

Implementation and Evaluation of a Reactive Multi-Robot System

P. Lucidarme, P. Rongier, A. Liégeois

LIRMM

University Montpellier 2 and CNRS, 161 rue Ada, 34392 Montpellier, France

{lucidarm,rongier,liegeois}@lirmm.fr

Abstract--This paper presents the implementation of an experimental setup to study the behavior of a group of purely reactive mobile robots subject to serious unrecoverable failures: some of them have lost mobility, others have lost infrared vision. To survive, each robot must reach a supply center. The eventually degraded mission uses complementary capabilities of the two types of robots by letting a blind robot, meeting by chance an invalid one, carry it. The latter is able to "see" a supply center within a wide range. The hardware and software of the experimental devices are described, using very simple and low-cost components. Experiments are run with the robots in different initial positions. The results are compared to those obtained by many computer simulations and by a Markovian process model, both are also described.

Index terms—Multi-Robots, Autonomous Mobile Robots, Distributed Intelligence, Reactive Agents, Modular Hardware

I. INTRODUCTION

More and more, mobile robotic systems are used or foreseen in many applications such as in manufacturing, work and rescue in hostile environments, underwater and planet exploration, as well as for leisure and games. Distributed solutions using multi-robot systems present many advantages such as robustness to unexpected disturbances, fault-tolerance, thanks to redundancy, self-adaptation and self-organization. Other advantages, such as simplified design, low cost and ease of maintenance, are exhibited by the so-called reactive systems: in this case, each robot neither needs map of the environment [1] nor *a priori* offline computations of complex plans [2]. It is equipped with crude, range limited, sensors and emitters, and with simple signal processing immediately selecting the elementary action to perform such as:

- Start, stop
- Wander
- Make a given turn
- Accelerate, decelerate
- Follow a given heading
- Push, pull
- Hear
- Emit
- Grope
- Take
- Deposit
- Get energy

Historically, the multi-robot systems have emerged for Space, Defense or Nuclear applications [3][4][5]. In parallel, a lot of work has been done to model the emergence of an intelligent cooperative behavior shown by societies of insects, herds of mammals, fleets of birds and fish shoals [6].

In this work, the cooperation and behavior requiring either complex modeling of the environment, sophisticated communications, negotiating, or task and motion planning have not been considered [7]. Such designs are called cognitive and deliberative, opposite to the reactive ones, defined by the elementary capabilities listed above. Deliberative systems can be efficient but also much more complex, expensive and slow: as their *a priori* information's quality decreases due to uncertainties and disturbances, they must update them and alter their plans.

Furthermore, "animats" modeling involving dozens to thousands of "agents" have been discarded: realistic robotic applications must be performed by a reduced number of agents due to cost, mass and energy limitations. However, several loosely coupled groups can be considered.

In this paper, the focus is put on study and implementation of a task-oriented multi-robot system [8], while elsewhere designing a more general reactive architecture [9].

Section II describes the application and the results showing the expected behavior computed by thousands of long and tedious simulations on the one hand, and by a stochastic automaton model on the other hand [10][11]. Section III describes the real robot implementation involving low-cost experimental robots and laboratory setup. In Section IV, experimental results are compared to the computer analyses. The conclusion includes also the ongoing and future theoretical and experimental research.

II. THE APPLICATION

The aim is to study the behavior of a group of purely reactive mobile robots subject to serious unrecoverable failures: some have lost their mobility and the others have lost their vision. The global goal for each member of the two groups is to reach a supply center (SC).

A. Description of the application

The system is composed of two kinds of robots with different capabilities

- A B-Robot can move, but can only sense its nearby environment by groping around. It can also hear call signals. It acts as if it was blind.
- A P-Robot cannot move, but can see. It is able to call a B-Robot by emitting a signal. It is as if this robot was paralyzed.

It is clear that a P-Robot cannot reach a SC without the help of a B-Robot. On the other hand, a B-Robot can find a SC by chance (it moves in random walk). But the cooperation of the two groups increases the success of the mission. When a B-Robot hears the call of a P-Robot, it moves toward it and 'loads it on its back'. This makes a new robot (called BP-Robot) with both capabilities: it is now able to move and also to see supply centers from a longer distance than a B-Robot alone.

The more robots of any kind that reach the SCs the more successful the mission.

In the following we simulate this mission with 3 P-Robots, 3 B-Robots and 2 Supply Centers.

B. The Simulations

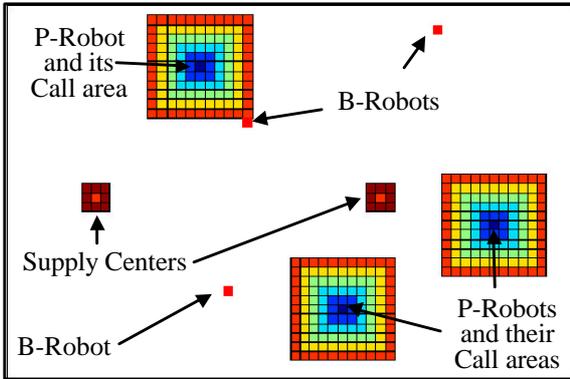


Figure 1: a snapshot of the simulation

To understand the global behavior of the system, we performed ten thousands simulations in a discretized area and noted for each simulation the different final robots locations. Zero, one, two or the three P-Robots may be found and carried by B-Robots to the SCs.

Figure 1 shows a snapshot of the simulation, where the system is still in its initial state: 3 B-Robots are searching randomly a SC.

The simulation results are discussed in Section IV.

In our system, random movements of a B-Robot occur until it senses a supply center or hears the call of a P-Robot, or until a P-Robot which is carried "sees" a supply center. Thousands of simulations are necessary for studying the behavior for a lot of various initial positions and numbers of robots. This takes many hours. Then a theoretical model

based on stochastic Petri nets and Markov chains has been proposed [8][12].

C. The Stochastic Model

The system can be modeled by the net of Figure 2.

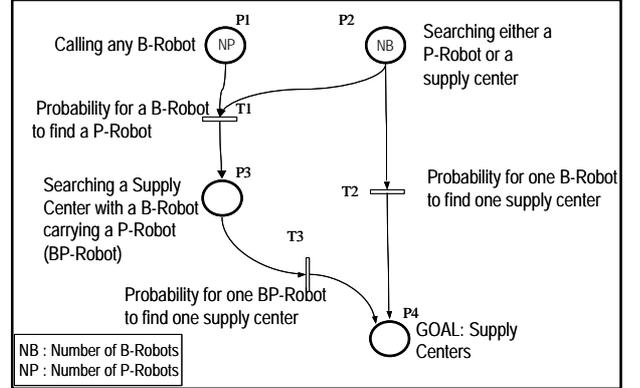


Figure 2: stochastic Petri net of the application

From the marked graph of the net we obtain the Markov chain with all the states in which the system could be. The state number k is described by the following vector:

$$E_k = \begin{bmatrix} M(P_1) \\ M(P_2) \\ M(P_3) \\ M(P_4) \end{bmatrix}$$

where $M(P_i)$ is the number of robots in the place P_i .

We obtain twenty different states for 3 P-Robots and 3 B-Robots (From E_1 to E_{20}). Each of the twenty states represents a different configuration of the process. The probabilities of state transitions are computed by simulations of random walks. The mission always begins in the initial state E_1 (Table 1): each B-Robot is searching either a SC or a P-Robot. The four different final states are also in Table 1. The best final state, the one we want to reach the more frequently is E_{20} where all the 3 P-Robots have been found by the 3 B-Robots and the 6 have reached the SCs.

$$E_1 = \begin{bmatrix} 3 \\ 3 \\ 0 \\ 0 \end{bmatrix} \quad E_{13} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 3 \end{bmatrix} \quad E_{17} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 3 \end{bmatrix} \quad E_{19} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 3 \end{bmatrix} \quad E_{20} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \end{bmatrix}$$

Table 1: initial (E_1) and final states (E_{13} , E_{17} , E_{19} and E_{20})

The worst final state is E_{13} , where none of the P-Robots has been carried to any SC and all the 3 B-Robots have finally found randomly a SC by themselves.

The Markov chain obtained allows us to compute the (20×20) Stochastic Matrix S of the system. Equation 1 gives

for any step time n the evolution of the row vector of the states probabilities.

$$\mathbf{p}(n) = \mathbf{p}(0) \cdot S^n$$

Equation 1: probabilistic state transition

$\mathbf{p}(0) = [1 \ 0 \ \dots \ 0]$ is the initial (1x20) state vector, the system is in E_1 , the probability to be in the state E_1 is equal to one. All the model results are presented in Section IV.

To enhance our studies we have made real robots for comparing the experiments to the simulations results and the predictions of the Markov analysis.

III. REAL IMPLEMENTATION

A. Purpose

Our goal is to design and realize small and low-cost robots and laboratory setup. The autonomous mobile robot is called “Servobot” and the cost of the electronic and mechatronic components does not exceed 150 USD. This robot can adopt the following basic behaviors: moving, looking for another robot, avoiding obstacles. We now describe the architecture chosen.

B. Hardware

First, we have built the chassis of the robot. We have intentionally constructed a very simple mechanical system using two motorized wheels. Each wheel is connected to a servomotor which is cheap, robust, and easy to control. To get unlimited rotations of the wheels, we have removed the position feedbacks. For that reason, we have printed stripes on the wheels and the information is obtained by an U-shaped system. There are 32 lines per wheel, which gives a resolution of three millimeters. The chassis is also equipped with two front switches. This allows the robot to avoid collisions with obstacles or other robots.

“Servobot” is a small autonomous mobile robot designed to be modular, as in [13]. It is composed of different boards which can be added one on top of the other. The minimal configuration is composed of the motherboard and of the motors control module. Two optional boards have been developed for the application described.

- The motherboard is based on a 68HC811E2 micro-controller. The 2Ko of EEPROM memory can be programmed through the serial link of a PC. This uses Assembly, Basic or C languages for example. This board has also 5 In/Out ports to interface with the others boards.
- The motors control module is built to interface the motherboard with the motors. It allows the user to

choose within a range of 65000 different speeds. It also manages the shaft encoders on the wheels.

- The infrared board is specially designed for our application. It can discriminate two infrared signals of different frequencies. It can also estimate the distance and the direction of the source. The micro-controller converts analogue signals into digital ones for each of the four IR receivers and frequencies.
- The monitor board allows the user to know the current state of the robot (It can also be useful when debugging the programs). It is composed of different LEDs which are connected to the micro-controller.

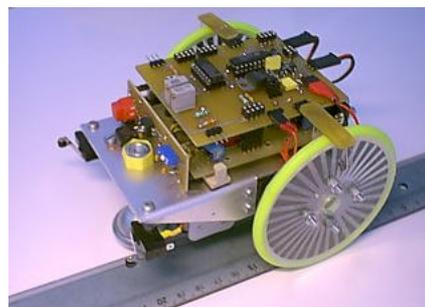


Figure 3: a “Servobot” in its basic configuration



Figure 4: an infrared beacon which represents a P-Robot or a SC

Figure 3 shows a robot in its minimal configuration: it is here equipped with the motors control board (the upper one) and with the motherboard. The power is supplied by two NiCd batteries that give about one hour of autonomy (The average duration of an experiment is 15 minutes). In our implementation, a P-Robot is simply an infrared beacon (Figure 4). In the original mission, P-Robots are carried by B-robots. The combination of the two agents improves their performance. In the application, B-Robots don’t carry the P-Robots. But when a B-Robot meets a P-Robot (an infrared beacon) the beacon stops emitting. Then the B-Robot behavior is modified: the main program switches to BP-Robot behavior by activating new sensing capabilities. For a better understanding, we now describe the algorithm and the different behaviors.

C. Algorithm and Behaviors

The mobile robot has a reactive architecture, and the algorithm is based on different behaviors such as random walk and moving towards a sensed target thanks to a simple heading in the maximal signal orientation (steepest ascent technique).

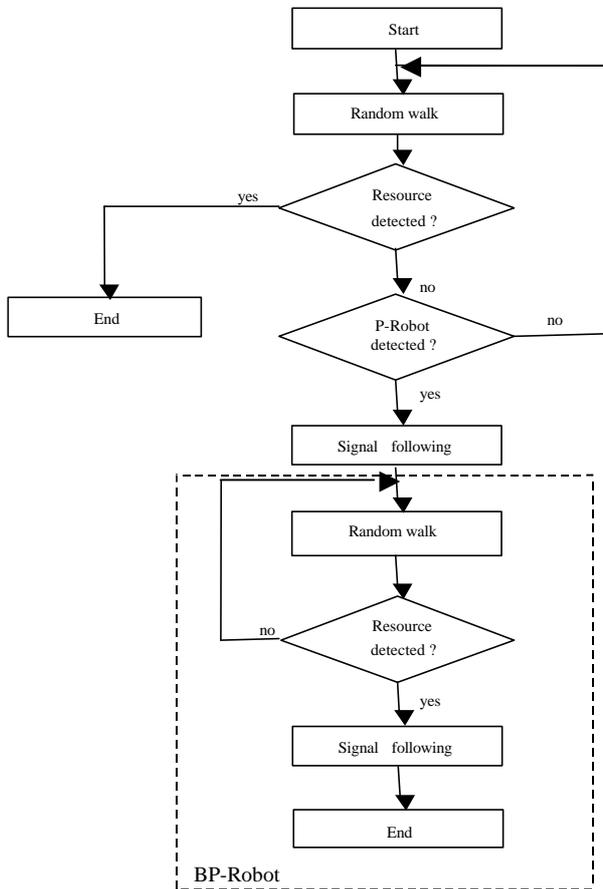


Figure 5: algorithm of a mobile robot

Figure 5 shows that the algorithm is based on simple behaviors, and that conditions depend on environmental signals. At the beginning, the B-Robot is walking randomly until it finds a supply center or a P-Robot. In that case, the robot finishes its mission when the resource is reached. Otherwise the formed BP-Robot looks for a SC, but can detect it from a longer distance. When the supply center is reached, its mission is ended. The different behaviors are described below:

- Random walk: periodically the robot chooses randomly one of the eight cardinal directions. It moves along this direction until the next step. A counter allows us to know the number of steps used to end its mission.
- Signal following: when a B-Robot finds a P-Robot or a supply center, it changes its own orientation to reach its target. This behavior is based on a proportional control. Each receiver decodes the frequency and transforms the

information into a 0-5 volt scale signal. Nearer from the source the receiver is, higher the signal. Figure 6 shows a top view of the robot trying to reach the supply. There are two values (one on each receiver) which are proportional to the distance between the beacon and the receiver. Let's call V_L and V_R the values of the left and right received signals. The controls implemented on the motors are:

$$M_L = \frac{M_{\max}}{2} + (V_R - V_L).K$$

$$M_R = \frac{M_{\max}}{2} + (V_L - V_R).K$$

Equation 2: controls implemented on each motor during signal following behavior.

Where:

- M_L is the command voltage on left motor.
- M_R is the command voltage on right motor.
- M_{\max} is the maximal voltage on any motor.
- K is the proportional gain.

With these simple controls, the robot can reach the beacon. When the perceived signals V_L and V_R are maximal and a front switch detects a collision, the goal is reached. Otherwise, if the signals are not maximal, the collision is due to an obstacle. Then the robot goes back and tries another way.

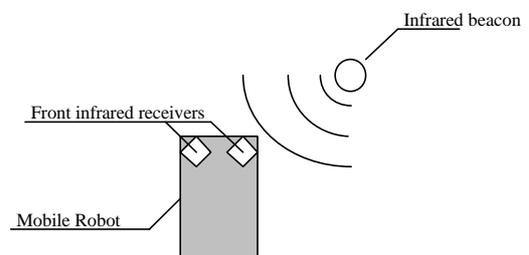


Figure 6: top view of signal following behavior

These two basic behaviors are sufficient to complete the mission. The algorithm represented in Figure 5 has been implemented on our robots. Note that the three tests in this algorithm are different by the sensing distance. To implement these tests, we have set three thresholds corresponding to different ranges: B-robot to SC, B-Robot to P-Robot, BP-Robot to SC .

IV. RESULTS AND DISCUSSION

This Section analyzes and compares the results obtained by the three approaches. We have run many experiments in different initial conditions. The particular case of 3 P-Robots and 3 B-Robots is interesting because the mission can be successfully ended if each B-Robot finds a P-Robot.

A. Simulation and Markov analysis

Let us focus only on the final states. These states are representative of the Markov Chain. They measure the performance of the mission. We represent the probabilities of all the possible ends.

First, we compare simulations and probabilistic results. 10000 simulations have been run with 3 P-Robots and 3 B-Robots. The results are drawn on Figure 7: the dotted line is the simulation results and the continuous one the Markovian results.

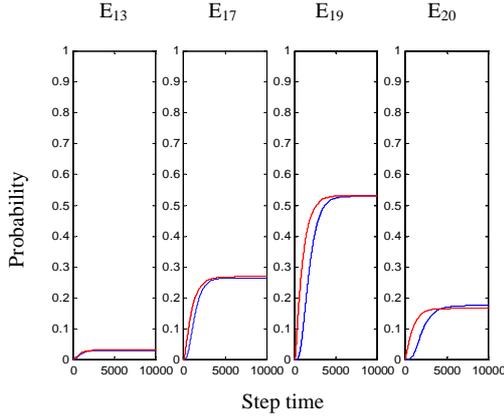


Figure 7: simulation and probabilistic results for 3 P-Robots and 3 B-Robots

We can notice on Figure 7 that, when compared to the simulations, the Markov process converges with a small delay. This is due to the approximation made when building the Markov chain. Actually, the system is modeled with only a 20 states Markov chain. We just need approximate results quickly. If we want to be more accurate, the chain must have as many states as the very large number of combinations of the positions.

We can conclude that 3 P-Robots and 3 B-Robots is not the optimal configuration to successfully complete the mission. The likely state is here E_{19} while the state we wish to reach the more frequently is E_{20} : the Markov analysis was able to predict quickly that three B-Robots are not sufficient to optimize the probability of full success.

It is demonstrated in [8] that, given 3 P-Robots, a good compromise between getting a better probability of success and avoiding complexity is obtained when 4 B-Robots are used.

B. Experimental results

Thirty experiments are here presented, using a real system (Figures 8-9). Of course, real experiments take longer than simulations. To get the exact results at the end of the experiment, the number of steps of each robot is stored in a register. The values are read after each experiment and processed to plot Figure 10.



Figure 8: beginning of one experiment



Figure 9: end of one experiment

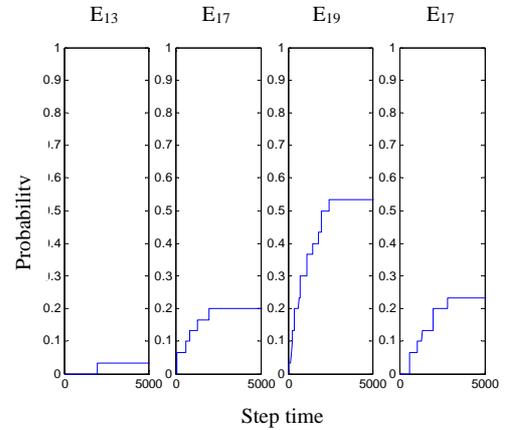


Figure 10: results for thirty real implementations with 3 P-Robots and 3 B-Robots

The measured probabilities of the four possible final states appear to be in good accordance with the results shown in figure 7. The following Table 2 confirms the validity of our approaches.

State	E_{13}	E_{17}	E_{19}	E_{20}
Markov	2.99 %	26.47 %	53.02 %	17.52 %
Simulations	3.32 %	26.87 %	53.18 %	16.63 %
Experiments	3.33 %	20 %	53.33 %	23.33 %

Table 2: comparison of the results

V. CONCLUSION

An experimental laboratory setup to study the particular behaviors of two heterogeneous complementary groups of agents under failure, has been presented. All the mobile mini-robots and the infrared environmental beacons are made of simple and low-cost modules easy to implement in a research laboratory. Hardware modularity and software modularity make the most sophisticated though reactive agents -the mobile "Servobots"- easily adaptable to many types of experiments for research on multiple mobile robots systems. Experimental results have proved to be in accordance with other design aids such as computer simulations and theoretical stochastic models. The latter have thus proved to be valid. The variations of the performance with respect to changes in the number of robots can then be checked rapidly as shown in [8] where groups of 4 and 5 B-Robot have also been considered. In addition, a new module is being designed for checking further a more general reactive architecture [14] involving agent's satisfaction and altruistic behavior.

References

- [1] O. Pinchard, A. Liégeois, T. Emmanuel, "A Genetic Algorithm for Outdoor Robot Path Planning", IAS'4, IOS Press, pp.413-419,1995.
- [2] S. Botelho and R. Alami., "Multi Robot Cooperation in the Martha Project", IEEE Int. Conf. On Robotics and Automation, pp.1234-1239, 1999.
- [3] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE J. on Robotics and Automation, RA-2, pp. 14-23, 1986.
- [4] R.C. Arkin, "Cooperation without Communication: Multiagent Schema-Based Robot Navigation", J. of Robotic Systems, Vol. 9 (3), pp. 351-354, 1992.
- [5] L.E. Parker, "ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation", IEEE Trans. on Robotics and Automation, Vol. 14, No. 2, pp. 220-240, 1998.
- [6] C. Reynolds, "Flocks, Herds and Schools : A Distributed Behavioral Model", Computer Graphics, 21(4), pp. 25-34, 1987.
- [7] F.R. Noreils, "Toward a Robot Architecture Integrating Cooperation between Mobile Robots: Application to Indoor Environment", The Int. J. of Robotics Research, Vol. 12. No 1, pp. 79-98, 1993.
- [8] P. Rongier, A. Liégeois and O. Simonin "Markovian Analysis of a Heterogeneous System: Application to a Cooperation Task for Multiple Consumer Robots", Proc. IEEE Int. Conf. on SMC, pp. 3033-3038, 2000.
- [9] O. Simonin, A. Liégeois and P. Rongier, "An Architecture for Reactive Cooperation of Mobile Distributed Robots", DARS'4, pp. 35-44, 2000.
- [10] A. Martinoli, A.J. Ijspeert and F. Mondada, "Understanding Collective Aggregation Mechanisms: From Probabilistic Modeling to Experiments with Real Robots ", Robotics and Autonomous Systems, No 1, pp. 51-63, 1999.
- [11] A. Mori and al., "A Robot-Controlling Agent Description with Finite States Machine", DARS 3, pp. 225-234, 1998.
- [12] P. Rongier and A. Liégeois, "Analysis and Prediction of the Behavior of one class of Multiple Foraging Robots with the help of Stochastic Petri Nets", Proc. IEEE Int Conf. on SMC., pp. 143-148, 1999.
- [13] K. Malmstrom and J. Sitte, "Experimental Setup for Multiple Agent Interaction", Proc. 3rd IFAC Symp. on Intelligent Autonomous Vehicles, pp. 37-41, 1998.
- [14] O. Simonin and J. Ferber, "Modeling Self Satisfaction and Altruism to handle Action Selection and Reactive Cooperation", in SAB'2000 Proceedings Supplement, pp. 314-323, 2000.