Electronics, Robotics and Automotive Mechanics Conference 2008

Cooperative adaptive behavior acquisition in mobile robot swarms using neural networks and genetic algorithms

Cesar Muñoz, Nicolás Navarro, Tomás Arredondo and Wolfgang Freund
Universidad Técnica Federico Santa María, Valparaíso, Chile,
Casilla 110V, Valparaíso, Chile.
{cmunoz, nnavarro, tarredondo, wfreund}@elo.utfsm.cl

Abstract

This paper describes the use of softcomputing based techniques toward the acquisition of adaptive behaviors to be used in mobile exploration by cooperating robots. Navigation within unknown environments and the obtaining of dynamic behavior require some method of unsupervised learning given the impossibility of programming strategies to follow for each individual case and for every possible situation the robot may face [1,5]. In this investigation in particular, it is intended to expose some of the benefits of cooperative learning robots using novel biologically inspired heuristic methods. Experiments were conducted using a Khepera mobile robot simulator which uses a neural network to generate behaviors based on robot sensor measurements. The training of this network was carried out with a Genetic Algorithm, where each individual is a neural network whose fitness function is the output of a function, proportional to the area covered by the robot.

1. Introduction

Individual robot navigation in unknown and unstructured environments is a fundamental problem in robotics. Extensive previous research has been made into this problem with the aid of various robotic architectures, sensors and processors. One such approach is behavior based robotics which follows an approach that in general does not consider the use of world models or complex symbolic knowledge [9]. Cooperative exploration is an extension of this problem which has received increasing interest by researchers [7]. An analogy with biological cooperative systems is used in this work, where cooperation generates a better performance for the individuals with this characteristic. A new methodology is introduced in order to achieve cooperation between the robots in order to generate a system where cooperative behaviors emerge given that the robots are provided with basic information such as base and relative robot distance. The main idea is to generate a system where cooperative behaviors emerge given essential information such as base and relative robot distance.

The results obtained with this approach show the benefit of cooperation compared with those that come from a non cooperative system, where each individual has only the information relative to its distance to the base. In our experiments the robots explore in a completely unknown environment because they have no information about the maps in which they navigate.

In the present work robot navigation is performed in the YAKS [3] Khepera simulator by providing sensor values directly into a neural network that drives left and right motors for each robot. In our experiments we used infrared sensors which gave limited information about the surroundings in which the robot was located, and a normalized Euclidean distance was used for the measurement of distance as inputs for the neural network.

The search space of suitable behaviors is huge and designing these behaviors by hand is very difficult therefore we have used action-based environmental modeling (AEM) [1,2] in order to reduce the search space. AEM was implemented with a small action set of four basic actions (e.g. go straight, turn left, turn right, turn around) in order to encode a sequence of actions based on sensor values.

This paper is organized as follows. Section 2 gives a description of the robotic system. Section 3 introduces the experiments performed. In section 4 we

978-0-7695-3320-9/08 $25.00 © 2008 IEEE
DOI 10.1109/CERMA.2008.89

describe and summarize our results. Finally, in section 5 some conclusions are drawn.

2. Robotics Simulation System Description

This section presents the main features of the robotic simulation system used for these studies. The simulation system has several different elements including: the robot simulator [3], AEM [1,2], neural networks [5], and GA [5].

2.1. Khepera Robot

For our simulations, a Khepera robot was chosen (Figure 1). The basic Khepera robot configuration has two DC motors and eight sensors of infrared proximity with which it is possible to detect objects up to a short distance. These low cost sensors provide the robot only with local data around itself. The robot does not have global position information (e.g. GPS). The infrared sensors provide 10 bit output values, which allow the robot to know in approximate form the distance to obstacles. In order to make the experiments more real, a 5% random noise was introduced to the readings. A zero input value indicates no obstacle is found, and a one indicates that the robot is close or in contact with an obstacle.

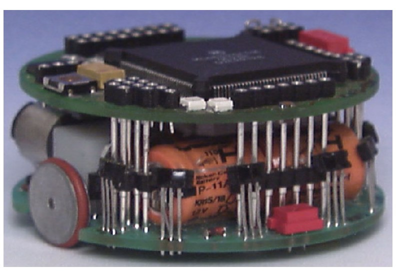

Figure 1. Khepera Robot

2.2. Simulation System and AEM

These experiments were made using the YAKS [3] Khepera simulator. The simulator has a map where the robot moves, and provides the readings for the sensors according to the current map (room). It also handles the information of areas visited, not visited and the various obstacles in the room. The rooms are square with 5,500 mm each side and are differentiated by the amount and type of obstacles. The rooms were divided into 10,000 zones (each 55 mm by 55 mm). The robot generates an internal map in which the zones are marked with various values: obstacles are indicated with a value of -1, those not visited by the robot are marked with 0 and the visited ones with 1. The robot executes 400 steps in each simulation but not every step produces forward motion as some only rotate the robot in place.

In order to reduce the search space of behaviors, we have used a limited number of actions for the robot to execute in each step. Using a similar encoding as in [1], four basic actions were used:

- A1: Go 55 mm straight on.
- A2: Turn 30º left.
- A3: Turn 30º right.
- A4: Turn 180º left.

2.3. Applied Artificial Neural Network (ANN)

As shown in Figure 3, the robot neural network [1,2] used in this research, have twelve input neurons (one for each infrared sensor, one for the distance to the start point and one for each distance to each robot), five neurons in the hidden layer, and two output neurons which are connected to the motors for robot displacement.

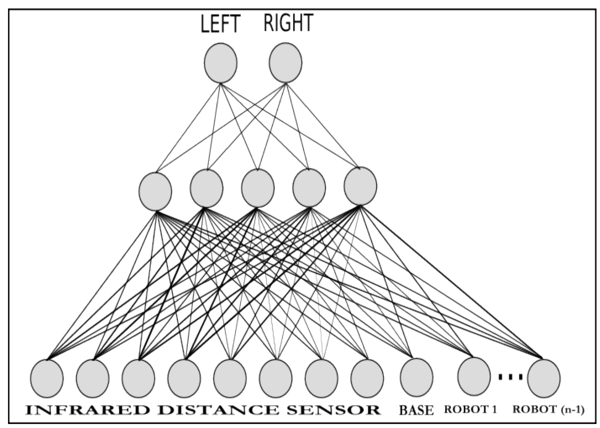

Figure 2. Configuration of the Neural Network

2.4. Genetic Algorithm

As previously mentioned, a GA [1,5] is used to find an optimal configuration of weights in the neural network [4]. Each individual in the GA represents a neural network which is evolving with the passing of different generations.

The GA uses the following parameters:

- Population size: 200.
- Crossover operator: Random crossover.
- Selection method: Elite strategy selection.
- Mutation rate P_{mut}: 1%.

- Generations: 100.

The GA procedure is as follows:

Step 1: Initializing population: An initial population $I_1,\ldots, I_n$ is randomly generated.
Step 2: Robot executes behavior: The behavior of each robot is simulated.
Step 3: Computing fitness: The fitness $f_1, \ldots, f_n$ for each individual $I_1,\ldots, I_n$ is computed based on complete behavior in the room.
Step 4: Selection: Using the fitness values $f_1, \ldots, f_n$, select an elite group of ten (10) individuals from the current population (C).
Step 5: Crossover: For each elite individual select ten mates from the rest of the population for the next generation. The rest of the individuals are randomly generated, forming C_1.
Step 6: Mutation: Mutate the individuals in C_1 based on mutation rate P_{mut}.
Step 7: Go to Step 2.

2.5. Fitness Function

The fitness function uses the information provided by the simulator and the capacity of exploration of each individual as follows:

$$f_i = Z_v / Zmax$$

Where:
f_i : is the fitness of an individual.
Z_v : visited zones.
$Zmax$: maximum number of possible visited zones.

The interaction of the complete system can be seen in Figure 3. The GA selects the best individuals in each generation and modifies the weights based on the f_i fitness function after each individual executes his behavior in each room. The GA selects the overall best individual in the last generation of the simulation.

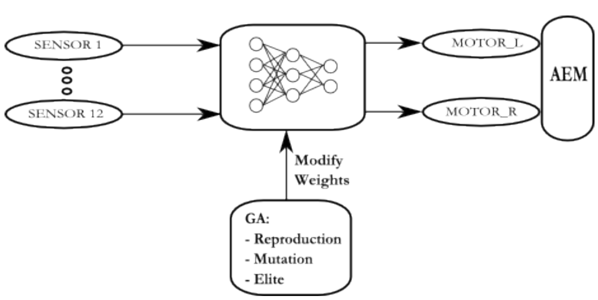

Figure 3. System Interaction

3. Experiments

The experiments performed included two main cases and four different rooms. System target was to optimize the robot neural networks in order to obtain the best behaviors for maximum area exploration, given specific energy constraints, represented by a limited number of actions. For this study, 400 actions were used.

The first case evaluated (case 1) consisted in obtaining simultaneous independent behaviors of four robots. Each robot counted with the information from their respective eight infrared distance sensors and the distance to the starting point. In this first case the relative distance towards the other robot was not considered.

The second case studied (case 2), consisted in obtaining independent behaviors for each of four robots; the difference with case 1 is the incorporation of tree neurons whose input signal is the distance towards the others robots. This allowed the generation of independent behaviors, but through a cooperative implicit (i.e. emergent behavior) communication between robots.

For all experiments, the starting point of exploration was the center of the lower wall.

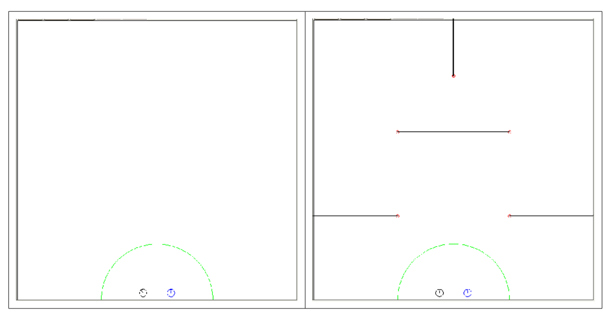

Figure 4. Square and Horizontal Rooms

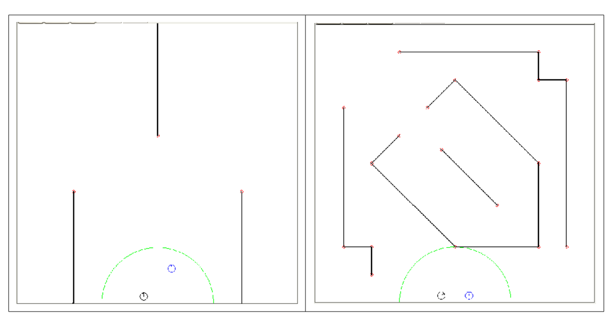

Figure 5. Vertical and Labyrinth Rooms

3.1. Experimental Results

In Figures 4 and 5 we show our four room environments. In these figures, straight lines represent maze walls.

In Figures 6 and 7 we show the evolution of the experiments as the average of individual exploration for the best individuals given 100 generations in all 4 rooms for cases 1 and 2 respectively.

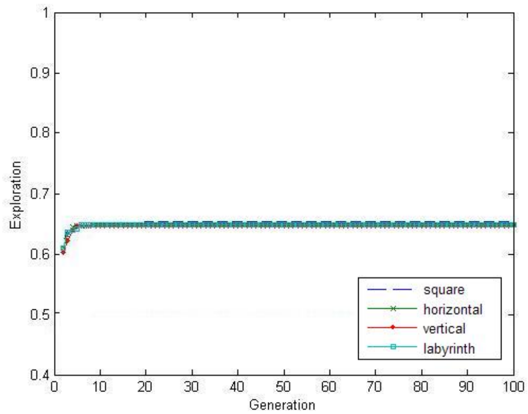

Figure 6. Exploration for Non Cooperative Behavior

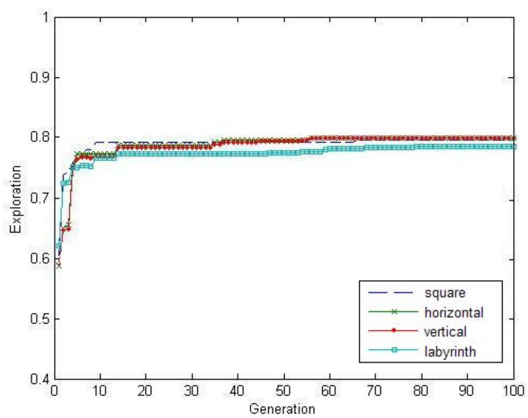

Figure 7. Exploration for Cooperative Behavior

4. Analysis and Evaluation

In Figures 4 and 5 we can observe the behavior of the robots in the second case (cooperative) compared with the non cooperative case. These results can be explained in terms of biological behavior given that we know that organisms which interact with each other can work together in order to achieve common objectives with higher efficiency than individuals who work alone.

In the simulation, this phenomenon could be attributed to the use of neural networks which when trained with variety of stimuli respond by interpolating solutions for unforeseen cases.

In Figure 7 it can be appreciated that the emergent behavior of the cooperative robots contributes significantly towards obtaining greater area exploration.

In Table 1 we show a more quantitative analysis of our results, where we show the average and standard deviation of explored area, normalized by the theoretical maximum navigable area (only ten different test runs were made, given that the results did not show much variation).

Table1. Exploration Percentage of the Best Individual

	Cooperative, Case 1		Non Cooperative, Case 2	
	Average	***Standard Deviation***	***Average***	***Standard Deviation***
Square	0.628	0.0008	0.802	0.0000
Horizontal	0.656	0.0000	0.788	0.0004
Vertical	0.645	0.0034	0.763	0.0008
Labyrinth	0.660	0.0029	0.749	0.0006

5. Conclusions and Future Work

The results presented in this paper are promising, based on the wide range of applicability. We show that the behavior of cooperative robots for navigation purposes is more efficient than the one obtained with an independent strategy.

Using our novel emergent cooperative approach we have validated that cooperative learning methods and the sharing of basic information can generate a wide range of useful behaviors in environment navigation.

These relative distances between robots and toward the base are variables which can be easily implemented in hardware based upon the strength of a radiofrequency signal. This means that with simple and low cost sensors we can characterize complex behaviors decreasing individual computational efforts.

We have realized from our experiments and previous results [8] that using cooperative learning methods and increasing shared information we can obtain a wide range of behaviors oriented to environments navigation and other kind of applications which requires this kind of adaptive behavior [7].

We are currently working in expanding the ANN of our simulator and improving the cooperation abilities of the robots by adding other message sets which can be shared in a similar vein as distance information.

6. Acknowledgements

This research was partially funded by the DGIP of UTFSM (230726).

We would like to thank Elena Villanueva a master student of automation at UTFSM.

7. References

[1] S. Yamada, "Evolutionary behavior learning for action-based environment modeling by a mobile robot", Applied Soft Computing 5 (2005) 245–257.

[2] S. Yamada, “Recognizing environments from action sequences using self-organizing maps”, Applied Soft Computing 4 (2004) 35–47.

[3] YAKS simulator website: http://r2d2.ida.his.se/

[4] Frank Hoffmann, “Soft computing techniques for the design of mobile robot behaviors”, Information Sciences 122 (2000) 241-258.

[5] Malrey Lee, “Evolution of behaviors in autonomous robot using artificial neural network and genetic algorithm”, Information Sciences 155 (2003) 43–60.

[6] Chapman E., Sahin F., “Application of Swarm Intelligence to the Mine Detection Problem”, IEEE Internacional Conference on Systems, Man and Cybernetics (2004) 5429- 5434.

[7] W. Erlhagen, A. Mukovskiy, E. Bicho, G. Panin, C. Kiss, A. Knoll, H. van Schie, H. Bekkering, “Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning”, Robotics and Autonomous Systems 54 (2006) 353–360.

[8] Arredondo, T., Freund,W., Muñoz, C., and Quirós, F.: "Learning Performance in Evolutionary Behavior Based Mobile Robot Navigation". Lecture Notes in Artificial Intelligence, Vol. 4827. Springer-Verlag, Berlin (2007) p. 811-820.

[9] Naoyuki Kubota, Toshihito Morioka, Fumio Kojima, Toshio Fukuda, “Learning of mobile robots using perception-based genetic algorithm”, Measurement 29 (2001) 237–248.