# Evolutionary Robotics

**J. A. Becerra**
*Universidade da Coruña, Spain*

**R. J. Duro**
*Universidade da Coruña, Spain*

## INTRODUCTION

**Evolutionary Robotics** is a field of **Autonomous Robotics** where the controllers that implement behaviours are obtained through some kind of Evolutionary Algorithm. The aim behind this technique is to obtain controllers minimizing human intervention. This is very interesting in order to achieve complex behaviours without introducing a "human bias". Sensors, body and actuators are usually different for a human being and for a robot, so it is reasonable to think that the best strategy obtained by the human designer is not necessarily the best one for the robot. This article will briefly describe Evolutionary Robotics and its advantages over other approaches to Autonomous Robotics as well as its problems and drawbacks.

## BACKGROUND

The firsts modern attempts to obtain a **robot** that could be called "autonomous", that is, with the ability of adapting to a non predefined environment and perform its tasks adequately, are from the late sixties and they basically tried to reproduce human reasoning in the robot. The reasoning process was divided into several steps (input data interpretation, environment modelling, planning and execution) that were performed sequentially. As time passed, robots were getting better thanks to better design and construction, more computational capabilities and improvements in the Artificial Intelligence techniques employed. But also some problems appeared and remained there: lack of reaction in real time, inability to handle dynamic environments and unmanaged complexity as desired behaviours become more complex.

In the late eighties a new approach, called **Behaviour Based Robotics**, was introduced. It emphasized the behaviour, no matter how it was obtained, as opposed to traditional (knowledge based) Autonomous Robotics where the emphasis was on modelling the knowledge needed to perform the behaviour. This new approach proposes a direct connection between sensors and actuators with no explicit environment modelling. Behaviour Based Robotics has proven to be very useful when implementing low level behaviours, but it has also shown problems when scaling to more complex behaviours. Phil Husbands (Phil Husbands et al., 1994) and Dave Cliff (Cliff et al., 1993a) have shown that it is not easy to design a system that connects sensors and actuators in order to achieve complex behaviours. Regardless of whether the system is monolithic (to design a complex system in just one step is never easy) or modular the design problem is difficult basically due to the fact that the possible interactions between modules grow exponentially. An additional problem is that human designed controllers for autonomous robots are not necessarily the best choice, sometimes they are simply not a good choice. A human designer cannot avoid perceiving the world with its own sensors and developing solutions for problems taking into account the perceptions and the actuations he / she can perform. Furthermore, humans tend to simplify and modularize problems and this is not always possible in complex environments.

Due to these drawbacks, in the early nineties some researchers started to use Evolutionary Algorithms in order to automatically obtain controllers for autonomous robots leading to a new robotics field: Evolutionary Robotics. Some examples of these research line are the papers by Irman Harvey (Harvey et al., 1993), Phil Husbands (Husbands et al, 1994), Dave Cliff (Cliff et al., 1993a) and Randall Beer and John Gallagher (Randall Beer and John Gallagher, 1992). The idea is very simple and very promising and, again, has shown it is very effective with simple behaviours. But, even if it solves some problems, it also has its own problems when dealing with complex behaviours. In the next section we will talk about those problems and, in general, about the main aspects to take into account when using evolution in Autonomous Robotics.

## EVOLUTIONARY ROBOTICS

The basis of Evolutionary Robotics is to use **evolutionary algorithms** to automatically obtain robot controllers. In order to do that, there are many decisions to be made. First of all, one must decide what to evolve (controllers, morphology, both?). Then, whatever is to be evolved has to be encoded in chromosomes. An evolutionary algorithm must be chosen. It has to be decided where and how to evaluate each individual, etc. These issues will be addressed in the following sections.

### What to Evolve

The first decision to be made is what to evolve. The most common choice is to evolve controllers for a given robot, but we can also evolve the morphology or both things together. If we choose to evolve only the controllers, we also have to decide how they will be implemented. The most usual choices are artificial neural networks, fuzzy logic systems and classifier systems.

Classifier systems are made up of rules (the classifier set). Each rule consists of a set of conditions and a message. If the conditions are accomplished, a message can produce an action on an actuator and is stored in a message list. Sensor values are also stored in this message list. Messages in the message list may change the state of conditions, leading to a different set of activated rules. There is an apportionment of credit system that changes the strength for each rule and a rule discovery system, where a genetic algorithm generates new rules using existing rules in the classifier set and their strength. An example of classifier systems is the work of Dorigo and Colombetti (Colombetti et al, 1996), (Dorigo and Colombetti, 1993, 1995, 1998).

Fuzzy logic has also been used to encode controllers. Possible sensed values and acting values are encoded into predefined fuzzy sets and the rules that relate both things can be evolved. Examples: (Cooper, 1995), (Hoffmann and Pfister, 1994), (Vicente Matellán et al, 1998).

Artificial neural networks are the most common way of implementing controllers in evolutionary robotics. On one hand, they are noise and failure tolerant and, on the other, they can be used as universal function approximators and can be easily integrated with an evolutionary algorithm to obtain a controller from scratch. Many researchers have used ANNs, just to mention some of them: (Beer and Gallagher, 1992), (Cliff et al, 1992), (Floreano and Mondada, 1998), (Harvey et al, 1993), (Kodjabachian and Meyer, 1995), (Lund and Hallam, 1996), (Nolfi et al, 1994) and (Santos and Duro, 1998).

### How to Encode What We are Evolving

When encoding a controller into the chromosome, the most obvious choice, and the most common one, is to make a direct encoding. That is, each controller parameter becomes a gene in the chromosome. For instance, if the controller is an ANN, each synaptic weight as well as the biases and other possible parameters that describe the ANN topology correspond to a gene, (Mataric and Cliff, 1996), (Miglino et al, 1995a). This can lead to very large chromosomes, as the chromosome size grows proportional to the square of the network size (in case of feedforward networks), increasing the dimensionality of the search space and making it more difficult to obtain a solution in reasonable time. Another problem is that the designer has to predefine the full topology (size, number of neurons, etc.) of the ANN, which is, in general, not obvious usually leading to a trial and error procedure. To address this problem, some researchers employ encoding schemes where the chromosome length may vary in time (Cliff et al., 1993b).

Another possibility is to encode elements that, following a set of rules, encode the development of the individual (Guillot and Meyer, 1997), (Angelo Cangelosi et al, 1994), (Kodjabachian and Meyer, 1998). Some authors even simultaneously evolve with this system both the controller and the morphology, but mostly for virtual organisms (Sims, 1994) or very simplified real robots.

### Where to Carry Out the Evolution Process

To determine how good an individual is, it is necessary to evaluate this individual in an environment during a given time interval. This **evaluation** has to be performed more than once in order to make the process indepen-