LEARNING FUZZY REACTIVE BEHAVIORS IN AUTONOMOUS ROBOTS *

VICENTE MATELLAN JOSE M. MOLINA JAVIER SANZ CAMINO FERNANDEZ

**Laboratorio de Agentes Inteligentes. Departamento de Informática. Avd. Butarque 15, Leganés (Madrid), SPAIN. Phone: +34 1 624 94 31, Fax: +34 1 624 94 30, E-mail: vmate@ia.uc3m.es. Universidad Carlos III de Madrid*

Abstract. *This paper is concerned with the learning of basic behaviors in autonomous robots. In this way, we present a method for the adaptation of basic reactive behaviors implemented as fuzzy controllers applying a genetic algorithm to the evolution of the fuzzy rule system. In this sense, we show our experiments in the evolution of control rules based on symbolic concepts represented as linguistic labels. The rules will be formulated in a fuzzy way and in order to test the rules obtained in each generation of the genetic algorithm a real robot has been used. The individual with the best performance is chosen to generate a new population: the elite strategy. All the new individuals were tested in the same real environment. In conclusion, the individuals of the last generation offer a set of rules that provides better performance than the ones designed by a non-expert designer.*

Key Words. Autonomous robots, fuzzy, genetic algorithm, learning.

1 Introduction

Artificial Life has been defined as "a scientific discipline that studies how behavior of agent emerges and becomes intelligent and adaptative" (**?**). Many experiments have been made using neural networks, finite state machines, etc. showing how these behaviors can emerge. However, artificial life community has not been very concerned about the emergence of symbolic concepts. On the other hand, the machine learning community has been mainly focussed on the improvement of the symbolic knowledge that a machine has about the world and how to operate in it. From this point of view, it has been forgotten how this symbolic knowledge can be obtained from raw data perceived in the real world. In this paper we present our work on the sub-symbolic learning of basic reactive behaviors implemented using a symbolic paradigm as a fuzzy controller (using IF-THEN rules and symbolic labels).

We have chosen fuzzy controllers as symbolic representation to study this emergence for many reasons. First, because fuzzy rules and fuzzy linguistic labels are close to the human way of expressing behavior rules. This means that we can easily evaluate the rules that have been obtained in a genetic way. Secondly, because fuzzy theories are susceptible to be shared among different agents, which is interesting in our further work. Thirdly, because fuzzy sets theory is a well-suited paradigm that has shown its efectiveness in many autonomous systems. And finally, fuzzy controllers are very flexible, which makes them adaptative easily.

This paper presents a genetic approach to the adaptation of fuzzy controllers in autonomous robots. In such controllers there are two main parts. The first one are linguistic labels and the second one are fuzzy rules. Linguistic labels can be viewed as low level ideas or symbolic concepts that can be interpreted in different ways. For instance, the concept *near* will not have the same physical meaning for a one meter diameter robot as for a 5 cm one. Rules express the relations among these concepts. In this paper we present a method for adapting these rules using genetic programming.

Our robot starts up with no information about the right rules to move around. From this situation, the robot is able to evolve to reach a set of rules that represent the highest adaptability grade to the sensors information. The only previous and fixed data are: the number of inputs (number of robot sensors), the partitions of the input domain (the range of the sensors), the number of outputs (number of robot motors) and its descriptions but it has no information about how do they relate to each other. This knowledge is obtained through the evolution of several generations.

The rest of this paper is organized as follows: The second section is concerned with the fuzzy description of the problem and the particularities of evolving a fuzzy controller. The third section deals with the description of our experiment. In the fourth section the result of these experiments is discussed and our future work is presented.

2 Problem Description

Fuzzy controllers have been widely used for controlling different autonomous robots. For instance, in (?) the LIFIA architecture is presented. It is a hierarchy of layers, each one working asynchronously with its own level of data abstraction, in which the navigation module consists of a reactive module based on fuzzy logic. Another classical example of an autonomous system controlled using fuzzy behaviors is (?). This system consists of a car which navigates autonomously thanks to a group of fuzzy rules sets.

In order to present the learning mechanism used to adapt the fuzzy control rules, we introduce in a first subsection the basic concepts of the fuzzy logic controllers. In a second subsection the strategy of design is presented and in the third one the adaptation process is described.

2.1 Fuzzy Controller

The first step in the design of a fuzzy controller shoud be to select adequate descriptions of the relevant inputs for the control, such as the *distance* to obstacles, "analogous" to those formulated by humans when they describe the perceived features. So, given the numerical distance d_i to an obstacle perceived by a sensor, D_i is defined as the range of all possible values of the computed distance d_i. To better cope with the intrinsic uncertainty that underlies the appearance of perceptual inputs (distorted after the acquisition process), the numerical values of the distances d_i can be mapped into qualitative symbolic labels through a fuzzification process (?), transforming the computed distances into linguistic variables.

A linguistic variable (?) is a variable whose values are sentences in a natural or artificial language, that is, a concatenation of atomic terms: labels (adjetives), hedges (modifiers such as very, much, slightly, etc.), the negation and makers (parentheses). The meaning of a linguistic variable is defined to be the fuzzy subset for which the value of the linguistic variable serves as a label. A fuzzy subset A of a universe of discourse U is characterized by a membership function $\mu : U \rightarrow [0,1]$ which associates with each element y of U, a number $\mu_A(y)$ which represents the degree of membership of y in A. The operation of fuzzification (application dependent) has the effect of transforming a nonfuzzy set or quantity into a fuzzy set. It is worth noting at this point, that the value of, for instance, the linguistic variable *distance* (a natural label such as *near*) represents a much less precise meaning than the numerical value of the inches to the obstacle.

Using these concepts, for each d_i, a linguistic variable Ld_i is introduced together with its set of values $ld_{i1}, ld_{i2}, \dots ld_{im_i}$, whose cardinality is m_i. Each term ld_{ij} in the set, labels a fuzzy subset in the universe of discourse D_i, with membership function $\mu_{ldij}(di)$. Values of the membership function of a label are related to the difficulty of attributing this label to a numerical value d_i obtained from the sensors of the robot. The fuzzyfication operation adopted, affecting the numerical values d_i, will result in their transformation into a fuzzy singleton (?) or fuzzy subset whose support is a single point in D_i, with membership function equal to one.

A Fuzzy Relational Algorithm (?) (FRA) will store the knowledge required to control the autonomous robot through a fuzzy reasoning process, based on the linguistic labels of the inputs. The FRA will be composed of a finite set of fuzzy conditional statements of the form **IF** Ld_i **IS** ld_{ij} **THEN** $LMOT_i$ **IS** lm_k, where their antecedent can be conjunctions and/or disjunctions about the linguistic variables Ld_i; and their fuzzy statements consequents about $LMOT$, the linguistic speed to apply to motor i, whose value set is $lm_1, lm_2, \dots lm_n$. The Mamdani implication (?) has been chosen to assign the meaning to these fuzzy conditional statements: the fuzzy subset of ordered pairs (d_i, s), with $d_i \in D_i$ and $m \in MOT$, of the Cartesian product of $(ld_{ij} \times lm_k)$ with degree of membership given by $min(\mu_{ld_{ij}}(d_i), \mu_{lm_k}(s))$. Where s is the defuzzification of $LMOT$ and MOT represents its numerical domain (universe of discourse of $LMOT$).

The final aspect that has to be considered is the inference strategy to manipulate the knowledge contained in the FRA. The compositional rule of inference (CRI) proposed by (?), (approximate extension of the familiar rule of *modus ponens*), serves as inference mechanism to obtain the fuzzy subset induced in MOT by a fuzzy statement of the form $(Ld_i\ is\ ld_{ir})$ through each conditional statement of the FRA. That is the fuzzy subset of MOT whose membership function is obtained after $max - min$ product of discretized versions of $\mu_{ld_{ir}}(d_i)$ and $\mu_{ld_{ij}}(d_i) \times ls_k(d_i, s)$, represented as (relational) matrices (?). As there can be several conditional statements forming the FRA, the meaning of $LMOT$ will be the intersection of the intermediate meanings resulting from each applic-

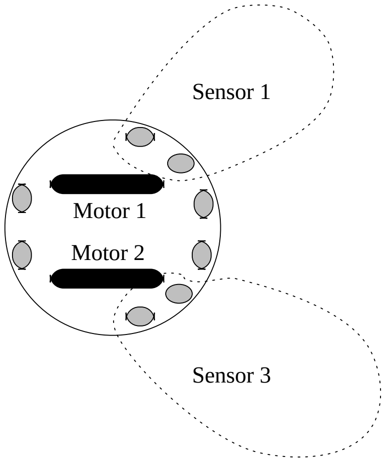

Figure 1 Robot for example 1.

ation of the CRI (*min* of all the induced consequent membership functions). Finally, the adopted defuzzification process on $LMOT$ will be a modified version of the Centre of Gravity procedure. This method treats the rules separately. Each rule produces a level of activation in the output labels λ_i. Let C_{ls_k} the numerical representatives of each label, ls_k (e.g. the centres of gravity). Then, the output is taken as a type of weighted sum: $sum = (\sum \lambda_i C_{ls_k})/(\sum \lambda_i)$.

2.2 Designing a FRA

The design of the rules to control an autonomous system using a FRA is not a trivial issue. Let us consider a system, similar to the robot we have used, with eight input signals (sensors) and two outputs (motors), codifying each one of the sensors inputs with only two linguistic labels (*near, far*), and each one of the outputs with five labels (*fast-advance, advance, stop, back, fast-back*); the number of possible fuzzy rules is 6,400. But if we had used a high level of granularity in our system, for instance, if every variable would had been codified with five linguistic labels, the number of rules would have been 9,765,625. Using a typical eight linguistic labels for variable, the number of rules would be in the range of billions (1,073,741,824).

Of course, only a few rules are needed (typically less than a hundred) to get a sophisticated behavior. The problem is how to choose the rules. Until now, the most commonly used method has been asking human experts the rules. But humans usually think in an antropomorphical way, which can cause some problems.

Let us consider a simple autonomous robot such as the one in Figure 1. This robot has only got two proximity sensors ($sensor1$ and $sensor2$), and two motors ($Motor1$ and $Motor2$). When a human is asked to write some rules that let the robot wander through the world without crashing, the obtained rules would be something like **IF** $sensor1$ **IS** *near* **THEN** $Motor1$ **IS** $fast$. This means that when an obstacle is perceived by the left sensor, the left motor speed is increased in order to go away from the obstacles by turning right. The symmetric rule will be written in the same way.

If this group of rules is tested on a robot, it would be proved that the robot begins to continuously increase its speed. The problem can be defined as a continuous increase of entropy. It can be easily corrected by writing the rules in an opposite way -**IF** $sensor1$ **IS** *near* **THEN** $M2$ **IS** *slow*- and adding a new rule to increase the speed when there are no obstacles. As a first goal, we want to test whether this problem would appear if the rules were obtained by genetic evolution.

Another problem with human rules is that they are designed in a theoretical world: the human mind. This means the rules and the linguistic labels have to be tuned many times till they reach an acceptable performance. This is caused by the differences among sensors due to the differences in the manufacturing process.

3 Adaptation of Fuzzy Behaviors

Some works can be bou about the use of genetic methods to learn plans for autonomous, for instance in (**?**) a genetic method to formulate new sets of low-level decision rules for robot movements and pushing techniques. Each rule checks to see that certain conditions are true (obstacles detected, goal position, etc.), and executes a number of corresponding operators (move forward, move backward, turn left or turn right). The genetic competition occurs among sets of behavior rules (plan) after testing the plans.

Some works have also been made in the genetic evolution of fuzzy rules. For instance, in (**?**) a genetic algorithm for the design of the fuzzy rules is presented to center a cart by applying a single force. Another example appears in (**?**) where the application of this method to three different physical systems is presented: a liquid-level system, a pH system and a satellite-rendezvous system. In each of these applications the genetically designed fuzzy controllers outperforms the human designed ones. However, few experiments have been made in the evolution of fuzzy linguistic labels. In (**?**) a genetic method that determines for a TSK fuzzy model each membership functions, the number of fuzzy rules and the rule consequent parameters;

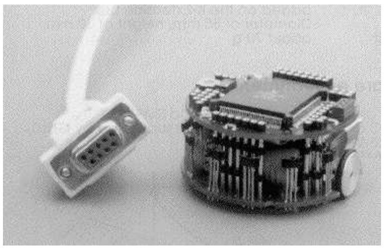

Figure 2 Khepera mini-robot.

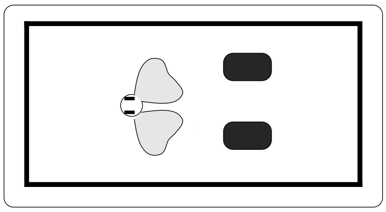

Figure 3 Experimental Enviroment.

this system is applied to the classical inverted pendulum control problem.

Finally, some works have been proposed in the evolution of fuzzy controllers applying to the control of autonomous vehicles. For instance, in (**?**) a general method for the evolution of rule-based fuzzy controllers in presented. Another methodology based in a hierarchical prioritized structure using a messy genetic algorithm is applied to the control of an autonomous vehicle, (**?**).

In the rest of this section we will briefly describe the robot used in our experiments, and then we will discuss our work: genetic evolution of a fuzzy controller for the control of the robot.

3.1 The Mini-Robot Khepera

The robot that has been used is the mini-robot KHEPERA (**?**), see Figure 2, which is a commercial mini-robot developed at LAMI (EPFL, Lausane Switzerland). This robot has a circular shape with a diameter of 5.5 cm., an height of 3 cm and a weight of 70 gr. It moves through two wheels and two small Teflon balls. The wheels are controlled by two motors that let the former move in both directions. The robot also has eight infrared proximity sensors.

The heart of the robot is the Motorola 68331 controller with 256 Kbytes of RAM and 512 Kbytes of ROM which manages all the input-output routines and can communicate via serial port with a host computer. It also has its own batteries which let it work autonomously. It is also possible for it to work attached to a workstation via the serial port. This lets the robot use the resources of the workstation.

We have preferred to use a real robot instead of a simulation for two reasons. First, because a perfect simulation of a simple robot as Khepera requires hard computations. For instance, the simulation of the sensors, taking care of the lab lightening, orientation of robot, etc; would require huge calculations. Second, even a very good simulation is unable to consider all the physical laws of a real robot such as inertia, friction, failures of the hardware, etc.

3.2 Adaptation of Fuzzy Rules

Our first goal was to test if the rules obtained by means of genetic evolution were able to control succesfully an autonomous robot. So, the robot was given some fixed concepts such as near, far, etc. for the sensors and slow and fast for the motors. The genetic program should be able to successfully combine these concepts. The robot was located in a simple environment. It consisted of a rectangular area with two obstacles situated as shown in figure 3. The walls were made of cardboard and the floor was the surface of a wooden table. This table was situated in an always artificially illuminated laboratory.

Each of the sensors of the robot returns an integer, whose range goes from 0 (no obstacle detected) to 1023 (an obstacle just in front of the sensor). The speed of both motors is also fixed by an integer. In this case, the range we have used goes from -10 (maximun speed backwards) to 10 (maximun speed forward). Though the robot has eight sensors, we have groupped some of them in two. So, the robot used in this experiment can be considered as equivalent to the one in Figure 1.

For this experiment we have chosen five fuzzy partitions of each one of the four variables ($sensor1$, $sensor2$, $motor1$ and $motor2$). The fuzzy partitions for the parameters have been chosen as indicated in Figure 4. In this way, the individual control strategies are two 5 x 5 tables, one for each motor. Each table relates the membership functions of both sensors with a motor one. The tables are coded as chromosomes with alleles {0, 1, 2, 3, 4, 5} corresponding to the membership functions *Far-Backward, Backward, Quiet, Forward, FarForward* and blank. The last one indicates that there is no relation between the two membership functions.

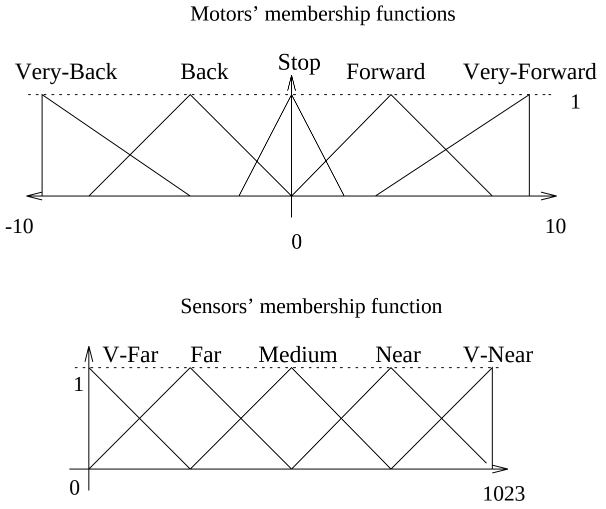

Figure 4 Fuzzy partions for sensors and motors

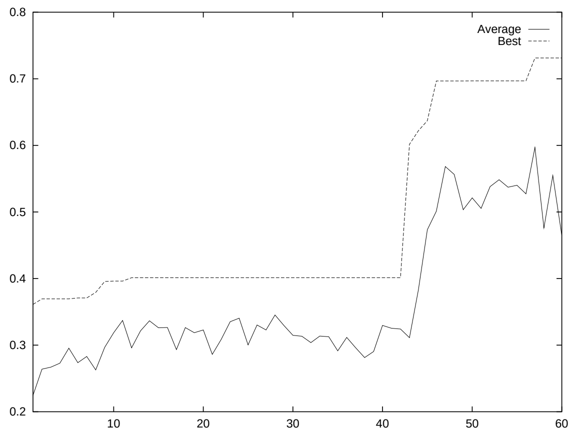

Figure 5 Average vs. Best *fitness.*

We consider both tables as a *genotype*. So, each individual has got a chromosome whose length is fifty. The alleles are the five fuzzy membership functions over the domain of the motors plus the blank code. The *phenotype* is the behavior that the fuzzy controller produces. The behavior is obtained applying the usual operations: fuzzification, max-composition and centroid defuzzification. Our fitness criterion, Θ, is function of three variables directly measured on the robot Khepera and another one on the genotype.

$$\Theta = \frac{V(1-\sqrt{D})(1-I)}{rules}$$

The variable I represents the normalized value of the sensor wich presents the highest level of activation:

$$I = \frac{sensor}{1023}$$

V is the rotation average speed of the two wheels:

$$V = \frac{average}{10}$$

and D is the normalized absolute value of the difference between the speed of the two wheels:

$$D = \frac{\mid v_1 - v_2 \mid}{20}$$

This makes function Θ be maximized by obstacle avoidance, straight direction, speed and few rules.

The evolutionary training was a standard genetic algorithm. It consists of 100 generations with a population of 100 individuals each. The mutation operator changes a fuzzy code either up or down a level, or to the blank code. The crossover operator is the standard two-point crossover. The life time of each individual has been set to 30 seconds. We have used an *elite* strategy, meaning that the best individual is automatically promoted to the next generation.

In Figure 5 we can see the evolution of the average fitness of each generation and also the fitness of the best individual of each generation in an experiment using the following configuration:

Obstacles	1
Mutation	0.2
Crossover	0.2

Figure 5 lets us evaluate the learning process. We have appreciated that, although we allow 100 generations, around the 60th generation in most of the experiments, the population has learned to avoid obstacles and that most of the individuals in the latest generations use less than 40 rules.

Figure 6 shows the same using a different configuration:

Obstacles	1
Mutation	0.4
Crossover	0.4

This figure shows that if both the probabilities of mutation and crossover rise, then the speed of the learning process increases. We have also made experiments increasing only one of these factors and varing the number of individuals in each generation and the number of generations.

In figure 7 whe show the contribution of the three different parts of the function Θ versus the total fitness. Both values (parts and total) showed in the figure correspond to the average of the individuals of each population. We have also tested the consequences of using a factor to improve one

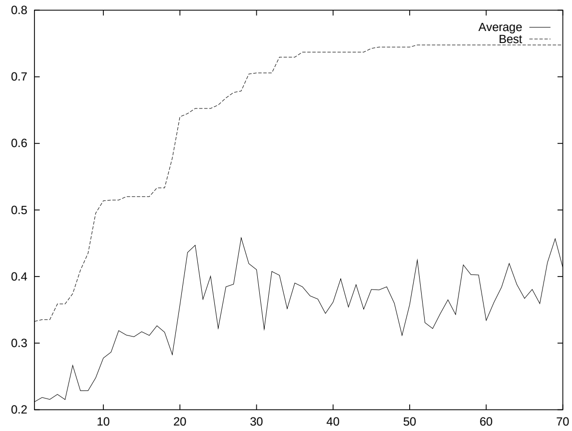

Figure 6 Average vs. Best *fitness.*

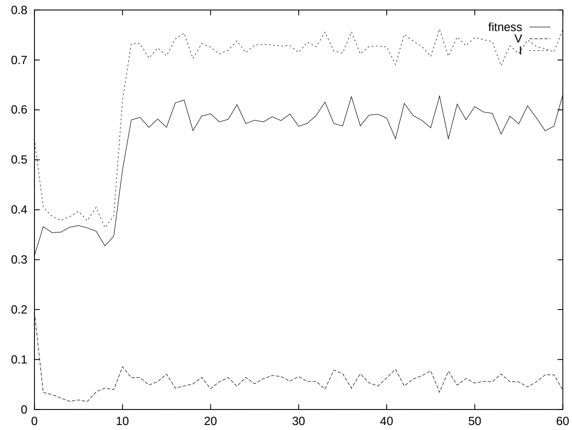

Figure 7 *Fitness* vs. two main component factors.

of the abilities, for instance, “to run faster” against “to keep safe”.

4 Conclusion

In this paper we have proposed a method to evolve high level rules using classical genetic algorithms. Through these evolutionary processes the mini-robot Khepera has been able to develop an autonomous behavior which allows it to survive in its environment. The role of investigators has been limited to provide the survival criterion and the structure of the fuzzy controller.

The average set of rules of the individuals in the last generation shows that this method provides solutions that are another main result is based on the number of rules, which has turned out to be lower than it was expected.

Our current work is aimed at using the same approach in the emergence of the fuzzy membership functions and in the evolution of both parts of the fuzzy controllers at the same time. We have already obtained some interesting results in the isolated evolution of the membership functions. We are also confident about the possibility of evolving both at the same time. We are also studying the possibility of mixing human designed controllers with genetically obtained ones.