

Agent Control Technology for Multi-Robot Systems

A. Caggiano^a, D. D'Addona^a, B. Siciliano^b, R. Teti^a

^a Dept. of Materials and Production Engineering, University of Naples Federico II
P.le Tecchio 80, 80125 Naples, Italy

^b Dept. of Computer and Systems Engineering, University of Naples Federico II
Via Claudio 21, 80125 Naples, Italy

Abstract

Nowadays, *Multi-Robot Systems (MRS)* control represents a great challenge in the research community: the interest in MRS is justified by the numerous advantages that a group of autonomous robots offers compared to a single robot, in particular when complex missions are assigned. However, these systems need a complex control architecture and a method of coordination of the autonomous robots. In this paper, a *Multi-Agent System (MAS)* based approach is proposed to solve this problem: the purpose is to realize a decentralized system, where each robot, using a *Reinforcement Learning (RL)* mechanism, autonomously acquires the appropriate behaviour and suitable rules through its experience of interaction with the environment and the other robots.

Keywords: Multi-Robot Systems, Agent-Based Control, Reinforcement Learning

1. Introduction

In recent years, several research efforts have been directed toward the control of groups of autonomous robots. The interest in this field is well justified by the several advantages that such systems present compared to single autonomous robots and it is well supported by the technology improvements that allow the interaction and integration among multiple systems [1-2].

For many complex tasks, a single autonomous robot may not be able to achieve the mission alone due to its functional limitations: in this case, it is essential to work cooperatively with other robots [3-5].

The applications of *Multi-Robot Systems (MRS)* involve different fields, e.g. industrial, military and service robotics, or research and study of biological systems, and they may concern largely different kinds of missions, e.g. exploration, box pushing, military operation, navigation in unstructured environment,

traffic control, entertainment, simulations of biological systems. When many autonomous robots work together at tasks, the system configuration can be classified into 2 types. The first one is a *centralized system* where all the robots obey to a master [6]. The second is a *decentralized system* where all the robots work cooperatively [7].

In the centralized system, the master robot is required to have many functions, and if it breaks down, the whole system becomes incapable to achieve any task. On the other hand, in the decentralized system, each robot has a function. As many robots work cooperatively, it is easier to deal with various tasks. Moreover, if one robot breaks down, other robots can help or replace it. Thus, the decentralized system is more fault-tolerant than the centralized one.

However, decentralized systems are difficult to realize due to interaction among robots. When there is no interaction, each robot has to work optimally for its

purpose, so that the total task should be achieved optimally, and it works selfishly. On the other hand, when using a public resource, conflicts among robots might occur. These conflicts may cause collisions and deadlock states among robots in a local area. To avoid conflicts, the robots should communicate at least locally and coordinate among themselves. Robot communication is as important as selfishness and, in particular, it is fundamental for coordination.

Flexible and powerful control architectures are desired to provide a structured approach to simplify and organize the design, implementation and validation of complex control schemes for multi-robot systems.

In recent years, multi-agent approaches to multi-robot control design have drawn much attention in the robotics community. In an agent-based architecture, agents can communicate, coordinate and negotiate to meet their goals in a framework suitable for task execution. *Multi-agent system* (MAS) architectures support both centralized (agent facilitator model) and distributed (agent to agent model) arrangements.

A typical approach to a MAS is to give a fixed behaviour rule set to each agent so that it can behave appropriately as soon as the system starts to move. In this case, the behaviour rule set has to be designed by a human programmer for all the possible situations that an agent can encounter during system operation. This approach seems effective when these situations are easily identified so that the programmer can provide an appropriate behaviour rule set. However, since in a MAS the number of situations to be considered rapidly increases with the number of agents, it is practically impossible, for most cases, to provide an appropriate behaviour rule set to each agent in advance. This is even clearer when agents are physical bodies, as in the case of MRS, because it may happen that robots are disturbed by sensor noise, unexpected obstacles, or accidents in motors or electronic wiring [8].

Considering it is impossible to give hand-crafted behaviour rules for all possible situations that a robot will encounter, the latter must develop cooperative behaviour autonomously, through its experience of interaction with other robots and the environment. That is, instead of providing knowledge in advance in top-down manner, robots should be given adaptation functions to unknown environments and learning functions to memorize experiences and utilize them later, in bottom-up manner. Thus, a methodology is needed to design an on-line autonomous behaviour acquisition mechanism capable of developing the robot's role in an embedded environment.

A traditional solution to overcome this difficulty is to provide a central controller to arbitrate competitive situations between robots or suggest appropriate behaviours when unknown situations occur to maintain the global stability in a MRS. However, such global controller is difficult to design and there are no widely applicable methods, though several trials were tested to design a central controller for dynamic task allocation.

This paper introduces a reinforcement learning approach to the described problem. A decentralized control architecture is developed and applied, whereby each robot autonomously develops a cooperative behaviour through its experience in a group of robots. An innovative MAS-based approach is used to solve this problem: some kind of social activity is expected to emerge in a group of robots through their interaction without the assistance of a central controller [9]. For this purpose, each robot is supposed to have a great adaptability and autonomy. The starting system is a homogeneous MRS because the purpose of the controller is to develop specialization in the system depending on the situation of each robot where the same type of learning function is provided.

2. Multi-Robot Systems

The term *Multi-Robot System* includes different typologies of robotic systems: multiple industrial manipulators, mobile robots with manipulators on board, or a team of autonomous vehicles. The reasons to employ MRS are widely different; however, one of the main motivations is that they can be used to increase the system effectiveness. That is, with respect to a single autonomous robot or a team of non-cooperating robots, a MRS can better perform a mission in terms of time and quality, achieve tasks not executable by a single robot, or take advantages of distributed sensing and actuation.

MRS can execute very complex missions since they can decompose a main task in elementary sub-tasks easier to understand and control. These sub-tasks can be distributed among multiple resources, and the overall behaviour of the system results from the cooperation of the different units. The word "cooperation" underlines the interaction or the integration among the robots, that means, they have to communicate, exchange information or interact in some way to achieve an overall mission.

As already mentioned, MRS offer numerous advantages compared to a traditional robot [10]. The

first one is the presence of autonomous and asynchronous robots in the system, that permits performing *parallel processing*: as a result, the time required for a mission is largely reduced, and many more tasks can be achieved in a single mission. The second advantage is *robustness* realized by the redundant situation where the system has a larger number of robots than the required: faults can be faced in a flexible manner, because if a robot breaks the system can execute the global mission without it. Another important characteristic of MRS is *scalability*, both in the static and dynamic sense. That is, a system is statically scalable when the control architecture can be kept exactly the same whether thousands of robots or only few are used, while it is dynamically scalable when robots can be added to or removed from the system on the fly, without compromising the system's performance. Furthermore, MRS are strongly *adaptable*, since they can be modified according to the different circumstances and missions, by forming teams of robots that have the necessary abilities. Finally, the single robots of a system can be designed independently, and the complexity of each robot can be reduced to a minimum, which means lower costs, simpler design, and faster computation [11].

These benefits, however, often come with drawbacks such as the increased complexity of software for control and coordination of multiple robots, the need for inter-robot communication, etc.

A flexible and powerful control architecture is therefore needed to simplify and organize the design, implementation and validation of such complex control system. The control architecture represents a central part in a MRS as it affects the missions success and strongly influences the system global performance.

3. Multi-Robot Systems Control Methodology

MRS can be seen as a particular instance of MAS [12]. A variety of approaches based on multi-agent coordination theories have been proposed to coordinate groups of robots. However, most of these approaches are only suitable for particular domains.

Generally, the essential component of a MAS is the agent having three fundamental capabilities [13]: *reactivity*, i.e. agents can perceive their environment and respond with a correct action in a timely fashion; *autonomy*, i.e. agents do not only act in response to their environment but can also take actions spontaneously without external intervention; *social*

communication, i.e. agents are capable of interacting with other agents or humans via some communication language. Besides these capabilities, *learning*, *mobility* and *coordination* are also important for agents.

MAS are aggregations of these advantaged capabilities, have self-restraint, intelligent and target driven attributes, and can apperceive and adapt to complex dynamic environments by various ways of sociality, learning and reasoning.

The concept of MAS is inspired by the biological society and it emphasizes coordination between agents having their own autonomy. Coordination can be defined as the use of particular mechanisms to manage inter-dependencies between collaborative activities of agents. In conventional systems, coordination is normally conducted by a top-down planning structure. However, MAS are capable of scheduling tasks, managing decisions, and social communication. These 3 capabilities are the basis to realize coordination.

This work's purpose is to develop a MAS-based design of a MRS composed of robots that gradually but steadily achieve a way to cooperate with each other to play suitable roles through interaction and reach the global goal. Each robot gradually attains a diverse behaviour and the right role based on its experience: this process is called *autonomous specialization*.

The present approach to realize autonomous specialization is based on *Reinforcement Learning* (RL) [14], achieved through an algorithm to find an appropriate set of IF-THEN rules that maximizes reward signals acquired through interaction with the environment. The IF part of a rule is a condition dependent on all sensory information. The THEN part is an action to be executed if that condition is met. From a theoretical point of view, conventional RL may not work effectively for a MRS due to the fact that RL has problems acquiring a strategy behaviour set in a noisy and dynamic environment. In fact, RL requires the assumption that the environment is static, i.e. Markovian. Moreover, in a MRS, the state space and the action space are generally continuous. To solve these difficulties, an instance-based RL is used, called *Bayesian Reinforcement Learning* (BRL), that realizes the segmentation of the two spaces simultaneously in the process of learning, and *Neural Networks* (NNs) are adopted to predict information on the near future.

3.1. Bayesian Reinforcement Learning

BRL operates on a set, R , of behaviour rules generated using a Bayesian discrimination method: at

the beginning, this set is composed of a single indefinite rule. A rule $r \in R$ is formally defined as:

$$r := \langle v, u, a, f, \Sigma, \Phi \rangle$$

where v is the state vector associated with the rule, u is the rule utility, a is the action related to the rule, f is the prior probability of the rule, Σ is the covariance matrix of the rule, and Φ is the sample set associated to the rule. Each rule can be associated to a cluster in the state space where the rule's action is appropriate.

During the MRS activity, when a new sensory input is received by a robot, its control system must verify if there is a rule in R with a similar sensory input so as to execute the corresponding action. If such a rule does not exist, the system will generate a new rule that consists of the current sensory input and the action to be executed and will store it in the rule set R .

In order to compare the sensory input with the states associated to the memorized rules and segment the state space in clusters, a well known method of clustering input data has been adopted: the Bayesian discrimination method [15]. During the learning procedure, the robot, after the execution of an action, is transferred to next state: in specific states, it receives a payoff that is used to update the utilities of the rules.

3.1.1. Bayesian discrimination method

The rules in the rule set compete to trigger an action: using the posterior probability, the risk of misclassification of the sensory input x of each cluster of states, C_i , associated to a rule of the rule set, can be compared and the rule characterised by the minimal risk can be selected. The posterior probability is calculated according to the Bayes' theorem:

$$p(C_i|x) = \frac{P(C_i)p(x|C_i)}{p(x)}$$

Given the probability density of the i -th rule's cluster and the estimated value of f_i , the risk of misclassification of the sensory input x into other clusters can be defined as:

$$\begin{aligned} g_i &= -\log(f_i \cdot P(C_i|x)) \\ &= \frac{1}{2}(x - v_i)^T \Sigma_i^{-1}(x - v_i) \\ &\quad - \log \left\{ \frac{1}{(2\pi)^{\frac{ns}{2}} |\Sigma_i|^{\frac{1}{2}}} \right\} - \log f_i \end{aligned}$$

The rule with the minimum value of g_i is selected as winner and denoted as r_w . If the value g_w of the

winner risk is larger than a threshold g_{th} , then the winner is rejected and the robot performs a random action. Otherwise, the robot performs the action associated to the winner rule.

3.1.2. Utility adjustment

All the rules of the set have an utility: the objective is to keep only the rules with high utilities, since they represent the best behavior for the system. The utilities of the rules are always adjusted during the process: in fact, they are updated every time an action has been executed. The mechanism of utility adjustment is a fundamental part of the learning process, and it consists of the following four steps.

1. In specific states (i.e. when a robot collides against a wall or when it reaches the goal), the winner rules receive a direct payoff P . This payoff can be of two types: reward ($P > 0$) and punishment ($P < 0$). The payoff is transmitted to all the previous winner rules with a discount rate $\gamma > 0$:

$$u_w^{(k)} \leftarrow u_w^{(k)} + \gamma^k P \quad k = 0, \dots, N$$

2. The current winner rule r_w hands over a part of its utility, Δu , to the previous one $r_w^{(1)}$:

$$u_w^{(1)} \leftarrow u_w^{(1)} + \Delta u$$

3. Whenever a definite rule triggers its action, its utility is reduced at the rate of c_f :

$$u_w = (1 - c_f)u_w$$

4. The utilities of all the rules are reduced according to the evaporation rate $\eta < 1$ when the robot reaches the goal state:

$$u_w = \eta u_w$$

A rule whose utility decreases below a threshold u_{min} is removed.

3.2. Prediction of state transition

In a MRS, if a robot selects the same action in the same situation, it may generate different results as its action's effectiveness depends on the other robots' actions. If each robot could precisely predict the state in the near-future, it wouldn't have to consider the action of the others as dynamic information and the environment could be treated as static. With this purpose, NNs are provided to each robot to predict the other robots' moves at the next time step in continuous space. Each NN is a three-layered, feed-forward, back-propagation model, and its output signals are used as the sensory information of the BRL mechanism.

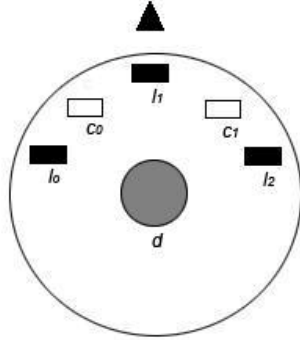


Fig. 1. Robot scheme and sensors location: $l_{1,2,3}$ = light sensor, $c_{0,1}$ = collision sensors, d = distance sensor.

4. Test Case Setting

The described approach was tested through a simulation of a MRS composed of 2 identical robots with the mission to reach a given goal, keeping the reciprocal distance constant. Each robot has two motors to rotate its two wheels to move around on a plane. Moreover, it is equipped with its own independent controller and several sensors to detect information from the environment (Fig. 1). In particular, it has two collision sensors, $c_{0,1}$, and three light sensors, $l_{1,2,3}$, at its front to perceive when it collides against the wall or it reaches the goal. In addition, it has a distance sensor, d , on its top to perceive the other robot's position and orientation, that represent a fundamental information for motion coordination. The distance sensors on the two robots are used to assure that the assigned reciprocal distance is respected or, otherwise, an action command is given to the robot to restore it. In other words, the 2 robots have to cooperate with each other to move around, but they do not employ a communication function in explicit form because they simply perceive each other through their distance sensors. Each robot can perceive the goal, in the form of a light source, only when it is within the range of the light sensors.

During the simulation, the 2 robots, starting from a given position, must reach the goal without hitting the walls. If a robot collides against a wall, it receives a punishment, whereas if it reaches the goal, both robots receive a reward. In this test, time is expressed through a *step* unit that comprises the process in which a robot gets the sensory inputs, decides and executes its action, and estimates the action. The experiments were conducted by *episodes*: an episode is updated when the robots reach the goal or after 200 steps in case they do not reach the goal. At each new episode the robots

initialize to the given starting position.

In this problem, the input to a robot that is conditioned by the partner robot is given by the angle measuring the orientation of the other robot, θ^p , as perceived through the distance sensor, therefore this is the value that the neural networks have to predict. The input and output of the prediction paradigm and the BRL mechanism are described in the following.

4.1. Prediction paradigm.

The *input* to the prediction mechanism is the angle history for three time steps. If θ^o is the angle of the robot and θ^p the angle of the partner, it is:

$$\{\cos(\theta^o(t-2)), \sin(\theta^o(t-2)), \cos(\theta^o(t-1)), \sin(\theta^o(t-1)), \cos(\theta^o(t)), \sin(\theta^o(t)), \cos(\theta^p(t-2)), \sin(\theta^p(t-2)), \cos(\theta^p(t-1)), \sin(\theta^p(t-1)), \cos(\theta^p(t)), \sin(\theta^p(t))\}$$

The *output* is the predicted value of the other robot's orientation, that is:

$$\{\cos \theta^{p'}(t+1), \sin \theta^{p'}(t+1)\}$$

4.2. Behaviour Learning Mechanism

The *input* to the BRL mechanism includes the robot current position and orientation, measured by the sensors, and the position of the other robot in the next future as predicted through NN processing, i.e.:

$$\{\cos(\theta^o(t)), \sin(\theta^o(t)), \cos(\theta^{p'}(t+1)), \sin(\theta^{p'}(t+1)), d_i(t), l_j(t)\}$$

The *output* is the rotation velocity of the robot's left and right wheels, namely:

$$\{m_l, m_r\}$$

A *reward* is given when the robot reaches the goal, i.e. when the following condition occurs:

$$l_j(t) > \theta_{goal}$$

A *punishment* is given when the robot collides against the wall, explicitly when:

$$d_i(t) > \theta_d$$

5. Results

During the simulation, the robots gradually learn the behaviour to reach the goal. In the earliest episodes, they cannot to achieve the mission within 200 steps, since the initially generated rules are not good enough.

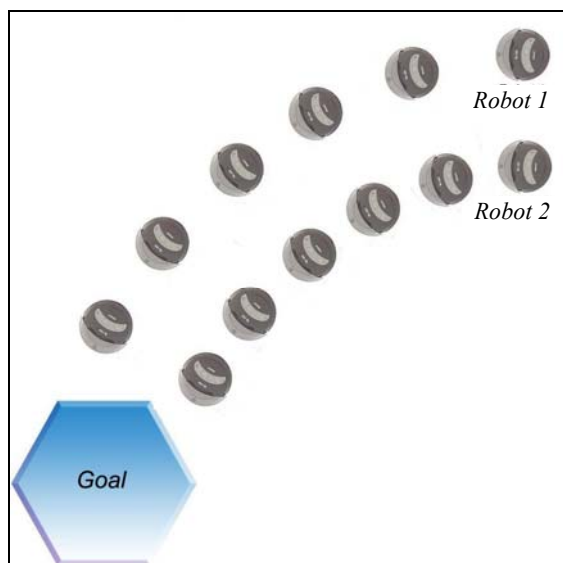


Fig. 2. System's behaviour during the 200th episode.

Proceeding with the learning process, new rules are generated and only the best ones are maintained. From the 150th episode on, the robots stably reach the goal in 200 steps, since they attained the right behaviour.

Fig. 2 shows the system's behaviour during the 200th episode. Depending on its experience, each robot gets specific rules and behaviour: in different tests, the same robot can take on the role of leader or follower. A leader robot changes rules more frequently to approach the goal, a follower robot simply moves to adjust the head direction.

6. Conclusions

A MAS-based approach using reinforcement learning was proposed to realize the acquisition of cooperative behaviour in multi-robot systems. The simulation showed that, starting from a homogeneous system, each robot acquires a different behaviour depending on experience and interaction with other robots and learns the way to achieve the assigned task.

Acknowledgements

This research work was carried out with support from the FP6 EC NoE on "I*PROMS" and the Italy-S. Africa MAE Project on "Design of Autonomous Multi-Robot Systems using Agent Technology and Reinforcement Learning".

References

- [1] Parker L.E. On the Design of Behavior-Based Multi-Robot Teams. *Ad. Robotics* (1996) Vol. 10, pp. 547-78.
- [2] Howard A., Parker L.E. and Sukhatme G.S. Experiments with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment and Detection. *Int. J. of Robotics Research* (2006) Vol. 25 pp. 431-448.
- [3] Fukuda T., Ueyama T. and Arai F. Control Strategy for a Network of Cellular Robots – Determination of a Master Cell for Cellular Robotic Network based on a Potential Energy. *ICRA '91 IEEE Int. Conf. on Robotics Automation*. Sacramento. Apr. 9-11 (1991) pp. 1616-21.
- [4] Fukuda T. and Kawauchi Y. The Cellular Robotic System (CEBOT): a Self-Organizing System. *Intelligent Robotic Systems*, ed. S.G. Tzafestas. M. Dekker, 1991.
- [5] Beni G. and Wang J. Theoretical Problems for the Realization of Distributed Robotic Systems. *ICRA '91 IEEE Int. Conf. on Robotics and Automation*. Sacramento. Apr. 9-11 (1991) pp. 1914-1920.
- [6] Antonelli G., Arrichiello F. and Chiaverini S. The Null-Space-based Behavioral Control for Autonomous Robotic Systems, *Journal of Intelligence Service Robotics* (2007), www.springerlink.com.
- [7] Feddema J.T., Lewis C. and Schoenwald D.A. Decentralized Control of Cooperative Robotic Vehicles: Theory and Application. *IEEE Trans. on Robotics & Automation* (2002) Vol. 18, pp. 852-864.
- [8] Stone P. and Veloso M. Multiagent Systems: A survey from a machine learning perspective. *Autonomous Robots*. (2000) Vol. 8 No. 3, pp. 345-383.
- [9] Ohkura K. and Ueda K. Autonomous Specialization: An Approach to Dynamic Role Allocation for a Multi-Robot System. *5th Int. Workshop on Emergent Synthesis – IWES '04*. Budapest. May 24-25 (2004) pp. 133-138.
- [10] Yasuda T., Ohkura K. and Ueda K. A Homogeneous Mobile Robot Team that is Fault-Tolerant. *5th Int. Workshop on Emergent Synthesis – IWES '04*. Budapest. May 24-25 (2004) pp. 139-146.
- [11] Flocchini P., Prencipe G., Santoro N. and Widmayer P., Hard Tasks for Weak Robots: the Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots. *10th Int. Symp. on Algorithm and Computation - ISAAC '99*. Chennai, India. Dec. 16-18 (1999) pp. 93-102.
- [12] Wang Z., Tianfield H. and Jiang P. A Framework for Coordination in Multi-Robot Systems. *IEEE Int. Conf. on Industrial Informatics – INDIN*. Banff, Alberta, Canada. Aug. 21-24 (2003) pp. 483-489.
- [13] Jennings N.R. and Wooldridge M.J. *Application of Intelligent Agents: Agent Technology - Foundations, Applications and Markets*, Springer-Verlag, 1998.
- [14] Sutton R.S. and Barto A.G. *Reinforcement Learning: an introduction*. MIT Press, 1998.
- [15] Duda R.O. and Hart P.E. *Pattern Classification and Scene Analysis*, John Wiley-Interscience, 1972.