to be 'nonlinear' when it can not be mathematically described as the linear sum of its components, i.e. it includes terms like products (x_1x_2) , powers (x^3) , and functions of x $(sinx, e^x)$. Almost all systems in nature behave nonlinearly and mathematical modeling of this type of systems is often very difficult and in some cases impossible. One option is to divide the problem in very small parts that behave linearly and find solutions for them, the major drawback of this approach is that it can not see changes of the dynamics at macro scales. Instead of trying to find exact solutions for nonlinear problems it is much easier to study the long-term behavior of the system and later focusing on interesting points of changing dynamics. Throughout this chapter we will present several of the tools that are used to understand the qualitative nature of nonlinear dynamical systems.

The following section presents the simplest case of nonlinear dynamical

systems, the linear case. By studying a linear function we will be able to compare the difficulties encountered when trying to solve nonlinear problems analytically. The other sections will focus on the road to chaos by perioddoubling starting from fixed points.

2.1 Linear Dynamical Systems

Linear dynamical systems could be considered as the simplest cases of nonlinear dynamical systems. This type of systems can be mathematically described as the linear sum of its components, i.e. a combination of elements to the first power only. It is always possible to find algebraic solutions for these systems; unfortunately, in nature there are not many examples of dynamical systems that behave linearly. To have a better picture of how they look and how they can be analyzed, we will use the general form of a one-dimensional function.

$$x(k+1) = a_0 + a_1 x(k);$$
 $x, a_i \in \Re, k \in \aleph.$ (2.1)

First, let's consider the case when $a_0 = 0$:

$$x(k+1) = a_1 x(k);$$

$$x(1) = a_1 x(0);$$

$$x(2) = a_1 (a_1 x(0)) = a_1^2 x(0);$$

...

$$x(k) = a_1^k x(0);$$

(2.2)

As k goes to infinity, the dynamics of this system depends on the value of constant a_1 and can fall in one of the following cases:

- 1. For $|a_1| < 1$, $a_1^k \to 0$ as $k \to \infty \Rightarrow x(k) \to 0$ as $k \to \infty$; our variable is attracted to zero as time goes by, Fig. 2.1a.
- 2. For $|a_1| > 1$, $a_1^k \to \infty$ as $k \to \infty \Rightarrow x(k) \to \infty$ as $k \to \infty$. Unless our initial value x(0) = 0, our variable becomes unstable and explodes to infinity, Fig. 2.1b.
- 3. For $a_1 = 1$, $a_1^k = 1 \Rightarrow x(k) = x(0)$; our variable x remains constant on its initial value as time goes by, Fig. 2.1c.

2.1. LINEAR DYNAMICAL SYSTEMS

4. For $a_1 = -1$, $x(0) = -x(1) = x(2) = -x(3) = x(4) \dots$ Our variable oscillates between a positive and a negative value of magnitude x_0 , Fig. 2.1d.

Finding fixed points and their stabilities by means of an analytical analysis gives us certain degree of confidence on our results. There is however a much simpler way of finding fixed points and their behaviors: graphically. The different plots in Fig. 2.1 are called *cobweb* plots and respond to a very simple principle of graphical iteration. The studied function is plotted together with a bisector and starting with x(0) on the x-axis we draw a vertical line from x(0) until hitting the function. Then a horizontal line is drawn until reaching the bisector, a new vertical to the function, and a new horizontal to the bisector, and so on. This kind of plot depicts the progress in time of the iterations within the function, thus showing how stable or unstable our fixed points are. Plots of activations versus time are also shown in each one of the examples.

Next, considering the case when $a_0 \neq 0$:

$$\begin{aligned} x(k+1) &= a_0 + a_1 x(k); \\ x(1) &= a_0 + a_1 x(0); \\ x(2) &= a_0 + a_1 x(1) = a_0 + a_1 (a_0 + a_1 x(0)) \\ &= a_0 + a_0 a_1 + a_1^2 x(0); \\ x(3) &= a_0 + a_1 x(2) = a_0 + a_1 (a_0 + a_0 a_1 + a_1^2 x(0)) \\ &= a_0 + a_0 a_1 + a_0 a_1^2 + a_1^3 x(0); \\ & \dots \\ x(k) &= a_0 (1 + a_1 + \dots + a_1^{k-2} + a_1^{k-1}) + a_1^k x(0); \end{aligned}$$
(2.3)

In order to further simplify the first term of the right-hand side of Eq.(2.3) we multiply and divide the terms within parenthesis by (a - 1):



Figure 2.1: A graphical method for finding fixed points and their stabilities when $a_0 = 0$ in Eq.(2.1)

$$\frac{a_{1}-1}{a_{1}-1}(1+a_{1}+\dots+a_{1}^{k-2}+a_{1}^{k-1}) = \\
= \frac{1}{a_{1}-1} \left[(a_{1}-1)1+(a_{1}-1)a_{1}+\dots+(a_{1}-1)\frac{a_{1}^{k}}{a_{1}^{2}}+(a_{1}-1)\frac{a_{1}^{k}}{a_{1}} \right]; \\
= \frac{1}{a_{1}-1} \left[a_{1}-1+a_{1}^{2}-a_{1}+\dots+\frac{a_{1}^{k}}{a_{1}}-\frac{a_{1}^{k}}{a_{1}^{2}}-\frac{a_{1}^{k}}{a_{1}}+a_{1}^{k} \right]; \\
= \frac{a_{1}^{k}-1}{a_{1}-1};$$
(2.4)

$$\hookrightarrow x(k) = \begin{cases} a_0 \frac{a_1^k - 1}{a_1 - 1} + a_1^k x(0); & a_1 \neq 1 \\ k a_0 + x(0); & a_1 = 1 \end{cases}$$
(2.5)

Here again the value taken by a_1 divides our analysis in four different cases:

- 1. For $|a_1| < 1$, $a_1^k \to 0$ as $k \to \infty \Rightarrow x(k) \to \frac{a_0}{1-a_1}$; our system is attracted to a constant value independent of x_0 , Fig. 2.2a.
- 2. For $|a_1| > 1$, $a_1^k \to \infty$ as $k \to \infty$. Let's reorganize Eq.(2.5) on a^k to analyze a bit further.

$$x(k) = a_0 \frac{a_1^k - 1}{a_1 - 1} + a_1^k x(0) = a_1^k \left(x(0) - \frac{a_0}{1 - a_1} \right) + \frac{a_0}{1 - a_1}$$

In this case, the final behavior of our system depends on the value of x(0): if $x(0) = \frac{a_0}{1-a_1}$ then our system remains constant at that value forever, otherwise it becomes unstable and explodes to infinity, Fig. 2.2b.

- 3. For $a_1 = 1$, the system will keep growing as k grows for any $a_0 \neq 0$; or will remain at x(0) otherwise, Fig. 2.2c.
- 4. For $a_1 = -1$, our variable oscillates between x_0 and $a_0 x(0)$, Fig. 2.2d.

Since the only effect of having $a_0 \neq 0$ is to shift our function away from the origin, it is possible to see qualitatively similar dynamics than when $a_0 = 0$. It is only when $a_0 \neq 0$ and $a_1 = 1$, Fig. 2.2c that the behavior changes from the previous case, now all initial values are taken to $+\infty$ if $a_0 > 0$ or to $-\infty$ if $a_0 < 0$ unless $x(0) = a_0$. When $x(0) = a_0$, x(k) = x(0) for all cases.

2.2 Fixed points

When analyzing the example in the previous section we saw that for some values of our parameters our system remained *fixed* (constant) in some specific value regardless of the initial value of x. In nonlinear dynamical system we will usually find this type of *attractors* and we call them *fixed points*. In a more strict mathematical sense we would say that a value x^* is called a fixed point if it satisfies $f(x^*) = x^*$.

In order to understand a bit more what the different *attractors* are and how they can be analyzed, we will use a special function from the family of quadratic maps, Eq.(2.6a). The logistic map, Eq.(2.6b), was studied



Figure 2.2: A graphical method for finding fixed points and their stabilities when $a_0 \neq 0$ in Eq.(2.1).

by the biologist Robert May as a model of population growth and is one of the simplest and most used examples of one-dimensional discrete maps. This second-order difference equation shows very rich and complex dynamics through the parameter α that controls the nonlinearity of the system. In order to keep the system bounded with normalized values, i.e. between 0 and 1, α takes values between 0 and 4. If $\alpha > 4$ the system just blows up to infinity very rapidly, if it is less than zero the system starts oscillating around zero including negative numbers in the overall dynamics until a point where it also explode to infinity.

$$x = a_2 x^2 + a_1 x + a_0; \quad a_i, x \in \Re.$$
(2.6a)

$$x = \alpha x(1-x); \quad 0 < \alpha < 4, \quad 0 < x < 1.$$
 (2.6b)



Figure 2.3: Fixed points representations: stable (left) and unstable (right).

The first step in our analysis is to find the fixed points of Eq.(2.6b). To do this we have to find those values where f(x) = x

$$f(x) = \alpha x(1-x) = x \to x(\alpha(1-x)-1) = 0$$
$$\hookrightarrow x^* = \begin{cases} 0\\ 1 - \frac{1}{\alpha} \end{cases}$$

Then we need to know how stable these points are. Fixed points come in two different flavors, and in order to see their differences we can compare them to being either a hill or a valley. If a ball is exactly at the top of the hill it will remain there, but we know that even the slightest force in any direction will make the ball roll over the hill and away from that steady point where it was, Fig. 2.3, right. This kind of fixed point is called a *repellor* or an unstable fixed point. On the other hand a ball resting at the bottom of a valley will remain there or will go back there even if a temporary force takes it out of its steady state, Fig. 2.3, left. This kind of point is called an *attractor* or a stable fixed point.

The stability of a point in a discrete system is found analytically by evaluating the first derivative of the function on that point, i.e. $f'(x^*)$. If the absolute value of this evaluation is less than one, i.e. $|f'(x^*)| < 1$ we say the fixed point is stable, and if it is greater than one, i.e. $|f'(x^*)| > 1$ we say it is unstable.

$$f'(x) = \alpha - 2\alpha x \tag{2.7}$$

$$\hookrightarrow |f'(x^*)| = \begin{cases} \alpha & ; x^* = 0\\ 2 - \alpha & ; x^* = 1 - \frac{1}{\alpha} \end{cases}$$
(2.8)

As we see, the stability of our map depends on the parameter α , and since we have constrained α to be greater than 0 but less than 4 we can start by saying that $x^* = 0$ is an attractor when $0 < \alpha < 1$ and a repellor otherwise. On the other hand, the point $x^* = 1 - 1/\alpha$ will always be a negative value for $0 < \alpha < 1$, hence not considered as part of the solution. In summary for $0 < \alpha < 1$ we have one single fixed point $(x^* = 0)$ and it is an attractor.

The second-order function of Eq.(2.6b) depicts a parabola with its vertex located at x = 0.5 and height proportional to α : $h = \alpha/4$, Fig. 2.4a. A cobweb plot is used to show the stability of attractor $x^* = 0$ by plotting three different initial conditions, Fig 2.4b.



Figure 2.4: A graphical method for finding fixed points and their stabilities. $0 < \alpha < 1$

A bifurcation¹ occurs at $\alpha = 1$ where $x^* = 0$ becomes a repellor and $x^* = 1 - 1/\alpha$ an attractor. Figure 2.5 give some examples of the new points

¹Bifurcation is the term given to sudden changes of the dynamics of a system when varying its control parameters. It is not within the scope of this thesis to analyze dynamical systems from the bifurcations point of view but to make use of stable states found around these points. A whole body of research has been developed around the different types of bifurcations found in nonlinear dynamical systems. The interested reader is suggested to ... [paper]

of intersection between the logistic function and the bisector. This type of bifurcation is called a *tangent* or *saddle node* bifurcation since it occurs when the tangent to the function crosses the bisector. Three different initial conditions are used to find the stability of the new fixed points, Fig. 2.5b. It is possible to see that values close to $x^* = 0$ are driven away with time making it a repellor, all initial conditions are attracted to the newly created fixed point at the crossing of both curves.



Figure 2.5: A graphical method for finding fixed points and their stabilities. Examples for $1 < \alpha < 3$.

2.3 Periodicity

The previous section gave us an initial analysis of fixed points found in the logistic map and their stabilities for values below and above $\alpha = 1$. However the condition for having stability changes once again as α goes beyond 3 since $|f'(x^*)| > 1$.

$$|f'(x^*)|_{x^*=1-\frac{1}{\alpha}} = |2-\alpha| > 1, \quad \text{for } \alpha > 3 \tag{2.9}$$

In order to better see what happens at this new bifurcation we will plot the second iteration of our function. Figure 2.6 shows two plots for $\alpha = 2.8$, Fig. 2.6a and for $\alpha = 3.2$, Fig. 2.6b. For $1 < \alpha < 3$, the second-iteration curve has the same stable fixed point as the first-iteration curve. Once α goes beyond 3, this fixed point looses stability and two new fixed points are created when the tangent of the fourth-order function, second-iteration curve, crosses the bisector curve. The main difference between these new fixed points and the studied in the previous section is that the function does not settle down to just one of them as time increases but it alternates between both on each time step. By doing so, an entirely new behavior is created known as *periodicity* and the name given to this type of bifurcation is *period-doubling* or *pitchfork* bifurcation.



Figure 2.6: First and second-iteration plots of Logistic map.

In order to find the location of the new fixed points we need to follow the same methodology as we did in the previous section, but in this case we will work with a second-iteration function, i.e. f(f(x)), since the second fixed point is located two steps ahead of our present state.

$$f(x) = \alpha x(1-x)$$

$$f(f(x)) = \alpha(f(x))(1-f(x)) = \alpha(\alpha x(1-x))(1-\alpha x(1-x))$$

$$= \alpha^2 x(1-x)(1-\alpha x(1-x))$$

$$= -\alpha^3 x^4 + 2\alpha^3 x^3 - (\alpha^2 + \alpha^3)x^2 + (\alpha^2 - 1)x \qquad (2.10)$$

Once again we use the condition to find the fixed points of a discrete map, f(x) = x. Equation (2.10) should give us four roots or fixed points, however we already know two solutions to this equation from the roots of the first-iteration function, namely $x^* = 0$ and $x^* = 1 - 1/\alpha$. We use these two already known roots to further simplify Eq. (2.10):

$$f(f(x)) = x$$

$$-\alpha^{3}x^{4} + 2\alpha^{3}x^{3} - (\alpha^{2} + \alpha^{3})x^{2} + (\alpha^{2} - 1)x = x$$

$$(-\alpha^{3}x^{3} + 2\alpha^{3}x^{2} - (\alpha^{2} + \alpha^{3})x + (\alpha^{2} - 1))x = 0$$

$$-\alpha^{3}x^{3} + 2\alpha^{3}x^{2} - (\alpha^{2} + \alpha^{3})x + (\alpha^{2} - 1) = 0$$
(2.11)

Dividing Eq. (2.11) by $x - (1 - 1/\alpha)$ we obtain:

$$-\alpha^{3}x^{2} + (\alpha^{2} + \alpha^{3})x - (\alpha^{2} + \alpha) = 0$$

$$\rightarrow x^{2} - \frac{\alpha + 1}{\alpha}x - \frac{\alpha + 1}{\alpha^{2}} = 0$$

$$\hookrightarrow x^{*} \begin{cases} x_{h}^{*} = \frac{\alpha + 1 + \sqrt{\alpha^{2} - 2\alpha - 3}}{2\alpha} \\ x_{l}^{*} = \frac{\alpha + 1 - \sqrt{\alpha^{2} - 2\alpha - 3}}{2\alpha} \end{cases}$$
(2.12)

These new solutions are defined only for $\alpha \geq 3$ and their stabilities could be found by the same methodology as before, i.e. by evaluating the derivative of the function on each solution:

$$|f'(f(x))|_{x^*} = -4\alpha^3 x^3 + 6\alpha^3 x^2 - 2(\alpha^2 + \alpha^3)x + (\alpha^2 - 1)$$
(2.13)

where

$$x^{*} = \begin{cases} x_{1}^{*} = 0\\ x_{2}^{*} = 1 - \frac{1}{\alpha}\\ x_{3}^{*} = \frac{\alpha + 1 + \sqrt{\alpha^{2} - 2\alpha - 3}}{2\alpha}\\ x_{4}^{*} = \frac{\alpha + 1 - \sqrt{\alpha^{2} - 2\alpha - 3}}{2\alpha} \end{cases}$$

Finding the stability of these points by algebraic means becomes very difficult already for a second-iterate function, not mentioning finding fixed points and their stabilities for higher-order periods. The principle of graphical iteration and the power of simulations becomes very helpful at this point allowing us to explore a wider space of the control parameter. By plotting a limited number of initial conditions and their iterative progress within the map, it is possible to find out how stable or unstable certain points are. Figure 2.7 shows the behavior of our logistic map for three different initial conditions when $\alpha > 3$. The new fixed points pull any initial condition toward them in an alternating pattern, thus becoming a limit cycle attractor,

Fig. 2.7a. Plotting our function versus time gives a better picture of what happens after a transient period, Fig. 2.7b. On the other hand, fixed point $x_1^* = 0$ is still a repellor since all orbits starting close to zero are still driven away from it.



Figure 2.7: Logistic map plots as a cobweb and in time for $\alpha = 3.2$.

Finally, Fig. 2.8 shows two very close initial conditions and their longterm behavior at fixed point $x_2^* = 1 - 1/\alpha$. When the initial point is exactly at $1 - 1/\alpha$ the system stays fixed at that point, however any infinitesimal variation from this point will make it lose stability and ultimately end up in the limit cycle create by the attractor points x_3^* and x_4^* . Hence we conclude that fixed point x_2^* has become a repellor for $\alpha > 3$.



Figure 2.8: Logistic map plots for $\alpha = 3.2$, showing the lack of stability of fixed point $x_2^* = 1 - \frac{1}{\alpha}$.

To study what happens for values of $\alpha > 3$ we will introduce a new

graphical tool that will simplify the rest of the analysis for any nonlinear map. Figure 2.9 is called a *bifurcation diagram* and consists on, after getting rid of transients, plotting as many random initial conditions as possible for each value of the control parameter, Fig. 2.9a. This plot gives a much more comprehensible picture of the whole dynamic space of the map being studied. Figure 2.9b shows the following period-doubling bifurcation points for the logistic map, giving us an immediate look into the dynamics of this function in the periodic regime.



Figure 2.9: Bifurcation diagram for the logistic map.

From these plots it is easy to see the values that any initial condition will take on the long term depending only on the value of α . Previously we chose three different initial conditions to see the behavior of the logistic map in Fig. 2.7, we can see the 2-cycle obtain in those figures represented in the bifurcation map at $\alpha = 3.2$. In the same way as it happened before, the increase of α will bring us new period-doubling bifurcations, thus creating limit cycles of periods 2, 4, 8, 16, etc. Figure 2.10a depicts the *return map* of our function, it shows the activations points only and not the cobweb lines as before, in this way it is much easier easier to see the different periods formed after the transients. Figure 2.10b shows a window of activations in time for limit cycles of periods 2, 4, 8, and 16.

2.4 Chaos

The previous sections showed us how rich dynamics can be obtained from a very simple equation such as the logistic map, Eq(2.6b). Starting from any



Figure 2.10: Bifurcation diagram for the logistic map.

point in our space, and depending on the value of our control parameter, we were able to decrease to zero, stay at constant values, or oscillate with different periods at a steady state. However, Fig. 2.9a shows a much more complex area to the right of our period-doubling tree, starting from somewhere around $x \approx 3.569945...$, the *Feigenbaum point*². The Feigenbaum point marks the end of the period-doubling tree and the beginning of *Chaos*, the richest and most difficult of all regimes in a nonlinear dynamical system.

After decades of research in chaos theory it is still difficult to find a universal definition of *chaos* or methodology to study and describe a chaotic system. Nevertheless most authors agree in some key properties of chaotic systems grouped in the following sentence. *Chaos* is the part of Nonlinear Dynamical Systems Theory that studies the **aperiodic long-term behavior** of systems governed by **deterministic rules** that exhibit **sensitive dependence on initial conditions**.

To better understand these and other properties of chaotic systems we will make use of simulations of the logistic map for values of α greater than the Feigenbaum point. Aperiodic long-term behavior means that there are trajectories that do not settle down to constant, periodic or quasi-periodic values as time goes to infinity, Fig. 2.11. As mentioned in the introduction of this thesis, deterministic rules are those free of noisy or random inputs or

²The *Feigenbaum constant*, $\delta = 4.6692...$, was discover by physicist Mitchel Feigenbaum in October of 1975 and is obtained as the relationship between the distance between two successive bifurcation points. This constant can be found not only in primitive mathematical models but also in real physical phenomena, a property known as *universality*. See M.J.Feigenbaum blabla...



Figure 2.11: Simulation of the Logistic map for $\alpha = 4$. Plotting 500 activations after 5000 time steps, no periodic dynamics are found.

parameters; the nonlinearity of the system is created by the internal dynamics alone. Finally, sensitivity to initial conditions refers to the property of a system where any arbitrarily small interval of initial values will be enlarged significantly with each iteration, Fig 2.12. In other words, the error of two nearby trajectories will have the same magnitude as the signal itself after few iterations, Fig 2.12b.



Figure 2.12: Simulation of the Logistic map for $\alpha = 4$. Plotting two initial conditions separated by 1e - 10. In a chaotic system this difference is amplified exponentially fast.

Chaos has two other central properties which are usually not mention in most textbooks: **mixing** and **dense periodic orbits**. A system is said to show *mixing behavior* when any point within an arbitrarily small subinterval of the state space, eventually reaches points in another arbitrarily small subinterval of the same space; in other words, we can get everywhere from anywhere. Besides sensitivity to initial conditions and mixing behavior, the existence of a dense set of *unstable periodic orbits* (UPO) is the other necessary condition for validating the presence of chaos. Within the chaotic regime, trajectories will not settle down to a single periodic orbit but will wander in a sequence of close approaches to these orbits.

So far we have been using the logistic map, Eq. (2.14a) to show that although simple this equation is rich with a great variety of dynamics. Other examples of one-dimensional discrete maps that present the kind of behaviors we have been studying so far, including chaotic dynamics are: shift map, Eq. (2.14c); tent map, Eq. (2.14d); sine map, Eq. (2.14b). Examples of twodimensional chaotic maps are: the tent map, Eq. (2.14e); the tent map, Eq. (2.14f).

$$f(x) = \alpha x (1 - x) \tag{2.14a}$$

$$f(x) = \alpha \sin(\pi x) \tag{2.14b}$$

$$f(x) = 2x, \quad mod(1) \tag{2.14c}$$

$$f(x) = \begin{cases} \alpha x, & 0.0 \le x \le 0.5\\ \alpha(1-x), & 0.5 < x \le 1.0 \end{cases}$$
(2.14d)

$$f(\mathbf{x}) = f\begin{pmatrix} x_1\\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 + 1 - \alpha x_1^2\\ \beta x_1 \end{pmatrix}$$
(2.14e)

$$f(\mathbf{x}) = f\begin{pmatrix} x_1\\ x_2 \end{pmatrix} = \begin{pmatrix} -\beta x_2 + \alpha x_1 - x_1^3\\ x_1 \end{pmatrix}$$
(2.14f)

Strange attractors

In the previous sections very simple kinds of attractors representing constant values (fixed points), and periodic oscillations (limits cycles) were studied. Within chaotic regimes some dynamical systems form very distinctive long term patterns called *strange attractors*. The shape of these kinds of attractors repeats over and over regardless of the initial conditions of the system. This is a surprising condition since we already saw how sensitive to initial conditions a chaotic system is, Fig. 2.12b. We would think that two different initial
conditions would create completely different patterns in the long term since already during the first few steps they diverge exponentially, but the surprise comes when analyzing the long term behavior of these orbits: both stay bounded within a very distinctive *basin of attraction*. Figure 2.13 shows two of the most known strange attractors, both of them are continuous threedimensional functions: Lorenz, Fig. 2.13a; and Rössler, Fig. 2.13b.



Figure 2.13: Strange attractors.

Strange attractors do not only challenge the notion of sensitivity to initial conditions but present other interesting properties like those of fractals and mixing behavior. Both, strange attractors and chaos have received a lot of attention during the last decades but they are still in their infancy when trying to find a definite mathematical description. Even so, their popularity is not only among mathematicians and physic scientists but for people in neural sciences, sociology or psychology to name a few.

Intermittency

Another very interesting property of chaotic systems that we will make use in this thesis is *intermittency*. *Intermittency* could be seen as the alternation of almost periodic dynamics and chaotic dynamics. When looking more carefully into some regions of the chaotic regime it is possible to find periodic windows that interrupt the chaotic behavior of the system for certain values of the control parameter, Fig. 2.14a. At the beginning of these windows it is possible to find values of the control parameter for which the system seems undecided between behaving periodically or chaotically, Fig. 2.14b.



gistic map.



Figure 2.14: Periodic windows within the chaotic regime of the logistic map contain points where chaos and periodic behavior interrupt each other.

2.5Summary

In this chapter we have studied the most basic concepts related to nonlinear dynamical system theory. All possible dynamics present in the generic form of linear functions were analyzed in the first section of this chapter. Solutions were found both algebraically and graphically. As we saw, some specific initial conditions drove this system to either stay constant, oscillate or blow away as time grows but always in a linear way. The following sections made use nonlinear functions to study those dynamics in more detail. In specific, a well known discrete nonlinear function was used to show the different behaviors that are possible to obtain by simple iterations. Section 2.2 studied the simplest case of long term behaviors presented in the linear case, fixed points. Section 2.3 focused on oscillatory dynamics of different periods and finally Sections 2.4 studied the most interesting and rich of all behaviors, *Chaos*.

Chaos theory is sometimes studied as an independent subject and has accumulated a great amount of studies and followers. It is not within the scope of this thesis to present all the concepts and tools used in Chaos theory since it would require a much larger amount of space and time, but to know the general ideas behind the dynamics of these kinds of systems. Those characteristics and tools of nonlinear dynamical systems studied so far are the ones that we will use when designing autonomous dynamics agents. In the following chapter we will come back to chaotic behaviors when studying some configurations of neural networks capable of generating nonlinear dynamics.

Chapter 3

Neural Networks

Neural Networks, both biological and artificial, have been studied for several decades and are a highly interdisciplinary topic difficult to describe in just one chapter of a book. A survey on general-purpose computation with neural networks done by Šíma&Orponen, [63] gives a general taxonomy on this topic, recreated in the following list. For our purposes a very simple classification of artificial neural networks based only on their architecture, and an introduction on their biological origin will be presented in this chapter.



3.1 Biological Neural Networks

The human brain is an extremely complex and sophisticated system that unlike any other organ, it performs thousands of functions simultaneously. It has been studied for several centuries at all scales and it does not seem likely that we will ever understand it completely. The smallest scale in the human brain for information processing is at the level of *neurons*. Neurons are the core component of the brain and in more general terms of the *central nervous system* (brain and spinal cord). However is in the brain where we will find most of them, an average of 100 billion; moreover the number of neurons in a person does not stay constant. There are several types of neurons depending on their function, size and location in the body, however a common characteristic of most of these cells is their tree like structure, Fig. 3.1. They receive signals from other neurons through their *dendrites* and after integrating all signals in the *soma* or body of the cell, they give their response through their axon. In average, each neuron has 1000 connections or synapses and convert electrical signals to chemical and back to electrical again. Finally this response could be either excitatory or inhibitory.



Figure 3.1: Diagram of a neuron.

Even though the speed of response of each neuron is just of a few miliseconds, the brain in its whole is infinitely more powerful, robust and adaptable than any digital computer. The main reason for this advantage in the human-machine battle is the greater connectivity of its processing units, it is a superb parallel computer. Indeed computers reaching record speeds on the teraFLOPS (10^{12} FLoating Operations Per Second) perform worse than, for example a 5-year-old child when recognizing a face or coordinating movements. Based on the structure and function of a biological neuron, a mathematical model of an artificial neuron was born, the *perceptron* Fig. 3.2. It consists of an input vector with their connection weights, a transfer function applied to the weighted summation of all inputs, and the output, $y_k = f(\mathbf{x}) = f(\sum x_i \omega_i + b_k)$.



Figure 3.2: Model of a neuron.

$$y_k = f(\mathbf{x}, \omega)$$

= $f\left(\sum_{i=0}^n x_i \omega_i + b_k\right)$ (3.1)

The transfer function f(.) is usually chosen such that the output y_k remains bounded between -1 and +1 (*bipolar*) or between 0 and +1 (*unipolar*). Equation (3.2) contains four different examples of transfer functions: the *step* (also known as *heaviside* or *hard limiter*) function, Eq. (3.2a); *piecewise-linear* or *saturated-linear*, Eq. (3.2b); *standard* (*logistic*) *sigmoid*, Eq. (3.2c); *gaussian*, Eq. (3.2d).

$$f(v) = \begin{cases} 1, & v \ge 0\\ -1, & v < 0 \end{cases}$$
(3.2a)

$$f(v) = \begin{cases} 1, & v \ge 0.5 \\ v, & -0.5 < v < 0.5 \\ -1, & v \le -0.5 \end{cases}$$
(3.2b)

$$f(v) = \frac{1}{1 + e^{-x}}$$
(3.2c)

$$f(v) = e^{-\frac{x^2}{2\sigma^2}}$$
 (3.2d)

Perceptrons are able to solve linearly separable problems only whereas a network of neurons have proved to find steady solutions for nonlinear problems.

3.2 Feedforward Networks

The general structure of feedforward neural network consists of signals that flow in one direction from n input neurons toward m output neurons, Fig. 3.3. All other nodes in between are called *hidden* neurons. The notation for a synaptic weight going from neuron i toward neuron j is ω_{ii} .



Figure 3.3: Diagram of the structure of a general feedforward neural network.

This type of networks have been successfully applied in *regression* problems like statistical estimation, optimization and control theory; as well as in *classification* problems like pattern recognition, image analysis and decision making.

The key point in any application of this type of network is the methodology used to change the values of the *synapses*, i.e. the amount of influence of one neuron over the other. This process is called *learning* and in general it can be classified in two different paradigms: supervised and unsupervised. Some authors differentiate also between unsupervised learning and reinforcement learning. The main difference between supervised or unsupervised learning is that in the former there is an *a priori* knowledge of the output whereas for the later this information is not available. Feedforward neural networks are being used in a broad range of practical applications and probably 90% of them use supervised learning algorithms. They have shown excellent results on tasks where the goal is to stimate the best option from a pre-defined set of solutions. In other words, supervised learning performs best with problems that have a finite number of solutions and the system has been trained with a large number of possible inputs for obtaining those solutions. This is not the case in some areas of, for example cognitive robotics where it is not possible to pre-define a set of solutions from dynamic environments. For instance, what would be the best sequence of movements if a walking robot trips and needs to protect the head? or can an agent trained with a limited number of classes propose the use of an object from certain class to be used as a member from another class?

3.3 Recurrent Networks

The structure of feedforward neural networks started with the model of a biological neuron but the structure *per se* is not biologically plausible. The main characteristic in biological neural networks is feedback. As mentioned earlier, the brain is an extremely complex system where millions of neurons have thousands of connections that, in their majority, does not follow a single direction but create cyclic circuits.

A network of neurons that presents feedback among its units is called a *recurrent neural network*, Fig. 3.4. There are several consequences of having connections going in two directions but in general we would say that the network could become unstable and present complex dynamics like the ones studied in the previous chapter, i.e. oscillations and chaos. Having a richer range of dynamics is both a positive and a negative characteristic of this type of networks. The positive side of a having single system containing not only steady states but periodic and chaotic dynamics as well is their potential for generalization, being able to solve different types of problems and include a larger space of inputs in their solutions. However, this constant change of states that could become unstable for certain conditions is also a negative property when trying to design solutions for the kind of applications engineers are used to work with. Specific tasks require in most cases specific, stable and accurate solutions and that is why feedforward networks are the ideal approach.

One of the most used configurations of recurrent nets is the one proposed



Figure 3.4: Two different representations of a recurrent neural network.



Figure 3.5: Possible outcomes from a retrieval trial in a Hopfield network.

by John Hopfield in 1982, Fig. 3.4. A Hopfield network works as an associative memory and more specifically as a content addressable memory, i.e. it can recall a memorized pattern given a distorted version of it. In order to guarantee the convergence to local minima, i.e. the memorized patterns, a Hopfield network requires its connections to be symmetric ($\omega_{ij} = \omega_{ji}$) and that no unit has a connection to itself ($\omega_{ii} = 0$). Even though the convergence to local minima is guaranteed by fulfilling these conditions, there is no guarantee that one of those minima is a memorized pattern. In practice we will see that the retrieved patterns are either the desired outcome, a reversed memory, a combination of memories, or just some other minimum with no correlation to the saved memories, Fig. 3.5.

Following Hopfield's work a vast body of research has been devoted to improve the drawbacks presented in the original version of this type of networks. Several algorithms have been proposed for cleaning the output space of spurious memories. Some others have gone beyond the learning and retrieving of a single set of patterns (auto-associativity), it has been proposed the association of patterns from several sets (hetero-associativity). With these achievements is now possible to talk about episodic memories since the presentation of a pattern, also for noisy or incomplete patterns, can induce the retrieval of a sequence of memories.

However a lot of work still needs to be done, specially for real-time online learning since most of the proposed algorithms assume the existence of training sets to be used in offline learning stages before the actual use of the system in real applications. This is not compatible with the concept of *cognition* since such methodology will surely limit the sensori-motor space of any agent in real environments.

Pasemann's neuromodules

Hopfield's model constrains the dynamics of the network by having symmetric connections and avoiding self-interactions. Biological neurons, on the other hand, are known to have self-connections both for excitatory and inhibitory neurons. Moreover, biological neural networks have rarely symmetric connections. More biological plausible models of recurrent neural networks were developed from the work of Amari [5, 6], Aihara [3], and Pasemann [38, 39, 40].

A model of the simplest case, namely a single unit with self-interaction and external inputs, Fig. 3.6, is described by Eq. (3.3). This neuro-module [38] has no restrictions on whether the self-connection ω has a positive (excitatory) or a negative (inhibitory) value and the sigmoidal transfer function, Eq. (3.2c) is chosen to active the state of the neuron. An external input together with its bias (θ) are also responsible of the final behavior of the neuron.

$$x(t+1) = \theta + \omega f(x(t)) \tag{3.3}$$

In contrast with the convergent dynamics of conventional neural networks, e.g. McCulloch-Pitts model, a single unit with recurrent connection is capable of obtaining interesting dynamics like bi-stable states (hysteresis effect, Fig. 3.7 II) and period-2 oscillations (Fig. 3.7 III) besides the steady responses of fixed points (Fig. 3.7 I).



Figure 3.6: Diagram of a single recurrent neuron.



Figure 3.7: Bifurcation diagram for a single recurrent neuron.

Pasemann later continued with the study of the dynamics of two and more neuro-modules with recurrent connections [41, 42]. In the two-neuron network there are six parameters that can be modified, Eq. (3.5), Fig. 3.8. Depending on their values it is possible to obtain all different kinds of attractors: fixed points, periodicity, quasi-periodicity, and chaos. Different types of hysteresis and harmonic oscillators can be created with a pair of recurrent neurons.

$$x_1(t+1) = \theta_1 + \omega_{11}f(x_1(t)) + \omega_{12}f(x_2(t))$$
(3.4)

$$x_2(t+1) = \theta_2 + \omega_{22}f(x_2(t)) + \omega_{21}f(x_1(t))$$
(3.5)



Figure 3.8: Diagram of two coupled recurrent neuron.

Hysteresis effects can be used as obstacle avoidance controllers, bi-stable fixed points, or short-term memory devices. Having the possibility of creating harmonic oscillators from a couple of neurons allows us to work with central pattern generators from the same circuitry, Fig. 3.9a.

Kaneko's Coupled Maps

Coupled Map Lattices (CML) and Globally Coupled Maps (GCM), were introduced by Kunihiko Kaneko in the middle of the 1980's as an alternative for the study of spatiotemporal chaos [27]. In short, this kind of dynamical systems use discrete partial difference equations to study the evolution of a process described by discrete steps in space and time but with continuous states. Equations (3.6) and (3.7) describe the dynamics of CML and GCM respectively.

$$x_i(n+1) = (1-\epsilon)f(x_i(n)) + \frac{\epsilon}{2} \{ f(x_{i+1}(n)) + f(x_{i-1}(n)) \}$$
(3.6)

$$x_i(n+1) = (1-\epsilon)f(x_i(n)) + \frac{\epsilon}{N} \sum_{j=1}^N f(x^j(n))$$
(3.7)

Where $x_i(n)$ is a variable at discrete time step n at lattice point i and represents a set of field variables which could be temperature, position measurements, velocities, etc. Function f(.) can be any chaotic mapping like the ones studied in the previous chapter, Eq. 2.14. There are two parameters: α



Figure 3.9: A minimal chaotic neuro-module, plots of two of the different dynamics that a two-neuron recurrent network is capable of: harmonic oscillator (top row) and chaos (bottom row).

controlling the level of chaoticity of the system and ϵ controlling the coupling level among neighboring elements. The state of each element is defined by its previous state and: by an average of its nearest neighbors in the case of CMLs, or by an average of the states of all the elements in the network in the case of GCM. In other words, while each node in a CML interact with specific points within the lattice, Fig. 3.10a, each of the nodes in a Globally Coupled Map (GCM) interact with all the others, Fig. 3.10b.

Due to the chaotic nature of the system, it is possible to see one of the main properties of chaotic systems: two slightly different initial conditions amplify their difference through time. On the other hand, the system tries to synchronize the activations of all its chaotic elements by coupling them. In between these two states of complete chaos and complete synchronization, interesting states emerge like the formation of clusters oscillating in different



Figure 3.10: Diagrams of coupled maps.

phases and amplitudes.

Both of these categories have been thoroughly studied during the last two decades with researchers trying to describe them both qualitatively and quantitatively. The effects of varying both chaoticity and the coupling factor in stand-alone CML and GCM systems were studied meticulously by Kaneko's group in the late 90's [24, 25]. Approximate phase diagrams were sketched covering the entire spectrum of synchronization among the interacting chaotic elements of a network.

The study of dynamically varying the connections among the elements in a GCM was done by Ito and Kaneko [26, 22]. The model is described by the set of equations in (3.8). The first equation correspond to a GCM, where f represents a chaotic map; (3.8b) updates each unit's connections coming from other units in the network; and (3.8c) specifies the hebbian rule governing the relationship between all units.

$$x_{n}^{i} = f\left((1-\epsilon)x_{n-1}^{i} + \epsilon \sum_{j=1}^{N} w_{n}^{ij}x_{n-1}^{j}\right),$$
(3.8a)

$$w_{n+1}^{ij} = \frac{\left[1 + \delta g(x_n^i, x_n^j)\right] w_n^{ij}}{\sum_{j=1}^N \left[1 + \delta g(x_n^i, x_n^j)\right] w_n^{ij}},$$
(3.8b)

$$g(x,y) = 1 - 2|x - y|$$
(3.8c)

In (3.8b), δ represents the degree of plasticity of the connections and ranges from 0 to 1. The weights w^{ij} in (3.8b) refer to the influence from unit j going into unit i. All self-connections were set to 0; and the initial condition for all remaining connections are equal to 1/(N-1), N being the number of chaotic units.

3.4 Summary

The research on artificial neural networks has been a mayor topic for more than 50 years. In this chapter we have described the most important parts of this subject but with an extra focus on those networks that give us the possibility to include in a single system both convergent and non-convergent dynamics. Feedforward neural networks are by far the most used models for finding stable solutions in a wide variety of problems, including specific tasks related to human cognition. However we argue that the dynamics present in recurrent neural networks give us the freedom we need to replicate the complex range of human behaviors in a more general sense. Hopfield networks were introduced in their original form but a large body of research can be found based on this model. The applications of this model and their variations have focused on the development of associative memories mainly. A more biological plausible model was introduced with the work of Pasemann and colleagues. The neurons in this model are simple enough to track their dynamics analytically and at the same time contain the rich dynamics of the whole spectrum of nonlinear dynamical systems. This model has been used successfully in robotic applications and kept its close relationship to the biological counterpart. Finally, the work of Kaneko was resumed at the end of the chapter. Although this model has no biological inspiration, it has provided the scientific community with important findings about the power of recurrent networks. In the second part of this book, we will be able to use this type of networks in a very practical application.

Part II

Applications in Robotics

Chapter 4

Visual-motor behaviors

This chapter is based on the following publications which can be found in Appendix B.

- B. Durán, G. Metta and G. Sandini. Emergence of smooth pursuit using chaos. In Proceedings of the First International Conference on Self-Organizing and Self-Adaptive Systems. July, 2007. pp: 269-272. Boston, MA. (2007)
- 2. B. Durán, G. Metta and G. Sandini. Towards a chaotic smooth pursuit. In *Proceedings of the IEEE-RAS 7th International Conference on Humanoid Robots*. December 2007. Pittsburg, PA. (To be published).
- B. Durán, Y. Kuniyoshi and G. Sandini. Eyes-neck coordination using chaos. Springer Tracts in Advanced Robotics, vol. 44, pp. 83 - 92. (2008).

4.1 Introduction

The study of nonlinear dynamical systems and chaos has a long history, however real applications that make direct use of chaos theory have not been fully developed. The purpose of this research is to demonstrate the feasibility of using coupled chaotic systems [27] within the area of cognitive developmental robotics. Based on the model of behavior emergence introduced by Kuniyoshi et al. [32], we study the coordination of multiple degrees of freedom in humanoid robots.

The task of tracking an object has been fully studied and many solutions presented before. Based either in position errors or velocity mismatches, some approaches try to control the activation of motors by means of robust PID controllers [37, 8, 11], while others base their controllers in fuzzy logic [4] or neural networks [31]. In any case, the common methodology in these approaches is to compute expensive Jacobian and kinematic expressions thinking in all the possible circumstances the system could encounter.

All these works comprehend the state of the art in motor control for tracking systems; therefore it would not be necessary to develop new solutions. However, the tracking problem represented the simplest test bed for the study of coupled chaotic systems, both in a simulated environment and for its implementation in a real platform. Our approach differs from previous work mainly in two aspects: first, our system does not need to deal with complex equations of kinematics and dynamics; second, the main goal behind our research is not to improve the performance of existing algorithms but, through our experiments, start building the basis of a dynamic model for motion emergence that embrace as a single entity body and environment. Following Esther Thelen and Linda Smith's suggestion that "action and cognition are also emergent and not designed" [58], another equally important goal of this research is the possibility of having new insights about how the coordination of multiple degrees of freedom emerges in human infants.

4.2 Coupled Chaotic Systems

A network of elements whose activation is defined by a chaotic map receives the name of Coupled Chaotic Systems. Depending on the level of interaction among their elements, it is possible to classify them in systems of local or global interaction. These two type of systems were studied in Chapter 3 as one type of recurrent neural networks with chaotic maps as transfer functions.

A Model for Behavior Emergence

The states of each of the elements in a GCM, or a CML, depend only in the internal dynamics of these systems; they are not influenced in any moment by an external force. When taking these concepts to a robotic application it is necessary to think of a way for including the environment within the dynamics of the system.



Figure 4.1: Body-environment interaction through coupled chaotic fields

The model used in this project is based on the approach followed by Kuniyoshi and Suzuki [32]. The main idea behind this model is to make use of the freedom given by the chaoticity of the system; and, on the other hand, the limitations imposed by the synchronization of all the elements. Their model uses both, the local interaction (CML) and the global interaction (GCM) but with the environment as the external force influencing the internal dynamics of the network. In our case, only GCM was used since no extra benefit was seen when including CML as well; nevertheless the overall approach is the same, Fig. 4.1. Each one of the blocks containing "chaotic" elements and their relationship constitute the core of the system and it is defined by (4.1).

$$u_n^i = f\left\{s_{n-1}^i + \epsilon_1(\bar{s}_{n-1} - s_{n-1}^i) + \epsilon_2\left(\frac{s_{n-1}^{i+1} + s_{n-1}^{i-1}}{2} - s_{n-1}^i\right)\right\}$$
(4.1a)

$$m_n^i = G_u(u_n^i + O_u) \tag{4.1b}$$

$$s_n^i = G_s(r_n^i + O_s) \tag{4.1c}$$

Where f(.) in (4.1a) is any chaotic function with values between -1and +1 since both sensor readings and motor activations are represented by positive and negative values. For this purpose a different version of the logistic map was chosen and was given by $f(x) = 1 - \alpha x^2$. Equations (4.1b) and (4.1c) help us to adjust the values from the chaotic field, here m is the output applied to each motor, s and u are inputs and outputs respectively of the chaotic field block, and r is the raw value coming from the sensors. Finally, G_u , G_r , O_u , and O_r are gains and offsets of the sensors and motors respectively; these values are applied in the same magnitude to all elements in the system.

4.3 Methodology

On a human-like robotic head, a tracking behavior involves the coordination of multiple degrees of freedom. In order to simplify this problem and to have an initial feeling on how feasible would be to implement a control based on coupled chaotic functions, a single eye tracking was decided to be the first challenge in this project. A single eye tracking behavior involves two degrees of freedom (DOF) in a robotic platform; once this first step was completed, a three DOF tracking was performed involving two eyes, and finally a five DOF tracking with the motion of the neck. In all cases a simulation and implementation of the algorithms was performed on a robotic platform.

Simulation

To simulate the dynamics of the real platform, a virtual environment called Webots was used [1]. This software is based on the Open Dynamics Engine libraries for reasonably accurate physics simulation such as the effect of gravity and friction. The time step for the simulations was fixed to 32 milliseconds and the experiments were done without the influence of gravity and with a minimum value of friction.

Each joint is also modeled by a spring and a damper, trying to replicate the physical characteristics of real muscles. A virtual camera mounted in front of the eye gives the visual input needed for modulating the chaotic field. The width and height were fixed to 32 x 32 pixels with a field of view of 0.5 radians. It is assumed that values of saliency are obtained from other visual components. One of these values to be tracked was simulated by a black circular shape moving on a white screen, Fig. 4.2b. The initial conditions were defined by having the object out of the field of view of the eye and generating a circular motion from zero to a maximum speed, and slowing down to zero again for two cycles, then changing direction for another two cycles of zero to maximum speed and so on. This motion was used as a basic test for the robustness of the system.

The input to the system is given by the difference between the center of the observed object within the field of view and the position of the center of



Figure 4.2: Simulation setup for visual tracking experiments with coupled chaotic systems.

the eye, for both the vertical and horizontal motions Fig. 4.2a. The outputs from the chaotic field are fed as torques to the motors after applying the respective offset and gain.

The methodology for tuning the gains and offsets was done by approximating the average of the raw output from the logistic map toward a zero average of the motor activation values. In other words, offsets and gains should be chosen in such a way that the activations from the logistic map oscillate around zero. The simulation worked with the following offsets and gains: $G_u = 1.5$, $G_s = 1.0$, $O_u = -0.92$, and $O_s = -0.2$, for $\alpha = 1.95$ and $\epsilon = 0.1$.

Implementation

The results from the simulation gave us enough confidence to implement this algorithm in a real platform. As mentioned before, the algorithm governing the dynamics for this part of the project was the original version of a globally coupled map, Eq. (3.7,4.2). The modified version from Kuniyoshi [32] was also tested in this part of the research but, since this version contains local couplings, it was necessary to consider the order of the chaotic units within the network. Whereas using global connections only, the system was much simpler to understand and design and no extra benefit or effect was observed when adding local couplings.

$$u_{n}^{i} = f \left\{ s_{n-1}^{i} + \epsilon (\bar{s}_{n-1} - s_{n-1}^{i}) \right\}$$
(4.2a)

$$s_n^i = G_s(r_n^i + O_s) \tag{4.2b}$$

$$m_n^i = G_u(u_n^i + O_u) \tag{4.2c}$$

The gains and offsets had to be modified compared to those in the simulation, mainly for two reasons. First, the implementation differs from the simulation in that the outputs of the chaotic field are fed as speed values into the motors of the head, whereas in the simulation they are used as torques. Second, the values representing mass, inertia, friction and gravity are different in both frameworks. Therefore, different values of gains and offsets needed to be found using the same methodology as before. The offsets and gains were fixed to: $G_u = 25.0$, $O_u = 0.0$, $G_s = 1.0$, and $O_s = -0.8$; $\alpha = 1.9$, and $\epsilon = 0.1$.

A copy of the iCub's head from the RobotCub project [48] was built to test all of our experiments. The following sections describe the software and hardware of this platform.

Hardware

The head has six degrees of freedom: yaw, pitch and roll for the neck, a single pitch motion for both eyes and independent yaw motors for each eye. DC-micromotors are used for moving the different joints; each motor contains an incremental encoder that provides the position of the joint at any time, Fig. 4.3. All motors and sensors are controlled by a group of DSP chips which channel data over a CAN bus to a computer in charge of iCub's high-level behavioral control [48].

Software

Due to the large amount of sensori-motor information generated within the platform the iCub's software was configured to run in parallel on a distributed system of computers. An open-source framework for robotics named YARP (Yet Another Robot Platform) was used for the implementation of the algorithms. The main features of YARP are: support for inter-process communication, image processing, and a class hierarchy to code reuse across



Figure 4.3: Picture of the experimental setup.

different hardware platforms [36]. The programming language in YARP is C++; however, a set of libraries has been developed to allow other programs, like Matlab, access YARP.

It is important to mention that the focus of this project is not the extraction of saliencies from moving images, which is in itself a hard problem in computer vision. A tracking algorithm already implemented and available from the YARP repository was used as the visual component in charge of providing us with the horizontal and vertical coordinates of a moving object. With this information we focus our efforts on the motor control problem.

4.4 Results

Two Degrees of Freedom

In the two-degrees of freedom case the left eye of the simulator and the physical platform were used for the experiments, Fig. 4.4. The relative positions of the objects within the image was used as the value that modifies the position of the motors before entering the coupled fields.

Simulation

The trajectory followed by the virtual eye can be observed in Fig. 4.5. The first reaction, once the object has entered into the eyes's field of view, is to move toward the object. As we can see, the adaptation to the path of motion



Figure 4.4: iCub's sensorimotor diagram, 2dof actuation.

is immediate, there are no overshooting oscillations.



Figure 4.5: Trajectory of the center of the eye ($\alpha = 1.95, \epsilon = 0.1$)

Even though both motors are trying to follow the object as smoothly as possible, a small trembling was observed specially at the maximum angle allowed in each direction. This trembling seems to be directly influenced by the physical characteristic of the hardware, in this case the simulated mass, inertia and friction of the eye. Note that there is no way of discerning the moments where the object is not moving or moving at full speed, this tell us how adaptive the system is to the changes in the environment.

Fig. 4.6 shows the root squared error of the whole system through time.



Figure 4.6: Motion of target and eye through time

The overshooting observed at the beginning of the plot is the result of the object entering into the field of view of the eye; in less than one second, the system adapts to the recent change in the environment. Once the eye keeps track of the object, this error is always between 0.0 and 0.08 radians which is an almost zero deviation for our smooth pursuit task.

The "smooth pursuit"-kind of motion was tested using different values of α (chaoticity factor) and ϵ (coupling factor). However when either α was lower than 0.1 or ϵ greater than 0.35 the system performs inconsistent movements, sometimes trying to follow the object and sometimes trying to escape from it. Fig. 4.7 shows the behavior of the 'eye' for $\alpha = 0.1$ and $\epsilon = 0.2$.

Implementation

Since it is not possible to quantify the motion of the target with respect to an absolute coordinate system, considering that the eye is also a moving framework, the motion of the eye as well as a reference of how the target is moving with respect to the motion of the eye is shown in Fig. (4.8). The target was moved in all possible directions and also with different speeds. It is possible to observe that the target is always within sight of the camera also when the physical limits of the robot are at their maximum. This plot shows how adaptive the system is, since it keeps track of the target independently of



Figure 4.7: Trajectory of the center of the eye ($\alpha = 0.1, \epsilon = 0.2$)

the direction or acceleration of the target. This can be seen in the smoothness of the motion when compared to several rapid changes in the environment.

The position of the target with respect to the center of the eye could be analyzed as an error of the performance of this approach, Fig. (4.9). However, it should be consider that the target was moved several times until reaching the mechanical constraints where the hardware is not allowed to continue the pursuit, thus increasing this 'difference' in positions up to 60%w.r.t. the position of the center of the eye. When working within the area far from the maximums, these errors were kept under 10% w.r.t. to the position of the eye.

When plotting the return map of this coupled chaotic system, Fig. (4.10), we observe the values of both chaotic units following the characteristic curve of a logistic map. The lack of activation values in the positive side of the plane is due to the offset applied to the raw sensors.

Five Degrees of Freedom

It is now time to couple the motion of the neck to the motion on both eyes. Only 2 degrees of freedom from the neck were added, pitch and yaw; mainly because the roll motion is not directly related to the visual information given by the tracking algorithm. In total 5 degrees of freedom will be actuated, leaving static the roll joint of the neck.

Again, each camera provides two quantities: the position of the target in each camera in vertical and horizontal directions. These values modify



Figure 4.8: Motion of the eye in the iCub's head.



Figure 4.9: Motion of left eye and target w.r.t. center of eye in the iCub's head.



Figure 4.10: Returning maps for both chaotic units: yaw (left) and pitch (right).

the position of each motor; thus generating a coupled chaotic system with 6 logistic maps, Fig. (4.11). Offsets and gains were kept the same as in the implementation of the 2-DOF case.



Figure 4.11: iCub's sensorimotor diagram, 5dof actuation.

Implementation

The motion of both eyes and the motion of the head is shown in Fig. (4.12). This plot shows the motion of the eyes relative to the head and the motion of the head relative to its fixed position; it also helps us to see the coordination between eyes and neck. The target was moved in random directions and at different speeds. Since the joints of the neck give approximately an extra 60 degrees on each side and on each direction, an object can be tracked in a wider space. It was also observed an increase of the tracking speed; the motors in the neck help the motors in the eyes to follow in a faster way the tracking object, especially in the yaw direction.

Even though the tracking in the pitch direction works very well, it was observed a strong influence of the weight of the head when moving the head up or down. This can be more easily appreciated in Fig. (4.13), while the position of the object does not exceed the previous 60% of distance from the center of the eye in the horizontal direction, the position of the object in the vertical direction was almost 80% when reaching one of the physical constraints. In all cases we observe a coordination of motion not only between eyes, but now with the respective degrees of freedom of the neck.



Figure 4.12: Motion of both eyes and neck.



Figure 4.13: Position of target w.r.t. center of eye



Figure 4.14: Return maps for all chaotic units.

Finally, the return maps of the different chaotic units show the same almost linear behavior observed on the simulation and implementation in the 2-DOF case, Fig. 4.14. However, in this case we can observe a saturation of values in the limits of the vertical motion. This values represent the nonlinearities found when all pitch motors reach their maximum torques when trying to keep up the the tracking task.

Neural Development

In order to study the influence of the environment in the development of connections within the network we use the approach followed by Ito and Kaneko [26, 22] for adaptive coupled maps. The model is described by the set of equations (3.8). Here the main idea is to start with a globally coupled map where all units are connected to all the others by weak synapses. The goal is to study the development of each of the connections within the network as time pass by.

The results of these experiments showed the development of weak and strong connections among the chaotic units depending on the level of interaction they have through time, Fig. 4.15. Even though all connections start with the same value the system takes only a few time steps to separate in groups of strong and weak connections. A very interesting observation



Figure 4.15: Development of connections in time.

from this plot is that after approximately 500 steps, the connections arriving to any unit oscillate around the middle of the permitted strength. Extreme cases are with pitch units in each eye LP and RP which develop a very strong influence from the pitch motion of the neck NP but a zero influence from one to another. Yaw units develop a more balanced influence in their network, oscillating always around 0.5.

In Fig. 4.16 the matrix of connections is presented in 6 different moments; whereas Fig. 4.17 shows the diagram of connections at time step 1 and time step 3500. At time step 3500 the system has entered in an almost fully developed state where its internal connections vary very little. In the end, each unit is influenced by no more than two other units within the whole network. As expected, two independent sub networks emerge after approximately 20 seconds. In one side all chaotic units fed by yaw motions strengthen their connections while weakening those toward and from 'pitch' units; and the same happens with those units fed by pitch motions when compared to 'yaw' units.

4.5 Summary

A very simple experiment for demonstrating the feasibility of applying coupled chaotic systems in the area of cognitive developmental robotics has been shown in this project. Tracking an object moving in front of a camera has



Figure 4.16: Connections matrix: Top left, step 1; top middle, step 100; top right, step 500; bottom left, step 1000; bottom middle, step 2000; bottom right, step 3500.



Figure 4.17: Diagram of weights at timestep 1 (left), and at timestep 3500 (right). Using the same colorbar specification as on Fig. 4.16. Notation of the chaotic units: (N)Neck, (Y)Yaw, (P)Pitch, (L)Left eye, (R)Right eye.

been solved in several ways previously, from using very simple trigonometric solutions to advanced control algorithms. However, this task represented the simplest test bed for the study of emergence of a reactive behavior in a real platform. A virtual setup consisting of only two rotational joints and a camera was created to replicate the sensori-motor configuration of a real eye. We have demonstrated that, once the object enters the field of view, this input is enough for the self-organization of the controller that generates the torques applied to each of the joints of our devices. No learning or specific coding of the task is needed, which results in a very fast reactive behavior.

By playing with the values of α (chaoticity factor) and ϵ (coupling factor) we saw that a smooth pursuit behavior can change to other 'non-tracking' patterns like following during some time and escaping from the target in some others, or a simple avoidance behavior similar to the way animals protect their eyes when being flashed by a bright light. This very simple action tells us that other visual behaviors can be achieved without much effort from the designing part to simulate specific cognitive actions.

The implementation of this algorithm in a real platform was straight forward. A copy of the iCub's head, a 6 DOF robotic platform part of the RobotCub project [48], was used with only minor changes in offsets and gains with respect to the simulation experiments. The tracking algorithm used in the implementation was taken from the YARP repository [36]. The control algorithm was tested by changing both the chaoticity of the system and the coupling among its elements. In both cases, simulation and implementation, the smooth pursuit behavior emerges when the system is highly chaotic and there is a weak coupling among its elements.

The experience obtained in previous experiments with the simulation and implementation of a single eye tracking [A.1] gave us enough confidence to increase the complexity of our model. Experiments were performed for two-eye (3 DOF) and eyes-neck (5 DOF) coordination problems with fixed parameters [A.2]; as well as the development of connections in eye-neck coordination [A.3]. A very simple Hebbian rule was used to study the development of connections within the core of the system, a globally coupled map. From normalized initial connections we saw them changing through time, restructuring the 'brain' according to the experiences with the environment. In the final stage, two independent sub networks were formed, one containing yawrelated chaotic units only and the other pitch-related chaotic units only. The smooth pursuit behavior emerged also during this process.

Chapter 5

Dynamic Field Theory

5.1 Introduction

Developmental Psychology is one of the main areas of scientific research from where cognitive robotics has been taking inspiration during the last decade. It studies the psychological changes that occur in human beings through their lives, being the first stages of infant development the most interesting for cognitive robotics. Among the most important works within this area, the one followed by Esther Thelen and Linda Smith [58] has been of great interest not only for psychologists but mathematicians and roboticists as well. The main reason for this multidisciplinary interest in their approach is based on their choice of nonlinear dynamical systems theory to explain the different paradigms found in infant development. According to Thelen and Smith, the different behaviors found in adult people are the result of actions that are not influenced by the dynamic responses to the environment only but have a strong component of previous experiences as well. They see human development as a landscape with an infinite number of valleys that represent different behaviors being created and reshaped continuously, Fig. 5.1. As time pass by, the previous 'history' in the formation of these valleys influences strongly their current tendencies and create in some cases deep and stable behaviors.

The formation of cognitive behaviors could be described in most cases by what is known as *habituation*. Habituation is the process by which the responses to certain stimulus vary less and less with each repetition of the experience during time. The mathematical model adopted by Thelen and col-



Figure 5.1: Schematics of an ontogenetic landscape for locomotion, from Thelen&Smith[58].

leagues for replicating this kind of behavior is called *Dynamic Field Theory*, [51].

This chapter describes the main characteristics of dynamic fields and show the results of its implementation in a classical infant paradigm known as the *A-not-B error paradigm*. A previous implementation of this paradigm was done by Gregor Schöner and colleagues using a khepera robot. A very simple sensori-motor loop was used to implement the dynamic fields in this robotic platform and simulate the typical response of infants in this paradigm. In our case, the implementation was done on a more human-like robotic platform, iCub's upper body; making use of its stereo vision and arms movements to implement the reaching motion performed by infants.

5.2 Dynamic Field Theory

Dynamic Field Theory is a mathematical framework based on the concepts of Dynamical Systems and the guidelines from Neurophysiology. In this theory, fields could be seen as populations of neurons and their activations as the continuous responses to external stimuli. A field has the same structure of a recurrent neural network and were studied first by Amari [6] as a class of bi-stable neural networks. They could be compared to globally coupled maps since all units influence each other in some degree, *global inhibition*. But they could also be seen as coupled map lattices since close neighbor units have a stronger influence than far neighbors, *local excitation*. Global inhibition


Figure 5.2: Typical activations in a dynamics field, from Schöner[50].

and local excitation are the two types of interactions among field sites x and having them embedded in the dynamics of the system allow the generation of single peaks of activation, Fig. 5.2.

The units of representation in DFT are the peaks of activation along the *dimension* being studied [50]. A *dimension* represents any perceptual feature, movement or cognitive decision, e.g. position, orientation, color, speed. Besides global and local interactions, the dynamics of a field depend also on inputs coming from external sources. In the absence of external inputs an stable "off" state is usually found as the resting position of a field. External inputs add lifting forces to this "off" state, and if sufficiently strong for trespassing the zero-point threshold, they bring the system into an unstable condition where local excitation and global inhibition will decide the new final stable state.

Local and global interactions are an implicit mechanism for sensor fusion and decision making. Local excitation makes neighboring field sites gain force once the input has become larger than the resting level of the field, moreover stabilizes the newly formed peaks against gradual decay. On the other hand, global inhibition makes that a strong input in a field site suppresses the activation of weaker input in another site, the weaker the input the less inhibitory influence will have on other field sites. As a result, the competition that exists among different inputs is dynamically computed with the help of these two types of interactions [50].

In DFT it is assumed that the different changes in the activations of the field occur continuously in time. Therefore, the different models that follow this theory are defined by differential equations, e.g. $\dot{u}(x,t) = f[u(x',t)]$.

Where u represents the activation field and the brackets defines a function of any field site x' and not just on field site x, [14]. The mathematical formulation of a single layer neural field is defined by a differential equation that describes the rate of change of each unit in the field, Eq. (5.1a). The four main components of this equation are a linear decay term -u, a constant resting level h < 0 that fixes the overall level of activation, the inputs to the field S that may contain both task and/or specific inputs shaped by a gaussian filter Eq. (5.1b), and finally a convolution term between the interaction kernel Eq. (5.1d) and a sigmoidal transfer function Eq. (5.1c) integrated over the whole field.

$$\tau \dot{u}(x,t) = -u(x,t) + h + S(x,t) + \int w(x-x')f[u(x',t)]dx'$$
(5.1a)

$$S(x) = C \exp\left[\frac{-(x-x')^2}{2\sigma_s^2}\right]$$
(5.1b)

$$f(u) = \frac{1}{1 + exp[-\beta(u - u_0)]}$$
(5.1c)

$$w(x - x') = w_e \, exp\left[\frac{-(x - x')^2}{2\sigma_w^2}\right] - w_i$$
 (5.1d)

A single layer dynamic neural field contains 8 different parameters that need to be tuned up, Eq. 5.1. First, parameter τ specifies the time scale over which the field gradually builds up or decay. As mentioned before, the resting level *h* defines the threshold required by the inputs to activate the field. External inputs *S* are shaped as gaussian bells therefore we need to specify their strength *C* and width σ_s . The sigmoidal transfer function f(.) is defined by a single parameter β that specifies its steepness. Finally, interaction kernel *w* works as the connection matrix of a recurrent neural network which weights are defined by a gaussian function of width σ_w , a local excitatory strength given by w_e and a global inhibitory level given by w_i .

The parameter space is able to generate a rich range of dynamics going from monostable and oscillatory behaviors to saturated activation fields. However, mono-stability and bi-stability are the most useful scenarios when trying to replicate human behavior. A system is said to be mono-stable when the only attractor is the sub-zero resting state in the absence or after the application of an input. A system may also present a bi-stability condition when, besides the sub-zero attractor, a self-stabilized peak remains after the input has vanished. The property of dynamic neural fields to sustain an activation peak in the absence of input is used as a working memory.

Dynamic Field Theory also studies the different configurations that could be created by grouping several fields and the possibility of having multidimensional fields. Fields may represent excitatory or inhibitory neural populations as well as long-term memory layers; for example, Simmering et al. [55] created a five-layer structure to study spatial-cognition. Bi-dimensional fields have been created to study visuo-spatial cognition where DFT provides an elegant solution to the "binding problem" [23]. Other applications of DFT include motor planning [14], saccadic eye movements [29, 64], infant perseverative reaching [57].

5.3 The "A-not-B" paradigm

The "A-not-B" error is an infant paradigm first studied by Jean Piaget in 1954 [44]. It could be described as a game designed to study the emergence of the concept "object" in infants between 7 and 12 months-of-age. Infants at this age present a partial knowledge of the location of a toy hidden at one of two locations [57]. They search repetitively for an object at a previously visited location, even though they see the desired object vanishing at another location.

Since Piaget's first description, this experiment has been repeated countless times and in different variations. Many different explanations has been proposed among developmental psychologists, thus reflecting the little consensus about the origin of this error. Thelen et al. [57] proposed an elegant modeling of this paradigm based on dynamic field theory, this approach will be the guideline for the implementation in our robotic platform.

The classic version of the A-not-B hiding task goes as follows [13]. Two marked locations (distinctive from the background) are presented to the infant. A few training trials consist of showing and hiding a small and attractive toy behind location 'A', after a delay the infant is allowed to reach and grab the toy from any of the two locations being 'A' the usual choice. However it is common to find experiments where the infant goes to location 'B' during these training trials. Following this training process the toy is shown and hidden behind location 'B' and again after the respective delay the infant is allowed to reach for one of the two locations. The 'A-not-B' error occurs when the reaching goes to location 'A' instead of 'B', Fig. 5.3.



Figure 5.3: Graphical representation of the hiding task experiment and its results.

5.4 Methodology

The hiding task deals with the location of objects in front of an agent, therefore a single dimension representing orientation was used. A one-dimensional neural field Eq. (5.2a) together with a memory trace field, Eq. (5.2b) were used as the core of the model for the 'A-not-B' error experiments, Eq. 5.2. The memory layer was active only when two conditions were fulfilled: the main layer was active $(u_x > 0)$, and a 'go signal' was set to 'on', Eq. 5.2d. This layer includes two parameters in charge of controlling how fast memories are built (λ_{build}) and how fast these memories decay (λ_{decay}) . The interaction kernel for the main field $w_{xx}(x - x')$ was given by Eq. 5.1d described in Section 5.2. The connection weights from the memory layer onto the main layer $w_{xm}(x - x')$ were also computed through an interaction kernel of gaussian type, Eq. 5.2c. Layers:

$$\tau \dot{u}_x(x,t) = -u_x(x,t) + h + S(x,t) + \int w_{xx}(x-x')f[u_x(x',t)]dx' + \int w_{xm}(x-x')f[u_m(x',t)]dx'$$
(5.2a)

$$\tau \dot{u}_m(x,t) = \begin{bmatrix} -u_m(x,t) + f[u_x(x,t)] \end{bmatrix} \\ \times [\lambda_{build} \Theta(u_x(x,t)) + \lambda_{decay}(1 - \Theta(u_x(x,t)))]$$
(5.2b)

Kernels:

$$w_{xx}(x - x') = C_e \exp\left[\frac{-(x - x')^2}{2\sigma_e^2}\right] - C_i \exp\left[\frac{-(x - x')^2}{2\sigma_e^2}\right] - G_i$$
$$w_{xm}(x - x') = C_m \exp\left[\frac{-(x - x')^2}{2\sigma_m^2}\right]$$
$$w_{\phi}(x) = -C_{\phi} x \exp\left[-2\sigma_{\phi}^2(x - x'_i)^2\right]$$
(5.2c)

Activation function for memory layer:

$$\Theta(u(x)) = \int \theta(u(x'))dx'; \qquad \theta(u(x')) = \begin{cases} 1, & u(x') > 0\\ 0, & \text{otherwise} \end{cases}$$
(5.2d)

Inputs:

$$S(x,t) = S_{task}(x) + S_{spec}(x,t) \rightarrow$$

$$S_{task}(x) = C_{AB} \left[exp \left[\frac{-(x - x'_A)^2}{2\sigma_s^2} \right] + exp \left[\frac{-(x - x'_B)^2}{2\sigma_s^2} \right] \right]$$

$$S_{spec}(x) = C_i exp \left[\frac{-(x - x'_i)^2}{2\sigma_s^2} \right]; \quad i \in \{A, B\}$$
(5.2e)

Output:

$$\dot{\phi}(x) = \sum u_x(x,t) \odot w_{\phi}(x) \tag{5.2f}$$

The input the the system is given by the sum of two terms representing the task and the specific input, Eq. (5.2e). The task input has two constant gaussian bells through all the experiment whereas the specific input changes its amplitude and position according to the trial, i.e. location 'A' or 'B'. Finally, the output to the motors was taken as the sum of the point-topoint multiplication between the activation layer and an attracting kernel, Eq. (5.2f). The output kernel w_{ϕ} dynamically computes the location and amplitude of the signal that activates the motor; a positive activation of $\dot{\phi}$ performs an action in one direction, a negative value performs the action in the opposite direction. The speed of motion is controlled by the steepness of the attracting area in the output kernel σ_{ϕ} .

Hardware

The iCub platform [48] was the physical layer where all of our experiments on DFT were implemented. Seven degrees-of-freedom (DOF) were actuated: 5DOF in the head (yaw and pitch for the neck, a single pitch motion for both eyes and independent yaw motors for each eye) and 2DOF in the shoulders, Fig. 5.4. The other degrees of freedom were blocked for these experiments. Three different Blade units were used in order to handle image analysis, motor control and dynamic fields respectively.

The task panel itself was simulated inside a computer screen in front of the robot. Two small rectangles representing the lids (task input) behind which the object would be hidden (specific input). Since the task deals only with the location of an object and not with the differentiation of lids and toys, the object itself was simulated by a rectangle of larger size than the lid but with the same color. The major advantage of simulating the 'A-not-B' panel and 'toy' in this way was the possibility of adapting the 'objects' to those sizes and colors better detected by the color segmentation algorithm depending on the room's luminance.

Software

The set of equations described above contains around 20 different parameters to be tuned. Most of these parameters remain constant and are easy to define, however some others produce important changes in the overall dynamics of the system. In order to have a 'live' feeling of the behavior of the fields when changing these parameters a graphical interface was implemented using Matlab¹. Figure 5.5 shows screen shots of the two interfaces designed for an

¹This code was an adaptation of the *Interactive Dynamic Field Simulator* written by Sebastian Schneegans, Ruhr-Universitaet Bochum, 2008.



Figure 5.4: Schematics of the experimental setup for the hiding task.

interactive simulation of the 'A-not-B' error task. Besides the parameters described previously, a local level of noise was added to the dynamics of the system. Two different sets of kernels representing the behavior of young and older infants were preset in the interface, however the user is also able to specify his/her own parameters.

For the implementation in the robotic platform, the same open-source framework used in the previous chapter was used in these experiments. The algorithms were implemented in C++ with the same parameters used in the simulation. Whenever a 'go-signal' was present, the activations of the main layer actuated the motors in the shoulders to simulate a reaching behavior. A color tracking algorithm was implemented in YARP using the Intel®Integrated Performance Primitive Libraries version 5.3. The eyes were at all times focused in the middle point between locations 'A' and 'B', in this way the task input to the field was given by two colored rectangles in front of the robot at each side of the middle point. Specific inputs were created by increasing the size of the rectangles whenever an 'A' or 'B' trial was performed. Thus, the values of Eq. (5.2e): $C_i \in A, B$ were given by the number of pixels at position 'A' or 'B', and the values for $x_i, i \in \{A, B\}$ were given by the center of the blobs in the image.



Figure 5.5: Screen shot of the field's simulator for the 'A-not-B' task.

Model Parameters

For comparison purposes two different sets of parameters were used in these experiments: the kind of response found in young infants when performing the hiding task was compared to the 'normal' behavior of adults or older children in the same task, Table 5.1.

The interaction kernels w_{xx} for both infants and adults was designed following the *spatial precision hypothesis*, [52]. This hypothesis states that the spatial precision of neural interactions becomes more precise and more stable over development. This development is represented by two factors: the increasing sharpness and strength of local excitatory interactions controlled

C_e	σ_e	C_i	σ_i	C_m	σ_m	C_s	σ_s	C_q	σ_q
11.0	5.0	0.0	10.0	4.0	5.0	7.0	5.0	0.04	1.0
21.0	5.0	10.0	10.0	4.0	5.0	0.7	5.0	0.04	1.0
$\operatorname{ers} \mid C$	$C_{\phi} = c$	$\tau_{\phi} = \tau$	λ_{bui}	$_{ld} \mid \lambda_{c}$	lecay	β	h	G_i	
nts 1	.0 0.	03 2	10^{4}	1 <u>4</u>	10^4	10.0	-5.0	-0.6	5
lts 1	.0 0.	03 2	10^{4}	1 4	10^{4}	10.0	-5.0	-0.9	5
1	$\begin{array}{c c} C_e \\ \hline 11.0 \\ 21.0 \\ \hline \\ ers & C \\ nts & 1. \\ llts & 1. \\ \end{array}$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$

Table 5.1: Parameters for the DFT model of the 'A-not-B' error paradigm found in young infants and for the control case of adults responses.

by σ_e and C_e respectively, and by the strengthening of lateral inhibitions controlled by C_i and σ_i . Figure 5.6a shows the interaction kernels w_{xx} created with these parameters as well as the output kernel used to obtain the motor response, Fig. 5.6b.



Figure 5.6: Plots of the kernels used in the DFT model of the 'A-not-B' error paradigm, Eq. (5.2c).

5.5 Results

In this section we present the results of the implementation of the model described by Eq. (5.2) with the conditions presented in Table 5.1.

Young infants

First, the kind of behavior found in infants between 7 and 12 months-ofage was simulated and implemented. Infants at this age show repeatedly the 'A-not-B' type of error described in Section 5.3. Figure 5.7 shows the evolution of both the main activation layer and the memory traces left when the reaching behavior is allowed.



Figure 5.7: The 'A-not-B' task performed by an 'infant' agent. Top row: 3D and top views of the main activation layer. Bottom row: 3D and top views of the memory layer.

The model is able to simulate the same kind of results found in human infants. When presenting the visual stimulus at location 'B', the pointing/reaching action goes back to location 'A'. Memory traces are created only when a motor activation is present, in turn the main layer will get an extra influence in those locations were the memory layer has values larger than zero. In this way a primitive way of *expectation* is created.

A single experiment consists on 3 training trials with the purpose of building a motor habituation at the 'A' location, and one testing trial. Each trial is composed of two parts separated by a 3 seconds delay: a visual stimulation where the 'toy' is showed and a motor activation where the robot is allowed to 'point/reach'. Further 'B' trials will end up activating location 'A' with the consequence of creating a higher memory trace at this location. Thus, a *motor habituation* is created at one location. The only way of breaking this *habit* is to wait for an internal decrease of these memory traces due to the effect of the memory decay term controlled by λ_{decay} . Depending on the experiment this value can be changed to induce a faster or slower 'forgetting' effect.

The model is able to replicate the spontaneous errors reported by 'A-not-B' experimenters. Spontaneous errors are described as the selection of 'B' location when showing an object at 'A' in the training stage. These type of errors are replicated by the DFT model when adding noise to the dynamics of the system.

Adult control

Several trials in both locations were performed continuously in order to test the response of 'adult' conditions, Fig. 5.8. Three 'A' trials were performed in a row before testing the response of 'B' locations. Then two trials at location 'B' gave successful responses and the creation of the respective memory traces. Trials at locations 'A' and 'B' were performed in single or multiple groups each to test the performance of the system. Only one 'A' trial is presented here due to the lack of space, however all trials went to their correct locations.

The plots depicting the memory layer in this experiment allow us to have a better look at the effect of λ_{decay} . This term represents the rate of decay of memories through time and the designer should decide how fast an old experience remains present. During a 'B' trial, the memory trace left by 'A' experiences starts its decay, this effect is better seen in the bottom left plot of Fig. 5.8.



Figure 5.8: The 'A-not-B' task performed by an 'older' agent. Top row: 3D and top views of the main activation layer. Bottom row: 3D and top views of the memory layer.

5.6 Summary

The goal of this project was to introduce the dynamic field theory and by making use of this approach, replicate the type of responses found in 7 to 12 month-old infants in a humanoid robot when performing the 'A-not-B' error paradigm. Both, the basic concepts of dynamic field theory and a brief description of the 'A-not-B task were introduced in this chapter. The iCub platform helped us to implement a well studied human behavior through an approach that is based completely in dynamical systems and neurophysiology.

Dynamic Field Theory has demonstrated to be capable of linking neural computation to real human behaviors. Even though it is still under development, this theory models in very elegant ways human paradigms that have been difficult to replicate with other approaches. This mathematical model embed in its dynamics very important and useful properties. The hysteresis property of bi-stable regimes implement a short-term memory, and longterm memories are created as field traces that feed-back past experiences. The way of seeing the world and its variables in terms of dimensions let us have implicit mechanisms for sensor fusion and decision making problems.

The large number of parameters becomes as usual one of the bigger challenges for most mathematical models. A single layer neural field with memory traces has already more than 20 parameters to be tuned. However in DFT, few experiences with the way fields work give the designer the feeling of which parameters create the most important dynamics. This reduces enormously the size of the parameter space.

Chapter 6 Conclusions

Nonlinear dynamical systems theory and the *enactive* paradigm of cognitive systems are the basis of our approach to cognition. The first part of this document studied the most important concepts of dynamical systems and their close relationship with neural networks, both biological and artificial. In the second part, practical applications of the theory were investigated both in simulations and implementation on real robotic platforms.

In the introduction of this thesis we argued that traditional approaches for controlling the behavior of artificial entities are based on the point of view of the designers. Consequently, unexpected circumstances at the moment of solving a task make these systems to either halt or continue with their programs without considering the new information. This is a crucial point for our proposal of a new dynamic, flexible and autonomous way of understanding and implementing human behaviors in robotic platforms.

Dynamical systems theory was our answer for a dynamic world. Dynamical systems theory let us see the body of an agent and the environment as two parts of single dynamic and continuous flow of information in a dynamic world. The body of an agent represents the apex of this world where information is condensed in their simplest forms (stable states, fixed points) through the use of sensors and motors. The further we go into the inner world of an agent or out into the environment the more complex dynamics will be found. With this in mind we could create the concept of a chaotic cognitive architecture for artificial agents where more complex attractors are created through the fusion of simpler stable states.

Any human behavior could be described by the integration of three big components interacting continuously among them. An input block: Different kinds of information are acquired through specific types of sensors installed in the physical layer (hardware) of an agent. An output block: Constitutes the set of devices, also part of the physical layer, used by the agent for the generation of specific actions within the environment; i.e. motors, displays, speakers. And finally, a 'mind' block: A more complex system made up of several parts but all belonging to a software layer (*mindware*); i.e. short and long term memories, emotions, attention cycles, decision making.

Input and output blocks were studied in our embodied approach to cognition. Since the main objective of this research is the simulation of human behaviors, a humanoid platform was the ideal tool for this purpose. In our experiments we had the opportunity to work with the iCub platform from the RobotCub project. This open source project gave us the possibility to test a novel control approach for the input/output blocks based on coupled chaotic systems. The study of visuo-motor behaviors in Chapter 4 showed the feasibility of applying nonlinear dynamical systems theory, and specifically, chaotic systems for controlling the input/output blocks of a robotic platform. What could be seen as a self-organizing recurrent neural network with chaotic maps as transfer functions worked as a reactive physical layer capable of simulating basic visuo-motor behaviors such as tracking, avoidance, and mixed reactions between those two opposites.

The strategic use of tools provided by dynamical systems theory, showed that neither the agent nor the environment require to be modeled since these two parts are seen as nonlinear systems; moreover, the way of solving a task is not specified in advance, as compared with the traditional control approach where the agent is told the steps to follow for solving a task or for overcoming certain problems. In short, if both approaches still need to know what to do, nonlinear dynamical systems theory free us from knowing how to solve a task; thus overcoming the slow or none reactions to unexpected circumstances in dynamic environments found in traditional control models.

The study of the *mindware* was the main focus of Chapter 5. Dynamic Field Theory was our choice for the simulation of higher level cognitive functions due to the different successful applications in developmental psychology. This theory has proved to have the potential to link neurophysiological studies with real human behaviors in a dynamic way. The most attractive properties of this approach are its implicit mechanisms for sensor fusion, decision making and dynamic way of building short- and long-term memories.

The 'A-not-B' error paradigm has been studied by developmental psychologists as a prove of a very elementary form of cognition. This game designed to test the emergence of the concept 'object'on infants between 7 and 12 months of age was modeled by DFT and implemented in the iCub platform. The model replicates the same responses found in infants and adults when faced with this kind of task. This approach showed us a new novel, powerful and dynamic way of implementing a human behavior in a robotic platform: motor habituation.

Further research is needed for DFT to replicate more complex human behaviors. In this approach fields represent static spaces that could take different shapes depending on the different experiences with the environment. However, most of the experiences in a time-space world are sequences of actions that can or can not be periodic. Periodic sequences are used in many human activities like walking, breathing, pattern recognition, etc.

Appendix A Published Articles

A.1 Paper I

B. Durán, G. Metta and G. Sandini. Emergence of smooth pursuit using chaos . In *Proceedings of the First International Conference on Self-Organizing and Self-Adaptive Systems*. July, 2007. pp: 269-272. Boston, MA. (2007).

Emergence of Smooth Pursuit using Chaos

Boris Duran, Giorgio Metta, Giulio Sandini Italian Institute of Technology - University of Genova, Italy boris@unige.it, giorgio.metta@iit.it, giulio.sandini@iit.it

Abstract

The task of tracking an object has been fully studied and many solutions presented before. However, it is a perfect test bed for the study of a novel model using Coupled Chaos Systems. Once an object appears in front of a camera, we demonstrate that the visual input is enough for the self-organization of the torques applied to each of the axes controlling the motion of a simulated eye. No learning or specific coding of the task is needed beforehand, which results in a very fast adaptation to perturbations.

1. Introduction

The research in modern humanoid robotics dates back approximately 30 years when the Bio-engineering group of Waseda University started the WABOT project. Some years later, Honda initiated their research, the result of which is one of the state-of-the-art humanoids of our time, ASIMO. Most of today's humanoid platforms follow a 50-year-old tradition of control theory that started with Industrial Automation at the beginning of the 1960s. Control theory gives us different tools for designing and evaluating the algorithms that will realize a desired motion or force application [2]. It is at this point where the problems start for the humanoids of the future.

In an industrial environment we are able to specify within centimeters, distances, area of motion, speed and acceleration of different links, force and torques, etc. But what happens when we want to move beyond this fixed framework? A more adaptive and flexible theory is needed when thinking of 'controlling' a device that is supposed to move within an ever-changing environment.

The study of nonlinear dynamics and chaos also has a long history; however, real applications that make direct use of chaos theory have not been fully developed. The purpose of this research is to demonstrate the feasibility of using coupled chaotic systems in a more realistic application by taking the model of behavior emergence introduced by Kuniyoshi et al. [8] within the area of humanoid robotics.

The task of smooth pursuit has been solved in many different and more accurate ways than the one presented here. However, this task presents a very simple and attractive challenge to use as test bed for coupled chaotic systems. Another interesting point to be considered is the notion of emergence and self-organization that characterize these systems. In fact, it is exactly at this point where a link between chaos and biological systems can be found. It would not be strange to think that this reactive-emergent kind of behavior generated by coupled chaos has its counterpart in the biological nervous system.

Walter Freeman and colleagues have done an extensive research on the dynamics found in EEG waves from the mammalian olfactory system [3]. He has demonstrated the existence of chaotic dynamics during perception at a mesoscopic level, which refers to the level in between the analysis of single neurons (microscopic) and the activity of whole brain areas (macroscopic). Since it has been shown that nature uses chaos to selforganize the information coming from our senses, we may assume that chaos is also used to organize our muscular responses. With this in mind a simple experiment of self-organizing behavior is studied in this project by using coupled chaotic systems.

The next section describes the basics of coupled chaotic systems together with the model of behavior emergence proposed in [8]. Next, a description of the setup used for smooth pursuit is presented together with the quantitative analysis of the experiment; and, finally, we present the conclusions and guidelines for future work.

2. Coupled Chaotic Systems

A network of elements whose activation is defined by a chaotic map receives the name of Coupled Chaotic Systems. Depending on the level of interaction among their elements, it is possible to classify them in systems of local or global interaction.

2.1 Coupled Map Lattices (CML)

CML were introduced by Kaneko in the middle of the 1980's as an alternative to the study of spatiotemporal chaos [7]. In short, this kind of dynamical systems uses discrete partial difference equations to study the evolution of a process described by discrete steps in space and time but with continuous states. Eq. (1) describes the dynamics of CML, whereas Eq. (2) represents the logistic map used in this work.

$$x_n^i = (1 - \epsilon)f(x_{n-1}^i) + \frac{\epsilon}{2} \{ f(x_{n-1}^{i+1}) + f(x_{n-1}^{i-1}) \}$$
(1)

$$f(x) = 1 - \alpha x^2 \tag{2}$$

Where x_n^i is a variable at discrete time step n and a lattice point *i*. x represents a set of field variables which could be temperature, position measurements, velocities, etc. There are two parameters: α controlling the level of chaoticity of the system and ϵ controlling the coupling level among neighboring elements.

2.2 Globally Coupled Maps (GCM)

These kinds of maps were also introduced by Kaneko and represent a network of chaotic elements with interactions among all of them. While CML interact with specific points within the lattice, each of the nodes in a Globally Coupled Map (GCM) interact with all the others, Eq (3). Due to the chaotic nature of the system, specified by α , it is possible to see one of the main properties of chaotic systems: two slightly different initial conditions amplify its difference through time. On the other hand, ϵ tries to synchronize the activations of all these chaotic elements by coupling them. In between these two states of complete chaos and complete synchronization, interesting states emerge like the formation of clusters oscillating in different phases and amplitudes.

$$x_{n}^{i} = (1 - \epsilon)f(x_{n-1}^{i}) + \frac{\epsilon}{N} \sum_{j=1}^{N} f(x_{n-1}^{j})$$
(3)

Both of these categories have been thoroughly studied during the last two decades with researchers trying to describe them both qualitatively and quantitatively. The effects of varying both chaoticity and the coupling factor in stand-alone CML and GCM systems were studied meticulously by Kaneko's group in the late 90's [6, 5]. Approximate phase diagrams were sketched covering the entire spectrum of synchronization among the interacting chaotic elements of a network.

2.3 Coupled Chaotic Fields

The model used in this project is based on the approach followed by Kuniyoshi and Suzuki [8]. The main idea behind this model is to make use of all those interesting states mentioned before that emerge when coupling chaotic elements and, in this case, the sensory information modified by the environment. This model uses both, the local interaction (CML) and the global interaction (GCM). The system is depicted in Fig. 1.



Figure 1. Body-environment interaction through coupled chaotic fields

Each one of the blocks containing "chaotic" elements and their relationship constitutes the core of the system and it is defined by Eq. (4) and (5). The function frepresents the logistic map, Eq. (2).

$$u_{n}^{i} = f \left\{ s_{n-1}^{i} + \epsilon_{1} (\bar{s}_{n-1} - s_{n-1}^{i}) + \epsilon_{2} \left(\frac{s_{n-1}^{i+1} + s_{n-1}^{i-1}}{2} - s_{n-1}^{i} \right) \right\}$$

$$m^{i} = G_{\nu} (u^{i} + O_{\nu})$$
(4)

$$m_n = G_u(u_n + O_u)$$

$$s_n^i = G_s(r_n^i + O_s)$$
(5)

Where *m* is the torque applied to each joint, *s* and *u* are inputs and outputs respectively of the chaotic field block, and *r* is the raw value coming from the plant. Finally, G_u , G_s , O_u , and O_s are gains and offsets of the sensors and motors in the body of the plant; these values are applied in the same magnitude to all the elements of the system.

3 A virtual eye

The virtual eye was created using two rotational joints, one perpendicular to the other in order to simulate the "pan" and "tilt" motions of a real eye, Fig. 2. Each joint is also modeled by a spring and a damper, trying to replicate the physical characteristics of real muscles. These two motors are actuating the virtual eye as the motion created by the main four muscles in biological eyes.



Figure 2. Biological eye and its virtual counterpart for the experiment

A virtual camera mounted in front of the eye gives the visual input needed for modulating the chaotic field. The width and height were fixed to 32 x 32 pixels with a field of view of 0.5 radians. It is assumed that values of saliency are obtained from other visual components. One of these values to be tracked was simulated by a black circular shape moving on a white screen, Fig. 3. Even though the trajectory followed by the object is circular all the time, it accelerates and decelerates several times to provide a basic test for the robustness of the system.



Figure 3. Screenshot of the virtual setup

The time step for the simulations was fixed to 32 milliseconds and the experiments were done without the influence of gravity and with a minimum value of friction. The offsets and gains were fixed to $G_u=1.5$, $G_s=1.0$, $O_u=-0.92$, and $O_s=-0.2$. The input to the system is given by the difference between the center of the observed object within the field of view and the

position of the center of the eye, for both the vertical and horizontal motions, Fig. 4. The outputs from the chaotic field are fed to the motors after applying the respective offset and gain.



Figure 4. Geometric description for inputs to the chaotic system

The trajectory followed by our virtual eye can be observed in Fig. 5. The first reaction, once the object has entered into the eyes's field of view, is to move toward the object. As we can see, the adaptation to the path of motion is immediate, there are no overshooting oscillations like the ones normally encountered in a PID controller.

Fig. 6 shows the orientation of the motors through time. Even though both motors are trying to follow the object as smoothly as possible, a small trembling was observed specially at the maximum angle allowed in each direction. This trembling seems to be directly influenced by the physical characteristic of the hardware. Note that the synchronization of both motors remains also during those moments when the object slows down and change direction (aprox. 4 seconds). Another important information from this plot is the short time elapsed until it reaches this "steady" state.



Figure 5. Trajectory of the center of the eye (α =1.95, ϵ =0.1)



Figure 6. Motor orientation in time

In less than one second, the system adapts to the recent change in the environment.

The "smooth pursuit"-kind of motion was tested using different values of α (chaoticity factor) and ϵ (coupling factor). However when either α was lower than 1.0 or ϵ greater than 0.3 the system performs inconsistent movements, sometimes trying to follow the object and sometimes trying to escape from it. Fig. 7 shows the behavior of the 'eye' for $\alpha = 0.1$ and $\epsilon = 0.2$. A more in-depth study on the effects of varying these two parameters, α and ϵ , would be very difficult because of the local and global interaction of elements at the same time; moreover, every observation would change from time to time due to the ever changing influence of the environment.



Figure 7. Trajectory of the center of the eye (α =0.1, ϵ =0.2)

4 Conclusions and Future work

A very simple experiment for demonstrating the feasibility of applying coupled chaos systems in humanoid robotics has been introduced in this project. Tracking an object moving in front of a camera has previously been solved in several different ways, from using very simple trigonometric solutions to advanced control algorithms. However, this task was simple enough to use it in the emergence of a reactive behavior that could have a better understanding in Neural Sciences.

According to neurosciences, all behavior is mediated by the central nervous system (brain and spinal cord) which is separate but functionally interconnected with the peripheral nervous system (continuous stream of sensory information about the environment). Simply put, the major difference between voluntary and reflexive movements is the intervention or not of the central nervous system [4]. In practice, it is not possible to separate the modulation signals coming from the brain into the muscles of our eyes. But according to the results of this experiment, we could speculate that visual tracking is just a reactive behavior given a saliency in our visual field. These saliencies are the necessary modulations given by our central nervous system and its areas of emotions, experiences, needs, etc.

Future work involves the coordination of motion with two eyes and finally the emergence of coordinated motion between eyes and head. The simulation environment saves a substantial amount of time and resources for these types of experiments; however, a notable amount of that saved time is dedicated only to tuning the physical parameters in these simulated environments. Most of the software for robotics simulation still creates instabilities influencing the way the algorithms are supposed to work. Therefore, an iCub head from the RobotCub project is being developed to test this and future experiments in a real environment [1].

References

- [1] Robotcub project. http://www.robotcub.org.
- [2] J. Craig. Introduction to Robotics: Mechanics and Control. Pearson Education, 2004.
- [3] W. Freeman. Mass Action in the Nervous System. Academic Press Inc., New York, 2004.
- [4] E. Kandel, J. Schwatrz, and T. Jesell. Principles of Neural Sciences. McGraw-Hill, 2000.
- [5] K. Kaneko. Clustering, coding, switching, hierarchical ordering and control in a network of chaotic elements. *Physica D41*, pages 137–172, 1990.
- [6] K. Kaneko. Pattern dynamics in spatiotemporal chaos. *Physica D34*, pages 137–172, 1990.
- [7] K. Kaneko and I. Tsuda. Complex Systems: Chaos and Beyond. Springer-Verlag, 2001.
- [8] Y. Kuniyoshi and S. Suzuki. Dynamic emergence and adaptation of behavior through embodiment as a coupled chaotic field. *International Conference on Intelli*gent Robots and Systems, pages 2042–2049, 2004.

A.2 Paper II

B. Durán, G. Metta and G. Sandini. Towards a chaotic smooth pursuit. In *Proceedings of the IEEE-RAS 7th International Conference on Humanoid Robots.* December 2007. Pittsburg, PA. (To be published).

Towards a "chaotic" smooth pursuit

Boris Duran¹, Giorgio Metta², Giulio Sandini³

University of Genova - Italian Institute of Technology Viale Francesco Causa 13, 16145 Genova, Italy. ¹boris@unige.it ²giorgio.metta@iit.it ²giulio.sandini@iit.it

Abstract—Real autonomous systems are very difficult to design, mainly due to the ever changing conditions of the environments where they are supposed to work. In the area of humanoid robotics these difficulties are increased not only because of the complexity of their mechanical structure, above all because they are supposed to work under the same dynamic conditions as we humans do. Our approach for the creation of real autonomy in artificial systems is based on the use of nonlinear dynamical systems. The purpose of this research is to demonstrate the feasibility of using coupled chaotic systems within the area of cognitive developmental robotics.

In our quest towards the design and implementation of a real self-adaptive autonomous cognitive architecture, we have decided to start with a simple application that will tell us how appropriate this approach can be for humanoid robots. Once an object appears in front of a camera, we demonstrate that the visual input is enough for the self-organization of the axes controlling the motion of a single eye, both in a virtual and a real platform. No learning or specific coding of the task is needed, which results in a very fast adaptation and robustness to perturbations. Another equally important goal of this research is the possibility of having new insights about how the coordination of multiple degrees of freedom emerges in human infants.

I. INTRODUCTION

Most of today's humanoid platforms follow an almost 50-year-old tradition of control theory that started with the industrial automation at the beginning of the 1960s. The methodology followed by this approach is based on modeling as precise as possible both the plant and the controller; and filtering or processing as noise the different unexpected circumstances that could occur during the operation of the system. This approach has worked pretty well when the system is in a fixed framework and the environmental conditions are known and controlled; however, this will not be the case for humanoid robots of the future. It is absolutely necessary to start working on a different approach if we want to design and build systems that move and act in the same kind of dynamic environments where humans move and act. A more adaptive and flexible theory is needed in order to 'control' a device that is supposed to move within an ever-changing environment. These are our first steps towards the design and implementation of a real autonomous cognitive architecture based on nonlinear dynamical systems.

Although the study of nonlinear dynamical systems and chaos has also a long history, real applications that make direct use of chaos theory have not been fully developed. The purpose of this research is to demonstrate the feasibility of using coupled chaotic systems [1] within the area of cognitive developmental robotics. Based on the model of behavior emergence introduced by Kuniyoshi et al. [2], we study the coordination of multiple degrees of freedom in humanoid robots.

The task of tracking an object has been fully studied and many solutions presented before. Based either in position errors or velocity mismatches, some approaches try to control the activation of motors by means of robust PID controllers [3], [4], [5], while others base their controllers in fuzzy logic [6] or neural networks [7]. In any case, the common methodology in these approaches is to compute expensive Jacobian and kinematic expressions thinking in all the possible circumstances the system could encounter.

All these works comprehend the state of the art in motor control for tracking systems; therefore it would not be necessary to develop new solutions. However, this problem represented the simplest test bed for the study of coupled chaotic systems, both in a simulated environment and for its implementation in a real platform. Another equally important goal of this research is the possibility of having new insights about how the coordination of multiple degrees of freedom emerges in human infants.

According to neurosciences, all behavior is mediated by the central nervous system (brain and spinal cord) which is separated but functionally interconnected with the peripheral nervous system (continuous stream of sensory information about the environment). Simply put, the major difference between voluntary and reflexive movements is the intervention or not of the central nervous system [8]. In practice, it is not possible to separate the modulation signals coming from the brain into the muscles of our eyes. But according to the results of this experiment, we could speculate that visual tracking is just a reactive behavior given a saliency in our visual field. These saliencies are the necessary modulations given by our central nervous system and its areas of emotions, experiences, needs, etc.

The next section describes the basics of coupled chaotic systems together with the model of behavior emergence proposed in [2]. In section III a description of the simulation setup used for smooth pursuit is presented together with the results. Section IV presents the physical platform and results of the implementation of this approach; and, finally, conclusions and guidelines for future work are summarized in section V.



Fig. 1. Bifurcation plot for logistic map

II. COUPLED CHAOTIC SYSTEMS

A. Introduction to chaos

The word 'chaos' has been used to represent a part of nonlinear dynamical systems theory that deals with the unpredictable behavior of a system governed by deterministic rules, [9]. It is often easier to understand what chaos is through the examples found in almost all the areas of sciences studying nature: it can be found in the way the weather changes every year (Lorentz); in the way the planets and all other celestial objects influence each other and move in space (Poincaré); in the dynamics of population grow (May); the turbulence generated in fluid systems (Libchaber); etc.

One of the most common, and probably the simplest, deterministic rule that generates chaos is the logistic map, Eq. (1). This second-order difference equation was studied by the biologist Robert May as a model of population growth. In this equation, the parameter α controls the nonlinearity of the system. In order to keep the system bounded between 0 and 1, α takes values between 0 and 2, Fig. (1).

$$f(x_n) = 1 - \alpha x_{n-1}^2 \tag{1}$$

A stand-alone logistic map (internal feedback without external influences) stabilizes in an specific behavior depending on its initial condition and the value of α . This very simple rule can generate fixed points, Fig. 2a; periodic oscillations of period two (Fig. 2b), period four (Fig. 2c); and following the period doubling path until reaching a chaotic behavior, Fig. 2d.

B. Coupled Map Lattices (CML)

CML were introduced by Kunihiko Kaneko in the middle of the 1980's as an alternative for the study of spatiotemporal chaos [1]. In short, this kind of dynamical systems use discrete partial difference equations to study the evolution of a process described by discrete steps in space and time but with continuous states. Equation (2) describes the dynamics of CML, whereas Eq. (1) represents the logistic map used in this work.



Fig. 2. Logistic map, different outputs depending on α

$$x_n^i = (1 - \epsilon)f(x_{n-1}^i) + \frac{\epsilon}{2} \{ f(x_{n-1}^{i+1}) + f(x_{n-1}^{i-1}) \}$$
(2)

Where x_n^i is a variable at discrete time step n and a lattice point *i*. x represents a set of field variables which could be temperature, position measurements, velocities, etc. There are two parameters: α controlling the level of chaoticity of the system and ϵ controlling the coupling level among neighbor elements.

C. Globally Coupled Maps (GCM)

These kinds of maps were also introduced by Kaneko and represent a network of chaotic elements with interactions among all of them. While CML interact with specific points within the lattice, each of the nodes in a Globally Coupled Map (GCM) interacts with all the others, Eq (3). Due to the chaotic nature of the system, specified by α , it is possible to see one of the main properties of chaotic systems: two slightly different initial conditions amplify their difference through time. On the other hand, ϵ tries to synchronize the activations of all these chaotic elements by coupling them. In between these two states of complete chaos and complete synchronization, interesting states emerge like the formation of clusters oscillating in different phases and amplitudes.

$$x_{n}^{i} = (1 - \epsilon)f(x_{n-1}^{i}) + \frac{\epsilon}{N} \sum_{j=1}^{N} f(x_{n-1}^{j})$$
(3)

During the last two decades, these two categories have been the subjects of thorough investigation with the aim of describing them both qualitatively and quantitatively. The effects of varying both, chaoticity and coupling factor, in standalone CML and GCM systems were studied meticulously by Kaneko's group in the late 1990's [10], [11]. Approximate phase diagrams were sketched covering the whole spectrum of synchronization among the interacting chaotic elements of a network.



Fig. 3. Body-environment interaction through coupled chaotic fields

D. Coupled Chaotic Fields

The model used in this project is based on the approach followed by Kuniyoshi and Suzuki [2]. The main idea behind this model is to make use of the freedom given by the chaoticity of the system; and, on the other hand, the limitations imposed by the synchronization of all the elements. It is being used both, the local interaction (CML) and the global interaction (GCM). The system is depicted in Fig. 3. Each one of the blocks containing "Chaotic" elements and their relationship constitute the core of the system and it is defined by Eq. (4) and (5). The function f represents the logistic map, Eq. (1).

$$u_{n}^{i} = f \left\{ s_{n-1}^{i} + \epsilon_{1} (\bar{s}_{n-1} - s_{n-1}^{i}) + \epsilon_{2} \left(\frac{s_{n-1}^{i+1} - s_{n-1}^{i-1}}{2} - s_{n-1}^{i} \right) \right\}$$
(4)

$$m_n^i = G_u(u_n^i + O_u)$$

$$s_n^i = G_s(r_n^i + O_s)$$
(5)

Where m is the output applied to each motor, s and u are inputs and outputs respectively of the chaotic field block, and r is the raw value coming from the sensors. Finally, G_u , G_r , O_u , and O_r are gains and offsets of the sensors and motors respectively; these values are applied in the same magnitude to all the elements of the system.

III. SIMULATION

A. Software

To simulate the dynamics of an artificial eye, a virtual environment named Webots has been used [12]. This software is based on the Open Dynamics Engine libraries for reasonably accurate physics simulation such as the effect of gravity and friction. The time step for the simulations was fixed to 32 milliseconds and the experiments were done without the influence of gravity and with a minimum value of friction.



Fig. 4. Biological eye and its virtual counterpart for the experiment



Fig. 5. Screenshot of the virtual setup

B. The setup

The virtual eye was created using two rotational joints, one perpendicular to the other in order to simulate the "yaw" and "pitch" motions of a real eye, Fig. 4. Each joint is also modeled by a spring and a damper, trying to replicate the physical characteristics of real muscles. These two motors are actuating the virtual eye in the same way as the main four muscles do in biological eyes.

A virtual camera mounted in front of the eye gives the visual input needed for modulating the chaotic field. The width and height were fixed to 32 x 32 pixels with a field of view of 0.5 radians. It is assumed that values of saliency are obtained from other visual components. One of these values to be tracked was simulated by a black circular shape moving on a white screen, Fig. 5. The initial conditions were defined by having the object out of the field of view of the eye and generating a circular motion from zero to a maximum speed, and slowing down to zero again for two cycles, then changing direction for another two cycles of zero to maximum speed and so on. This motion was used as a basic test for the robustness of the system.

The input to the system is given by the difference between the center of the observed object within the field of view and the position of the center of the eye, for both the vertical and horizontal motions. The outputs from the chaotic field are fed as speed values to the motors after applying the respective offset and gain.

The methodology for tuning the gains and offsets was done by approximating the average of the raw output from the logistic map towards a zero average of the motor activation values. In other words, offsets and gains should be chosen in such a way that the activations from the logistic map oscillate around zero. Our simulation worked with the following offsets and gains: $G_u=1.0$, $G_r=1.0$, $O_u=0.0$, and $G_s=-0.72$, for $\alpha=1.9$ and $\epsilon=0.1$.



Fig. 6. Motion of the eye ($\alpha = 1.9, \epsilon = 0.1$)

C. Results

As mentioned before, the outputs from our globally coupled map were treated as speeds; in other words, we wanted our system to be controlled in velocity. However, it was proven that the same configuration and parameters were enough for controlling our virtual eye either in position or in torque. The trajectory followed by our virtual eye can be observed in Fig. 6. The first reaction, once the object has entered into the eye's field of view, is to move toward the object. As we can see, the adaptation to the path of motion is immediate, there are no overshooting or oscillations.

Note that there is no way of discerning the moments where the object is not moving or moving at full speed, this tell us how adaptive the system is to changes in the environment. Even though the tracking is not accurate, the object remains inside the field of view and almost in the center of the eye throughout the simulation time.

The simulation was run over more than two cycles of increasing and decreasing the object's velocity, Fig. 7. The overshooting observed at the beginning of the plot is the result of the object entering into the field of view of the eye; in less than one second, the system adapts to the recent change in the environment. It is possible to observe those moments when the object is changing direction, its speed decreases to zero; the relative displacement of the center of the eye and the object decreases even more.

Finally, Fig. 8 represents the relationship between the input to the GCM and its output on both chaotic units, yaw and pitch. The inputs to this function are the errors in both horizontal and vertical directions after being modified by their previous states and the influence among each other. The linearity of both units is necessary for the tracking to occur.

The "smooth pursuit"-kind of motion was tested using different values of α (chaoticity factor) and ϵ (coupling factor). However the tracking behavior was found to be optimal for a high chaoticity and small coupling. When decreasing α to values smaller than the critical point for being inside the desynchronized areas ($\alpha \approx 1.34$), it was possible to see the appearance of other interesting behaviors like avoidance of the



Fig. 7. Relative displacement between eye and object.



Fig. 8. Return maps for both chaotic units. The dotted line is shown as refence for the logistic map.

target or a sort of "boring" tracking, following the target for a short time but relaxing after a while. The motion of eye and target for the later case is showed in Fig. 9.

IV. IMPLEMENTATION

The results from the simulation gave us enough confidence to implement this algorithm in a real platform. The following subsections describe the experimental setup and the results of implementing the GCM algorithm for the activation of two degrees of freedom.



Fig. 9. A "boring"-kind of tracking ($\alpha = 1.05, \epsilon = 0.1$)



Fig. 10. Picture of the iCub's head

A. Software

An open-source framework for robotics named YARP (Yet Another Robot Platform) was used for the implementation of the algorithms. The main features of YARP are: support for inter-process communication, image processing, and a class hierarchy to code reuse across different hardware platforms [13]. The programming language in YARP is C++; however, a set of libraries has been developed to allow other programs, like Matlab, access YARP.

As mentioned before, the focus of this project is not the extraction of saliencies from the image, which is in itself a hard problem in computer vision. A tracking algorithm available in the YARP repository was used as the visual component in charge of providing us with the horizontal and vertical coordinates of a moving object. With this information we focus our efforts on the motor control problem.

B. Hardware

A copy of the iCub's head from the RobotCub project [14] was built to test this and future experiments. The head has six degrees of freedom: yaw, pitch and roll for the neck, a single pitch motion for both eyes and independent yaw motors for each eye. Three Faulhaber DC-micromotors [15] are used for moving the eyes; each motor contains an incremental encoder that provides the position of the joint at any time. We invite the reader to visit the project's webpage for having more information about hardware and software. The RobotCub project has been thought to be distributed as an open platform both in hardware and software. Fig. 10 shows a picture of the platform.

Again, the difference between the center of the observed object within the field of view and the position of the center of the eye, for both vertical and horizontal motions, was the value that modified the way the chaotic system behaves. The algorithm governing the dynamics for this part of the project was the same as in the simulation; however, the gains and offsets had to be modified, Eq. (4, 5) since the values representing mass, inertia, friction and gravity are different in both frameworks. Using the same methodology for adjusting



Fig. 11. Motion of the eye



gains and offsets as before, these values were fixed to: $G_u=1.0$, $G_s=1.0$, $O_u=0.0$, and $O_s=-0.8$; $\alpha = 1.9$, and $\epsilon = 0.1$.

C. Results

Figure 11 depicts the motion of the eye as well as the motion of the target. When analyzing these results, it is important to remember the following working conditions: first, the target was moved in all possible directions and with different speeds; second, the target was moved in several occasions into the limits of the visual field after reaching the mechanical constraints. Independently of the direction or the acceleration of the target, it remains inside the field of view within a maximum error of 10 degrees, Fig. 12.

The return maps are depicted in Fig. 13 and describe the relationship of input and output in the GCM. In contrast with the simulation, the use of a larger area in the logistic map is seen. These plots show activations in the left side of the logistic map; however, activations reaching and trespassing the top of the map were also observed in several trials. In those experiments, the target was moved at very low speed around the mechanical and visual limits of the system.



Fig. 13. Return maps for both chaotic units. The dotted line is shown as refence for the logistic map.

V. CONCLUSIONS AND FUTURE WORK

A. Conclusions

A very simple experiment for demonstrating the feasibility of applying coupled chaotic systems in the area of cognitive developmental robotics has been shown in this project. Tracking an object moving in front of a camera has been solved in several ways previously, from using very simple trigonometric solutions to advanced control algorithms. However, this task represented the simplest test bed for the study of emergence of a reactive behavior, both in a simulated environment and for its implementation in a real platform.

A virtual setup consisting of only two rotational joints and a camera was created to replicate the sensori-motor configuration of a real eye. We have demonstrated that, once the object enters the field of view, this input is enough for the self-organization of the controller that generates the torques applied to each of the joints of our devices. No learning or specific coding of the task is needed, which results in a very fast reactive behavior.

By playing with the values of α (chaoticity factor) and ϵ (coupling factor) we saw that a smooth pursuit behavior can change to other 'non-tracking' patterns like following during some time and escaping from the target in some others. This very simple action tells us that other visual behaviors can be achieved without much effort from the designing part to simulate specific cognitive actions.

The implementation of this algorithm in a real platform was straight forward. A copy of the iCub's head from the RobotCub project [14] was used with only minor changes in offsets and gains. The tracking algorithm used in the implementation was taken from the YARP repository [16]. The algorithm was tested by changing both the chaoticity of the system and the coupling among its elements. In both cases, simulation and implementation, the smooth pursuit behavior emerges when the system is highly chaotic and there is a weak coupling among its elements.

The focus of the present research is the coordination of several degrees of freedom for smooth pursuit but we found that other cognitive behaviors are also possible by changing a single parameter in our system. The work reported in this article represents the ground for building a more complex architecture for sensori-motor integration and cognitive development.

B. Future work

The iCub's head has also an inertial sensor which will be included in the future as another element influencing the chaotic field. Future work involves the emergence of a coordinated motion between both eyes and finally among the motors representing the three degrees of freedom of neck (yaw, pitch and roll) and the three degrees of freedom of the eyes (left eye yaw, right eye yaw, and both eyes pitch). Several questions should be addressed regarding the correspondences between this research and the biological counterpart; for example, if a smooth pursuit behavior emerged from the interaction of chaotic units, could it be possible to obtain other visual behaviors like vestibulo-ocular reflex (VOR), vergence or saccades in the same way?

ACKNOWLEDGMENTS

The work presented in this paper has been in part supported by the ROBOTCUB project (IST-2004-004370), funded by the European Commission through the Unit E5 -Cognitive Systems. Boris Duran would like to thank also to Paul Fitzpatrick for all comments, suggestions, and especially for the hand given when working with YARP.

References

- K. Kaneko and I. Tsuda, Complex Systems: Chaos and Beyond. Springer-Verlag, 2001.
- [2] Y. Kuniyoshi and S. Suzuki, "Dynamic emergence and adaptation of behavior through embodiment as a coupled chaotic field," in *Proceedings* of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, conference, pp. 2042–2049.
- [3] G. Metta, A. Gasteratos, and G. Sandini, "Learning to track colored objects with log-polar vision," *Mechatronics*, vol. 14, no. 9, pp. 989– 1006, November 2004.
- [4] A. Bernardino and J. Santos-Victor, "Binocular visual tracking: Integration of perception and control," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 1080–1094, December 1999.
- [5] D. Coombs and C. Brown, "Real-time binocular smooth pursuit," *International Journal of Computer Vision*, vol. 11, no. 2, pp. 147–164, 1993.
- [6] S. Aja Fernandez, C. Alberola Lopez, and J. Ruiz Alzola, "A fuzzycontrolled kalman filter applied to stereo-visual tracking schemes," *Signal Processing*, vol. 83, no. 1, pp. 101–120, January 2003.
- [7] S. Kumarawadu, K. Watanabe, K. Kiguchi, and K. Izumi, "Self-adaptive output tracking with applications to active binocular tracking," *Journal* of Intelligent and Robotics Systems, vol. 36, no. 2, pp. 129–147, 2003.
- [8] E. Kandel, J. Schwatrz, and T. Jesell, *Principles of Neural Sciences*. McGraw-Hill, 2000.
- [9] S. Strogatz, *Nonlinear dynamics and chaos*. New York: Addison Wesley, 1994.
- [10] K. Kaneko, "Pattern dynamics in spatiotemporal chaos," *Physica D: Nonlinear Phenomena*, vol. 34, pp. 1–41, Jan-Feb 1989.
- [11] —, "Clustering, coding, switching, hierarchical ordering and control in a network of chaotic elements," *Physica D: Nonlinear Phenomena*, vol. 41, pp. 137–172, March 1990.
- [12] (2002) Cyberbotics, webots. [Online]. Available: http://www.cyberbotics.com/
- [13] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, pp. 043–048, 2006.
- [14] (2004) The robotcub project. [Online]. Available: http://www.robotcub.org/
- [15] (2002) Faulhaber group. [Online]. Available: http://www.minimotor.ch/
 [16] (2003) Yet another robot platform. [Online]. Available: http://yarp0.sourceforge.net/

A.3 Paper III

B. Durán, Y. Kuniyoshi and G. Sandini. Eyes-neck coordination using chaos. Springer Tracts in Advanced Robotics, vol. 44, pp. 83 - 92. (2008).

Eyes-Neck Coordination Using Chaos

Boris Durán¹, Yasuo Kuniyoshi², and Giulio Sandini¹

 Italian Institute of Technology - University of Genova, Via Morego 30, 16145 Genova, Italy, boris@unige.it, giulio.sandini@iit.it,
 ² University of Tokyo, Laboratory for Intelligent Systems and Informatics,

Eng. Bldg. 2. 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan kuniyosh@isi.imi.i.u-tokyo.ac.jp,

Abstract. The increasing complexity of humanoid robots and their expected performance in real dynamic environments demand an equally complex, autonomous and dynamic solution. Our approach for the creation of real autonomy in artificial systems is based on the use of nonlinear dynamical systems. The purpose of this research is to demonstrate the feasibility of using coupled chaotic systems within the area of cognitive developmental robotics.

Using a robotic head, we demonstrate that the visual input coming into the head's eyes is enough for the self-organization of the axes controlling the motion of eyes and neck. No specific coding of the task is needed, which results in a very fast adaptation and robustness to perturbations. Another equally important goal of this research is the possibility of having new insights about how the coordination of multiple degrees of freedom emerges in human infants. We show that the interaction between body and environment modifies the inner connections of the controlling network resulting in the emergence of a tracking behavior.

1 INTRODUCTION

Most of today's humanoid platforms follow an almost 50-year-old tradition of control theory that started with the industrial automation at the beginning of the 1960s. The methodology followed by this approach is based on modeling as precise as possible both the plant and the controller; and filtering or processing as noise the different unexpected circumstances that could occur during the operation of the system. This approach has worked pretty well when the system is in a fixed framework and the environmental conditions are known and controlled; however, this will not be the case for humanoid robots of the future. It is absolutely necessary to start working on a different approach if we want to design and build systems that move and act in the same kind of dynamic environments where humans move and act. A more adaptive and flexible theory is needed in order to 'control' a device that is supposed to move within an ever-changing environment. These are our first steps towards the design and implementation of a real autonomous cognitive architecture based on nonlinear dynamical systems. Although the study of nonlinear dynamical systems and chaos has also a long history, real applications that make direct use of chaos theory have not been fully developed. The purpose of this research is to demonstrate the feasibility of using coupled chaotic systems [1] within the area of cognitive developmental robotics. Based on the model of behavior emergence introduced by Kuniyoshi et al. [2], we study the coordination of multiple degrees of freedom in humanoid robots.

The task of tracking an object has been fully studied and many solutions presented before. Based either in position errors or velocity mismatches, some approaches try to control the activation of motors by means of robust PID controllers [3–5], while others base their controllers in fuzzy logic [6] or neural networks [7]. In any case, the common methodology in these approaches is to compute expensive Jacobian and kinematic expressions thinking in all the possible circumstances the system could encounter.

All these works comprehend the state of the art in motor control for tracking systems; therefore it would not be necessary to develop new solutions. However, the tracking problem represented the simplest test bed for the study of coupled chaotic systems, both in a simulated environment and for its implementation in a real platform. Our approach differs from previous work mainly in two aspects: first, our system does not need to deal with complex equations of kinematics and dynamics; second, the main goal behind our research is not to improve the performance of existing algorithms but, through our experiments, start building the basis of a dynamic model for motion emergence that embrace as a single entity body and environment. Following Esther Thelen and Linda Smith's suggestion that "action and cognition are also emergent and not designed" [8], another equally important goal of this research is the possibility of having new insights about how the coordination of multiple degrees of freedom emerges in human infants.

The following section contains a short introduction on chaos and coupled chaotic systems; as well as a description of the model of behavior emergence proposed in [2]. Section III describes the experimental setup and the results of our experiments from the implementation of our model when working with constant parameters. In Section IV it is presented the results of a developmental process in a five degree of freedom implementation of our approach. Finally, conclusions and guidelines for future work are summarized in section V.

2 Coupled Chaotic Systems

2.1 A Short Introduction to Chaos

The word 'chaos' has been used to represent a part of nonlinear dynamical systems theory that deals with the unpredictable behavior of a system governed by deterministic rules, [9]. One of the most common, and probably the simplest, deterministic rule that generates chaos is the logistic map (1). This second-order difference equation was studied by the biologist Robert May as a model of population growth [10]. In this equation, the parameter α controls the nonlinearity of the system. In order to keep the system bounded between -1 and 1, α takes values between 0 and 2, Fig. 1.

$$f(x_n) = 1 - \alpha x_{n-1}^2 \tag{1}$$

A stand-alone logistic map (internal feedback whitout external influences) stabilizes in an specific behavior depending on its initial condition and the value of α . This very simple rule can generate fixed points, Fig. 1a; periodic oscillations of period two, Fig. 1b; period four, Fig. 1c; and following the period doubling path until reaching a choatic behavior, Fig. 1d.



Fig. 1. Left, bifurcation plot for logistic map. Right, different outputs for Logistic Map depending on α

2.2 Coupled Maps with Adaptive Connections

Coupled Map Lattices (CML) and Globally Coupled Maps (GCM), were introduced by Kunihiko Kaneko in the middle of the 1980's as an alternative for the study of spatiotemporal chaos [1]. In short, this kind of dynamical systems use discrete partial difference equations to study the evolution of a process described by discrete steps in space and time but with continuous states. Two parameters control the dynamics of these maps: a chaoticity factor and the strength of connections among their elements.

Due to the chaotic nature of the system, it is possible to see one of the main properties of chaotic systems: two slightly different initial conditions amplify their difference through time. On the other hand, the system tries to synchronize the activations of all its chaotic elements by coupling them. In between these two states of complete chaos and complete synchronization, interesting states emerge like the formation of clusters oscillating in different phases and amplitudes.

The study of dynamically varying the connections among the elements in a GCM was done by Ito and Kaneko [11, 12]. The model is described by the set of equations in (2). The first equation correspond to a GCM, where f represents

a chaotic map; (2b) updates each unit's connections coming from other units in the network; and (2c) specifies the hebbian rule governing the relationship between all units.

$$x_{n}^{i} = f\left((1-\epsilon)x_{n-1}^{i} + \epsilon \sum_{j=1}^{N} w_{n}^{ij}x_{n-1}^{j}\right),$$
(2a)

$$w_{n+1}^{ij} = \frac{\left[1 + \delta g(x_n^i, x_n^j)\right] w_n^{ij}}{\sum_{j=1}^N \left[1 + \delta g(x_n^i, x_n^j)\right] w_n^{ij}},$$
(2b)

$$g(x,y) = 1 - 2|x - y|$$
(2c)

In (2b), δ represents the degree of plasticity of the connections and ranges from 0 to 1. The weights w^{ij} in (2b) refer to the influence from unit j going into unit i. All self-connections were set to 0; and the initial condition for all remaining connections are equal to 1/(N-1), N being the number of chaotic units.

2.3 A Model for Behavior Emergence

The states of each of the elements in a GCM, or a CML, depend only on the internal dynamics of these systems; they are not influenced in any moment by an external force. When taking these concepts to robotic applications it is necessary to think in a way of including the environment within the dynamics of the system.

The model used in this project is based on the approach followed by Kuniyoshi and Suzuki [2]. Their model uses both, the local interaction (CML) and the global interaction (GCM) but with the environment as the external force influencing the internal dynamics of the network. In our case, only GCM was used since no extra benefit was seen when including local connections; nevertheless the overall approach is the same, Fig. 2.



Fig. 2. Body-environment interaction through coupled chaotic fields

3 Implementation

A copy of the iCub's head, the humanoid platform of the Robotcub's project [13], was used in the present work. The head's hardware and software components will be described in the following subsections together with the implementation of the algorithms used to create a dynamic smooth pursuit.

3.1 Hardware and Software

The head has six degrees of freedom: yaw, pitch and roll for the neck, a single pitch motion for both eyes and independent yaw motors for each eye. DCmicromotors are used for moving the different joints; each motor contains an incremental encoder that provides the position of the joint at any time. All motors and sensors are controlled by a suite of DSP chips which channel data over a CAN bus to a computer in charge of iCub's high-level behavioral control [14].

Due to the large amount of sensori-motor information generated within the platform the iCub's software was configured to run in parallel on a distributed system of computers. An open-source framework for robotics named YARP (Yet Another Robot Platform) [15] was used for the implementation of the algorithms. It is important to mention that the focus of this project is not the extraction of saliencies from moving images, which is in itself a hard problem in computer vision. A tracking algorithm available in the YARP repository was used as the visual component in charge of providing us with the horizontal and vertical coordinates of a moving object. With this information we focus our efforts on the motor control problem.

3.2 Methodology

Each camera provides two quantities: the position of the target in vertical and horizontal directions. These values modify the position of each motor; thus generating a coupled chaotic system with 6 logistic maps, Fig. 3. The algorithm governing the dynamics of the system is governed by (3).

$$u_{n}^{i} = f\left((1-\epsilon)s_{n-1}^{i} + \epsilon \sum_{j=1}^{N} w_{n}^{ij}s_{n-1}^{j}\right)$$
(3a)

$$m_n^i = G_m(u_n^i + O_m)$$

$$s_n^i = G_s(r_n^i + O_s)$$
(3b)

Where m is the output applied to each motor as speed values, s and u are inputs and outputs respectively of the chaotic field block, and r is the raw value coming from the sensors. Finally, G_m , G_s , O_m , and O_s are gains and offsets of the sensors and motors respectively; these values are applied in the same magnitude to all elements in the system.

The methodology for tuning offsets was done by approximating the average of the raw output from the logistic map towards a zero average of the motor


Fig. 3. iCub's sensorimotor diagram, 5dof actuation.

activation values. In other words, offsets should be chosen in such a way that the activations from the logistic map oscillate around zero. Gains G_m were chosen depending on the speed limits of the motors. The following parameters were fixed during all experiments: $G_s=1.0$, $O_s=-0.8$, $G_{LY} = G_{RY} = G_{EP}=25.0$, $G_{NY}=70$, $G_{NP}=35$, and $O_m=0.0$; $\alpha = 1.9$, and $\epsilon = 0.1$.

3.3 Results

The motion of both eyes and the motion of the head is shown in Fig. 4. This plot shows the motion of the eyes relative to the head and the motion of the head relative to its fixed position. In this plot is possible to see the coordination between eyes and neck. The target was moved in random directions and at different speeds. Since the joints of the neck give approximately an extra 60 degrees on each side and on each direction, an object can be tracked in a wider space. It was also observed an increase of the tracking speed when compared to the 3dof case (2-eye tracking). The motors in the neck help the motors in the eyes to follow the object in a faster way, especially in the yaw direction.

The coordination between both eyes and between eyes and neck in each direction can be more easily appreciated in Fig. 5. Since the tracking algorithm works on independent threads in each camera, different points in space are delivered to the GCM. This 'computer vision' problem creates the errors observed during some points during the experiments.

The activations of all units grouped in yaw, Fig. 6, and pitch Fig. 6 directions show the dynamics of the system. Here is also possible to see the coordination of chaotic units since all activations are gathered along the diagonal of each plot. The nonlinearity of the chaotic units give them enough freedom to use the rest of the space when needed but always staying and returning back to this diagonal.

The development of weak and strong connections among the chaotic units depend on the level of interaction they have through time, Fig. 7. Even though all



 ${\bf Fig.}\, {\bf 4.}$ Motion of both eyes and neck.



Fig. 5. Position of target w.r.t. center of eye: yaw motion, top; and pitch motion, bottom.



Fig. 6. Left, phase space (yaw). Right, phase space (pitch)

connections start with the same value, the system takes only a few time steps to separate in groups of strong and weak connections. A very interesting observation from this plot is that after approximately 500 steps, the connections arriving to any unit oscillate around the middle of the permitted strength. Extreme cases are with pitch units in each eye LP and RP which develop a very strong influence from the pitch motion of the neck NP but a zero influence from one to another. Yaw units develop a more balanced influence in their network, oscillating always around 0.5.



Fig. 7. Development of connections in time.

At time step 3500 the system has entered in an almost fully developed state where its internal connections vary very little. In the end, each unit is influenced by no more than two other units within the whole network, Fig. 8. As expected, two independent sub networks emerge after approximately 20 seconds. In one side all chaotic units fed by yaw motions strengthen their connections while weakening those towards and from 'pitch' units; and the same happens with those units fed by pitch motions when compared to 'yaw' units.

4 Conclusions and Future work

Conclusions

A very simple experiment for demonstrating the feasibility of applying coupled chaotic systems in the area of cognitive developmental robotics has been shown in this project. Tracking an object moving in front of a camera has been solved in several ways previously, from using very simple trigonometric solutions to advanced control algorithms. However, this task represented the simplest test bed for the study of emergence of a reactive behavior in a real platform.



Fig. 8. Initial (left), and final (right) configurations of the GCM.

A copy of the iCub's head [13], a 6 DOF robotic platform, was used to replicate the sensori-motor configuration of a real head. The tracking algorithm used in all experiments was taken from the YARP repository [15]. The experience obtained in previous experiments with the simulation and implementation of a single eye tracking [16] gave us enough confidence to increase the complexity of our model. The present work contains the results on the development of connections in the eyes-neck coordination problem (5 DOF).

We have demonstrated that a visual input is enough for the self-organization of a globally coupled map whose outputs are used as speed values activating each of the joints of our device. No specific coding of the task is needed, which results in a very fast reactive behavior. A very simple Hebbian rule was used to study the development of connections within the core of the system, a globally coupled map. From normalized initial connections we saw them changing through time, restructuring the 'brain' according to the experiences with the environment. In the final stage, two independent sub networks were formed, one containing yawrelated chaotic units only and the other pitch-related chaotic units only. The smooth pursuit behavior emerged during this process.

Future work

The iCub's head includes also an inertial sensor which will be used in the future as another element influencing the chaotic field. Several questions should be addressed regarding the correspondences between this research and the biological counterpart; for example, if a smooth pursuit behavior emerged from the interaction of chaotic units, could it be possible to obtain other visual behaviors like vestibulo-ocular reflex (VOR), vergence or saccades in the same way?

The tracking algorithm used in all experiments does not focus on the same point in both cameras; consequently a displacement is observed when comparing the centers of both images. Therefore, this algorithm will be modified in order to visually track the same point in space.

Acknowledgments

The work presented in this paper has been in part supported by the ROBOTCUB project (IST-2004-004370), funded by the European Commissionthrough the Unit E5 -Cognitive Systems.

References

- Kaneko, K., Tsuda, I.: Complex Systems: Chaos and Beyond. Springer-Verlag (2001)
- Kuniyoshi, Y., Suzuki, S.: Dynamic emergence and adaptation of behavior through embodiment as a coupled chaotic field. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems. (2004) 2042–2049
- Metta, G., Gasteratos, A., Sandini, G.: Learning to track colored objects with log-polar vision. Mechatronics 14(9) (November 2004) 989–1006
- Bernardino, A., Santos-Victor, J.: Binocular visual tracking: Integration of perception and control. IEEE Transactions on Robotics and Automation 15(6) (December 1999) 1080–1094
- Coombs, D., Brown, C.: Real-time binocular smooth pursuit. International Journal of Computer Vision 11(2) (1993) 147–164
- Aja Fernandez, S., Alberola Lopez, C., Ruiz Alzola, J.: A fuzzy-controlled kalman filter applied to stereo-visual tracking schemes. Signal Processing 83(1) (January 2003) 101–120
- Kumarawadu, S., Watanabe, K., Kiguchi, K., Izumi, K.: Self-adaptive output tracking with applications to active binocular tracking. Journal of Intelligent and Robotics Systems 36(2) (2003) 129–147
- 8. Thelen, E., Smith, L.: A dynamic systems approach to the development of cognition and action. The MIT Press Inc., Cambridge, MA (1994)
- 9. Strogatz, S.: Nonlinear dynamics and chaos. Addison Wesley, New York (1994)
- 10. May, R.: Simple mathematical models with complicated dynamics. Nature ${\bf 261}$ (June 1976) 459–467
- Kaneko, K.: Relevance of dynamic clustering to biological networks. Physica D: Nonlinear Phenomena 75(1) (Aug 1994) 55–73
- Ito, J., Kaneko, K.: Spontaneous structure formation in a network of dynamic elements. Phys. Rev. E 67(4) (Apr 2003) 046226
- 13. : Robotcub project. http://www.robotcub.org
- Sandini, G., Metta, G., Vernon, D.: The icub cognitive humanoid robot: An opensystem research platform for enactive cognition. Journal of Bionics Engineering 1(3) (2004) 191 – 198
- Metta, G., Fitzpatrick, P., Natale, L.: Yarp: Yet another robot platform. International Journal of Advanced Robotic Systems 3(1) (2006) 043–048
- Duran, B., Sandini, G.: Towards the implementation of a "chaotic" smooth pursuit. In: IEEE/RAS International Conference on Humanoid Robots. (2007) Accepted

Bibliography

- [1] Cyberbotics, webots, 2002.
- [2] Karen Adolph. Learning to learn in the development of action, volume 33, chapter 3, pages 91–122. Lawrence Erlbaum Associates, Inc., New Jersey, 2005. In the Minnesota Symposa on Child Psychology.
- [3] Kazuyuki Aihara. Chaotic neural networks. *Physics Letters A*, 144(6,7):333–340, Mar 1990.
- [4] S. Aja Fernandez, C. Alberola Lopez, and J. Ruiz Alzola. A fuzzycontrolled kalman filter applied to stereo-visual tracking schemes. *Signal Processing*, 83(1):101–120, January 2003.
- [5] Shun-Ichi Amari. Characteristics of randomly connected thresholdelement networks and network systems. *Proceedings of the IEEE*, 59(1):35–47, Jan. 1971.
- [6] Shun-Ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.
- [7] A. Babloyantz, J. M. Salazar, and C. Nicolis. Evidence of chaotic dynamics of brain activity during the sleep cycle. *Physics Letters A*, 111(3):152– 156, 1985.
- [8] A. Bernardino and J. Santos-Victor. Binocular visual tracking: Integration of perception and control. *IEEE Transactions on Robotics and Automation*, 15(6):1080–1094, December 1999.
- [9] Rodney Brooks. A robust programming scheme for a mobile robot. In Proc. of the NATO Advanced Research Workshop on Languages for sensor-based control in robotics, pages 509–522, London, UK, 1987. Springer-Verlag.

- [10] Wayne Christensen and Clifford Hooker. An interactivist constructivist approach to intelligence: self-directed anticipative learning. *Philosophi*cal Psychology, 13(2):5–45, 2000.
- [11] David Coombs and Christopher Brown. Real-time binocular smooth pursuit. International Journal of Computer Vision, 11(2):147–164, 1993.
- [12] Keith Davids, Chris Button, and Bennett S. Modeling human motor systems in nonlinear dynamics: Intentionality and discrete movement behaviors. Nonlinear Dynamics, Psychology, and Life Sciences, 3(1):3– 30, 1999.
- [13] Gregor Schöner Evelina Dineva. Dynamic instabilities as mechanisms for emergence. *Developmental Sciences*, 10(1):69–74, 2007.
- [14] Wolfram Erlhagen and Gregor Schöner. Dynamic field theory of movement preparation. *Psychological Review*, 109(3):545–572, 2002.
- [15] Dario Floreano, Phil Husbands, and Stefano Nolfi. Evolutionary robotics. In *Handbook of Robotics*. Springer Verlag, Berlin, 2008.
- [16] Walter Freeman. Mass Action in the Nervous System. Academic Press Inc., New York, 2004.
- [17] Walter Freeman and Giuseppe Vitiello. Nonlinear brain dynamics as macroscopic manifestation of underlying many-body field dynamics. *Physics of Life Reviews*, 3(2):93–118, 2006.
- [18] Stephen Guastello. Motor control research requires nonlinear dynamics. American Psychologist, 61(1):77–78, 2006.
- [19] Derek Harter. Towards a model of basic intentional systems: Nonlinear dynamics for perception memory and action in autonomous adaptive agents. PhD thesis, The University of Memphis, May 2004.
- [20] Robert Hristovski, Keith Davids, Duarte Araujo, and Chris Button. How boxers decide to punch a target: Emergent behaviour in nonlinear dynamical movement systems. *Journal of Sports Science and Medicine*, 5:60–73, 2006.

- [21] Michael Hutchinson and Phillip D. Swanson. Chaos theory and the treatment of refractory status epilepticus: Who benefits from prolonged anesthesia, and is there a better way? *Medical Hypotheses*, 68(2):439– 441, 2007.
- [22] Junji Ito and Kunihiko Kaneko. Spontaneous structure formation in a network of dynamic elements. *Phys. Rev. E*, 67(4):046226, Apr 2003.
- [23] Jeffrey Johnson, John P. Spencer, and Gregor Schöner. Moving to higher ground: The dynamic field theory and the dynamics of visual cognition. Dynamics and Psychology [special issue]. New Ideas in Psychology, 26:227-251, 2008.
- [24] Kunihiko Kaneko. Pattern dynamics in spatiotemporal chaos. Physica D: Nonlinear Phenomena, 34:1–41, Jan-Feb 1989.
- [25] Kunihiko Kaneko. Clustering, coding, switching, hierarchical ordering and control in a network of chaotic elements. *Physica D: Nonlinear Phenomena*, 41:137–172, March 1990.
- [26] Kunihiko Kaneko. Relevance of dynamic clustering to biological networks. *Physica D: Nonlinear Phenomena*, 75(1):55–73, Aug 1994.
- [27] Kunihiko Kaneko and Ichiro Tsuda. Complex Systems: Chaos and Beyond. Springer-Verlag, 2001.
- [28] Leslie Kay, Koji Shimoide, and Walter Freeman. Comparison of eeg time series from rat olfactory system with model composed of nonlinear coupled oscillators. *International Journal of Bifurcation and Chaos*, 5(3):849–858, 1995.
- [29] Klaus Kopecz and Gregor Schöner. Saccadic motor planning by integrating visual information and pre-information on neural dynamic fields. *Biological Cybernetics*, 73(1):49–60, 1995.
- [30] Jeffrey Krichmar and Gerald Edelman. Brain-based devices for the study of nervous systems and the development of intelligent machines. Artificial Life, 11(1-2):63–77, 2005.

- [31] Sisil Kumarawadu, Keigo Watanabe, Kazuo Kiguchi, and Kiyotaka Izumi. Self-adaptive output tracking with applications to active binocular tracking. *Journal of Intelligent and Robotics Systems*, 36(2):129–147, 2003.
- [32] Yasuo Kuniyoshi and Shinsuke Suzuki. Dynamic emergence and adaptation of behavior through embodiment as a coupled chaotic field. In Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2042–2049, 2004.
- [33] Mark Latash, John Scholz, and Gregor Schöner. Toward a new theory of motor synergies. *Motor control*, 11(3):276–308, 2007.
- [34] WANG Le, LI Guang, LI Xu, GUO Hongji, and Walter J. Freeman. A chaotic neural network mimicking an olfactory system and its application on image recognition. *Journal of Bionics Engineering*, 1(3):191–198, 2004.
- [35] Poramate Manoonpong, Frank Pasemann, and Hubert Roth. Modular reactive neurocontrol for biologically-inspired walking machines. *International Journal of Robotics Research*, 26(3):301–331, 2007.
- [36] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. Yarp: Yet another robot platform. International Journal of Advanced Robotic Systems, 3(1):043-048, 2006.
- [37] Giorgio Metta, Antonios Gasteratos, and Giulio Sandini. Learning to track colored objects with log-polar vision. *Mechatronics*, 14(9):989– 1006, November 2004.
- [38] Frank Pasemann. Dynamics of a single model neuron. Int. J. Bifurcation and Chaos, 3(2):271–278, 1993.
- [39] Frank Pasemann. Neuromodules: A dynamical systems approach to brain modelling. In *Supercomputing in brain research: From tomography* to neural networks, pages 331–348. World Scientific, 1995.
- [40] Frank Pasemann. A simple chaotic neuron. Physica D: Nonlinear Phenomena, 104(2):205–211, 1997.

- [41] Frank Pasemann. Synchronized chaos and other coherent states for two coupled neurons. *Physica D*, 128:236–249, 1999.
- [42] Frank Pasemann, Manfred Hild, and Keyan Zahedi. So(2)-networks as neural oscillators. In IWANN '03: Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks, volume 2686-1 of Lecture Notes in Computer Science, pages 144–151. Springer Berlin/Heidelberg, 2003.
- [43] Frank Pasemann, Uli Steinmetz, Martin Hülse, and Bruno Lara. Evolving brain structures for robot control. In *IWANN '01: Proceedings of* the 6th International Work-Conference on Artificial and Natural Neural Networks, volume 2085 of Lecture Notes in Computer Science, pages 410–417. Springer-Verlag, 2001.
- [44] Jean Piaget. The Construction of Reality In The Child. Basic Books, Inc., New York, 1954.
- [45] David Pincus. Dynamical systems theory and pain imagery: Bridging the gap between research and practice. Journal of Mental Imagery, 30(1-2):93-112, 2006.
- [46] David Pincus and Stephen Guastello. Nonlinear dynamics and interpersonal correlates of verbal turn-taking patterns in group therapy. *Small Group Research*, 36(6):635–677, 2005.
- [47] Mikhail Rabinovich, Pablo Varona, Allen Selverston, and Henry Abarbanel. Dynamical principles in neuroscience. *Reviews of Modern Physics*, 78(4):1213–1265, 2006.
- [48] Giulio Sandini, Giorgio Metta, and David Vernon. The icub cognitive humanoid robot: An open-system research platform for enactive cognition. Journal of Bionics Engineering, 1(3):191–198, 2004.
- [49] S. Schaal, P. Mohajerian, and A.J. Ijspeert. Dynamics systems vs. optimal control a unifying view. *Progress in Brain Research*, 165:425–445, 2007.
- [50] Gregor Schöner. Toward a New Grand Theory of Development? Connectionism and Dynamic Systems Theory Re-Considered., chapter Development as Change of System Dynamics: Stability, Instability, and Emergence., page In press. Oxford University Press, New York, 2008.

- [51] Gregor Schöner and Esther Thelen. Using dynamic field theory to rethink infant habituation. *Psychological Review*, 113(2):273–299, 2006.
- [52] Anne Schutte, John Spencer, and Gregor Schöner. Testing the dynamic field theory: Working memory for locations becomes more spatially precise over development. *Child Development*, 74(5):1393–1417, 2003.
- [53] Murray Shanahan. A cognitive architecture that combines internal simulation with a global workspace. *Consciousness and Cognition*, 15(2):433– 449, 2006.
- [54] D.-S. Shiau, W. Chaovalitwongse, L. D. Iasemidis, P. M. Pardalos, P. R. Carney, and J. C. Sackellares. Nonlinear dynamical and statistical approaches to investigate state transitions before epileptic seizures. pages 239–249, 2004.
- [55] Vanessa Simmering, Anne Schutte, and John P. Spencer. Generalizing the dynamic field theory of spatial cognition across real and developmental time scales. *Computational Cognitive Neuroscience [special issue]. Brain Research*, 1202:68–86, April 2008.
- [56] Elizabeth Spelke and Katherine Kinzler. Core knowledge. Developmental Science, 10(1):89–96, 2007.
- [57] Esther Thelen, Gregor Schöner, Christian Scheier, and Linda Smith. The dynamics of embodiment: A field theory of infant perseverative reaching. *Behavioral and Brain Sciences*, 24:1–86, 2001.
- [58] Esther Thelen and Linda Smith. A dynamic systems approach to the development of cognition and action. The MIT Press Inc., Cambridge, MA, 1994.
- [59] Ichiro Tsuda. Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behavioral and Brain Sciences*, 24(5):793–847, 2001.
- [60] Ichiro Tsuda and Hiroshi Fujii. A complex systems approach to an interpretation of dynamic brain activity i: Chaotic itinerancy can provide a mathematical basis for information processing in cortical transitory and nonstationary dynamics. In *Summer School on Neural Networks*, volume

3146 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2003.

- [61] David Vernon. Cognitive vision: The case for embodied perception. Image Vision Comput., 26(1):127–140, 2008.
- [62] David Vernon and Dermot Furlong. Philosophical foundations of ai. In 50 Years of Artificial Intelligence, volume 4850 of Lecture Notes in Artificial Intelligence, pages 53–62. Springer-Verlag, 2007.
- [63] Jiří Šíma and Pekka Orponen. General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation*, 15:2727–2778, 2003.
- [64] Claudia Wilimzig, Stefan Schneider, and Gregor Schöner. The time course of saccadic decision making: dynamic field theory. *Neural Net*works, 19(8):1059–1074, 2006.