Full paper

Emergent behaviors of a robot team performing cooperative tasks

E. PAGELLO[1,3], A. D'ANGELO[2,*], C. FERRARI[1], R. POLESEL[1],
R. ROSATI[1] and A. SPERANZON[4]

[1] *Intelligent Autonomous Systems Laboratory, Department of Electronics and Engineering, University of Padua, Padua, Italy*
[2] *Department of Mathematics and Computer Science, University of Udine, Udine, Italy*
[3] *LADSEB-CNR, Padua, Italy*
[4] *Division of Automatic Control, Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, Sweden*

Received 2 August 2001; accepted 18 October 2001

Abstract—We investigate the problem of how to make a multi-robot system performing a cooperative task by inducing a set of emergent actions. We model the environment dynamics by considering some parameters that express the ability of each robot to perform its task. Thus, the members of a group of robots become aware of their ability to realize some tasks by simply computing some quality function Q of the configuration pattern of the environment. A role assignment schema allows roles to be swapped among the robots of the group in order to select the best behaviors able to perform the task cooperatively. We illustrate this approach by showing how two soccer robots were able to exchange a ball, during a real game, by combining the use of efficient collision avoidance algorithms with role swapping triggered by the value of the above quality function Q.

Keywords: Multi-robot; emergent behavior engineering; RoboCup.

1. INTRODUCTION

In the last decade robotic applications have been moving from industrial to civil environments, including home and social entertainments. At the same time, the design and building of robots have switched the goals of controlled speed, high accuracy and repeatability toward new targets of flexibility, reliability and safety in human–machine interaction. Thus, the emergence of cooperative abilities is the key issue to successfully perform such kinds of advanced tasks.

The evaluation of the aptitude of a group of robots to work cooperatively should start by considering two alternative types of robot societies: the *differentiating* and

---

[*] To whom correspondence should be addressed. E-mail: antonio@dimi.uniud.it

the *integrating* robot societies [1]. The former show a large number of homogeneous individuals with limited abilities, whereas the latter refers to a small number of heterogeneous individuals with highly specialized skills. Both societies depend on individuals having well-defined peculiarities, such as role to play inside the group, the aptitude to modify dynamically their behavior while performing an assigned task, any kind of hierarchy eventually rearranging the group to cope with collective goals and the ability to settle down in open spaces.

For example, a group of robots could be characterized by some aspects like its size, composition and reconfigurability as well as its communication topology, availability and range [2]. However, a group of robots becomes a robotic team only if it shows some specialized aptitudes to perform collective tasks cooperatively. In some sense, a robotic team can be viewed as a group of robots which provides better performance than their individual components and carries out complex tasks taking advantage of the distributed sensing and acting capabilities.

Collective intelligence [3] can also emerge as a result of global behaviors involving two or more agents. Fault tolerance is usually supplied by agent redundancy, while group cohesion can be obtained by suitable robot formation motion-planning algorithms. Thus, an intelligent multi-robot system should evolve from a group of mobile robots that cooperate to solve a given complex task by allowing communication among individuals and dynamic group reconfigurability.

Soccer robots international games, like RoboCup [4], have been shown to be a very helpful field of experimentation to test the various approaches to these issues. The solution, designed at the IAS Laboratory of Padua University to coordinate our soccer robots, has been successfully adopted in the ART team in RoboCup 1999 [5], and RoboCup 2000 [6], and now in the new team Artisti Veneti [7].

In these games, very effective emergent cooperative ability was achieved by our soccer robots through the use of efficient collision avoidance algorithms activated while they were mutually exchanging their play roles. Basic behaviors, like *find_ball*, *go_to_ball* and *carry_ball*, were used along with a smart collision avoidance ability, and this was the reason why the collective emergent behaviors were forced into the two robots.

The rest of this paper deals with the problem of how to endow cooperative abilities into individual robots in a group by forcing collective actions. The organization of the paper includes a general discussion on multi-robot systems with some general accepted definitions covering our approach to emergent behavior engineering (Section 2). Then, in Section 3, we clarify what we mean with the term quality function Q and its use to allow the reconfiguration of a group of robots. The dynamic role assignment, triggered by the estimation of the environment configuration made by each individual component, is also discussed. Finally, in Section 4, we illustrate how some complex emergent cooperative abilities have been forced in the RoboCup domain, yielding a ball-exchange behavior as a result of a playing a real game in a competitive environment.

2. MULTI-ROBOT SYSTEMS

Robotic team design and implementation address a number of issues. For example, we should specify if each individual robot shares a common goal or not. The choice between distributed and centralized control implies that any decision is taken locally by individual robots or it is forced by some external authority. Nevertheless, the importance of communication among individuals inside a group cannot be ignored and, depending on whether *explicit* or *implicit*, or a combination of both, a distinctive character of the group can be observed.

In the next sections we shall differentiate between *explicit communication*, where signals are intentionally shared between two or more individual robots, and *implicit communication*, where the information is acquired by observing other robot actions. Thus, in Ref. [8] a cooperative ability without communication is achieved through the use of a BDI approach, whereas in Ref. [9] the same ability is obtained by using explicit communication.

Counter-intuitively, intelligent cooperation does not necessarily require explicit communication among robots, both because the amount of exchanged information can practically saturate the communication channel and because, from a theoretical point of view, there should exist some common sense knowledge about the task that all the robot would share. For example, in our preceding papers [10, 11] we exploited the case of forcing collective behaviors inside a group of agents through implicit communication.

There, the idea of perceptive patterns, recognizable by evaluating a set of scalar quantities, termed *macroparameters*, was introduced. Every agent inside the group was provided with a set of basic behaviors and, moreover, each behavior was defined with its *complementary* behaviour. Individual agent arbiters were able to activate complementary behaviors with respect to the observed one whenever the collective behavior of the group had to be enforced.

2.1. Behavior-based approach

Shaping robot agents includes both the design of physical components and the development of new software architectures. The former mainly concerns the introduction of innovative sensors and actuators as well as new arrangements of their physical structures. The latter should investigate the issues that arise from the integration of different software components able to support the *decide–sense–adapt behavior cycle*, i.e. the major activity of a robotics system.

The classical serial hierarchy of low level, intermediate level and high or decision level is not well suited for modeling the complex relations that result for on-line re-planning, especially if more robots are involved. On the contrary, a robotics system should be equipped with a set of behaviors that allow the system to act properly in the environment.

Starting from the pioneeristic work of Brooks [12], the *behavior-based approach* has become very popular to cope with several robotic applications, also including service robotics. Also termed *reactive control*, it refers to the direct coupling of perception to action as a specific technique which provides time-bound responses to robots moving in dynamic, unstructured and partially unknown environments.

A behavior is defined to be a control law for *achieving* and/or *maintaining* a particular goal. Usually, robot agents have multiple goals, including at least one achievement goal and one or more maintenance goals. This requires robot agents to be equipped with a number of behaviors, whose activation or inhibition must be triggered by a specialized module — the *arbiter*.

Depending on its sensor data and/or information coming from an external supervisor, it provides either spatial or temporal ordering of behaviors. The former causes the concurrent activation of a set of primitive reflexive behaviors, also referred to as static arbitration; the latter brings about a sequential activation of different sets of primitive reflexive behaviors, also referred to as dynamic arbitration.

A behavior-based approach assumes a robot to be *situated* in, and surrounded by, its environment. This means that a robot interacts with the world on its own, without any human intervention, i.e. its perspective is different from that of the observer. Moreover, since robots are not merely information processing systems, its *embodiment* requires that both all acquired information and all delivered effector commands must be transmitted through their physical structure.

Much of the behavior-based design has been inspired by different research areas such as ethology, biology, cognitive psychology and, especially, by looking at animal behavior as a model for robot control [13]. From this point of view all animals possess a set of innate behaviors, while complex behaviors can occur as a result of applying different basic behaviors. Animals typically respond only to a small subset of the total amount of sensory information available. They always live in a particular ecological niche, where their properties of autonomy can really work.

2.2. Communication and cooperation

Certainly, the principle of *cheap design* stems from looking at animal behavior, but the most important source of inspiration, while developing a group of interacting agents, is the understanding of the underlying mechanisms which are responsible for the emergence of collective behaviors inside animal societies.

The key feature which gives an account of agent societies is the *interaction* among individuals, defined as the *mutual influence on behavior*. Thus, *collective behavior* refers to *patterns of interaction between individuals of a group of agents detected by an external observer along with the time*.

Communication is the most common form of interaction which can result in either a *direct* or an *indirect* pattern. The former refers to a merely communicative act having the only purpose of transmitting information. The latter is based on the observed behavior of other agents and it is termed ‘*stigmergic*’ in the biological literature.

Cooperation is a form of interaction based on some form of communication. Thus, the distinction between collective and cooperative behavior is made on the basis of communication. If cooperative behaviors require negotiation between agents, then direct communication is also required.

In many cases, it could be useful to distinguish between *explicit* and *implicit* communication. The former refers to a kind of interaction which involves exchanging information or performing actions so that other agents can take advantage. This is analogous to the definition of cooperative behavior as it appears in [14]. The latter refers to the behaviors of an agent, whose effects could be useful for other agents achieving their own goals.

We can also understand cooperative behavior as a kind of collective behavior which has the property of increasing the total utility of the group, provided that some definition of utility is given. Collective behaviors can be also understood as emergent behaviors inside a group of individual agents, but the descriptive categories used to explain them are not those describing its constituent components [15].

While cooperating soley on a complex task, intelligent multi-robot systems must cope with *interference*, which *results in opposing or blocking an agent's behavior.* When we consider societies whose agents have identical goals, interference appears as *competition for shared resources*, whereas the situation is much more complicated if agents have different goals. In this case the type of conflicts includes *goal clobbering*, *deadlocks* and *oscillations* [16].

Interference covers *resource competition* as well as *goal completion*. The former stems from any interaction among individual agents competing for common resources such as object, space and information, and it requires *social rules* when the number of individual agents become too large. It can appear in both homogeneous and heterogeneous groups.

On the contrary, *goal competition* refers to a group consisting of individual agents with different goals, provided that they have compatible high-level goals. A possible reason for this situation stems from the heterogeneous functionality of agents.

2.3. Collective emergent behaviors

The implementation of a multi-robot system usually requires a careful design of the behaviors that any individual component should be equipped with. This is the first step in the emergence of a collective behavior, as a result of the actions provided by the individual agents, whose activities must be coordinated to cooperate and solve the global task. The same idea applies to a typical service robotics scenario [17], involving a multi-robot system.

In addition to the inherent complexity of the design and arrangement of multi-robot systems, the principal drawback is their high sensitivity to small and limited faults. Thus, individual robots should be simple in their basic architectures to be easily maintainable. This requires robots to be provided with intrinsic functionalities for motion and manipulation tasks.

Moreover, each robot can be equipped with different tools to add specialized functionalities and give rise to a kind of modular system. The modularity of the hardware components should also be reflected in the software components.

Usually multi-robot systems are designed to accomplish very complex tasks, that can be difficult and, even, completely unfeasible for a single robot system. The low sensitivity to any fault in its individual components stems from the intrinsic redundancy of the system. Changing the arrangement of its individual components at run-time results in performance improvement.

The emergence of cooperative behavior depends on a number of circumstances. First, we need to specify which tasks a group of individual robots can accomplish. Second, we must devise the basic skills any individual robot should be provided with. Of course, such individual behaviors have been already implemented during the phase of designing individual robots. Third, we need a mechanism to initialize the cooperative behavior, eventually considering the level of cooperative strategies the robots must follow to collectively solve given tasks.

Our implementation policy, to solve the problem of coordinating a group of individual robots to perform a cooperative task, evolves from our past experience in analogous situations [10]. Remember that cooperation requires communication which, in turn, can be supplied as an interaction. Thus, if we assume indirect communication, information can be exchanged as *mutual influences between the behaviors exerted by different individual robots*.

In Ref. [10] the notion of '*macroparameter*' has been introduced to monitor behavior influences. However, a cooperative behavior usually requires evaluation of two or more macroparameters so that a *performance index* could be more appropriate to directly *evalute the quality of the configuration* the actual task is exploiting. A further measure can be done over the environment to estimate other robots' work.

In the following section, we show how to use the constraints from the required task to evaluate such quality function and how to use the ability of planning collision-free paths to trigger the emergency of a collective behavior from the observation of the environment.

Then, in the next section, we show how to achieve an emergent cooperative ability through the use of efficient collision-avoidance algorithms activated by a couple of robots while mutually exchanging their play roles.

Several examples, taken from our experience in the RoboCup competitions during the past games of the middle size-league, within the Italian national ART team [5, 6], and within the Artisti Veneti team [7], show how a set of collective emergent behaviors can be easily induced into a team of real robots.

Thus, we illustrate the coordination abilities of a group of robots, related to the task of moving around a field, both without interfering with the other allied robots and by contrasting the opponents robots. We have also experimented with some other kind of cooperative abilities depending on the assigned task in order to test this approach for other possible tasks, like transporting objects by multi-robot systems.

3. DYNAMIC ROLE ASSIGNMENT

While building multi-robot systems, proper design of the behaviors for each individual component can result in a global cooperative emerging behavior, at the lowest architectural level, if the coordination ability among the basic behaviors is implemented in some suitable way. Our soccer robots were programmed by using a behavior-based approach [12] with the aim of obtaining robotics social organization [18].

Each robot must take into consideration a collective profit instead of individual profit only. Thus, particular emphasis must be put to the problem of coordinating the actions of each robot with the other members of the team. A set of different robot roles has been introduced, according to the definition of a role given in [19], by specifying a set of behaviors. An arbiter activates the basic behaviors according to the data received from the sensor module, based on a simple FSA. Each basic behavior was first realized as an expert in the real-time kernel Ethnos [20] and, then, as a thread in the ADE environment [7].

3.1. Function Q

A measure of quality Q, able to trigger the proprole was first introduced on 1999 at the IAS Laboratory of Padua University for evaluating how much work must be done by a robot to get the ball in the best position to score [21]. At any given time, the value taken by the Q_i, with reference to the ith robot, depends on the following quantities:

- Its distance from the ball.
- Its relative position with respect to a correct configuration to approach the ball.
- The last visible position of the ball, if the ball is not currently visible.
- The position of other robots if there are any toward the goal.
- The number of failure while it is trying to move around collision-free.
- Its previous role.

Each robot i computes independently Q_i based on its local estimation. The robot sends this value of Q_i to its team mates 10 times per second and decides autonomously how to behave comparing its own estimation of Q_i with the other value of Q_i. The design of Q takes into account some patterns in the environment, and results in a trade-off between the need to make the robots able to swap their roles dynamically and the requirement to hold the system in a sufficient stable state. The value of Q is computed summing up the partial values computed for each of the items appearing below.

Players may assume three different roles:

- *Master*, when the robot holds the ball, either as a defender or an attacker.
- *Active supporter*, when the robot cooperates with the *master*, avoiding interfering with it and protecting it from opponents.

- *Passive supporter*, when the robot is located far from the ball, but it is ready to enter the game.

This protocol guarantees fault tolerance and flexibility, and prevents ambiguity. The most interesting result, however, is the possibility of achieving an emergent cooperative behavior among the different roles assumed by each robot by introducing a suitable collision-avoidance algorithm.

Function Q was then used in Stockholm 1999, in Amsterdam 2000, and in Melbourne 2000, by the whole ART Team, to distribute among its team members a specific role set depending from its value communicated using Ethnos.

Bart and Homer were able to shows a cooperative action, like ball exchange, by coordinating their basic behaviors through the dynamic assignment of the above three roles realized by a set of behaviors that exploit some smart collision-free motion strategies, based on the computation of field vectors. An obstacle avoidance module implemented these motion-planning algorithms as an Ethnos expert.

A generalization of the function Q has been introduced in [22], as a set of utility functions, able to give some utility values that indicate the usefulness of each role through an explicit communication. Thus, ART robots may decide to distribute among its team members a specific role set depending on several values assumed by a set of functions transmitted among the members.

The same approach was used by our new heterogeneous team Artisti Veneti [7] where robots were again able to realize ball exchange actions during RoboCup 2001 in Seattle.

4. EMERGENT COLLECTIVE BEHAVIORS WITHIN COMPETITIVE ENVIRONMENTS

Since each ART member was allowed to specialize the basic behaviors for the robots of their local team, at IAS Laboratory we developed an original design of the behaviors for our robots, Bart and Homer. They played with ART and also with our team Artisti Veneti.

4.1. Field vector-based collision avoidance

The actual implementation of our approach is based on a collision-avoidance algorithm which makes use of field vectors to generate schema-based behaviors [23].

Both *target* (the attractor) and *obstacles* (the repulsors) generate their own specific vectors. The target generates a purely attractive field, proportional to the distance, while the obstacles generate a rotational field. The direction along the rotational vector is positive or negative according to the shortest path toward the target. The computed sum of all environmental vectors is updated every 0.1 s.

All active behaviors send the coordinates of a target to the Obstacle Avoidance ExperT (OAET). The attractive field vector is bound to the maximum robot velocity.

This gives a well formed acceleration to robot motion. Then, OAET realizes the motion toward the target according to the corrections caused by obstacles on the attractive field vectors. Each obstacle is delimited by a circular area, centered on it and denoting its *Affected_area* (OA), whose radius is taken equal to 100 cm.

In the same way, the *Robot_affected area* (RA) is also computed, but in this case the robot dimension is summed up with its current speed, to take into account the robot motion. So, we have:

$$\begin{aligned} OA\_radius &= 100\text{ cm} \\ RA\_radius &= V\_robot + Dim\_robot \end{aligned}$$

as it appears in Fig. 1.

The rotational field is computed accordingly to Coulomb Laws:

$$I = \frac{K * H}{r^2},$$

where K is a constant, whereas H is a quantity whose value depends on the distance from the obstacle. Formally, they are defined as follow

$$K \gg 1,$$

$$H = \begin{cases} 0, & \text{when } d \leqslant D_1 \\ \dfrac{1}{d}, & \text{when } D_1 < d \leqslant D_2 \\ 1, & \text{when } d > D_2, \end{cases}$$

with d a parameter denoting the distance from the obstacle, whereas D_1 and D_2 are two empirical quantities which take the values 15 and 55 cm, respectively.

That is, if the robot is too close to an obstacle, then it must be forced to do a back-step since the game rules penalizes a robot if it collides with other robots.

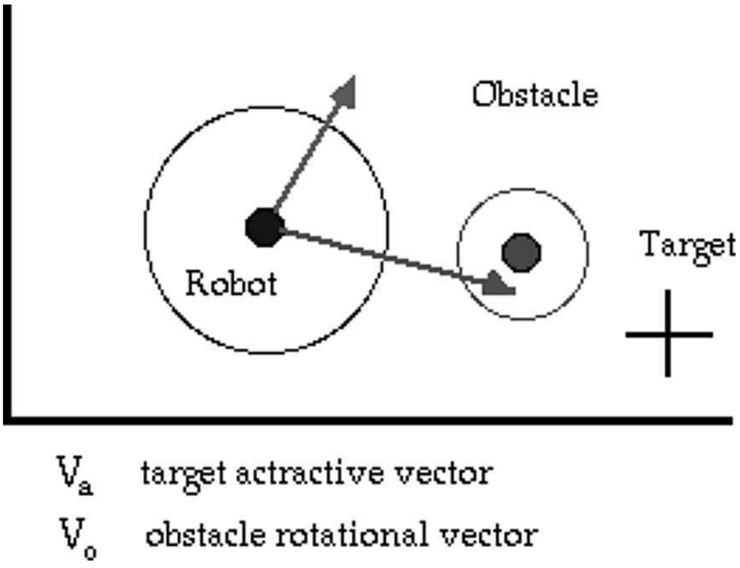


Figure 1. Obstacle and robot affected areas.

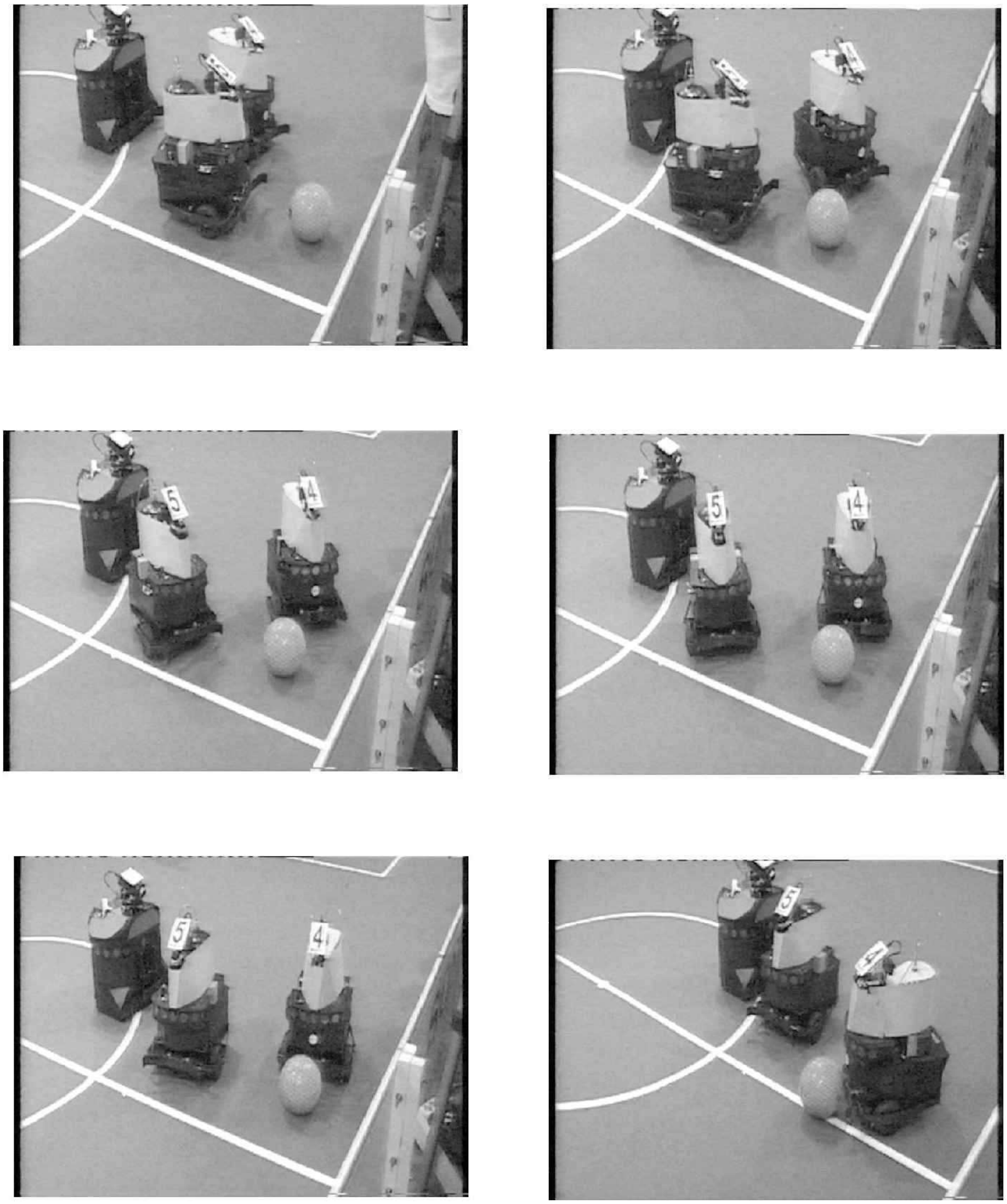


Figure 2.

Then, the obstacle's field vector becomes repulsive instead of rotational, if the distance between the robot and obstacle is $d \leqslant D_2$.

In RoboCup games, usually only two or three robots play nearby (Fig. 2). Thus, sometimes two alternative paths can be equally selected to get the target, as in Fig. 3, but only one is safe due to other constraints, as in Fig. 4.

A decision is achieved by blocking a direction on one rotational vector in order to force all other vector directions to follow.

However, some delicate configuration may happen. The most frequent is the one depicted in Fig. 3, where the robot must decide if it is more convenient, or even

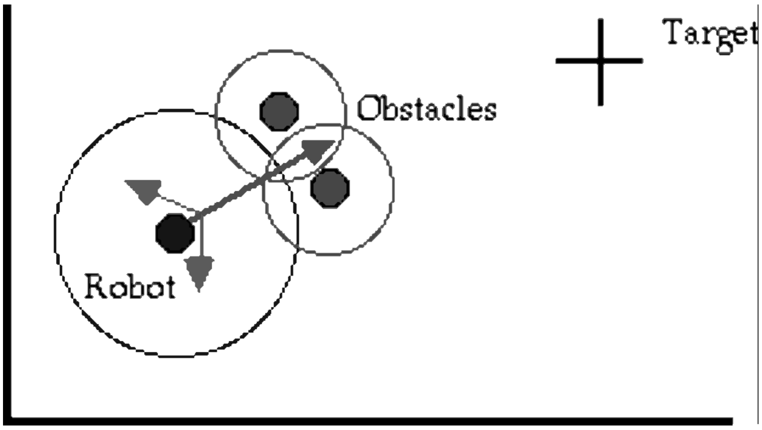


Figure 3.

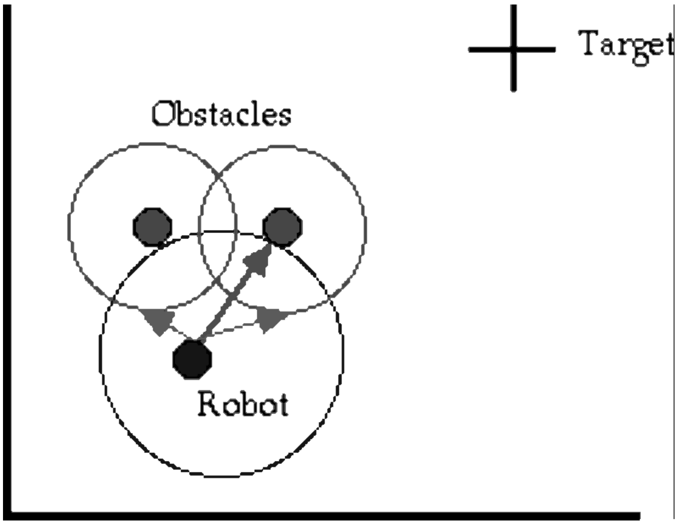


Figure 4.

possible (e.g. sliding along a wall), to turn around the obstacle, instead of passing through, and which direction to select for this turnabout. In this case, the presence of the wall may suggest to turn right toward the free area by blocking the direction of one obstacle vector and forcing the other to follow the same choice.

However, we have also considered walls as particular kinds of obstacles. To avoid collisions of the robot against walls, the component of the robot velocity orthogonal to the wall itself becomes null, so that the resultant robot trajectory becomes parallel to the wall itself. Let us consider, for example, Fig. 5, where the component of the obstacle rotational vector oriented toward the wall is made null by the component of the wall repulsive vector, so that the robot acquires a motion parallel to the wall.

A serious problem may arise if both of two opposite directions are blocked due to some difficult configuration like the one of Fig. 6 (and its symmetric configuration where the target and the robot are swapped). Due to the environmental constraints, we decided the robot does not move for a while, waiting the opponents' move or a game restart by the arbiter.

To enhance role swapping robustness and avoid system instability, Q was made sensitive to the previous role played by robot R_i. Thus, if robot R_i does not play as a *master* in the previous move, its actual value of Q is penalized in order to make

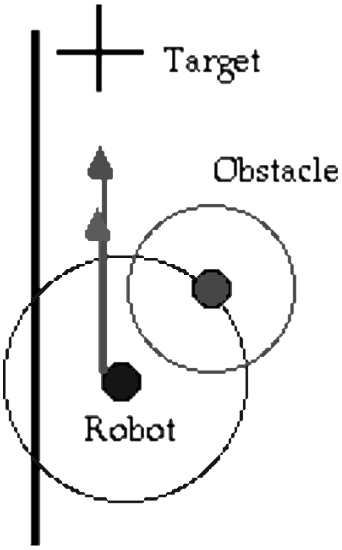


Figure 5.

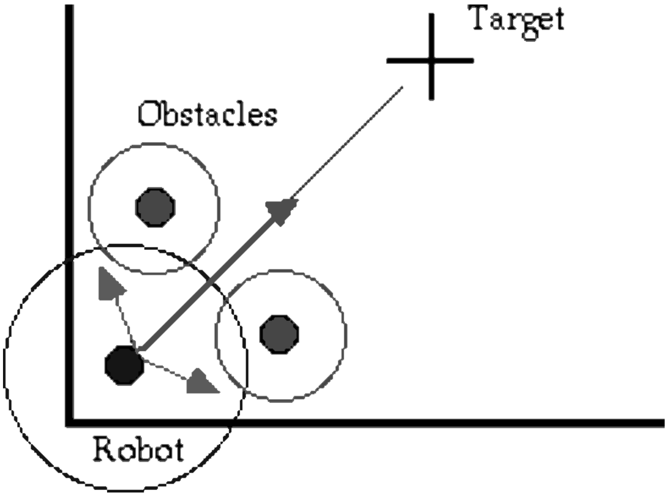


Figure 6.

it more difficult to move from an *active supporter* to a *master* in order to avoid oscillation. Of course, it must not be penalized too much to avoid inertia becoming too large.

To enforce the effectiveness of the *active supporter*, the implementation at IAS Laboratory has been realized as follows. More specifically, when a robot is playing as an *active supporter*, the following statements are always true.

- It must never interfere with the robot playing as the *master*.
- It must quickly try to take the ball if the *master* fails to perform its task.
- It must keep itself close to the *master* to eventually recover the ball if the *master* looses it.
- It must avoid any position on the straight line connecting the *master* with the opponent's goal.

4.2. Exchanging the ball by combining collision avoidance and robot role swapping

To reinforce the robustness of robot role swapping, robot navigation algorithms were based on a different evaluation of the attractive and repulsive potential fields of a robot that plays as an *active supporter* versus the robot which plays as a *master*. The former is much more influenced by obstacles than the latter.

Then, an *active supporter* also moves directly toward the ball, but it does not restrain the *master* robot, since it is affected by a stronger repulsive force than the *master* robot.

Thus, if the *active supporter* meets the *master* robot when it is moving to the ball, it handles the *master* robot as an obstacle and its repulsive force prevents it from interfering with the *master*'s action. If, for any reason, the *active supporter* does not meet the *master* robot along its path to the ball, because the *master* robot was faced with some unexpected difficulties while performing its task, then the *active supporter* is able to become the *master*.

When the *master* meets an opponent, while keeping the ball, often it makes a back-step, to avoid collision, while the *active supporter* succeeds to move to a better approach position to the ball. Thus, the *master* robot makes room to its *active supporter* that may take the ball, because it comes to be in a better position to score and thus swaps its role.

If the *master* robot keeps the ball while it is far from the opponent's goal, but the *active supporter* is in a better scoring position, although far from the ball, the value of Q computed by the *master* robot becomes lower than the value of Q computed by the *active supporter*, because the *active supporter* is able to develop some *limited competitive behavior* against the *master*. Thus, the two robots are able to swap their roles, with ball exchanging, as depicted in Fig. 8.

There, we can see that the *master* robot keeps the ball but, since the *active supporter* is in a better scoring position but far from the ball, the value of Q computed by the *master* becomes lower than the that computed by the *active supporter*. Thus, the two robots swap their roles and they *succeed in exchanging the ball*, showing an emergent behavior. This configuration really happened several times when Bart and Homer had the opportunity to play together with the ART team and within the heterogeneous team Artisti Veneti.

We have verified it (and video recorded in www.dei.unipd.it/∼robocup) several times, for example, in Stockholm (August 1999) during the quarter-finals of ART against the University of Ulm, as shown in Fig. 8, where Bart and Homer played together, as well as in Seattle (August 2001) during the preliminary game of Artisti Veneti against the University of Friburg, as shown in Figs 2 and 7, where Bart and a Golem Robot, borrowed from the Golem Team [24], played together.

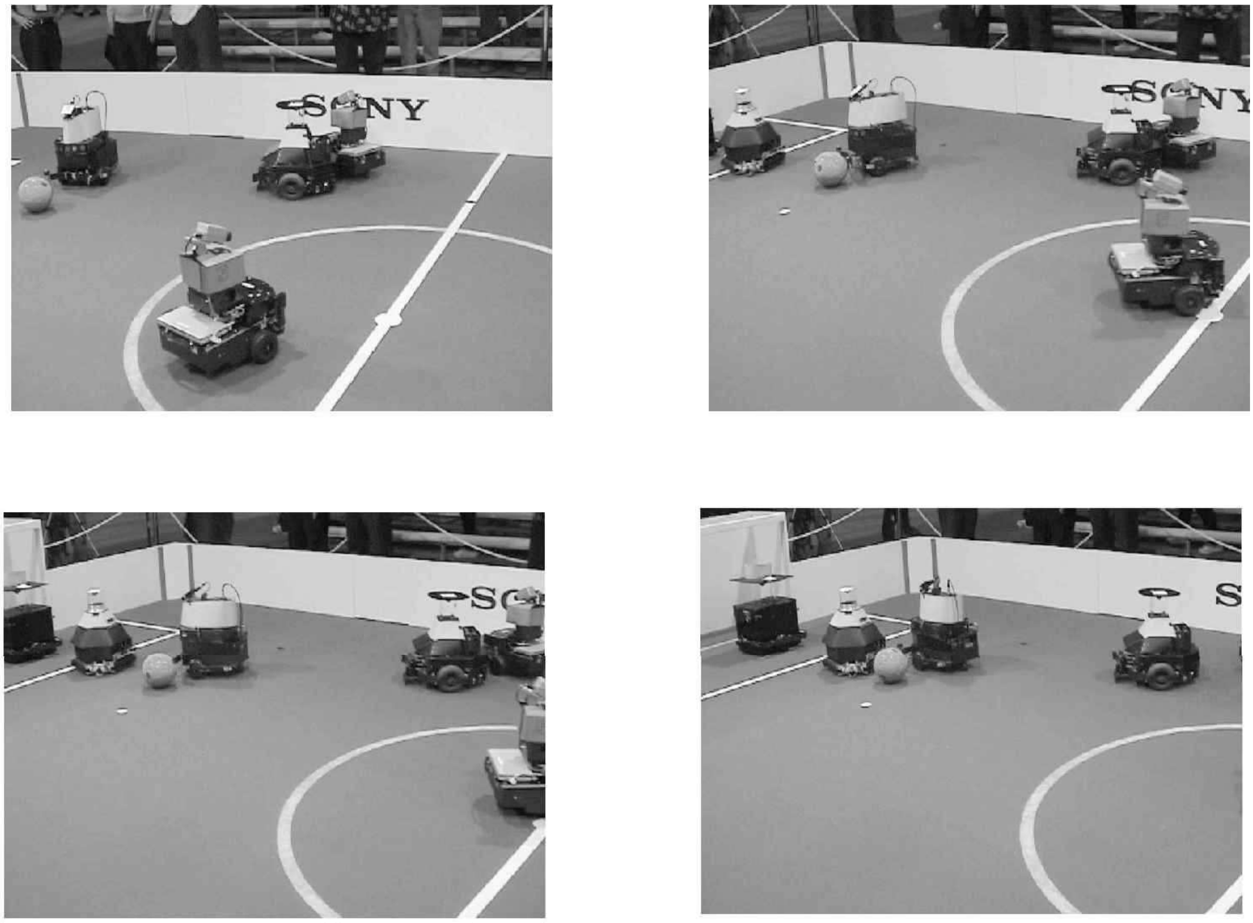


Figure 7.

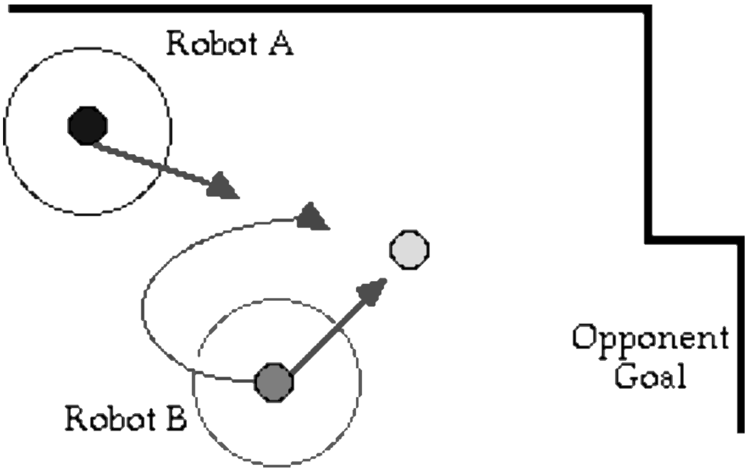


Figure 8.

5. CONCLUSIONS

We have illustrated our research project aimed at using an approach based on emergent behaviors engineering for designing multi-robot systems by testing its performance in the research field of edutainment robotics. We discussed the problem of how to give autonomy to each single individual robot in the group by introducing some suitable basic behaviors based on efficient collision-avoidance algorithms. Thus, we showed how to achieve an emergent cooperative ability by moving around, without conflicting with other potentially cooperative robots, realizing a mutual exchange of the roles played by the robots. We have illustrated our approach through some examples of these emergent collective actions performed by our real robots, Bart and Homer, developed at the IAS Laboratory of the University of Padua, that played successfully in past RoboCup competitions.

Acknowledgements

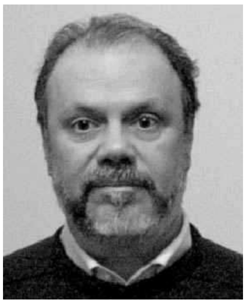
This research was supported partially by MURST (Certamen Project), by CNR (Special Project) and by the ENEA (Parallel Computing Project). We acknowledge the collaboration to the Research Project on RoboCup at the IAS Laboratory by M. Barbon, M. Bert, P. De Pascalis, M. Ferraresso, P. Filippin, F. Garelli, M. Lorenzetti, F. Montesello, C. Moroni, A. Modolo, P. Patuelli, C. Pellizzari, M. Peluso, N. Scattolin, D. Spagnoli, A. Vaglio, S. Zaffalon and W. Zanette, all students at the Enginering School of Padua University. We also acknowledge the members of ART, in cooperation with which we have developed and tested the dynamic role assignment, in particular G. Adorni, A. Bonarini, D. Nardi and M. Piaggio.

REFERENCES

1. L. Parker, From social animals to teams of cooperating robots, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Workshop on Multi-robot Cooperation: Current Trends and Key Issues, Grenoble (1997).
2. G. Dudek, M. Jenkin, E. Milios and D. Wilkes, A taxonomy for swarm robotics, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robot and Systems*, pp. 441–447 (1993).
3. D. Kurabayashi, Toward realization of collective intelligence and emergent robotics, in: *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, Tokyo, pp. 748–753 (1999).
4. M. Asada, A. Birk, E. Pagello, M. Fujita, I. Noda, S. Tadokoro, D. Duhaut, P. Stone, M. Veloso, T. Balch, H. Kitano and B. Thomas, Progress in robocup soccer research in 2000, in: *Proc. Int. Symp. on Experimental Robotics*, Honolulu (2000).
5. D. Nardi, G. Adorni, A. Bonarini, A. Chella, G. Clemente, E. Pagello and M. Piaggio, Art99 azzurra robot team, in: *RoboCup 1999: Robot Soccer World Cup III*, M. Veloso, E. Pagello and H. Kitano (Eds), Lecture Notes in Artificial Intelligence, Vol. 1856, pp. 695–698. Springer, Berlin (2000).
6. G. Adorni, A. Bonarini, G. Clemente, D. Nardi, E. Pagello and M. Piaggio, Art'00 — azzurra robot team for the year 2000, in: *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone, T. Balch and G. Kraetszchmar (Eds), Lecture Notes in Artificial Intelligence, Vol. 2019, pp. 559–562. Springer, Berlin (2001).
7. E. Pagello, M. Bert, M. Barbon, E. Menegatti, C. Moroni, C. Pellizzari, D. Spagnoli and M. Zaffalon, Artisti veneti: a new entry in 2001 middle-size league, in: *RoboCup 2001: Robot Soccer World Cup V*, A. Birk, S. Coradeschi and S. Tadokoro (Eds), Lecture Notes in Artificial Intelligence. Springer, Berlin (in press).
8. M. Hannebauer, J. Wendler, P. Gugenberger and H. D. Burkhard, Emergent cooperation in a virtual soccer environment, in: *Proc. DARS 3*, T. Lueth *et al.* (Eds), pp. 341–350. Springer, Berlin (1998).
9. K. Yokota, K. Ozaki, A. Matsumoto, K. Kawabata, H. Kaetsu and H. Asama, Modeling environment and tasks for cooperative team play, in: *Proc. DARS 3*, T. Lueth *et al.* (Eds), pp. 361–370. Springer, Berlin (1998).

10. E. Pagello, A. D'Angelo, F. Montesello, F. Garelli and C. Ferrari, Cooperative behaviors in multi-robot systems through implicit communication, *Robotics Autonomous Syst.* **29** (1), 65–77 (1999).
11. E. Pagello, C. Ferrari, A. D'Angelo and F. Montesello, Intelligent multirobot systems performing cooperative tasks, in: *Proc. Special Session on Emergent Systems — Challenge for New System Paradigm*, Tokyo, pp. 754–760 (1999).
12. R. Brooks, A robust layered control system for a mobile robot, *IEEE J. Robotics Automat.* **2**, 14–23 (1986).
13. T. Anderson and M. Donath, Animal behavior as a paradigm for developing robot autonomy, *Robotics Autonomous Syst.* **6**, 145–168 (1990).
14. Y. U. Cao, Cooperative mobile robotics: Antecendents and directions, *Autonomous Robots, Special Issues on Robot Colonies* **4** (1), March (1997).
15. L. Steels, The artificial life roots of artificial intelligence, Technical report, Artificial Intelligence Laboratory, Vrije Universiteit Brussel (1993).
16. M. Mataric, Learning social behavior, *Robotics Autonomous Syst.* **20**, 191–204 (1997).
17. T. Arai and J. Ota, Let-us work together — task planning of multiple mobile robots, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robot and Systems*, Pittsburgh, PA, pp. 298–303 (1995).
18. L. E. Parker, Alliance: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Munich, pp. 776–783 (1994).
19. P. Stone and M. Veloso, Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork, *Artificial Intelligence* **110**, 241–273 (1999).
20. M. Piaggio, A. Sgorbissa and R. Zaccaria, Ethnos: a light architecture for real-time mobile robotics, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Kyonju (1999).
21. M. Ferraresso, C. Ferrari, E. Pagello, R. Polesel, R. Rosati, A. Speranzon and W. Zanette, Collaborative emergent actions between real soccer robots, in: *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone, T. Balch and G. Kraetzschmar (Eds), Lecture Notes in Artificial Intelligence, pp. 297–302. Springer, Berlin (2001).
22. C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo and A. Sgorbissa, Communication and coordination among heterogeneous mid-size players: Art99, in: *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone, T. Balch and G. Kraetzschmar (Eds), Lecture Notes in Artificial Intelligence, pp. 86–95. Springer, Berlin (2001).
23. R. C. Arkin, *Behavior-Based Robotics*. MIT Press, Cambridge, MA (1998).
24. P. de Pascalis, M. Ferraresso, M. Lorenzetti, A. Modolo, M. Peluso, R. Polesel, R. Rosati, N. Scattolin, A. Speranzon and W. Zanette, Golem team in middle-sized robots league, in: *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone, T. Balch and G. Kraetzschmar (Eds), Lecture Notes in Artificial Intelligence, pp. 603–606. Springer, Berlin (2001).

ABOUT THE AUTHORS



Enrico Pagello received the Laurea in Electronic Engineering from the University of Padua in 1973. From 1973 to 1983, he was a Research Associate at the Institute on System Science and Biomedical Engineering of the National Research Council of Italy, where now he is a part-time collaborator. Since 1983 he has been an Associate Professor of Computer Science at the Department of Electronics and Informatics of the University of Padua. During 1977/1978, he was a Visiting Scholar at the Laboratory of Artificial Intelligence of Stanford University. He has regularly visited the Department of Precision Engineering of the University of Tokyo, since 1994, in the frame of a joint scientific agreement between Padua and Tokyo Universities. He was the General Chair of the Sixth Internatiol Conference on Intelligent Autonomous Systems in

July 2000 and a member of Editorial Board of IEEE *Transaction on Robotics and Automation.* He is a Vice-president of the RoboCup International Federation, and has been appointed as a General Chairman of RoboCup 2003, that will be held in Padua on July 2003. His current research interests are on applying artificial intelligence to robotics with particular regard to the multi-robot systems area.

Antonio D'Angelo is an Assistant Professor of Computer Science at the University of Udine. He received a MS in Electrical Engineering from Padua University in 1981 and since 1984 he has been working at the Laboratory of Artificial Intelligence and Robotics at the Department of Mathematics and Computer Science at the University of Udine. His current research covers multi-agent autonomous system coordination, behaviour-based planning and control including complex system models for autonomous robots.

Carlo Ferrari received the Laurea in Electronics Engineering from the University of Genua in 1985, and the Doctoral degree in Computer Engineering and Industrial Electronics from the University of Padua in 1992. He visited the University of California at Berkeley from 1990 to 1991. He has been a Research Associate at the Depertment of Electronics and Informatics of the University of Padua since 1992, and an Associate Professor of Computer Science since 2002.

Roberto Polesel is a student at the Undergraduate Division of the School of Engineering of the University of Padua. He has been a member of both the Golem and ART soccer robot teams that participated in the RoboCup Competitions. He is progressing towards his degree in Biomedical Engineering.

Robert Rosati is a student at the Undergraduate Division at the School of Engineering, University of Padua. He participated in research projects on multi-robot systems at the Department of Electronics and Informatics. He has been a member of the Golem team which ranked second at RoboCup 2000 Middle-size International Competitions and formerly a member of the ART Team, ranked second at RoboCup 1999, Stockholm. He is progressing towards his degree in Computer Engineering. His current research interests include artificial intelligence and bioinformatics.

Alberto Speranzon received his Laurea degree in Computer Engineering from the University of Padua in 2000. In the same year he joined the Automatic Control Group at the Signals, Sensors and Systems Department, Royal Institute of Technology, Stockholm, Sweden, as a PhD student. He has been member of the ART and Golem RoboCup teams, both ranked second at the RoboCup World Championships in 1999 and 2000, respectively. His current research interests include control systems with communication constraints, non-linear control and hybrid systems.