ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Scuola di Ingegneria e Architettura
Campus di Cesena
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

ANALYSIS OF ROBOT DYNAMICS THROUGH COMPLEXITY MEASURES: AN APPLICATION IN AUTOMATIC DESIGN

Elaborata nel corso di: Sistemi Intelligenti Robotici

Tesi di Laurea di:
MATTIA BALDUCCI

Relatore:
Prof. ANDREA ROLI

ANNO ACCADEMICO 2014–2015
SESSIONE III

KEYWORDS

Boolean Networks
Complexity Measures
Automatic Design
Robot Dynamics

A me...
Alla mia famiglia...
Ai miei nonni Giuseppe e Luciano...

Abstract

EN: The application of measures deriving from information theory to complex systems provide useful tools to quantify some of systems properties. The same tools can be applied to robotics in order to improve the analysis and the syntesis of automatic designed robot control systems. In this thesis, the correlation of complexity measures with the fitness of robots trained for three different experiments have been investigated. Results obtained suggest that complexity measures are a promising tool for robotics, but their employement may be not trivial for composite tasks.

IT: L'applicazione di misure, derivanti dalla teoria dell'informazione, fornisce un valido strumento per quantificare alcune delle proprietà dei sistemi complessi. Le stesse misure possono essere utilizzate in robotica per favorire l'analisi e la sintesi di sistemi di controllo per robot. In questa tesi si è analizzata la correlazione tra alcune misure di complessità e la capacità dei robot di portare a termine, con successo, tre differenti task. I risultati ottenuti suggeriscono che tali misure di complessità rappresentano uno strumento promettente anche nel campo della robotica, ma che il loro utilizzo può diventare difficoltoso quando applicate a task compositi.

Contents

Introduction

Manually designing a robot controller is not a trivial task especially when the robot has to operate in a noisy and unpredictable environment. A potential alternative relies on the employment of automatic design techniques, which usually make use of complex networks as evolvable controllers. Typical examples of such complex networks are Artificial Neural Networks (ANNs) and Boolean Networks (BNs).

While a qualitative description of a robot behaviour can be easily performed by means of an objective function or a visual inspection, a way to formally describe internal network dynamics is still elusive. A common approach, adopted in complex system science (CSS), to try quantifying complex systems properties makes use of complexity measures deriving from information theory. Recently, this approach has been also employed in robotics to investigate if well performing robots share some kind of desired dynamics.

This thesis aims to investigate the existence of a correlation between complexity measures and fitness values of robots designed by means of an automatic design process. The objective is to improve the analysis and synthesis of robot control systems providing: 1) a methodology to formally describing robots dynamics, and 2) an high-level task-independent utility function to be used in combination with classical task-dependent ones.

Outline of the work

This thesis is organised as follows:

In Chapter 1, we introduce the main concepts of complex systems and complex system science. Moreover, we present the complexity measures that will be used in our analysis.

In Chapter 2, we explain how an automatic design process works and which is its structure. The last section of this chapter is dedicated to genetic algorithms which represent the search algorithm we chose to train our robots.

In Chapter 3, we present the model we used as robots controller: Boolean Networks. We provide an exhaustive description of their structure and dynamics, but we will focus on Random Boolean Networks which are a variation of particular interest for this work. Finally, we describe how these networks can be employed on robots.

Chapter 4 describes potential advantages deriving from the employment of complexity in robotics, and illustrates some previous important works related to this argument. The last section specifies how we implemented complexity measures to perform our analysis.

In Chapters 5 and 7 are discussed the first two experiments we performed, *phototaxis* and *obstacle avoidance*. We provide a description of experimental settings and a brief presentation of obtained results in terms of performance.

Chapters 6 and 8 contain respectively the analysis on robot trajectories, by means of complexity measures, for phototaxis and obstacle avoidance. Results will be presented and discussed.

In Chapter 9 we present the third, and last, experiment of this thesis: the *T-Maze* task. This task presents some important differences from the other two, such as the requirement of memory and the presence of different phases during the same run. Considerations, and conjectures about analysis results will be made in Chapter 10.

Considerations deriving from a comparison of the three tasks are provided in Chapter 11. In this analysis, we examine all the obtained results looking for phenomena that could not be discovered by observing tasks separately.

Chapter 1

Complex systems

Complex system science (*CSS*) represents a new scientific approach to study how parts of a system give rise to the collective behaviours of a system, and how the system itself interacts with the surrounding environment. Complex systems (CSs) exists in nature, examples can be the brain, a cell, or ant colonies, moreover many of the human built systems such as the economical system or the city traffic can be defined *complex*.

This Chapter provides a briefly introduction to CSs and CSS, a major part of the material presented has been taken from [31], reworked, and integrated.

The importance of studying complex systems is of immediate understanding, the capacity of creating reliable models of these systems and controlling them can potentially bring to huge advancements in many fields. For this reason CSS is multidisciplinary and involves mathematics, physics, computer science, biology, economy, philosophy, neurology, chemistry, anthropology, meteorology and many others.

The definition of complex system is elusive but an informal definition can be provided by summarizing the characteristics that CSs usually exhibit [23], notice that all or only some of this characteristics can be present:

- Numerosity (composed of many elements)
- Nonlinear dynamics
- Positive and negative feedbacks

- Some degree of spontaneous order in system behaviour (e.g. simmetry, periodicity, pattern and more)
- Robustness and lack of central control
- Emergence of global behaviours
- Hierarchical organization

A key feature of complex systems, which is at the basis of the concept of *emergence*, is that local rules of individual components produce a dynamics that is not possible to understand only by observing the parts composing the system. Another important concept directly connected to emergence is *self-organisation*, which refers to phenomena where some spatio-temporal structures emerge in a system without a direct external control, a typical example is given by snow crystals.

Before proceeding in the description of complex systems, it is important clarifying the mean of word *system*. *System* is used to identify entities (concrete or abstract) for which it is possible to identify some kind of boundaries, therefore it is possible to distinguish what belongs to the system and what is outside the boundaries (the environment). Of course, the identification of a system depends on the level of abstraction chosen. An example is the human body that can be certainly considered a system, but that can be decomposed in other systems like muscular or nervous systems, which are composed of cells and so on. In this case, the identification of system boundaries depends on the observer specific viewpoint.

The definition of *system* leads automatically to the concept of *model*. A *model* is a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system. A model only represents a delimited portion of the whole system and captures only some characteristics considered of interest. Thus, the creation of a model requires an abstraction process which involves simplification, aggregation, and omission of details. In CSS, models are widely used with the following objectives:

- understanding the system and investigating some of its properties;
- being able to control the system;
- making prediction about the future of the system;

In this thesis, for example, real robots have been modelled through virtual models and tested by means of computer simulations.

1.1 Order and chaos in dynamical systems

The term *dynamical systems* identifies a class of systems that evolve in time, thus a model of a system that belongs to this class usually contains the rules that govern the evolution in time of the system state. The system state is usually defined by a set of values considered of relevance for the system and that are measurable. Some important concept for dynamical systems are:

- *Trajectory*: represents the sequence of states encountered during the system evolution in time. Trajectories end in an *attractor* that can be a *fixed point*, a *cycle* or a more complex attractor;

- *Phase space*: is the space of the variables needed to characterise the system or, in other words, is a space in which all possible states of a system are represented;

- *State space*: is the space of the states of the system. It coincides with the phase space when the set of variables needed to characterise the system coincides with the system state.

Typical and widely studied models for dynamical systems are *ordinary differential equations* and *difference equations*, due to their immediate understanding and predisposition for computer simulation only *difference equations* (DE) will be briefly explained with the objective of introducing the concept of *order* and *chaos* in dynamical systems.
A typical DE is in the form:

$$x(t+1) = f[x(t), P, t] \tag{1.1}$$

where x, f and P can be vectors and represent respectively the system state, the transition functions, and the parameters of the model. When a system does not depend upon t it is named autonomous.

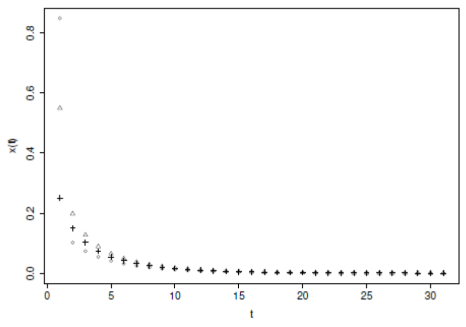
One popular difference equation is the *logistic map*:

$$x(t+1) = Rx_t(1 - x_t) \tag{1.2}$$

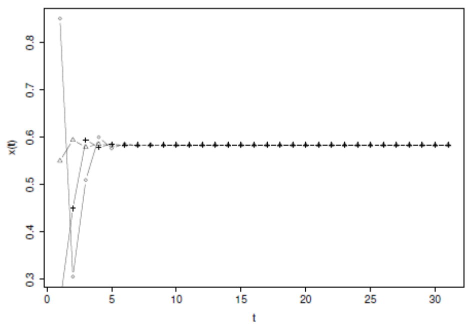
that can be seen as a discrete-time demographic model where: x_t is a number between 0 and 1 that represents the ratio of existing population to the maximum possible population, and R is a control parameter that permits to modify the equation behaviour. By iterating the equation and plotting the results some different behaviours can be observed depending on the value assigned to parameter R:

- for $R < 1$ the equation tends to 0, i.e. the population goes extinct. In this case the attractor is a *fixed point*, see fig.1.1(a);
- for $1 < R < 3$ the equation value oscillates in a first phase, then it eventually reaches a stable value. Also in this case the attractor is a *fixed point*, but the value changes with R, see fig.1.1(b);
- for $R > 3$ the equation never reaches a stable value but oscillates instead. In this case, the attractor is a *cycle* and its period starts from 2 and doubles while R increases, see fig.1.1(c).

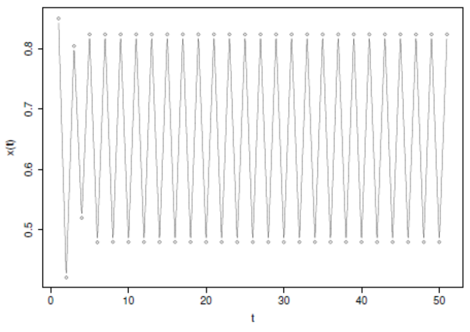
The equation results move from a fixed point attractor to a cycle attractor of increasing period. With values of R around 3.56995 no more finite period oscillations are visible (see fig.1.1(d)) and slight variations in the initial value of x_0 bring to dramatically different results over time, however, for some values of R after this threshold, it is still possible to have finite period attractors called *islands of stability*. Finally for $R > 4$ no more *islands of stability* exist, and a *chaotic* behaviour can be observed (see fig.1.2). Due to the deterministic nature of the *logistic map*, observed phenomena can not be accounted as a random behaviour also if it is almost unpredictable, this kind of attractor is defined *strange attractor*.



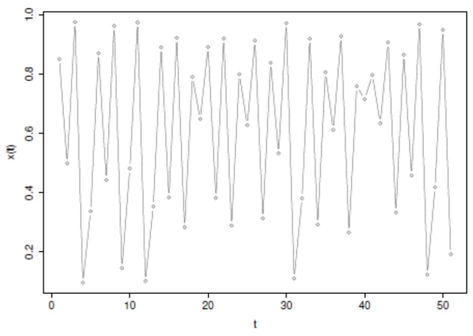
(a) R=0.8 and $x_0 \in \{0.25, 0.55, 0.85\}$



(b) R=2.2 and $x_0 \in \{0.25, 0.55, 0.85\}$



(c) R=3.3 and $x_0 = 0.85$



(d) R=3.9 and $x_0 = 0.85$

Figure 1.1: Graphical representations of Eq.1.2 for different values of R

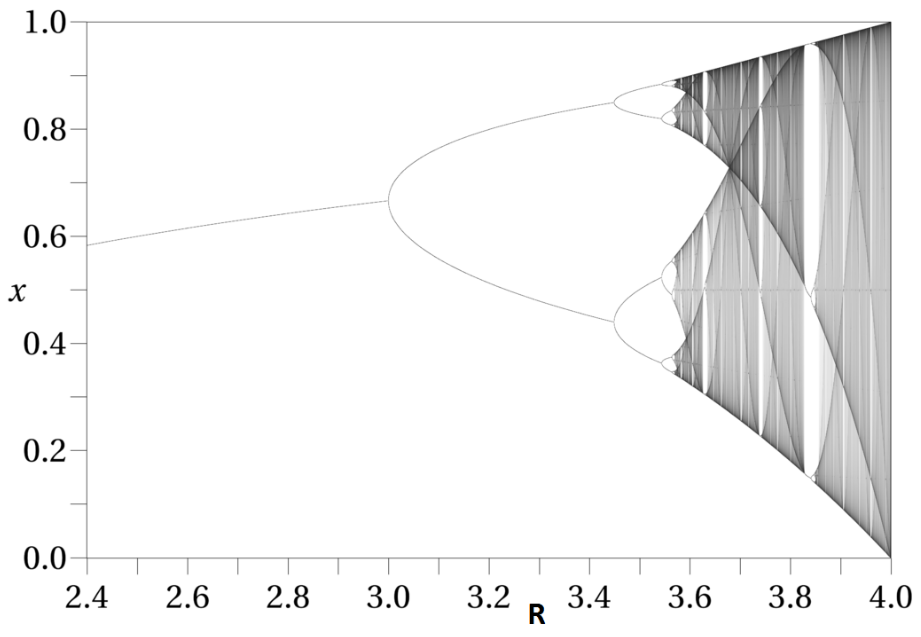


Figure 1.2: Bifurcation diagram for the logistic map. The attractor for any value of the parameter R is shown on the vertical line at that R.

1.2 Complexity and measures

In the previous section we pointed out how systems can behave in two different ways: they can be in an ordered regime, attaining a fixed point or cycling attractor and presenting regular behaviour, or they can be chaotic, following a strange attractor and showing an apparently random behaviour. Between these regime there is the realm of *complexity*, where systems exhibit a behaviour that is neither totally regular, nor completely chaotic. CSS tries to characterise, end even quantify, complexity.

Given a complex system, it is common to make use of qualitative descriptions to identify some of its properties such as *emergence* or *self-organisation*, this because does not exist a formal way to define them as for the *complexity* itself.

A useful way to quantify some properties of dynamical systems makes use of *complexity measures*. At the base of these measures there is the information theory, this link with complex systems is sustained by the hypothesis that a generic computational process can be thought as the evolution of a dynamical system in time, thus systems exposing complex behaviours should be characterised by an higher computational capability and, consequently, by higher values of some kind of complexity measure.

In the following we give a description of measures employed for this thesis, but details on their implementations will be provided in Chapter 4

1.2.1 Shannon Entropy

Entropy, also called *Shannon Entropy* due to the name of its inventor, can be defined as a measure of *unpredictability* of information content. It was developed as a measure to quantify the information content of a message. Considering, for example, a system under observation, its state can be modelled as a discrete random variable X with values $x \in \mathcal{X}$. The information content about the observation of x, with an outcome probability $P(x) = Pr\{X = x\}$, can be measured as $\frac{1}{\log_2 P(x)} = -\log_2 P(x)$, thus an improbable observation carries more information than one with high probability. The system entropy is given by averaging the information of each observation over all possible outcomes:

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log_2 P(x) \tag{1.3}$$

Notice that in case $P(x_i) = 0$ for some i, the value of $0\log(0)$ is taken to be 0.

As measure of unpredictability, $H(X)$ will be maximum for an X with uniform probability distribution, while it will assume value near to 0 when one of the possible values has probability $P(x_i) \approx 1$. Therefore it is expected to find high entropy values in systems with a chaotic regime and low values in ordered systems.

1.2.2 Disequilibrium and LMC complexity

As pointed out at the begin of this section, the notion of *complexity* refers to a system that is neither in an ordered regime nor in a chaotic one, but the entropy measure by itself does not really capture this concept. For example, a random generated sequence of symbols 0 and 1 will return an high entropy value, while it should have low complexity instead. A solution to this problem has been proposed in [26] with a measure called *LMC complexity*, by the name of its inventors. The idea is that a measure for complexity should reach its maximum in a region between order and chaos, therefore to identify this region it is necessary a measure that assumes higher values for more chaotic systems (entropy) and one that is higher for more ordered systems. The measure proposed for measuring order is called *Disequilibrium*, and it is defined as:

$$D(X) = \sum_{x \in \mathcal{X}} \left(P(x) - \frac{1}{|\mathcal{X}|} \right)^2 \tag{1.4}$$

On the contrary of entropy, Eq.1.4 assumes value 0 when X has a uniform probability distribution.

Once both measures has been described, *LMC complexity* can be simply defined as the product of entropy and disequilibrium:

$$LMC(X) = H(X) \cdot D(X) \tag{1.5}$$

1.2.3 Mutual Information

Another important *information-theoretic* measure is the *mutual information* between two random variables X and Y. This measure is based on entropy and quantifies the amount of information gained on a variable by observing the other. Mutual information is defined as follows:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{1.6}$$

where $H(X,Y)$ is the joint entropy of X and Y, defined on the basis of joint probability $P(x,y)$.

1.2.4 Dynamic Correlation

One last measure to quantify complexity, proposed by Beggs and Timme[4], is the *dynamic correlation*:

$$C_{ij} = \langle (i - \langle i \rangle)(j - \langle j \rangle) \rangle \tag{1.7}$$

This measure quantifies the correlation between states of two separated system components, the angled brackets indicate the time average while i and j are the values assumed by the respective components at the observation moment. The word *dynamic* suggests that to obtain high values of this measure each component must change its state in time, in fact the term $(i - \langle i \rangle)$ will assume a value of 0 if at each instant of time i is equal to its average. Moreover, to obtain a positive value the components must fluctuate in a coordinated manner, i.e. they must be both over or under their averages, otherwise their product will be negative. Applying this measure to two random generated sequences of symbols 0 and 1 it will return a correlation value of 0, because contributes of coordinated fluctuations will be nullified from uncoordinated ones.

The idea of this measure is to capture information flowing among the system and, coherently with the hypothesis made at the begin of this section, it is expected to be higher in a system that shows a complex behaviour. In their article Beggs and Timme applied dynamic correlation to the Ising model[10][12], and demonstrated how it reaches higher values and maximum distance when the system is working at the critical temperature.

Chapter 2

Automatic design

The most typical approach for developing an autonomous robot controller is based on manual design processes. This methodology relies entirely on the abilities and knowledge of researchers and engineers designing the robot. This practice suits perfectly for some kind of robotic applications, in particular when robots are operating in a very simple environment and/or are performing a repetitive task, and often permits to reach optimal solutions in terms of effectiveness and efficiency. An example is given by industrial robots where, even for really complicated tasks, the robot controller can be defined in terms of well-defined procedures, this possibility is given by the controlled environment in which they are operating.

When the environment and tasks complexity increases, the difficulty of designing a well performing robot controller will rapidly become unsustainable, limiting the degree of functional complexity that can be achieved, this because designers must anticipates every possible situation that the robot may encounter and, in a complex environment like the real world, they can be simply too much. A potential solution to the problem is to develop methods that allow robots to learn how performing complex tasks in an automatic way. The term *automatic design* includes all the methodologies known to design a system, in this case a robot controller, with a minimum human intervention, thus represents an appealing alternative to the manual design processes described above. In addition to the one presented, there are many motivations supporting the *automatic design* idea:

- this practice can potentially save many hours of human work and resources by moving the solution design process to computers;

- may find solutions not imagined by the designers by exploiting dynamical couplings between robots and the environment;
- may be used to investigate the emergence of some kind of behaviour;
- is very useful in robotic fields, such as *swarm robotics*[9], where traditional methodologies are not effective;

The increasingly interest in this field is demonstrated by the recently selection for founding by the *European Research Council* of project "*DEMIURGE: automatic design of robot swarms*"[6].

While there are many successful examples of automatic design applied to robot controllers, it is worth to mention also some works where automatic design techniques have been utilized to develop robot morphology. In [25] an automatic design process develops simultaneously the robot controller and his morphology by adding, modifying or removing basic building blocks. Another example of automatic design applied to hardware, but not to a robot, is reported in [20] where researchers from NASA were able to crate, in an automated way, an antenna with better performance than manually designed ones.

2.1 Structure of an automatic design process

In [27], Matteini provides an interesting overview of the most used automatic robot design techniques, in this section we described an automatic design process by following the structure of his work.

The elements necessary to apply an automatic design process can be summarized in:

- a configurable or evolvable controller;
- a genotype-phenotype mapping to transfer controller information to the physical robot;
- a fitness function for evaluating robots;
- a search algorithm to find a good solution for the specified controller;

2.1.1 Evolvable controller

The decision of the controller to use is a choice of great impact on the whole design process, there is in fact a great variety of controller typologies that can be summarized in:

- *Programs (executable code)*: This type of controller requires a special set of instructions usually defined for a specific experiment. In Busch et al. [11] researchers programmed different morphologies of walking robots using a sequence of instructions as genotype, and applied *genetic algorithms* in order to obtain better performing controllers.

- *Arificial Neural Networks (ANNs)*: ANNs are a family of models inspired by biological neural networks, such as the brain, and are widely used in the field of *Artificial Intelligence* for many purposes. An ANN is defined by the number of neurons composing the network, the connections topology, the weight of connections between neurons and the activation threshold value of each neuron. An example of ANN controlling a robot can be found in [17] where a *Kephera* is trained to perform *obstacle avoidance*, and the evolution process has been carried out entirely on a real robot.

- *Finite State Automata (FSA)*: An example of FSA used as robot controller is reported in Francesca et al. [18], where the performances of an automatic designed FSA controller are compared, on different tasks, with the ones obtained by an ANN controller and two human designed controllers. In that experiment, the genotype was composed of two different modules called *behaviours*, representing activities the robot could perform, and *transitions* that were criterion to regulate the change of behaviour in response to events.

- *Boolean Networks (BNs)*: BNs are a model of *genetic regulatory network* (GRN) that have been adopted in robotics only recently. This is the controller chosen for the experiments presented in this work, for this reason a deeper explanation of the model will be given in Chapter 3 .

2.1.2 Genotype-phenotype mapping

The word *genotype* refers to the form chosen to represent the controller, usually only the variables are taken in consideration. Considering for example an ANN whose topology is kept fixed during all the design process, the genotype will be composed only of threshold values and connection weights. Once defined the *genotype* it is necessary to map it with the system, in terms of the previous example this means building an ANN with all its properties coded in the *genotype*, and specifying how it interacts with the robot sensors and actuators. The result obtained by coupling genotype with system is the *phenotype*, and may refer to both, behaviour and morphology.

2.1.3 Fitness function

The fitness function is responsible for distinguishing well performing individuals from the others and, even more important, it can be considered as the task objective, therefore it is what specify the task itself. For this reason, defining the fitness function is the phase with the major human intervention required, and thus with the higher probability of introducing biases in the automated process.

A well-defined fitness function should be as *task-independent* as possible but often this means slowing down the design process. This is particularly true in case of complex tasks where a too general function could not lead to the desired behaviour. In Nelson et al. [28] fitness functions has been categorized on the base of the degree of *a priori knowledge* that they introduce into solutions.

2.1.4 Search algorithm

The search algorithm objective is to search for a genotype able to produce a phenotype that better fits the desired task, this means a research in the genotype space that can be both, finite or infinite depending on the controller chosen. Notice that in the case of finite search space it can be too large to be completely explored anyway.
Often, the search of a more performing individual can be reduced to an optimization problem, therefore some kind of heuristic algorithms are used to find a solution that, in this case, will be a *suboptimal* one. Probably,

the most used algorithms in the field of automatic robot design are *stochastic descent*(SD) and *genetic algorithms*(GAs). As this last category is the one used for the experiments reported in the following chapters, it will be described in details during the next section.

2.2 Genetic algorithms

GAs are widespread employed because of their ability to solve a wide range of real world problems. This category of algorithms takes inspiration from the Darwinian ideas of natural selection and "*survival of the fittest*". In nature, individuals in a population compete with each other for resources such as food and water, moreover members of the same species compete to attract a mate. Those individuals which are most successful in surviving and attracting mates, i.e. the ones with an higher fitness, will have a larger number of offspring than poorly performing individuals, i.e. the ones with a lower fitness. This means that the individuals of the successive generation will have "*better*"genes and, with them, an higher chance to *fit* well in the environment. Sometimes a particular combination of genes gives origin to individuals that are better then their parents, moreover the *DNA* of an individual may suffer a mutation with positive consequences in its phenotype. These events bring the species to evolve, making them more and more well suited with their environment. GAs are based on the same natural principles, in fact they are characterised by the following common elements:

- *Initial population*: this is the initial ensemble of *genomes* (genotypes), each one represents a possible solution to the problem to solve. Each genome is evaluated through the fitness function described above and a score is assigned. The dimension of the initial population can vary: in general more individuals means more phenotypic variety, and consequently higher probabilities to find different problem solutions. The number of individuals in the successive generations can be kept fixed or can be modified during the evolution depending on the specific algorithm chosen, usually it is kept fixed.

- *Selection criteria*: once all the individuals of a population have been evaluated, some of them are chosen for "*reproduction*", giving them

the opportunity to transmit their genes to a new generation of individuals. A criteria for the selection of individuals chosen for reproduction is thus necessary, for example only the *fittest* individual of each generation can be selected, or the N ones with higher scores. Other techniques are based on assigning to each individual a probability value to being chosen based on the fitness value obtained. This criteria is typically referred as *roulette wheel selection.* The choice of the selection criteria can have an high impact on the whole evolution result, on the one hand a restrictive criteria can lead to a solution in a minor time but with an higher probability of being stuck in *local minima*, on the other hand a criteria that select too many bad individuals may be responsible for a non-convergent evolution.
A typical mechanism adopted to facilitate the evolution convergence is called *elitism* and, when applied, it lets the best individual ever found to pass in the successive generations and spreading it genes, this technique prevents the lost of a "*good*" genome caused by few children with bad performances.

- *Crossover operator*: taken two individuals chosen for reproduction, their genomes are cut into two parts at some random point. The genome "*stripes*" produced are then swapped to create a new individual that have parts of genome from both parents. This technique is called *single point crossover*; many variations of crossover exists, but all of them relies on the same principle.
The crossover idea is to take the best from two good individuals, ideally if a part of the genome identifies a desired characteristic in one individual and another part determines something good on a second one, the crossover should give birth to a child exposing traits from both. Practically this is unlikely to happen due to the complex dynamics between the components identified by the genome, but it is still useful to promote new solutions, especially when some distinct "*families*" emerge inside a population. This operator is not applied to all the individuals of a new generation, usually a probability p is defined.

- *Mutation operator*: crossover operators do not introduce novelties in the population, therefore if a particular gene value is not present in at least one individual it will never appear in new generations. For this

reason mutation operators are needed. Their implementations heavily depends on the genome type, for example in the case of boolean genomes a mutation can consist in a bit-flip, while for an ANN can be the variation in a connection weight or threshold value.
As for crossover, it is defined a probability for applying mutation. This probability may refer to a single gene or to an entire individual, in the second case the number of mutations per individual must be specified in some way.

Once specified the problem formulation and the elements described, the evolution can start. The algorithm will end on the basis of a termination criteria which can be the maximum number of generations to evolve, or a particular fitness value to reach, it can also not be defined at all letting the evolution run until manually interrupted.

Chapter 3

Boolean Networks

Boolean networks (BNs) have been introduced by Kauffman [22] as a genetic regulatory network (GRN) model. BNs raised a considerable interest in both, biology and complex systems communities, due to their capabilities to reproduce some aspects and mechanisms of living systems.

BNs are of great interest due to their unique attributes: they are characterized by the compactness and dynamics richness typical of GRNs combined with the simplicity of Boolean logic that make this model particularly easy to compute. A considerable part of the robotic landscape is heading to miniaturization, leading to the creation of robots with dimensions of few centimetres [35], or even smaller. For this kind of robots a small and easily computable software controller is crucial, however BNs remain a promising model for many other applications.

In this chapter we describe Boolean networks in terms of structure and dynamics, and we illustrate how they can be employed as robots controller. A major part of the information reported has been taken from [33], reworked and integrated with other material.

3.1 Structure

A BN can be seen as a directed graph of N nodes where each node can only assume Boolean values (*discrete-state*) and the whole system state evolves at precise intervals of time (*time-discrete*). Therefore each node is associated to a Boolean variable $x_i, i = 1, ..., N$, and a Boolean function $f_i(x_{i_1}, ..., x_{i_{K_i}})$, where K_i is the number of inputs of node i. The arguments of the Boolean

t			$t+1$		
x_1	x_2	x_3	x_1	x_2	x_3
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

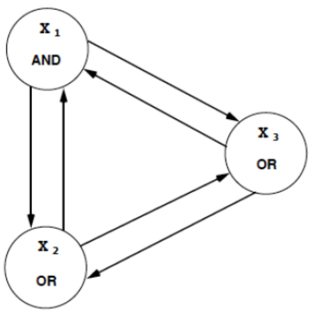


Figure 3.1: Example of a simple BN with deterministic update functions and synchronous dynamics

function f_i are the values of the nodes whose outgoing arcs are connected to node i. The set of all the Boolean functions composes the network *truth table*. At time $t, t \in \mathbb{N}$, the network state is completely defined by the N Boolean variable values: $s(t) \equiv (x_1(t), ..., x_N(t))$. Since each node can only assume values 0 or 1, the state space size is 2^N. The model used in this work is characterized by *synchronous* dynamics and *deterministic* functions, which is the most studied BN model (see fig.3.1). However many variants have been proposed in literature, some of these make use of *asynchronous* and *probabilistic* update rules [36].

3.2 Dynamics

Despite their simple definition, BNs can exhibit complex dynamics which can be studied by means of usual dynamical system methods such as *state (or phase) space*, *trajectories*, *attractors*, and *basins of attraction*. Every BN starts from an *initial state* which can be both, randomly generated or externally imposed, and evolves in time according to its update functions. In the case of *autonomous* BNs, where no external events can interfere with the dynamics, the network evolution will eventually ends in an *attractor* that, how has been said in Chapter 1, can be a *fixed point* or a *cycle* with length $\leq 2^N$ (see fig.3.2).

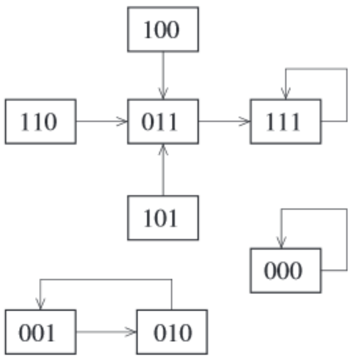


Figure 3.2: State space and possible trajectories of the network described in fig.3.1

A special category of BNs of particular interest are called *Random Boolean Networks (RBNs)*. RBNs are obtained by randomly generating both, the network morphology and truth table, by keeping fixed the value K. Each of the 2^K values composing the truth table is set by assigning a value of 1 with probability p, and 0 with probability $1-p$. Parameter p is called *bias*.

RBNs, and BNs in general, are indeed complex systems and, as described in Chapter 1, can show ordered or chaotic behaviours. It has been demonstrated that RBNs behaviour changes depending on the values of p and K selected [14][37]. In particular, an ordered behaviour has been observed for:

$$2p(1-p)K < 1 \tag{3.1}$$

while the network shown, on average, to be chaotic for:

$$2p(1-p)K > 1 \tag{3.2}$$

In the ordered regime attractors are small or even fixed point. Initially there may be a short transient where many states may change, but the network will quickly stabilize (fig.3.3(a)). A this regime, a perturbation such as a bit-flip on a node state value dies out quickly with a very little propagation. Networks at this regime are also very robust to initial conditions which means that similar states have an high probability to end in the same attractor. On the contrary, at the chaotic regime the attractors have a really long period, and successive states may appear as completely random with no recurrent patterns recognisable (fig.3.3(c)). Moreover chaotic networks are extremely sensitive to initial condition, thus slightly different states tend to diverge.

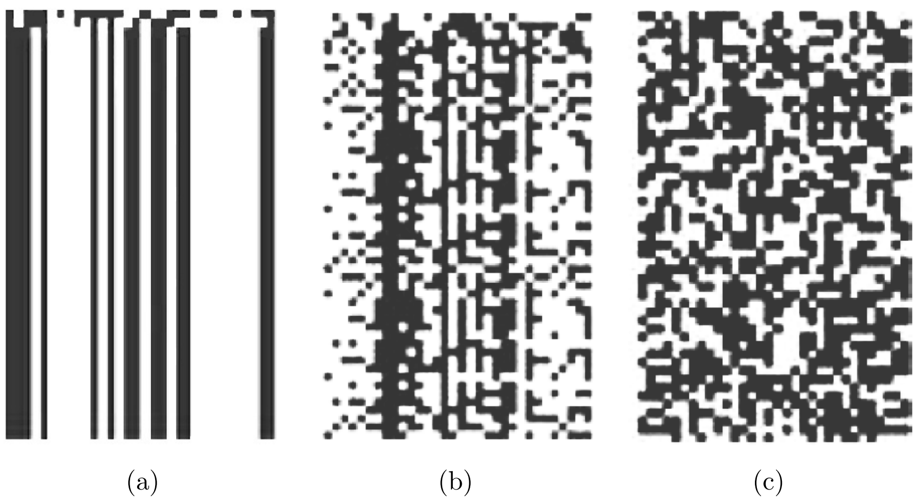


Figure 3.3: Trajectories of RBNs in different regimes: (a) ordered, (b) critical, (c) chaotic.

3.2.1 Criticality in RBNs

What has been observed in the field of CSS suggests that systems that exhibit a complex behaviour are usually in a particular dynamical condition that lies between order and chaos, taking advantages from the robustness of ordered regimes and the flexibility of chaotic ones. This conjecture, first introduced by Packard[29], Langton [24], and Crutchfield[13], is at the base of the so-called criticality hypothesis that defines this particular condition, i.e. operating between order and chaos, as *dynamical criticality* or also *"the edge of chaos"*. The concept of *critical state* has been introduced in the theory of phase transitions, that describes how modifying a suitable *control parameter* can bring a system to undergo strong qualitative changes in its macroscopic properties, usually defined in terms of an *order parameter*[38].

In the case of RBNs the critical regime is achieved when the equation $2p(1-p)K = 1$ is satisfied, therefore when the network lies between order and chaos. In a network at the critical regime, a perturbation can spread but its effects tend to remain isolated to a portion of the whole system. Moreover, as can be seen in figure 3.3(b), some recurrent patterns are visible in the network dynamics.

3.3 Boolean Network Robotics

RBNs, and BNs in general, have been mainly studied as isolated system. To being able to use BNs as robot controllers, it is first necessary defining a mapping between network nodes and robot sensors/actuators. A possible solution is proposed in Roli et al.[33] where the values of a set of nodes (BN input nodes) is imposed on the base of robot sensor readings, thus the state of these nodes is not determined by the network dynamics. In a similar way, a set of nodes (BN output nodes), whose states depend on the network dynamics, is chosen to pilot robot actuators. This model for coupling network nodes with robots sensors and actuators requires some tailored functions to encode continuous sensor values into boolean ones, and vice-versa (see fig.3.4).

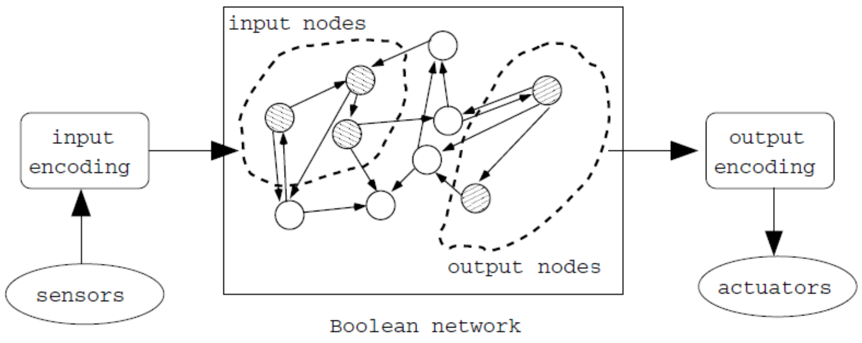


Figure 3.4: The *coupling* between BN and robot

An automatic design method (see Chapter 2) can be used to evolve the RBN controller and obtaining well performing robots. For the designing of RBNs employed in the various experiments conduced in this work, we decided to use genetic algorithms (see Section2.2); the specific evolution parameters will be reported for each task separately. The evolution process can be applied on the network truth table, by performing one or more *bit-flips* on its values, or even on the network topology by *rewiring* some of the node connections. The first of these two described methods is the one we applied in this work experiments since it is widely employed in literature with good results.

Despite RBNs has only recently being used as robot controllers, some experiments have been already performed. In Roli et al.[33] a RBN-controlled robot is trained to alternate phototaxis and anti-phototaxis behaviours on the base of an external command, represented by a sharp sound (a "*clap*"), given at a random time during the experiment.

A sequence learning task has been performed in [32], in this experiment a robot equipped with a floor colour sensor is trained to recognize a specific sequence of black and grey rectangles, separated by white ones, painted along a straight line. When a rectangle placed in the right sequence order is found, the robot must light its LEDs on, otherwise the LEDs must be turned off.

An application in the field of swarm robotics is provided by Vichi in his thesis [41], where a swarm of *E-pucks* are placed on a floor with a particular pattern painted on it. Each robot can only sense the ground below itself, and communications are limited to the neighbourhood. In case the recognised pattern is equal to a given one all the robots must light their LEDs on, otherwise all LEDs must be switched off.

Chapter 4

Complexity measures in robotics

In Chapter 1 we introduced *complexity measures* as a useful tool to quantify some complex systems properties such as *emergence*, *self-organization*, and even *complexity*. Given the fact that robots are themselves complex systems, an entire branch of research has been started to investigate how to make use of such measures to quantify properties of robots behaviour.

One of the main problem deriving from the employment of complex controllers, such as ANNs or BNs, is the impossibility to describe their dynamics by means of a set of clear and formal rules, therefore it is not possible to predict robot behaviours. Moreover, a robot is not an isolated entity and its dynamics, therefore its behaviour, is the result of a continuous interaction with the surrounding environment. Thus, if a robot performs well in a specific environment it can poorly behave in a slightly modified one, in addition its controller can exhibit a total different dynamics if extracted from its embodiment and analysed as an autonomous system. The only way we have to classify a robot behaviour is to make use of a *task-specific* objective function, or performing a visual inspection and assigning a qualitative and subjective evaluation, but neither of the two methodologies gives us any information on controller properties and on dynamics characterising it.

Advantages deriving from the application of complexity measures to automatic design of robot controllers can be summarized as follows:

- complexity measures can be adopted to provide quantitative information on robot behaviours, thus giving an evaluation that does not depend on subjective interpretations;

- they will be employed to increase our understanding of robot dynamics and how complex behaviours emerge from the interaction with the environment;
- the usage of information theory principles let us to abstract from the specific controller employed. For example, information collected on ANN-robots dynamics through complexity measures remain valid for BN-robots;
- can be used in automatic design as an high-level task-independent utility functions for the development of useful behaviours. This particular way to use information-theoretic measure gave origin to a novel methodology in evolutionary robotics named *information-driven evolution*[39];
- if the correlation between such measures and well performing robot behaviours will be demonstrated, they can be combined with classical task-dependent fitness functions to improve automatic design processes in terms of both, efficiency and effectiveness, by promoting internal complex network dynamics;

In the following sections we present some related works about this branch of research, then we provide an overview of the work done giving details on the specific implementation adopted for measures presented in Chapter 1.

4.1 Related works

Complexity measures have been already adopted with different purposes in robotics.

In Ay et al.[2] researchers used *predictive information*, evaluated as mutual information on successive sensor states, to describe the behaviour of a two wheels robot. The experiment consists in placing a robot, moved by a purely reactive controller, in a square arena surrounded by walls with some obstacles in it and observing its exploration capacity. The robot does not have any proximity sensors, and it can sense the environment only through the true velocity of its wheels (if the robot hits an obstacle, the wheels may get totally or partially blocked). The robot behaviour is modified by means of a control parameter c that determines the wheels target velocities at $t+1$ on the bases of real velocities sensed at time t. Results shown that the value of c corresponding to the maximum value of predictive information agrees almost exactly with the maximum explorative robot behaviour.

In [39] Sperati, Trianni and Nolfi simultaneously evolved a group of 3 *e-puck* robots by maximizing the mutual information of motor states between all possible robot pairs. Each robot is controlled by an ANN, and is equipped with a large variety of sensors (light, proximity, visual, and signal sensors). In addition to motors, robots have at their disposition LEDs and a sound emitting actuator that can be used for communication. Researchers define the behaviours obtained as "*structured, periodic, and coordinated*". Moreover, different evolutionary runs led to qualitative different behaviours. For example, one of the obtained behaviour consists in a sort of "dance" around a light bulb placed in the middle of the arena, each robot circles anticlockwise alternating four atomic movements: (i) forward motion on the circle, (ii) clockwise turn on the spot, (iii) backward motion on the circle, and (iv) anticlockwise turn on the spot. Moreover, to maintain the synchronisation robots learnt to make use of the sound signal emitter.

A deep analysis of the correlation between a set of complexity measures and animats fitness is reported in [16]. In this experiment, agents controlled by Markov networks must navigate and pass through a maze, composed of a series of walls having a single door, along the shortest possible path connecting the entrance with the exit. At every maze door, the agent is instructed about the position of the next door through a light signal, which is available only while the agent is standing in the doorway. The agent has only few time instants to perceive and memorizing the information about the next door before the light signal disappears. Six typologies of information processing and information integration measures have been calculated along the whole evolutionary process. Results showed a positive and highly significant correlation of all the measures proposed with fitness, but researchers pointed out how integrated information measures tend to better predict agent fitness when memory is involved.

One last work, strictly connected with the one performed in this thesis, is described in Roli et al. [34] where various analysis of BN-robots dynamics were performed. In the task presented, robots have to alternate phototaxis and anti-phototaxis as already described in Section 3.3. Among the analysis performed, researchers calculated entropy, disequilibrium and LMC complexity on robot trajectories collected during the entire evolution process. Results led to an important observation: LMC complexity of successful individuals increases during the training process, whilst remains almost constant for unsuccessful ones.

4.2 Application of complexity measures

All the results reported by the works presented in the previous section support the hypothesis that there is a strong correlation between complexity measures and robot behaviours.

The aim of this thesis is to investigate such correlation in RBN-controlled robots. To do so, we implemented 3 experiments with different characteristics and different complexity degrees. A detailed description of the work performed and of obtained results will be provided in the next chapters. This section reports how we applied complexity measures presented in Chapter 1 to our robotics experiments (i.e. to RBN trajectories).

For the calculation of complexity measures, we decided to discard values of nodes related to sensor groups because externally imposed and not directly associated with the network dynamics. On the other hand, it is also true that the motor commands are related with the sensor readings through the environment, so that the robot with its "*brain*"forms a feedback system which also includes the environment itself. We took this decision to maintain the analysis focus on the network internal dynamics, although an alternative analysis considering only sensor and actuator nodes has been made for the T-Maze task (see Chapter 10). Nodes connected to robot actuators have been kept instead, thus measures are calculated on trajectories composed of states of 16 or 15 bits depending on the number of sensors employed in each task.

Entropy and Disequilibrium

Both the entropy equation (eq. 1.3) and the disequilibrium one (eq. 1.4) contain the term $P(x)$, which is the probability for the discrete random variable X of assuming value x. To implement these measures on trajectories, the following assumptions have been made:

- x represents one of the state values that appear in the trajectory;

- $P(x)$ is obtained by dividing the number of times each x appears in the trajectory by the trajectory length;
- $|\mathcal{X}|$ is the number of different states that appear in the trajectory. We also tried to use the number of possible states that can be obtained (i.e. 2^{16}), but results obtained in the analysis did not highlight appreciable differences, therefore we decided to report only results obtained with the first definition of $|\mathcal{X}|$.

LMC complexity is the product result of measures above, for this reason does not need any further explanation.

Average Mutual Information

The main decision on how to apply mutual information to trajectories concerns the choice of the random variables to evaluate. One of the possible ways to apply this measure is considering single nodes as variables, then evaluating mutual information on every possible pair. The term "*average*"indicates that the final value is obtained by making the sum of all mutual informations, and diving by the number of different node couples. Thus average mutual information is defined as follows:

$$\langle I \rangle = \left(\frac{\sum_{i=0}^{N-2} \sum_{j=i+1}^{N-1} M_{ij}}{\binom{N}{2}} \right) \tag{4.1}$$

where N is the number of nodes composing a state, M_{ij} is the mutual information for nodes i and j as defined in eq.1.6, and $\binom{N}{2}$ is the number of possible node couples.

Dynamic Correlation

The dynamic correlation requires the definition of some kind of *system components*. In the case of RBNs, network nodes are the obvious choice. Equation 1.7 is then implemented as follows:

- i is the value of the *i-th* node at a precise instant of time, i.e. the value assumed by the node in one of the state composing the trajectory;
- $\langle i \rangle$ is calculated as sum of the values assumed by the *i-th* node in all the trajectory states divided by the trajectory length;

- $C_{ij} = \langle (i - \langle i \rangle)(j - \langle j \rangle) \rangle$ is obtained by the sum of $(i - \langle i \rangle)(j - \langle j \rangle)$ iterated over all the states and divided by the trajectory length;
- C_{ij} refers to a single pair of nodes. The final value of dynamic correlation is given by the sum of C_{ij} calculated for every possible couple of nodes with $i \neq j$. Notice that $C_{ij} = C_{ji}$ and is considered only one time.

Chapter 5

Phototaxis task

This chapter presents the first task we implemented for this thesis. We focus on giving an exhaustive description of the experimental set-up and of the results obtained in terms of fitness values.

5.1 Task definition

The *Phototaxis* task is one of the simplest task used in robotics. We chose this to study how complexity measures behaves in a task of minimal complexity. In this task the robot objective is to move towards a light source, reaching it, and remaining as near as possible.

Many variants of this experiment have been proposed in literature, for example in [19] the light source change its position once reached by the robot. In [40] phototaxis is part of a much more complex task where the robot has to discriminate between two scenarios, in the first it is able to reach the light source while in the second the path is blocked and must be signalled by emitting a sound. The latter task requires the emergence of *memory*, while the version we decided to implement for this work can be accomplished by a purely reactive robot like a *Braitenberg vehicle* [8].

This task, as all the tasks of this thesis, has been implemented using an open source modular multi-robot simulator called *ARGoS* [30]. The robot used is a *Foot-bot* [15], but despite the large variety of sensors available only light sensors have been used.

The environment is defined by an arena (*12m x 12m*) with a light source positioned in the centre (see figure 5.1). Each experiment run is composed

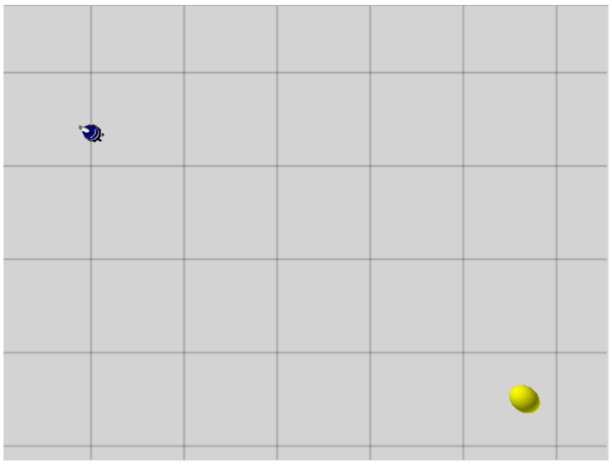


Figure 5.1: Portion of the arena used in phototaxis experiment

of 5 trials with a duration of 120 seconds, at the start of each trial the robot is placed 4.5 metres away from the light at angles going from $\frac{1}{12}\pi$ to $\frac{5}{12}\pi$, moreover the robot rotation is randomly chosen from a uniform distribution. During the whole process a Gaussian distributed noise, with standard deviation=0.02 and mean=0.0, is added to robot actuators.

The evolutionary process is carried out using GAlib simple algorithm [1], the evolution concerns a population of 20 individuals evolved over 100 generations. At each generation the best individual remains unchanged (*elitism*) while the others are selected through a *roulette wheel* mechanism. A bit-flip mutation with probability $p = 0.02$ per bit is applied to generate the new population, but no crossover operators are used. The fitness function is defined by minimizing an error value represented by the robot distance from the light when the trial terminates, averaged over the 5 trials.

The RBNs controlling the robot are composed of 20 nodes (N=20) and the function of each node depends on the value of 3 other nodes (k=3). Only two constraints have been applied during the networks generation: 1) the nodes belonging to one node boolean function are different, 2) no self-connections.

The evolution process concerns only the network truth table, while the networks physical topology is kept fixed. The genome is composed of a 20x8 bits matrix: one row for each node, one column for every possible input nodes combination (with k=3 there are 2^3 combinations). Notice that the rows corresponding to the BN input nodes are kept to maintaining the model as general as possible, but are never used. Each run of the experiment involves a RBN, the truth tables of the initial population of 20 individuals are automatically generated with $bias = 0.5$ (i.e. each bit of the genome has an equal probability to assume value 0 or 1). During the whole experiment a total of 30 RBNs have been trained.

Sensors and actuators mapping

Each robot is equipped with 24 light sensors equally distributed around the *Foot-bot* body and divided into four groups (see figure 5.2), each group corresponds to an RBN input node. When the sum of values read by a group of sensors reach an empirically defined threshold, the corresponding node state is imposed to 1, otherwise it is set to 0. This configuration makes possible to have more sensor groups active at the same time, but practically only two of them can be simultaneously turned on until the robot is really near to the light. Finally, each of the robot actuators is connected with an RBN output node, when the node state is 1 the relative wheel moves forward with a speed of 5 cm/s, when the node state is 0 the wheel is idle.

By convention, sensors have been mapped into the first network nodes $(x_0, .., x_3)$, while actuators have been assigned to the last two nodes (x_{18}, x_{19}). Notice that in an RBN all nodes are equals and there are no hierarchies between them, neither exists the concept of order. The terms "*first nodes*" and "*last nodes*" are used for ease of description and are the result of a vectorial representation of the network nodes. Any other choice of nodes may have been made without adding, or losing, any a priori information on the resulting robots behaviour.

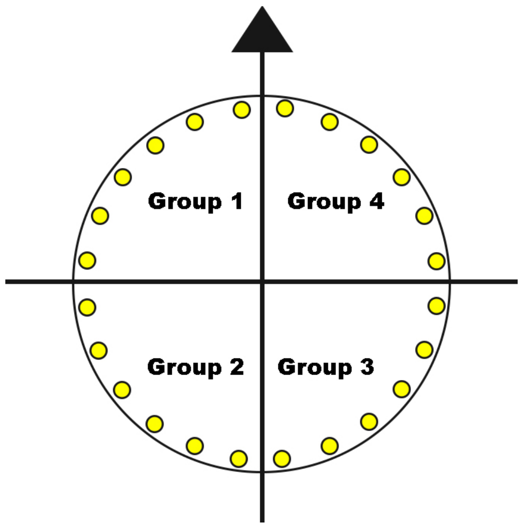


Figure 5.2: Light sensors configuration employed in phototaxis task, each yellow dot represents a light sensor.

5.2 Evolution results

Because of its extreme simplicity, all the trained networks were able to perform the task successfully.

In figure 5.3 the error value of the best individual of each generation is reported, averaged over the 30 experiment runs (30 different RBNs). As can be seen, the evolution process is very fast, and after only 20 generations almost all the best individuals are capable to finish the task with a distance from the light source lower than 0.5m. After generation 60 the improvement is on average of only 2cm in 40 generations, thus the evolution process can be considered terminated.

During the test phase all the evolved individuals demonstrated to be able to reproduce good results with regularity.

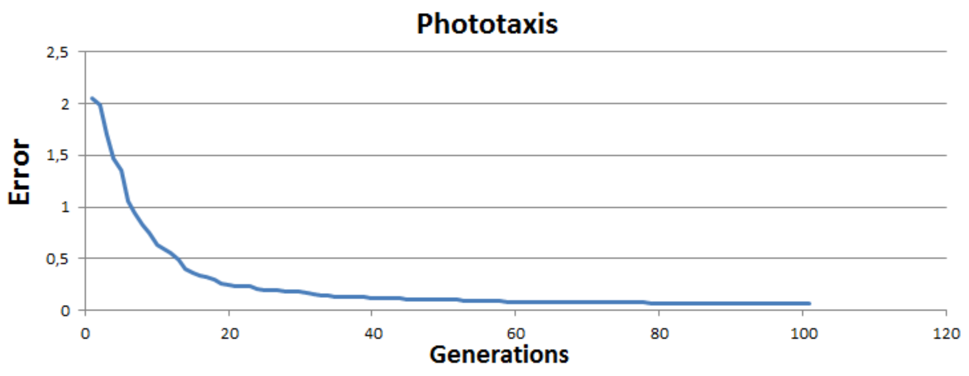


Figure 5.3: Error value of the best individual of each generation averaged over the 30 evolved networks.

Chapter 6

Phototaxis complexity measures

The analysis of robot dynamics through complexity measures requires, as first step, to collect the robots trajectories. We performed this trajectories collection operation simultaneously with the evolution process. Each time a BN-robot is evaluated by the genetic algorithm its trajectory is recorded, trajectories deriving from different trials of the same run are saved in separated files. A particular file naming convention has been adopted to distinguish trajectories of the same individual from the others. No filters have been applied during the trajectories recording (e.g. saving only the best individuals trajectories) to maintain the collected data as complete as possible. The entire evolution process lasts for 100 generations, each generation is composed of 20 individuals tested on 5 trials for a total of 10.000 trajectories for each RBN trained, and 300.000 for the whole experiment. Each trajectory is composed of a sequence of 1.200 states of 20 bits: 4 bits are used for sensors and were not considered in the analysis.

6.1 Analysis

The analysis has been conduced only on the best individual of each generation. On the trajectory registered for each trial *entropy*, *disequilibrium*, *LMC complexity*, *average mutual information*, and *dynamic correlation* are calculated, then an average over all the 5 trials is made. In this section we report the results obtained for each measure as the average over the 30

RBNs evolved in the experiment. In addition, also the *coefficient of variation* (CV) of such measures, defined as the ratio of standard deviation to the mean in percentage, is provided.

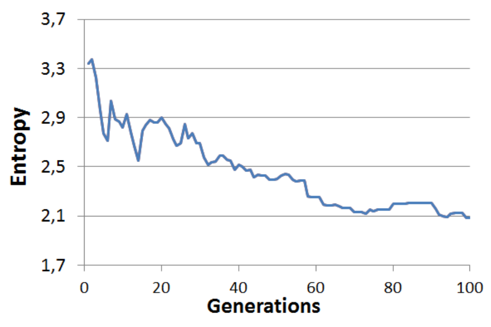
Following equation 3.2, with $k = 3$ and $p = 0.5$ the network should be initially chaotic, thus is expected to find high entropy and low disequilibrium values in individuals of first generations even if that equation has been derived for *autonomous* RBNs. The idea is that a network in a chaotic regime keep being chaotic also if few of its nodes value are imposed.

As can be seen in fig.6.1(a), individuals from first generations have on average an higher value of entropy which tends to decrease gradually with fitness (fig.5.3). On the contrary, the disequilibrium value (fig.6.1(c)) is low in an early phase of the evolution and increases during the process. This is a quite obvious result given the fact that entropy and disequilibrium actually measure two almost complementary phenomena. These trends of entropy and disequilibrium are an evidence that the adaptive process successfully achieved generalisation of the task, moving from a more chaotic regime to a more ordered one, and learning how to efficiently make use of information coming from the environment.

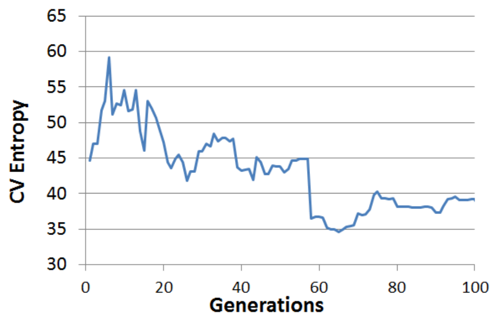
An interesting result comes from LMC complexity, as shown in fig.6.1(e) this measure increases during the training process until reaching a stable value around 0.8. Some works about criticality shown that critical systems are characterised by a peak of LMC complexity, the result obtained here, instead, only means that the dynamics of the trained RBNs moved toward a critical regime, and not that it has been reached, in fact the phototaxis task we proposed may be not sufficiently challenging to drive networks to their maximum complexity behaviour. A similar hypothesis, but applied to autonomous BNs, has been proposed in [5].

Another important result is highlighted from coefficients of variation of the measures mentioned above, all of them show a clear downward trend. This is a sign of convergence of the RBN-robots dynamics towards an optimal value for this task that does not depend on the specific networks structures.

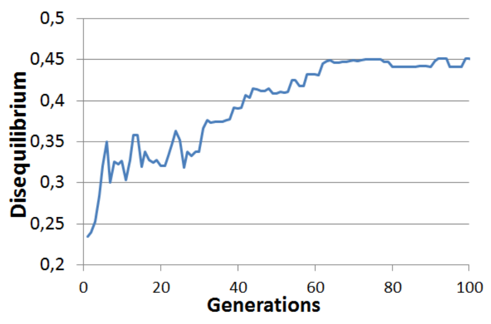
Also the average mutual information increases during generations (fig. 6.1(g)) but, contrarily to the other measures, its CV only slightly decreases during the evolution. Although, we still consider this result enough significative to suppose that a faint convergence process took place also for this measure.



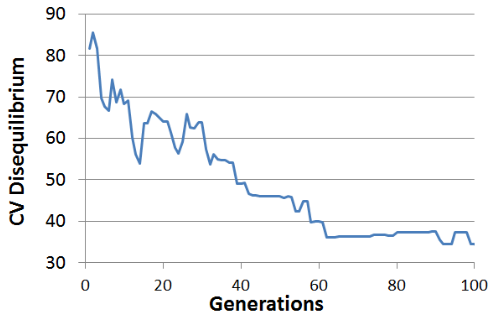
(a)



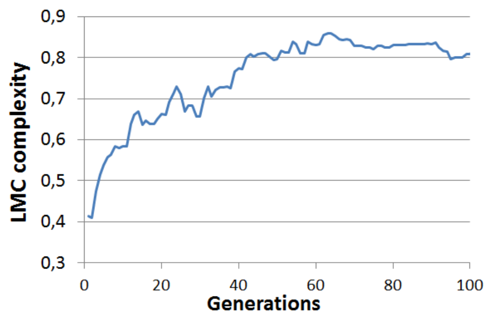
(b)



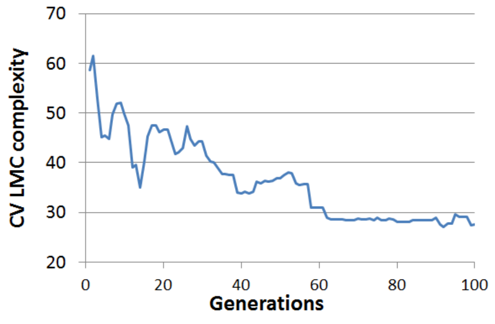
(c)



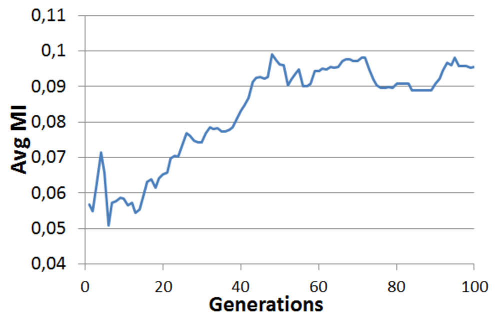
(d)



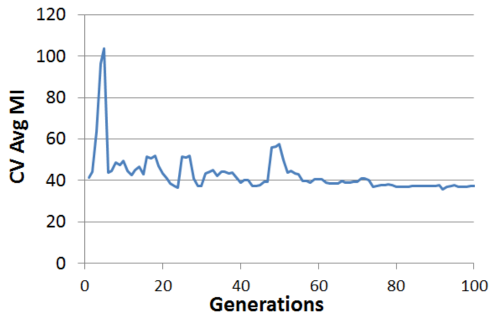
(e)



(f)



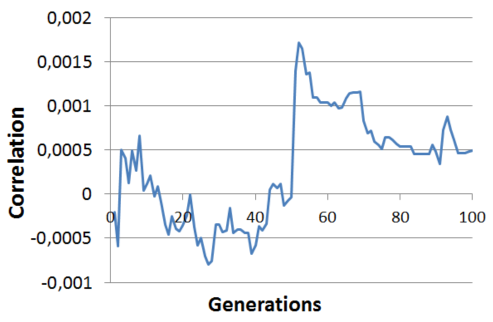
(g)



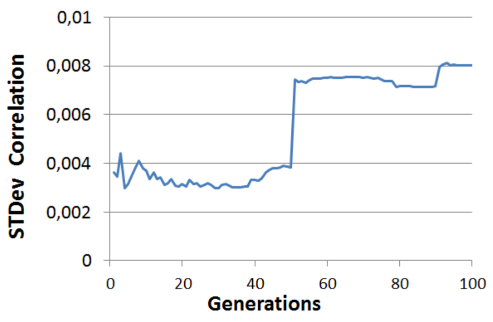
(h)

Figure 6.1: Average values of complexity measures and their coefficients of variation, both calculated considering the best individual of each generation in the 30 RBNs trained.

Finally, dynamic correlation 6.2(a) does not provide useful information about the robots dynamics, it fluctuates around 0 during the first half of the process and then has a positive peak that gradually decreases to its initial values. A major proof that dynamic correlation is not meaningful for this task is given by its standard deviation, in fact it dramatically increases in correspondence of the positive peak around generation 50, and values assumed are about one order of magnitude higher than the correlation ones, meaning that the result reported is representative of a very sparse values distribution. Notice that, in this case, we did not report CV because it tends to really high values when the mean tends to 0.



(a)



(b)

Figure 6.2: Average value and standard deviation of dynamic correlation, calculated considering the best individual of each generation in the 30 RBNs trained.

Chapter 7

Obstacle Avoidance task

This chapter describes the second task we implemented for this thesis. As for the previous one, we will describe the task and qualitative results obtained here, while complexity measures evaluated on the robots trajectories will be discussed in the next chapter.

7.1 Task definition

Obstacle avoidance is another classic task in robotics. Here, the robot has to learn how to move inside a circuit without hitting the walls.

This task shares an important aspect with the photaxis one: it does not require memory. Once again, a *Braitenberg vehicle* can accomplish this task in an efficiently way. A major difference between phototaxis and obstacle avoidance can be found, instead, in the interaction with the environment: in the former the robot has to locate and following a fixed reference point, resulting in a more constant and gradual sensors solicitation, in the latter it has to react once an obstacle is found, alternating phases of high sensors activity with idle ones. Moreover, the rapidity with which information coming from sensors starts to influence actuator actions is crucial.

The specific version of the task we decided to implement is inspired to the one proposed by Floreano and Mondada in [17]. The robot is put in an arena consisting in a sort of circular corridor of about *2m x 1.5m* (see fig.7.1). The walls are disposed in order to create some difficult situations and making the task more challenging, for example the upper-right corner constitutes a real trap for many robots.

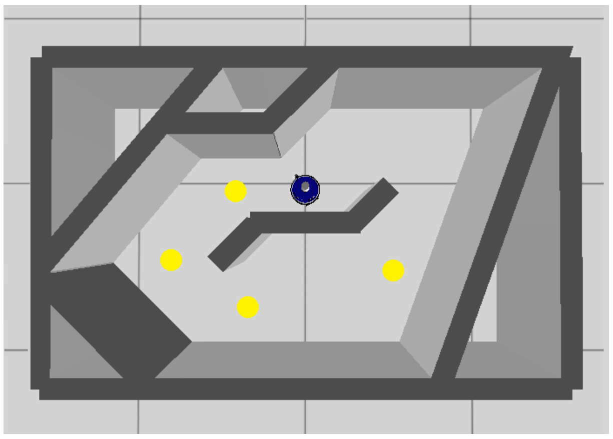


Figure 7.1: Arena employed in the obstacle avoidance task, the yellow dots show the others 4 initial positions where the robot can be placed.

The experiment is composed of 5 trials of 60 seconds each; the trial duration should be sufficient for a robot to perform about an half lap of the circuit. At each trial, the robot starting position is changed on the bases of 5 pre-defined locations shown in fig.5.1, moreover the robot rotation is randomly chosen from a uniform distribution. As for the previous task, a Gaussian distributed noise, with standard deviation=0.02 and mean=0.0, is added to robot actuators.

Genetic algorithm parameters are the same employed for photaxis with an only difference in the algorithm termination criteria, for this task the evolution is let run for 250 generations.

The fitness function proposed by Floreano and Mondada, thought for ANNs, is defined as follows:

$$\Phi = V(1 - \sqrt{|\Delta V|})(1 - i) \tag{7.1}$$

where V is the average speed of the wheels, $|\Delta V|$ is the absolute value of the wheels speed difference, and i is the highest value read by proximity sensors (the values range is $[0, 1]$). Equation 7.1 is evaluated at each *tick* of time, the sum of these values constitutes the final robot score. Thus defined, the fitness function value is maximum when the robot moves straight at maximum velocity, but every time an obstacle is detected the score decreases proportionally with the distance from it.

Due to the restrictions imposed by the boolean logic of RBNs, we modified the fitness function trying to keep the same concepts. Actuators can only be *active* or *idle*, consequently term V, that now refers to the average actuators activation, can only assume values 0, 0.5, or 1. The contribute given by the difference of wheels speed looses relevance because of the few values it can assume, the information it should give are fully provided by term V, for this reason term $(1 - \sqrt{|\Delta V|})$ has been eliminated. Finally, $(1 - i)$ is kept without any modifications, notice that sensor readings are continuous values even for a BN-controlled robot. The resulting fitness function is then defined as:

$$\Phi = V(1 - i) \tag{7.2}$$

As the robot is evaluated 10 times each second, the maximum theoretical score reachable in each trial is 600. That limit is obviously impossible to reach because, in the entire arena, does not exist a zone where the robot can move without running into a wall. The final robot score is then averaged over the 5 trials.

Sensors and actuators mapping

The sensory set-up is very similar to the one adopted in the phototaxis task with the only difference that infrared proximity sensors have been employed instead of light ones (see fig.7.2).

The same groups division has been maintained, but the activation threshold value has been considerably lowered. As said, reaction rapidity is crucial for this task, and a little threshold value has been introduced only to make the robot behaviour more robust in case of sensors noise (that has not been adopted).

7.2 Evolution results

Even if the evolved robot scores are on average good, results obtained show that not all the RBNs were able to produce well performing individuals. However this is a quite common result in the field of automatic design, and there is an high probability that the genetic algorithm remained stuck in a local maxima. In general, it is unlikely for network topologies having a

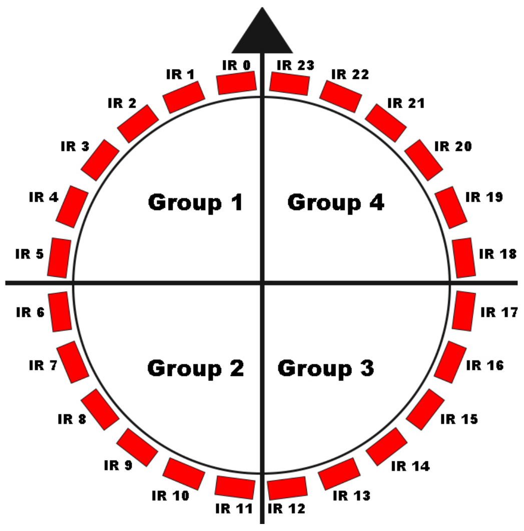


Figure 7.2: Proximity sensors configuration.

major influence on robots performance, the only exception is represented by pathological cases (e.g. one or more actuators completely disconnected from sensors).

Considering that the best evolved individual obtained a score of 410, and the worst achieved 222, a totally arbitrary threshold has been fixed at 300 to distinguish well performing individuals from the others. Only 3 RBNs did not produce individuals capable to reach such reference score. Figure 7.3 reports the score obtained by the best individual of each generation, averaged over the 30 experiment runs.

A visual inspection of best individuals confirmed that robot behaviours are qualitatively good, however even the best individual may sometimes hit a wall. This problem is mainly caused by the peculiar shape of the arena and the space for manoeuvre that a robot needs to change its direction (wheels can not go backwards).

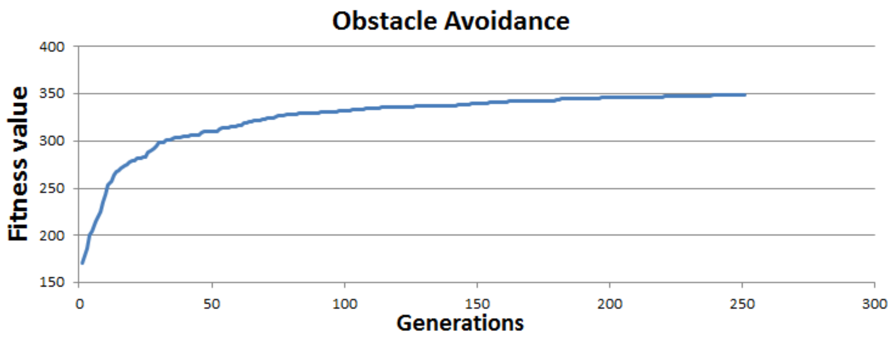


Figure 7.3: Fitness value of the best individual of each generation averaged over the 30 evolved networks.

Chapter 8

Obstacle Avoidance complexity measures

The same concepts applied to phototaxis trajectories have been applied also for this task, therefore measures are evaluated on trajectories with states of 16bits long. In total we recorded 750.000 trajectories, but only the ones relative to the best individual of each generation has been employed for complexity measures calculation, therefore data presented are obtained by the analysis of 37.500 trajectories.

8.1 Analysis

This section reports the results obtained on *entropy*, *disequilibrium*, *LMC complexity*, *average mutual information*, and *dynamic correlation*. Results will be discussed here, but a comparison with other tasks is demanded at Chapter11. Once again, the values reported are representative of the 30 trained RBNs.

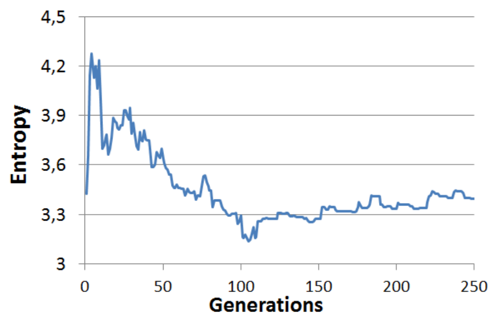
The first interesting result is given by the initial values of entropy and disequilibrium. As said, a more chaotic regime is expected in initial generations, but in a very first phase of the evolution process many RBNs manifest a more ordered regime. This phenomena can be easily explained by observing the fitness function definition (eq.7.2) and the arena structure (fig.7.1). Robots acquire fitness value only if they keep moving, for this reason at the first generations, when the evolution process may still not had any effects, robots that only move forward or spin around (i.e. with more ordered dynamics) will be able to achieve an higher fitness score and, consequently, being selected as the bests of their generation. Notice that RBNs that satisfy equation 3.2 have a chaotic behaviour only on average, thus it is perfectly admissible that some of the randomly generated individuals have more ordered behaviour than the others.

After some initial heavy oscillations, entropy and disequilibrium values follow the expected trend: the former decreases while the latter increases. Also in this case, trends show how evolved and well performing robots tend to have more ordered dynamics than the ones from earlier generations. LMC complexity obviously follows the same trend and increases during the evolution process upon reaching an almost stable value at 0.56.

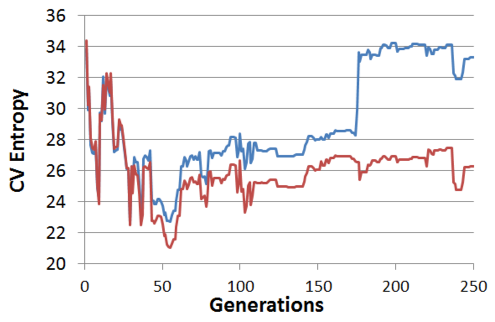
Less clear is the information coming from the standard deviations of such measures. For disequilibrium and LMC complexity, standard deviations tend to slightly increase, but their coefficients of variation show a clear downward trend(fig.8.1(d) and 8.1(f)). Entropy CV shows instead a clear downward trend until generation 175 when, suddenly, its value increases up to its initial value. A deeper inspection highlighted that this anomaly is caused by the values of a single RBN, by considering such network values as *non-representative* and eliminating them from the measures analysis, the peak in entropy CV completely disappears. Figure 8.1(b) reports CV values with and without considering values deriving from such network.

Average mutual information grows considerably during the first 100 generations, achieving an increment of about 45% with respect to its initial value, then starts to decrease until reaching the same values assumed at generation 15. Neither the CV gives any useful information, it heavily oscillates without decreasing, and does not provide any element for hypothesizing a convergence of average mutual information towards an optimal value. Given these results, we consider that average mutual information does not provide any kind of correlation with robots fitness for what concerns this task.

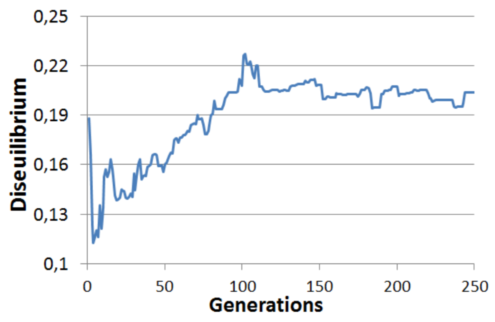
Finally, dynamic correlation confirms its low capacity to predict robot behaviours. Its mean oscillates without showing any clear trend, and its standard deviation values are three time higher than the mean ones, even if it decreases during the evolution.



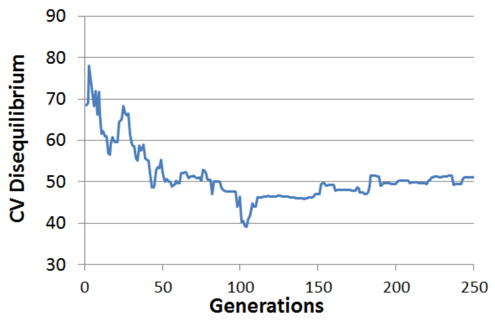
(a)



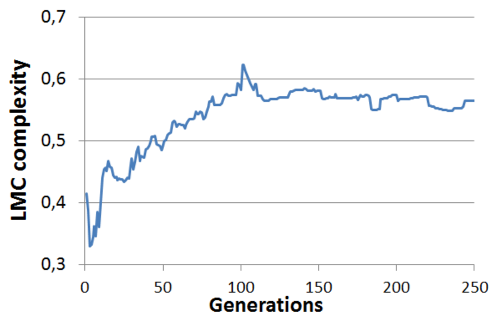
(b)



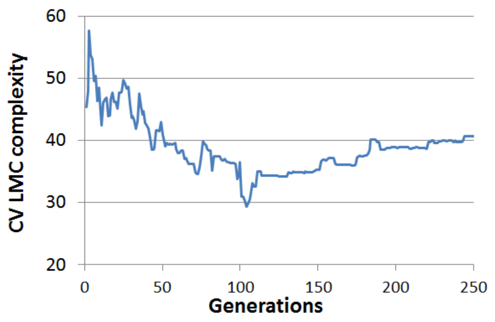
(c)



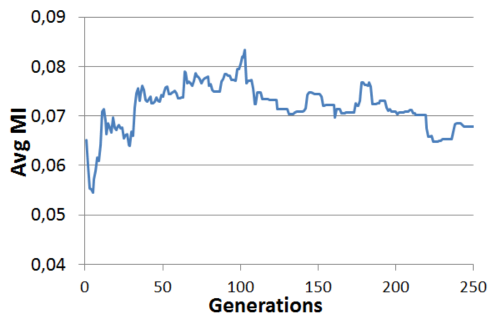
(d)



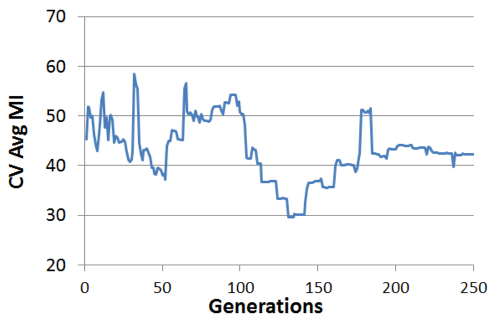
(e)



(f)

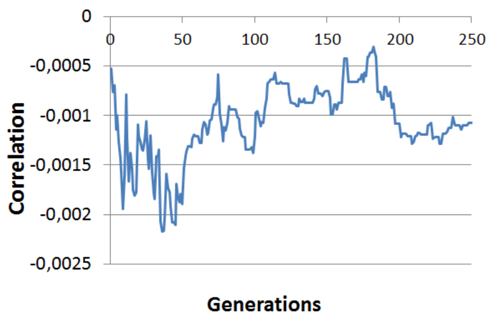


(g)

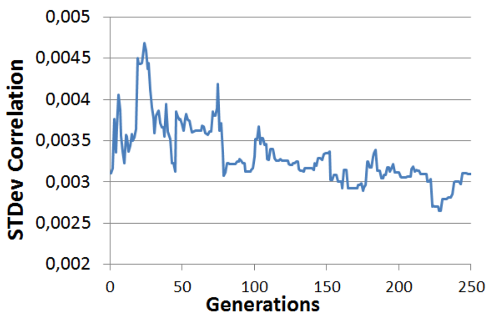


(h)

Figure 8.1: Average values of complexity measures and their CVs both calculated considering the best individual of each generation in the 30 RBNs trained. Figure 8.1(b) also reports the CV evaluated excluding the non-representative individual (red).



(a)



(b)

Figure 8.2: Average value and standard deviation of dynamic correlation calculated considering the best individual of each generation in the 30 RBNs trained.

Chapter 9

T-Maze task

The last task we realized to conduce this investigation is the *T-Maze* task, which is indeed more complicated and complex than the previous ones. In this chapter we describe the task and the variations applied, with respect to its original formulation, in order to improve robots performance.

9.1 Task definition

Many versions of this task has been proposed in literature, but we chose the one named *Simple T-Maze* described by Blynel and Floreano [7]. Despite its apparent simplicity, this task requires elements like exploration and memory which are typically desired also in real world applications. The main difference from the previous tasks is the memory requirement, in fact, contrarily to phototaxis and obstacle avoidance, the T-Maze task can't be successfully accomplished by a purely reactive robot. Moreover, this task includes features such as an environment that varies during the experiment, and a limited time to accomplish the task, which surely contribute to increase the overall task difficulty.

The T-Maze task is designed to test the capability of a robot to retain information over successive trials. The most important characteristic of this experiment is that the "*learning*" comes from *internal network dynamics*, and not from modifications of the network topology or truth table. It can be said that memory *emerges* from internal network dynamics. A previous successful example of memory emergence in a BN-controlled robot dynamic state can be found in [33].

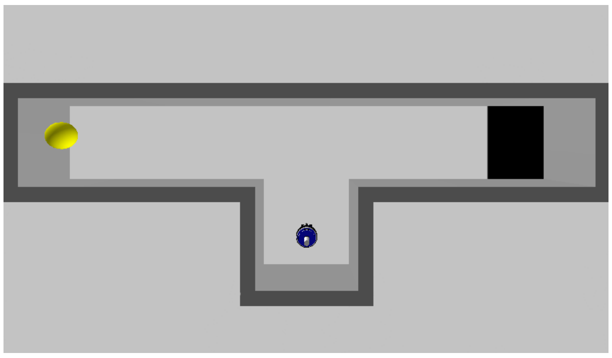


Figure 9.1: Arena employed in the T-Maze task, the black square represents the reward-zone while the yellow sphere is the light source employed as fixed reference point.

In the *Simple T-Maze* task a robot is initially positioned at the base of a *T-shaped* maze, and his objective is to find and stay on a reward-zone represented by a black square on the floor.

Each robot is tested for 4 epochs composed of 5 trials each, the reward-zone position depends on the current epoch, but is guaranteed to have the reward-zone 2 times on the left maze arm and 2 times on the right arm. At the beginning of each epoch, the network state is always initialized with the same values, this means that network state is kept during different trials, and information about the reward-zone position can be potentially stored in the network dynamic state.

The first trial (*exploration trial*) of each epoch is longer than the others and lasts enough to let the robot exploring all the maze, a well evolved robot should use this trial to find the reward-zone and memorizing its position. Successive trials (*memory trials*) are shorter and the robot can reach the reward-zone only if it turns in the correct direction at the T-junction, ideally the choice on where to turn should be based on information collected during the exploration trial (the reward-zone position do not change during the same epoch).

The *fitness function* used to evaluate the robots is simply the number of times a robot ends a trial on the reward-zone. The maximum fitness value a robot can obtain is 20 (4 epochs of 5 trials each). Notice that this fitness function is not designed for obtaining behaviours like fast moving robots or obstacle avoidance, thus a very little a priori knowledge is introduced into solutions.

In a previous work [3], the original task has been tested on BN-robots, results shown how the impossibility for robots to understand their position inside the maze heavily influences their behaviours robustness. With the objective of giving to robots the capability of recognizing left from right in a more reliable way, a light source has been added on the left arm of the maze (see fig.9.1). Notice that light position never changes during the experiment, its purpose is to provide a fixed reference point to identify the left maze arm, it basically works as the *North Star*. This choice can be legitimized by the fact that usage of an external source of information is quite common in real robot applications (e.g. some kind of markers positioned inside a warehouse), moreover adding this kind of information to the environment is usually a cheap operation.

Previous works made use of a light source in a T-Maze task [21][42], but in these experiments the light purpose was to indicate the right turning direction, which is a completely different use.

Successive tests, conduced on the modified task, highlighted an increment in robots performance in terms of robustness and number of individuals capable to reach the maximum score.

The experiment is carried out using GAlib steady-state algorithm with a population of 30 individuals per generation. At each generation, the best 3 individuals remain unchanged while for the remaining population single-point crossover, with a 0.1 probability, and bit-flip mutation, with 0.02 probability per bit, are applied. The evolution lasts for 400 generations, during the whole process the same noise level employed in the previous tasks is applied to robot actuators.

It is important to specify that the mechanism used by GAlib to implement elitism does not evaluate best individuals at each generation: if an individual hit a good score by luck it will not be tested again, and will be brought through generations even if its behaviour is generally bad. Moreover the fitness function for this task is bounded to a maximum value of 20: this means that if two individuals reach this limit value they are considered equivalent, therefore there is no way to distinguish which one is "*better*" (e.g. faster, manifests a more robust behaviour, ..).

To partially overcome this problem, the genetic algorithm output is composed of the best 5 individuals encountered throughout the evolution process, during the test phase these individuals are individually tested and the best one is picked for statistics.

Sensors and actuators mapping

For this task, robots were equipped with three different kind of sensors (see fig.9.2):

- ground sensor: this sensor, placed below the robot, detects the floor colour. When the robot steps on the reward-zone the corresponding node state is set to 1, in all the other cases is set to 0;
- proximity sensors: 3 groups of 6 sensors are placed on the front and sides of the robot. The corresponding nodes activation criteria are the same employed in the obstacle avoidance task;
- light sensors: all the 24 light sensors are used, the highest value read is compared to a threshold value representing the middle of the maze, if the value read is higher than the threshold means that the robot is near the light (left arm of the maze), if lower means that it is far (right arm);

9.2 Evolution results

The results obtained during the training phase are the following:

- 19 BNs reached the maximum fitness value of 20/20;
- 1 BN reached 19/20;
- 4 BNs reached 18/20;
- 2 BNs reached 16/20;
- 1 BN reached 15/20;
- 3 BNs reached 12/20;

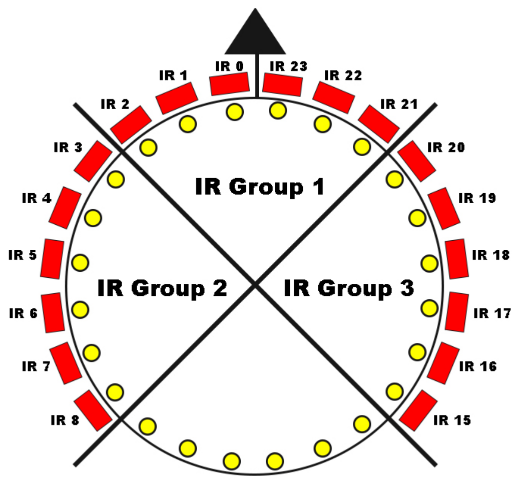


Figure 9.2: Proximity and light sensors configuration employed in the T-Maze task.

A considerable number of RBNs were able to produce individuals with high fitness values, and about 63% of them achieved the maximum value of 20/20. All evolved individuals manifest a preferred turning direction once arrived at the T-junction: this means that when no information about the reward-zone position has been collected (e.g during the exploration trial) the robot always turns on the same side. From now on, for ease of description, when an epoch has the reward-zone on robot preferred side it will be called "*concordant epoch*", when the reward-zone is placed on the opposite side it will be called "*discordant epoch*".

Results from training phase are obtained from one single run of the experiment, to verify the robustness of robots behaviour and the truthfulness of the obtained results, a test on the best individual of each RBN has been performed. The test consists in 30 runs of the experiment, results obtained are then reported as box-plots in figure 9.3. Notice that the *best* robot of each network is chosen after a preliminary analysis from the best 5 returned by the genetic algorithm.

Test results highlight how some individuals are capable to reproduce its scores on almost all the 30 runs, while others obtain results that may

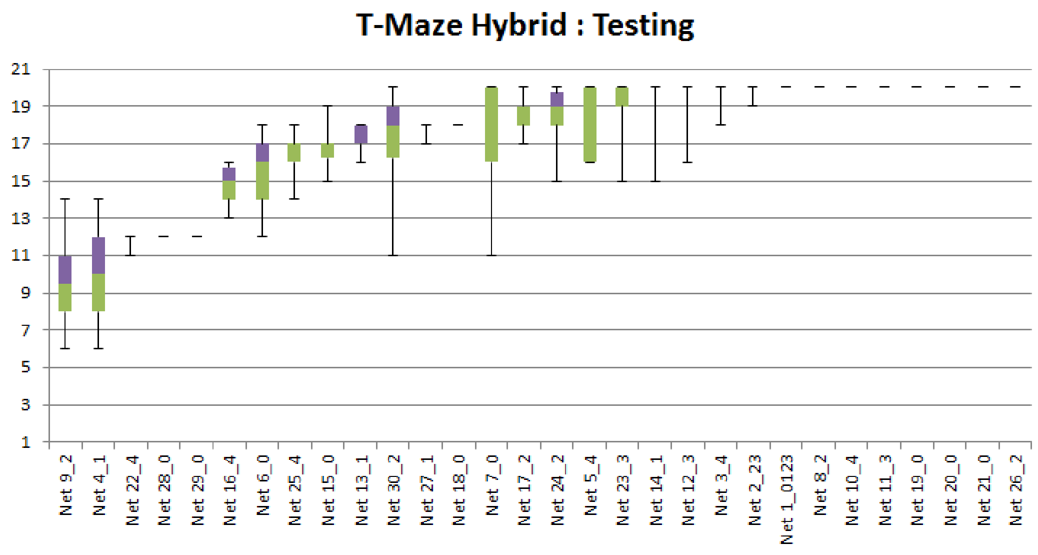


Figure 9.3: Results of the best robot of each RBN obtained during the test phase and ordered by mean. Each box-plot represents 30 runs of the experiment.

vary in a wide range. The formers developed a solid strategy that gives origin to robust behaviours, e.g. individuals with an highest score of 12/20 simply perform wall following, this simple strategy allow them to always reach the reward-zone during concordant epochs, and scoring one point in the exploration trial of discordant ones. The latter did not developed a precise strategy, and their behaviours are heavily influenced by actuators noise. These last individuals may be the result of GAlib elitism mechanism combined with the limit imposed to the fitness function maximum value.

Chapter 10

T-Maze complexity measures

Given the different formulation of this task, some changes in the trajectories structure have been necessary. Trajectories from different trials of the same epoch are saved in the same file, but separated by an identifier. At each epoch corresponds a trajectory file, therefore there are 120 files per generation, 48.000 per network trained, and a total of 1.440.000 for the whole experiment each one containing 5 trials. The sensors configuration employed for this task required 5 BN input nodes, thus complexity measures are calculated on trajectories with states of 15bits.

It is important to point out that T-Maze is a composite task:

- first trial differs from successive trials, the former is based on exploration while the latter relies on memory;
- each trial can be divided into two phases where, in the first, the robot has to find the reward-zone and, in the second, it has to maintain his position. Also in this case, the dynamics characterizing these phases can be different;
- all evolved robots manifest a preferred turning direction and, for this reason, the difference between concordant and discordant epochs constitutes another distinction element;

Given these differences in the task structure, we performed three different typologies of analysis to better and deeper investigate robot dynamics. In the following sections, a description of the analysis performed and a discussion of the results obtained are provided.

10.1 Evolution analysis

This analysis is the same performed for the other tasks: complexity measures are evaluated during the whole evolution process, and results reported are the average values over the best individuals of trained RBNs. Notice that we decided to include in the analysis only those networks that reached at least a score of 18/20 (24 in total).

As previously said, each trial is composed of a searching phase and a position maintaining phase which potentially require really different dynamics. Moreover, the duration of each phase depends on the robot under observation and on the specific path followed during the trial. With the objective of providing clearer information about robot dynamics, we decided to limit the measures evaluation at the first phase of each trial, therefore states collected after the reward-zone is found are discarded during the trajectories analysis. Some preliminary analysis were performed including also the position maintaining phase, results obtained shown how it badly influences information deriving from complexity measures. In addition, also the exploration trial has been discarded from this analysis, a direct comparison between exploration trial and memory trials will be provided in the next section.

Complexity measures are calculated on each trial separately, and an average over all the 4 epochs composing a run (i.e. 16 trials) is performed. We also performed an analysis variation where the trials of each epoch were considered as a unique trajectory, results obtained does not present any appreciable difference from the other ones, therefore only the first version of the analysis is reported in the following.

Results do not provide clear information as in the previous tasks, but some interesting considerations can be made anyway. The first observation concerns the measures CV values which, despite the oscillating measures means, show evident downward trends (see figures 10.1(b)10.1(d)10.1(h)) with the only exception of LMC complexity (fig.10.1(f)). Once again, results highlight a convergence of robot dynamics towards what can be considered an optimal value, conjectures about this phenomena will be provided in the next Chapter.

How can be seen in figures 10.1(a)10.1(c)10.1(e), trends of entropy, disequilibrium and LMC complexity are characterised by an irregularity around generation 100. After having inspected all individuals singularly, we discovered that such irregularity is correlated with two particular evolutionary path followed by some networks. First of all, the importance of a fitness score of 12/20 must be pointed out. This score is in fact the highest value reachable, with regularity, without memory, and may represent a strict division in a network evolution process. Some early generation individuals base their strategies on simple behaviours that require more ordered regimes, e.g turning always in the same direction without caring of sensor readings, or performing wall following. Thus, a slight increment in the fitness value after a score of 12/20 may mean a sharp change in the robot dynamics towards a more chaotic regime.

On the contrary, many individuals present a very chaotic dynamics during first generations, meaning that their behaviours are heavily influenced by actuators noise, and their results are consequences of lucky runs. The development of a more performing strategy requires the presence of some kind of pattern in the robots behaviour, thus a more ordered regime. The different timing with which these changes happen, and the sharp variations they cause in robot dynamics, give origin to the observable irregularity in complexity measure means.

Finally, we decided to not report dynamic correlation because, as in the previous tasks, it did not provide any useful information.

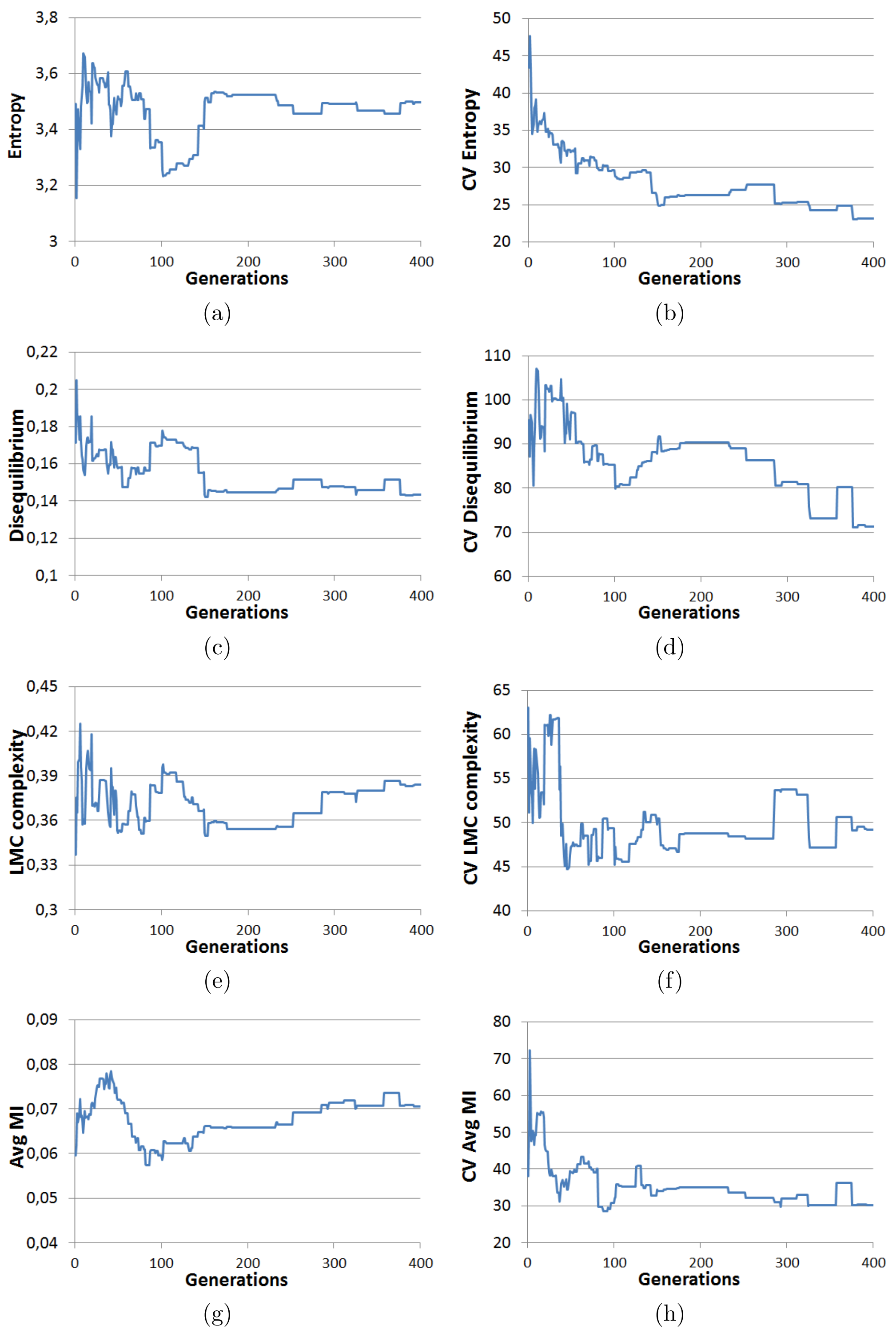


Figure 10.1: Average values of complexity measures and their coefficients of variation, both calculated considering the best individual of each generation in the 24 RBNs selected for the analysis.

10.2 Single run analysis

This kind of analysis aims to detect potential differences in robot dynamics that may be present between concordant and discordant epochs in both, well performing individuals and poor performing ones. To provide this distinction, a group composed of 10 individuals that achieve the best results during the test phase has been selected, and their behaviours have been visually inspected to determine their preferred turning direction. Then, a complete run (i.e. 4 epochs) for each individual has been performed and relative trajectories collected. Knowing the reward-zone position and the preferred turning direction of each robot, epochs have been divided into concordant and discordant ones. Finally, complexity measures have been calculated separately for these two groups.

Poorly performing individuals have been selected to provide a wide range of possible undesired behaviours. In this group there are individuals that achieved a fairly-high result but were not able to reproduce it during tests, individuals with low maximum results, and also individuals with a maximum score of 0 or 1.

Moreover, with the objective of giving more detailed information, we decided to separate exploration trials from memory trials providing two different analysis. As previously done, we considered only states collected before the reward-zone is found.

Graphs presented in the following report complexity measures values evaluated during concordant and discordant epochs. Each point on the graph corresponds to the value obtained by a robot, and is calculated as the average of 2 epochs (two epochs are concordant e two discordant). Notice that some of the *bad* individuals examined did not developed the concept of *preferred turning direction*, in this case, by convention, we treated them as if they prefer turning left, but no qualitative differences were visible in their behaviours among epochs.

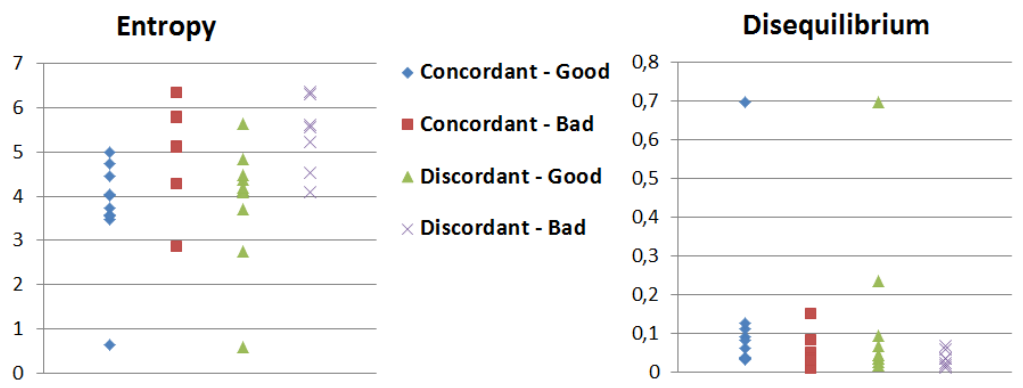
Analysis on the exploring trials shows how, for well performing individuals, there are apparently no differences in the robot dynamics between concordant and discordant epochs. We hypothesized that this result can be explained by the fact that no real changes of behaviour occur between the two types of epochs, i.e. robots always explore the maze, but in the case of a discordant epoch they just need more time. Notice that values slightly change when passing from a category of epoch to the other, but they do it in an uncoordinated manner remaining stable on average (i.e. for a robot increase while for another decrase).

Another result comes from a *strange* but well performing individual, which exhibits an incredibly ordered regime. This result, observed in entropy and disequilibrium (see fig.10.2(a)), has been confirmed by visually inspecting its behaviour. This particular robot moves straight until reaching the wall, then turns right of 90 degrees and proceeds to the end of the maze: if it does not find the reward-zone, it performs a *U turn* and follows the wall until the opposite side.

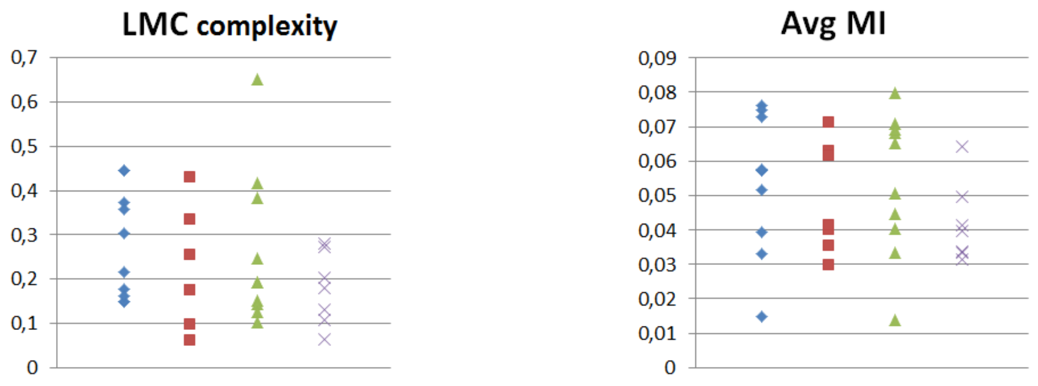
A great result comes from the analysis of memory trials. In this case, all the four complexity measures emphasize a difference between concordant and discordant epochs. We consider this result particularly interesting because shows how a qualitative behaviour change is reflected into the complexity measures proposed. Moreover, during concordant epochs, measure values are very similar to the ones observed in exploring trials, we hypothesized that, in those epochs, robots are still performing the same exploration behaviour. Confirming this hypothesis, entropy and disequilibrium of the *strange* individual maintain the same values assumed during exploration trials. On the contrary, during memory trials, entropy and disequilibrium of such individual changed drastically highlighting a behaviour change. We conjectured that discordant epochs show a more ordered regime because, in this case, networks have to adopt a sort of reactive-like behaviour that is usually associated to a lower richness of dynamics.

Also average mutual information highlights fairly well, with the exception of a single individual, the change of behaviour between concordant and discordant epochs, and it is also the only measure that always increases in this transition.

One last observation must be done on LMC complexity, it is on average higher for good individuals, and reaches its maximum during discordant epochs, which surely represent the most difficult part of the task.

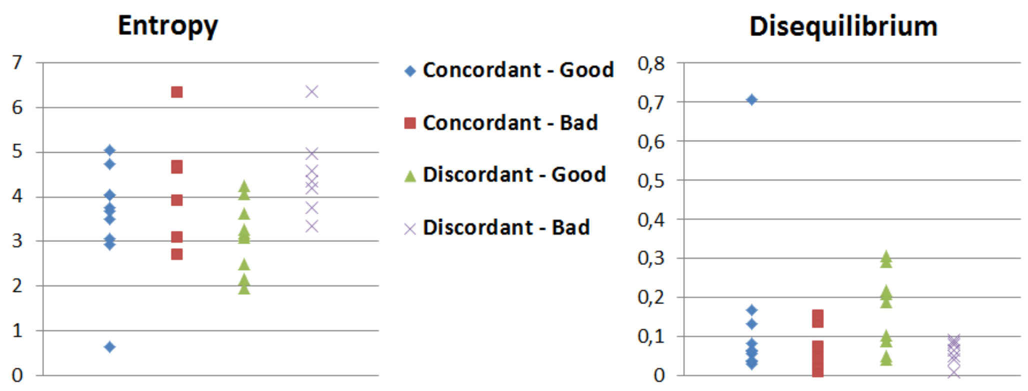


(a)

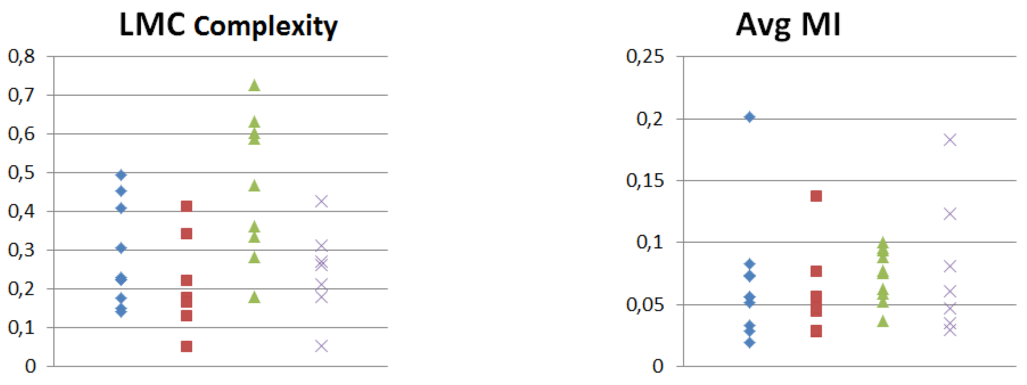


(b)

Figure 10.2: Measures obtained by well performing (*good*) and poorly performing (*bad*) individuals during exploration trials. Each point is representative of a single robot, and measures have been separated between concordant and discordant epochs.



(a)



(b)

Figure 10.3: Measures obtained during memory trials separated between concordant and discordant epochs.

10.3 Sliding window analysis

We performed one last type on analysis aimed to investigate what happen in the robot dynamics, at each instant of time, during the task execution. Contrarily to the previous analysis, we do not calculate measures on the entire robot trajectory, but we evaluate them through a *sliding window.* The idea is to look at the dynamics associated to a short time period, evaluating complexity measures on it, and then moving to the successive period. This process can be though as a window sliding through the trajectory.

Fundamental parameters for sliding window analysis are:

- *window width*: this is the number of states considered at each measures evaluation. Basically, measures are calculated on a sequence of "*mini*" trajectories whose lengths are determined by this parameter;

- *offset*: determines of how many states the window must be moved forward between evaluations. If smaller than window width, successive windows will partially overlap;

A good parameters selection is crucial, therefore we repeated this analysis with 2 different parameters sets:

- width=50 and offset=30;

- width=20 and offset=10;

Notice that we want windows to overlap because this situation let us to capture events that, without overlapping, would have been split between successive windows.

For this analysis, trajectories deriving from 7 individuals with different performance levels have been selected. For each measure and individual, a graph reporting the value of such measure during the 2 epochs (one concordant and one discordant) has been plotted. After having inspected a total of 70 graphs, no useful information have been discovered, therefore we decided to only report on of them as example of the work performed (see fig.10.4). Despite the poor results, this analysis provided us a deeper understanding of robot dynamics during the task highlighting how it is heavily influenced by sensor activities.

Finally, we want to mention that the same analysis has been performed considering only states relative to sensors and actuators. Also in this case we did not obtain any interesting result.

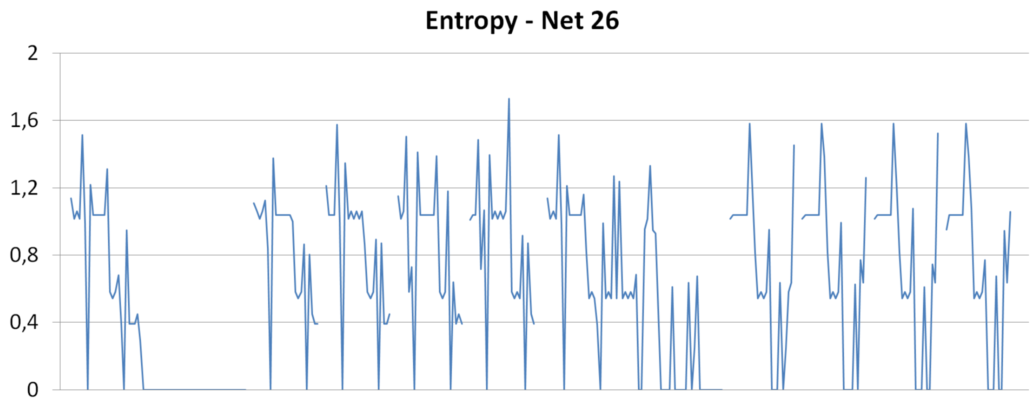


Figure 10.4: Example of entropy values evaluated during the run of a well performing robot with window width=50 and offset=30. This graph is representative of 2 epochs (10 trials): the first is concordant and the second discordant. Each spike represents a change in sensor readings.

Chapter 11

Comparison and considerations

In previous chapters, we presented the experiments made to investigate the correlation between complexity measures and robots fitness. Moreover, we discussed results obtained for each task separately. The objective of this chapter is to provide a comparison between results of all the three tasks proposed, discussing how they have been affected by tasks differences, and looking for phenomena that could not be discovered by observing tasks separately.

Given the extremely poor results of dynamic correlation, we decided to exclude such measure from our comparison, thus all the considerations we will make in the following refer only to entropy, disequilibrium, LMC complexity, and average mutual information.

The first observation we can do concerns the correlation between complexity measures proposed and robots fitness, assumed during the training process, among the three tasks. In general, all measures have an extremely significant correlation with fitness during the phototaxis task, they show clear trends in both, average values and coefficients of variation. In the obstacle avoidance task, all measures tend to have less clear CV trends, and the average mutual information does not show any correlation with fitness at all. Finally, in the T-Maze task only entropy, disequilibrium and average mutual information showed a faint correlation with fitness and fairly good CV trends, but more significant results emerged from a more detailed analysis. We conjectured that this gradual, and general, loss of clearness in measure graphs may be owed to the increasingly composite nature of the tasks, i.e. tasks require the alternating of different kind of behaviours. This conjecture is supported by the significant results obtained from separating T-Maze trials.

Another comparison can be made on the dynamical regimes adopted from robots to accomplish the different tasks proposed. As we would expect, robots performing phototaxis exhibit a much more ordered regime than the others. This result derives, with any probability, from the extreme task simplicity and from the low necessity to rapidly react to sensor solicitations. On the contrary, robots performing obstacle avoidance show more chaotic dynamics. The obtained results perfectly fit with the idea that a more chaotic system has an higher sensitivity to external stimuli, thus robots evolved towards such desired condition. Comparing results obtained from the T-Maze task is not trivial. On average, evolved individuals shown the most chaotic dynamics over the three tasks, but results deriving from *single run analisys* have shown that, during memory trials of discordant epochs, robots assume a dynamical regime at least as ordered as the one assumed during obstacle avoidance. This suggests that a robot may vary its dynamics if the task requires it.

We highlighted how, with only few exceptions, all the complexity measures coefficients of variation tend to decrease during the evolution. This result leads us to hypothesize the existence of a sort of "optimal value", towards which networks converge, that characterises the task itself (i.e. the behaviour necessary to successfully accomplish it). Notice that when we speak about "*convergence*", we are not referring to the convergence of a genetic algorithm towards an homogeneous population, but we are referring to a group of different networks that evolve independently from each other, and that tend to expose the same properties.

Some final dedicated observations can be made on measures adopted:

- *entropy* and *disequilibrium*: these measures always demonstrated a tight correlation with robots fitness, with a little exception in the T-Maze task explained in Section 10.1. For this reason, between the complexity measures proposed, we consider them to be the best ones to identify qualitative changes in robots behaviour. Despite this result, we do not consider these measures to be a viable option for a possible employment, in an automatic design process, as task-independent utility functions. The main reason is that they do not identify a desirable network dynamics but they simply describe it, thus their maximization (or minimization) will lead to an highly undesired result.

- *LMC complexity*: this measures gave useful information about robots fitness in phototaxis and obstacle avoidance, but behaved poorly in the T-Maze evolution analysis. This last result highlights how this measure is sensible to task composed of different kind of behaviours. Despite everything, we think that it is one of the most promising measures, between the ones proposed, to be employed combined with task-dependent fitness function. We hypothesize that results have been highly influenced from the analysis typology we performed. This hypothesis is supported by results deriving from single run analysis (see Section 10.2). Notice that an higher value of LMC complexity does not mean an higher intrinsic task complexity, and should not be used to perform comparisons between different tasks.

- *average mutual information*: despite this measure performed fairly bad on average, its values are more significant in the T-Maze task than in obstacle avoidance, which is a unexpected result. Moreover, it is the one that more clearly shown the difference between concordant and discordant epochs in the T-Maze single run analysis. A deeper inspection, based on data deriving from the three tasks, shows how this measure tends to be higher in ordered regimes and follows the disequilibrium graphs, but in the T-Maze task it starts to increase against disequilibrium exactly in the moment we hypothesized corresponding to the emergence of memory in the majority of individuals.

Conclusion

In this thesis we investigated the correlation between a group of complexity measures we selected and robots fitness. The motivation for such investigation derives from the application of complexity measures, designed in information theory, to quantify complex systems properties, and from the successive idea of employing such measures in robotics to quantify properties of robots behaviour.

The objective was to understand how these measures can be used to improve the analysis, by providing an evaluation criteria that does not depend on subjective interpretations of behaviours, and synthesis of robot control systems, by adopting such measures as an high-level task-independent utility functions to be employed alone, or in combination with classical task-dependent ones.

To do so, we applied an automatic design methodology (Chapter 2), based on genetic algorithms, in order to develop robot controllers for three different tasks implemented specifically for this thesis. Boolean networks (Chapter 3), more precisely random boolean networks, has been chosen as evolvable controller for our robots due to their extreme simplicity and richness of dynamics.

We analysed trajectories collected on three tasks of increasing complexity: *phototaxis*, *obstacle avoidance*, and *T-maze*. The last one, is a composite task that requires the emergence of memory in the internal network dynamics to be successfully accomplished.

While *dynamic correlation* immediately showed to be not employable for our purposes, *entropy* and *disequilibrium* showed instead a tight correlation with robots fitness during the whole training process in all the experiments conduced: they increase (or decrease) coherently with the average fitness of individuals examined.

Results obtained by *LMC complexity* during the T-maze task suggested that a too general analysis may not capture all the robot dynamics properties in case of composite tasks. For this reason we performed two additional analysis, the first (see Section 10.2) aimed to capture differences that may be present between different trials of the same experiment, the second (see Section 10.3) focused on studying the robot dynamics, at different time instants, during the task execution by means of a sliding window analysis. Single run analysis highlighted some important differences in robot dynamics between concordant epochs, where robots are free to follow their internal dynamics, and discordant epochs, where they must adopt another behaviour based on memory. On the contrary, sliding window analysis did not provide any significant results. On the basis of what has been observed, we think that the choice of the analysis detail level is crucial when applying complexity measures to robotics.

Average mutual information gave bad results in obstacle avoidance but shown an interesting possible correlation with the emergence of memory in the robot dynamics. This conjecture obviously need further investigations.

We believe that this thesis can provide insights for some future works. A first option concerns the employment of LMC complexity, or average mutual information, as part of the fitness function in an automatic design process of robot controllers. For example, in Chapter 10 we pointed out how the combination of GAlib elitism mechanism, jointed with the bounded fitness function of T-maze task, makes impossible to distinguish between two robots which reached a score of 20/20, resulting in some evolved robots with a poor behaviour robustness. A possible future work may investigate if the integration of a complexity measure into the fitness function improves the evolutionary process in terms of robots robustness, number of network able to reach the maximum score, or even number of generations needed to generate well-performing individuals.

Another future work will be focused on studying how tasks composed of heterogeneous phases affect robot dynamics. This kind of analysis may provide useful information on how employing complexity measures more effectively.

As final consideration, we want to point out that in this work we studied complexity measures on BN-controlled robot, but, given the fact that such measure are based on information theory, all the results obtained and considerations done may be extended to other types of robot control systems.

Ringraziamenti

Vorrei innanzitutto rigraziare i miei genitori, che mi hanno sostenuto in questa mia scelta di proseguire gli studi fornendomi il loro appoggio incondizionato.

Ringrazio la mia ragazza, Alessia, con cui ho condiviso le gioie e le ansie di questo percorso e che ha avuto l'arduo compito di "*farmi dare una mossa*".

Ringrazio i miei amici di sempre, che nonostante tutto sono ancora li e sono dei grandi.

Ringrazio Cornel, perchè in lui ho trovato un amico con cui condividere tante ore di lezione, tanti progetti e tante corse in stazione, ma con il quale mi sono fatto anche tante risate che hanno reso tutto ciò molto più piacevole.

Ringrazio i ragazzi di Twitch, che, con piccoli gesti e semplici parole, hanno fatto e continuano a fare veramente tanto per me.

Ringrazio, infine, il Professor Roli, che mi ha ispirato, seguito, ed aiutato in questa lunga avventura che è stata la tesi.

Bibliography

[1] Galib a c++ library of genetic algorithm components. `http://lancet.mit.edu/ga/`.

[2] N. Ay, N. Bertschinger, R. Der, F. Güttler, and E. Olbrich. Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63(3):329–339, 2008.

[3] M. Balducci. *Evolution of Boolean networks trained for T-Maze task.*

[4] J. M. Beggs and N. Timme. Being critical of criticality in the brain. *Frontiers in Physiology*, 3(163), 2012.

[5] S. Benedettini, M. Villani, A. Roli, R. Serra, M. Manfroni, A. Gagliardi, C. Pinciroli, and M. Birattari. Dynamical regimes and learning properties of evolved boolean networks. 99:111–123, 2013.

[6] M. Birattari. Demiurge: automatic design of robot swarms. `http://www.demiurge.be/`.

[7] J. Blynel and D. Floreano. Exploring the t-maze: Evolving learning-like robot behaviors using ctrnns. In *Applications of Evolutionary Computing*, pages 593–604. Springer Berlin Heidelberg, 2003.

[8] V. Braitenberg. *Vehicles: Experiments in synthetic psychology.* MIT press, 1986.

[9] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[10] S. G. BRUSH. History of the lenz-ising model. *Rev. Mod. Phys.*, 39:883–893, Oct 1967.

[11] J. Busch, J. Ziegler, C. Aue, A. Ross, D. Sawitzki, and W. Banzhaf. *Automatic generation of control programs for walking robots using genetic programming*, pages 258–267. Springer, 2002.

[12] B. A. Cipra. An introduction to the ising model. *Am. Math. Monthly*, 94(10):937–959, Dec. 1987.

[13] J. P. Crutchfield and K. Young. Computation at the onset of chaos. In *The Santa Fe Institute, Westview*, 1988.

[14] B. Derrida and Y. Pomeau. Random networks of automata: a simple annealed approximation. *EPL (Europhysics Letters)*, 1(2):45, 1986.

[15] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, et al. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. 20(4):60–71, 2013.

[16] J. A. Edlund, N. Chaumont, A. Hintze, C. Koch, G. Tononi, and C. Adami. Integrated information increases with fitness in the evolution of animats. 7(10), 2011.

[17] D. Floreano and F. Mondada. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior : From Animals to Animats 3: From Animals to Animats 3*, SAB94, pages 421–430. MIT Press, 1994.

[18] G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pinciroli, et al. Automode-chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 9(2-3):125–152, 2015.

[19] I. Harvey, E. Di Paolo, R. Wood, M. Quinn, and E. Tuci. Evolutionary robotics: A new scientific tool for studying cognition. *Artificial life*, 11(1-2):79–98, 2005.

[20] G. S. Hornby, A. Globus, D. S. Linden, and J. D. Lohn. Automated antenna design with evolutionary algorithms. In *AIAA Space*, pages 19–21, 2006.

[21] N. Jakobi. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adapt. Behav.*, pages 325–368.

[22] S. A. Kauffman. Metabolic stability and epigenesis in randomly connected nets. *Journal of Theoretical Biology*, pages 437–467, 1969.

[23] J. Ladyman, J. Lambert, and K. Wiesner. What is a complex system? *European Journal for Philosophy of Science*, 3(1):33–67, 2012.

[24] C. G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D*, 42:12 – 37, 1990.

[25] H. Lipson and J. P. Pollack. Automatic design and manufacture of robotic life forms. pages 260–267. Cambridge University Press, 2010.

[26] R. Lopez-Ruiz, H. Mancini, and X. Calbet. A Statistical Measure of Complexity. *ArXiv e-prints*, Sept. 2010.

[27] M. Matteini. *An overview on automatic design of robot controllers for complex tasks.* Seconda facoltà di Ingegneria, Università di Bologna.

[28] A. L. Nelson, G. J. Barlow, and L. Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345 – 370, 2009.

[29] N. H. Packard. *Adaptation toward the edge of chaos.* 1988.

[30] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, et al. Argos: a modular, multi-engine simulator for heterogeneous swarm robotics. pages 5027–5034, 2011.

[31] A. Roli. *An introduction to complex system science.* Lecture notes at Università di Bologna.

[32] A. Roli, M. Amaducci, L. Garattoni, C. Pinciroli, and M. Birattari. State space properties of boolean networks trained for sequence tasks. pages 235–240. Springer International Publishing, 2013.

[33] A. Roli, M. Manfroni, C. Pinciroli, and M. Birattari. On the design of boolean network robots. In *Applications of Evolutionary Computation*, volume 6624, pages 43–52. Springer Berlin Heidelberg, 2011.

[34] A. Roli, M. Villani, R. Serra, S. Benedettini, C. Pinciroli, and M. Birattari. Dynamical properties of artificially evolved boolean network robots. pages 45–57. Springer, 2015.

[35] M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3293–3298. IEEE, 2012.

[36] I. Shmulevich and E. Dougherty. *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks.* SIAM, 2010.

[37] I. Shmulevich and S. A. Kauffman. Activities and sensitivities in boolean network models. *Phys. Rev. Lett.*, 93, Jul 2004.

[38] R. V. Solé, S. C. Manrubia, B. Luque, J. Delgado, and J. Bascompte. Phase transitions and complex systems: Simple, nonlinear models capture complex systems at the edge of chaos. *Complexity*, 1(4):13–26, 1996.

[39] V. Sperati, V. Trianni, and S. Nolfi. Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, 2(2-4):73–95, 2008.

[40] E. Tuci, V. Trianni, and M. Dorigo. Feeling the flow of time through sensory-motor coordination, 2004.

[41] D. Vichi, A. Roli, D. I. M. Birattari, C. Pinciroli, and M. Dorigo. *On the design of a boolean-network robot swarm.* PhD thesis, Thesis, Università Di Bologna, 2012.

[42] T. Ziemke and M. Thieme. Neuromodulation of reactive sensorimotor mappings as a short-term memory mechanism in delayed response tasks. In *Adaptive Behavior*, 2002.