Anticipation: A Key for Collaboration in a Team of Agents*

Manuela Veloso, Peter Stone, and Michael Bowling
Computer Science Department
Carnegie Mellon University
Pittsburgh PA 15213

Submitted to the 3rd International Conference on Autonomous Agents, October 1998

Abstract

We investigate *teams* of complete autonomous agents that can collaborate towards achieving precise objectives in an adversarial dynamic environment. We have pursued this work in the context of robotic soccer both in simulation and with real physical robots. We briefly present these two frameworks emphasizing their different technical challenges. Creating *effective* members of a team is a challenging research problem. We first address this issue by introducing a team architecture organization which allows for a rich task decomposition between team members. The main contribution of this paper is our recent introduction of an action selection algorithm that allows for a teammate to *anticipate* the needs of other teammates. Anticipation is critical for maximizing the probability of successful collaboration in teams of agents. We present our team organization architecture and the anticipation algorithm. We show how our contribution applies to the two concrete robotic soccer frameworks. Anticipation was used in both our CMUnited-98 simulator and CMUnited-98 small-robot teams in the RoboCup-98 competition held jointly with ICMAS in July 1998. The two teams are RoboCup world champions each in its own league. Anticipation was one of major differences between our team and the other teams.

keywords: multi-agent coordination and collaboration, multi-agent teams, autonomous robots, coordinating perception, thought, and action

*This research is sponsored in part by the Defense Advanced Research Projects Agency (DARPA), and Rome Laboratory, Air Force Materiel Command, USAF, under agreement numbers F30602-97-2-0250 and F30602-98-2-0135. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory (AFRL) or the U.S. Government.

1 Introduction

We have been pursuing research in the development of *teams* of autonomous agents that need to act in adversarial environments. In these domains, single agents cannot achieve the overall team goals individually. Goal achievement necessarily requires the collaboration between the members of the team. We have used three different testbeds in the robotic soccer domain to pursue this investigation: a rich simulation environment using the RoboCup soccer server [18], our own-built small wheeled robots [19], and Sony's fully autonomous legged robots [22]. We participated at the RoboCup-98 competitions at ICMAS in Paris, and we came in first place in each of these three leagues.

Although the three platforms are in the domain of robotic soccer, the technical challenges presented by each one for building effective multi-agent teams are quite different. In particular, in both the simulator and the small-sized robots we have been capable of developing robust teamwork approaches. Instead, with the Sony autonomous legged robots, we have so far concentrated primarily on an automated color calibration algorithm and probabilistic localization to allow individual agents to perceive the surrounding world effectively [22]. Teamwork is still minimal in this platform. In this paper, therefore we focus on the team organization and teamwork of our CMUnited-98 simulation and small-robot teams.

One main focus of our research is on algorithms for collaboration between agents in a team. An agent, as a member of a team, needs to be capable of individual autonomous decisions while, at the same time, its decisions must contribute towards the team goals.

We introduce a flexible team architecture in which agents are organized in *formations* and *units*. Each agent plays a *role* in a unit and in a formation. In many multi-agent systems, one or a few agents are assigned, or assign themselves, the specific task to be solved at a particular moment. We view these agents as the *active* agents. Other team members are *passive* waiting to be needed to achieve some task. Concretely, in the robotic soccer domain, we view the agent that goes to the ball as the active agent, while the other teammates are passive. While the active agent has a clear task assigned and therefore a clear plan to follow (e.g. move towards the ball), it is less clear what is the plan for the passive agents. As the team agents most probably will need to collaborate, it seemed to us that passive agents could not simply be "passive."

Our initial team architecture allowed for the passive agents to flexibly vary their positions within their role only as a function of the position of the ball. In so doing, their goal was to *anticipate* where they would be most likely to find the ball in the near future. In our CMUnited-97 teams, both simulation and real robots, we effectively used this ball-dependent role-adjustment strategy. This is a first-level of single-agent anticipation towards a better individual goal achievement.

However we recently investigated a more elaborate team behavior for the passive agents. For this year's CMUnited-98 teams, we introduced a team-based notion of *anticipation*, which goes beyond individual single-agent anticipation. The passive team agents position themselves strategically so as to optimize the chances that their teammates can successfully collaborate with them, in particular pass to them. By considering the positions of other agents and the attacking goal, in addition to only that of the ball, they are able to position themselves more usefully: they *anticipate* their future contributions to the team. This strategic anticipation is the main contribution of this paper. We believe that this new team collaboration algorithm was one of the main improvements from our last year's CMUnited-97 robot champion team [21, 20].

The paper is organized as follows. Section 2 describes the simulation and the robotic soccer frameworks. Section 3 describes our base initial team architecture contributing a flexible role-based team organization. Section 4 contributes the anticipation algorithm as a key behavior for the success of team of agents, and reports on the results obtained at RoboCup-98. Section 5 concludes the paper.

2 Simulation and Real Robotic Soccer

Robotic soccer is a very good domain for studying real-time multi-agent coordination techniques: agents must act quickly and autonomously while contributing to the achievement of the team's overall goal [4, 7]. Robotic soccer systems have been recently developed both in simulation [9, 14, 15] and with real robots [1, 3, 12, 13, 20]. The research presented in this paper was developed jointly in simulation and on our real robot team.

2.1 Simulator

The RoboCup soccer server [10] has been used as the basis for successful international competitions [11] and research challenges [5]. Though not directly based upon any single robotic system, the soccer server, as pictured in Figure 1, captures several real-world complexities:

- the players' vision is limited;
- the players can communicate by posting to a blackboard that is visible to all players;
- all players are controlled by separate processes;
- each team has 11 members;
- each player has limited stamina;
- actions and sensors are noisy; and
- play occurs in *real time*: the agents must react to their sensory inputs at roughly the same speed as human or robotic soccer players.

The simulator, acting as a server, provides a domain and supports users who wish to build their own agents (clients).

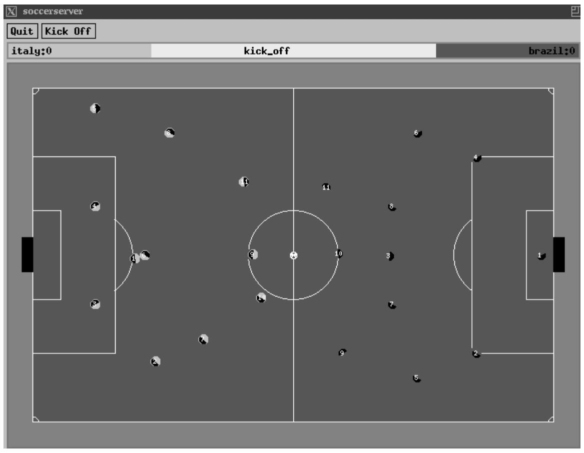


Figure 1: The Soccer Server system

By abstracting away the low-level perception and action complexities inherent in robotics, the simulator allows researchers to focus quickly on the multi-agent coordination issues. In this regard, the fact that the simulator enforces a completely distributed approach (each player must be controlled by a separate program) is a crucial feature.

Perception in the simulator is distributed. Each agent sees a portion of the world depending on the direction it is facing. All of the information it receives is in polar coordinates relative to its own position (instead of global Cartesian coordinates). Objects that are farther away are seen with less precision. Actions available to the clients are parameterized movement commands (turn/dash/kick) as well as a communicate "say" command. The effects of all actions are non-deterministic.

2.2 Real Robots

Our small-size robot team is a fully autonomous system consisting of a global perception system and decision-making individual clients. The team is made up of five robots that we have built [19]. Figure 2.2 shows our CMUnited-98 robots.

Figure 2: CMUnited-98 Small Robot Team: RoboCup-98 World Champions

The decision-making clients select actions based on the perceived world state and send these commands to the physical robots through radio communication. Perception is accomplished through a camera over-looking the playing field. The full view of the world is processed by our vision algorithm to find and track the position and orientation of the team's agents, the position of the opponents, and the position and trajectory of the ball [2]. This processing provides a global view of the environment that is shared by all of the agents.

Each physical robot is controlled by a separate client program that makes decisions using the information obtained from the vision system. Each agent is assigned a role and therefore behaves differently. There are three roles used in our system: goal-tender, defender, and attacker. Even with this partition of agents, it is still not desirable for all of the agents to be "actively" filling their roles. For example, it is rarely successful for all of the attackers to chase the ball since they will often be hindering each other's progress. Multiple attackers need therefore to coordinate. Each attacking agent uses the perceived world to calculate the value of its own and its teammates' possible actions (e.g. shooting or passing to a teammate). The agent with the highest valued action becomes active and attempts to achieve this action. The other agents become passive. These agents then act in a way to anticipate where they may be useful for future collaboration with the active agent.

3 Team Organization Architecture

Our base teamwork structure is situated within a team member architecture suitable for real-time multi-agent domains in which individual agents can cooperate with teammates towards common goals while still acting autonomously. Based on a standard agent paradigm, our team member architecture allows agents to sense the environment, to reason about and select their actions, and to act in the real world. At team synchronization opportunities, the team also makes a "locker-room agreement" for use by all agents during periods of low communication. The locker-room agreement specifies team conventions and collaboration protocols, alleviating the need for negotiation during time-pressured situations.

The agent keeps track of three different types of state: the *world state*, the *locker-room agreement*, and the *internal state*. The agent also has two different types of behaviors: *internal behaviors* and *external behaviors*.

Internal and external behaviors are similar in structure, as they are both sets of condition/action pairs where conditions are logical expressions over the inputs and actions are themselves behaviors as illustrated in Figure 3. In both cases, a behavior is a directed acyclic graph (DAG) of arbitrary depth. The leaves of the DAGs are the behavior types' respective outputs: internal state changes for internal behaviors and action primitives for external behaviors.

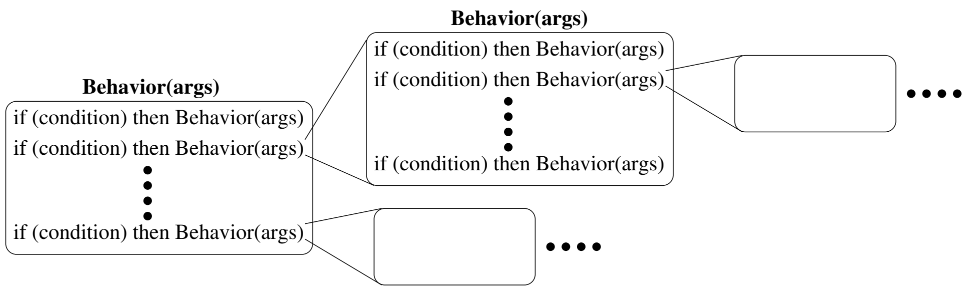


Figure 3: Internal and external behaviors are organized in a directed acyclic graph.

Our notion of behavior is consistent with that laid out in [8]. In particular, behaviors can be nested at different levels: selection among lower-level behaviors can be considered a higher-level behavior, with the overall agent behavior considered a single "do-the-task" behavior. There is one such *top-level* internal behavior and one top-level external behavior; they are called when it is time to update the internal state or act in the world, respectively. We now introduce the team structure that builds upon this team member architecture.

3.1 Formations and Roles

Figure 4 shows a sample top-level external behavior used by a team agent. The agent's top priority is to locate the ball. If the ball's location is known, it moves towards the ball or goes to its position (i.e., to assume its role), depending on its internal state. It also responds to any requested communications from teammates. It is when it goes to its position that it makes use of the anticipation mechanism presented here.

External Behavior: Play Soccer()

If (Ball Lost)	Find Ball()
If (Ball known AND Chasing)	Handle Ball(args1)
If (Ball known AND Not Chasing)	Play Position(args2)
If (Commuincate Flag Set)	Communicate()

Figure 4: An example of a top-level external behavior for a robotic soccer player.

The referenced "Handle Ball" and "Play Position" behaviors may be affected by the agent's current role and/or formation. Such effects are realized by references to the internal state either at the level of function arguments (args1, args2), or within sub-behaviors. None of the actions in the condition-action pairs here are action primitives; rather, they are calls to lower level behaviors.

The definition of a position includes *home coordinates*, a *home range*, and a *maximum range*, as illustrated in Figure 5.

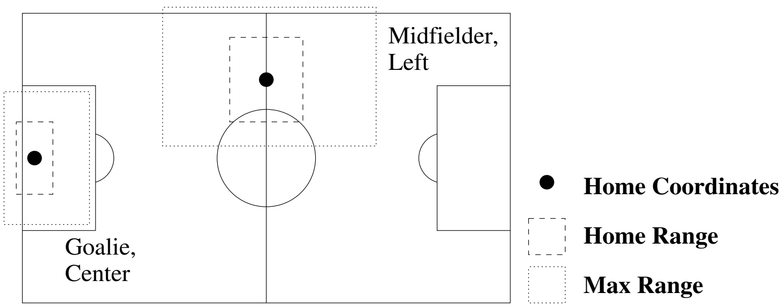
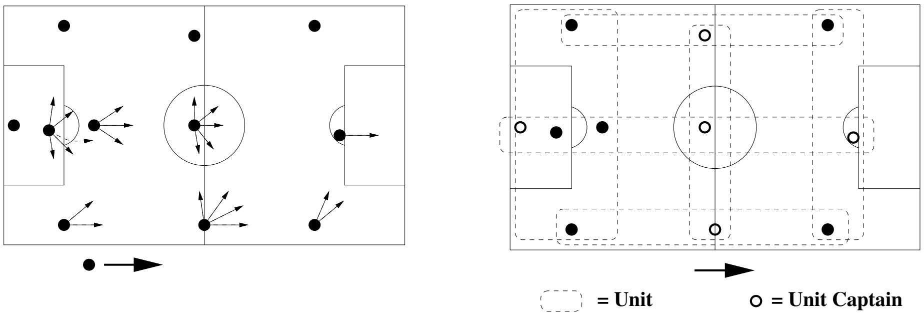


Figure 5: Different positions with home coordinates and home and max ranges.

The position's home coordinates are the default location to which the agent should go. However, the agent has some flexibility, being able to set its actual home position anywhere within the home range. It is this flexibility that is exploited by the anticipation mechanism. When moving outside of the max range, the agent is no longer considered to be in the position. The home and max ranges of different positions can overlap, even if they are part of the same formations.

A formation consists of a set of positions and a set of units. The formation and each of the units can also specify inter-position behavior specifications for the member positions, as illustrated in Figure 6(a). In this case, the formations specify inter-role interactions, namely the positions to which a player should consider passing the ball [16]. Figure 6(b) illustrates the units, the roles involved, and their captains. Here, the units contain defenders, mid-fielders, forwards, left players, center players, and right players.



(a) (b)

Figure 6: (a) A possible formation (4-3-3) for a team of 11 players. Arrows represent passing options. (b) Positions can belong to more than one unit.

Since the players are all autonomous, in addition to knowing its own role, each one has its own belief of the team's current formation along with the time at which that formation was adopted, and a map of teammates to positions. Ideally, the players have consistent beliefs as to the team's state, but this condition cannot be guaranteed between synchronization opportunities. Another offshoot of the player's autonomy is that each is free to leave its position unilaterally, leaving the team to adjust behind it. Thus, players are not bound by their positions when presented with unexpected action opportunities.

Our team structure allows for several features in our robotic soccer team. These features are: (i) the definition of and switching among multiple formations with units; (ii) flexible position adjustment and position switching; (iii) and pre-defined special purpose plays (set plays) [17].

3.2 Flexible Positions

In our multi-agent approach, the player positions itself flexibly such that it *anticipates* that it will be useful to the team, either offensively or defensively.

One way in which agents can use the position flexibility is to react exclusively to the ball's position. When reacting to the ball's position, the agent moves to a location within its range that minimizes its distance to the ball. This was the mechanism used successfully by the CMUnited-97 teams [17].

The flexible roles defined in the CMUnited-97 software were an improvement over the concept of rigid roles. Rather than associating fixed (x, y) coordinates with each position, an agent filling a particular role was given a range of coordinates in which it could position itself. Based on the ball's position on the field, the agent would position itself so as to increase the likelihood of being useful to the team in the future.

However, by taking into account the positions of other agents as well as that of the ball, an even more informed positioning decision can be made. The idea of strategic position by attraction and repulsion (SPAR) is one of the novel contributions of the CMUnited-98 software.

4 Anticipation

Our anticipation approach in the robotic soccer domain, as presented below, could be easily generalized.

Consider that for each agent, for each state, and at each time, there is a computable value for the probability that an active agent could successfully collaborate with a passive agent. As the world is constantly changing, the values for the probability of collaboration are computed as a function of the dynamic world.

Assuming that the transitions between states for each agent take time (or other type of cost), then anticipation consists of the selection of a new state that maximizes the probability of future collaboration.

Anticipation therefore allows for a flexible adjustment of a team agent towards the increase of the probability of being useful for the team. We now formally present our anticipation algorithm within the robotic soccer domain.

4.1 Anticipation in a Robotic Soccer Team

In a robotic soccer team there is a crucial resource, namely the ball, that one single agent is in control of or aiming at being in control of. [1] A precise objective in robotic soccer is the scoring of goals, i.e. kicking the ball into the opponent's goal. We say that the agent in control is *active* if it is the team member currently in charge of directly controlling the achievement of the team goal. All the other teammates are said to be *passive*.

We have previously created an action selection algorithm for an active agent in a team that allows for the run-time evaluation of what action to take when in possession of the ball: passing the ball to one of the teammates or shooting it directly towards the goal [16]. The interesting question we address here is what should the passive agents do? Anticipation allows the passive agents to actually not be "passive," but to position themselves with the concrete objective of trying to maximize the chances of a successful pass in case the active agent chooses to pass to them.

This strategic position takes into account the position of the other robots—both teammates and adversaries—and the positions of the ball (and active teammate) and of the opponent's goal. This becomes a multi-objective function with repulsion and attraction points.

Let's introduce the following variables:

- P - the desired position for the passive agent in anticipation of a passing need of its active teammate;
- n - the number of agents on each team;
- O_i - the current position of each opponent, $i = 1, \ldots, n$;
- T_i - the current position of each teammate, $i = 1, \ldots, (n-1)$;
- B - the current position of the active teammate and ball;
- G - the position of the opponent's goal.

Given these defined variables, we can then formalize the concrete robotic soccer anticipation algorithm, which we call SPAR for Strategic Positioning with Attraction and Repulsion. This extends similar approaches of using potential fields for highly dynamic, multi-agent domains [6]. The probability of collaboration in the robotic soccer domain is directly related to how "open" a position is to allow for a successful pass. SPAR then maximizes the repulsion from other robots and

[1] In a general case, there could be several crucial resources and several agents in control of these resources.

minimizes attraction to the ball and to the goal, namely:

- *Repulsion* from opponents, i.e., maximize the distance to each opponent: $\forall i$, max $dist(P, O_i)$
- *Repulsion* from teammates, i.e., maximize the distance to other passive teammates: $\forall i$, max $dist(P, T_i)$
- *Attraction* to the active teammate and ball: min $dist(P, B)$
- *Attraction* to the opponent's goal: min $dist(P, G)$

This is a multiple-objective function. To solve this optimization problem, we restate this function into a single-objective function. As each term may have a different relevance (e.g. staying close to the goal may be more important than staying away from opponents), we want to consider each term with a different weight, namely w_{O_i}, w_{T_i}, w_B, and w_G, for the weights for opponents, teammates, the ball, and the goal, respectively. Our anticipation algorithm uses then a weighted single-objective function:

$$\max(\forall i w_{O_i} dist(P, O_i) + \forall i w_{T_i} dist(P, T_i) - w_B dist(P, B) - w_G dist(P, G))$$

This optimization problem is then solved under the constraints which are specific to each team environment. We now present the set of constraints for both the simulator and the real robots environments. We used this anticipation algorithm in both of these platforms.

4.2 Constraints in the Simulator Team

One constraint in the simulator team relates to the position, or role, that the passive agent is playing relative to the position of the ball. The agent only considers positions that are within a rectangle whose corner is on the ball that is closest to the position home of the position that it is currently playing. This constraint helps ensure that the player with the ball will have several different passing options in different parts of the field. In addition, players don't need to consider moving too far from their positions to support the ball. The four possible quadrants are illustrated in Figure 4.2.

In addition to this first constraint, the agents observe three additional constraints. In total, the constraints in the simulator team are:

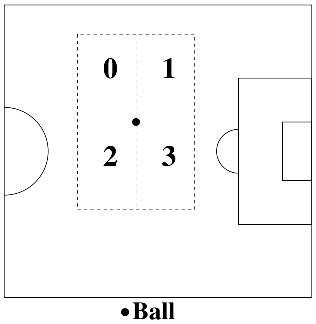


Figure 7: The four possible quadrants with corner at the ball considered for positioning by simulator agents.

- Stay in an area near home position
- Stay within the field boundaries
- Avoid being in an offsides position
- Stay in a position in which it would be possible to receive a pass.

This last constraint is evaluated by checking that there are no opponents in a cone with vertex at the ball and extending to the point in consideration.

4.3 Constraints in the Real Robot Team

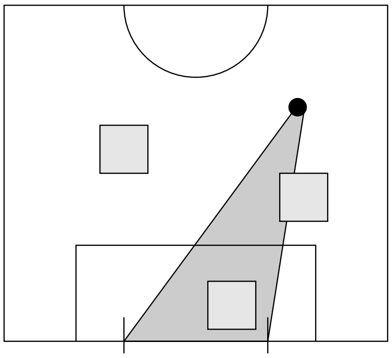
The real robotic environment is different from the simulator in several aspects. The main differences result from the fact that the agents here are real physical artifacts that occupy space, do not control the ball carefully, and have rather unreliable accurate motion. The robots' motion is quite crude manipulating the ball, as they cannot hold it, but act on the ball simply by pushing it with their mostly flat sides.

The constraints in the real robot team therefore involve two main aspects on the position of the passive agent:

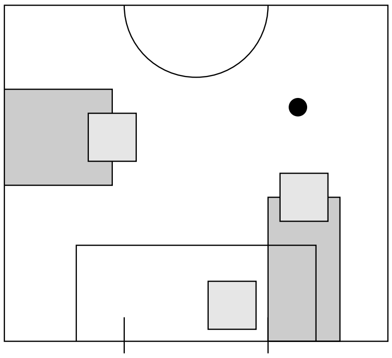
- Do not block possible direct shot from active teammate to the goal.

- Do not stand behind other robots, because these are difficult positions to receive passes in case the active team decides to do so.

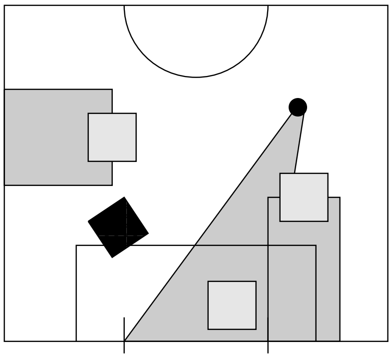
Figure 4.3(a) and (b) illustrate these two constraints and Figure 4.3(c) shows the combination of these two set of constraints and the resulting position of the anticipating passive teammate.



(a) Don't block goal shot.



(b) Avoid difficult collaboration.



(c) Anticipate optimal position for collaboration

Figure 8: Constraints for the anticipation algorithm for the CMUnited-98 small robot team; in (a) and (b) we show three opponents robots, and the current position of the ball corresponding also to the position of the active teammate; in (c) we illustrate the position of the passive agent, dark square, as returned by our anticipation algorithm.

Using this anticipation algorithm, the attacking team agents behaved in an exemplary collaborative fashion. Their motion on the field was a beautiful response to the dynamically changing adversarial environment. The active and passive agents moved in great coordination using the anticipation algorithm increasing very significantly successful collaboration.

4.4 Results

We will be investigating systematic ways to empirically evaluate the performance of our anticipation algorithm. In this paper, we provide the results of all of our real RoboCup-98 games. To note that the environments are highly adversarial and that our agents had never seen the opponent teams before.

Tables 1 and 2 show the scores of the games of CMUnited-98 simulator and real robot teams.

Opponent Name	Affiliation	Score
UU	Utrecht University, The Netherlands	22–0
TUM / TUMSA	Technical University Munich, Germany	2–0
Kasuga-Bitos II	Chubu University, Japan	5–0
Andhill'98	NEC, Japan	8–0
ISIS	Information Sciences Institute (USC), USA	12–0
Rolling Brains	Johannes Gutenberg-University Mainz, Germany	13–0
Windmill Wanderers	University of Amsterdam, The Netherlands	1–0
AT-Humboldt'98	Humboldt University of Berlin, Germany	3–0
TOTAL		66–0

Table 1: The scores of CMUnited-98's games in the simulator league of RoboCup-98. CMUnited-98 won all 8 games.

Opponent Name	Affiliation	Score
iXS	iXs Inc.	16–2
5DPO	University of Porto, Portugal	0–3
Paris-8	University of Paris-8	3–0
Cambridge	University of Cambridge, UK	3–0
Roboroos	University of Queensland, Australia	3–1
TOTAL		25–6

Table 2: The scores of CMUnited-98's games in the small-robot league of RoboCup-98. CMUnited-98 won 4 of its 5 games.

5 Conclusion

In this paper we have presented our work investigating and introducing the concept of anticipation to increase collaboration in teams of agents. We have been working in the concrete domain of robotic soccer.

We have presented our basic flexible team architecture in which individual agents flexibly move within their role as a function of the position of the ball to try to optimize their individual actions towards the ball. We then introduce a novel anticipation algorithm that allows for team agents to strategically position themselves in anticipation of possible collaboration needs from other teammates. We used this new anticipation algorithm in our RoboCup-98 teams achieving very successful results.

Given that we have developed algorithms for individual action and team collaboration anticipation, the current on-going step on our research agenda is to develop algorithms to anticipate the actions of the opponent agents and effectively react to them.

References

[1] Minoru Asada, Eiji Uchibe, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda. Coordination of multiple behaviors acquired by vision-based reinforcement learning. In *Proc. of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994 (IROS '94)*, pages 917–924, 1994.

[2] Kwun Han and Manuela Veloso. Reactive visual control of multiple non-holonomic robotic agents. In *Proceedings of the International Conference on Robotics and Automation*, Leuven,Belgium, May 1998.

[3] Jong-Hwan Kim, editor. *Proceedings of the Micro-Robot World Cup Soccer Tournament*, Taejon, Korea, November 1996.

[4] Hiroaki Kitano, Yasuo Kuniyoshi, Itsuki Noda, Minoru Asada, Hitoshi Matsubara, and Eiichi Osawa. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):73–85, Spring 1997.

[5] Hiroaki Kitano, Milind Tambe, Peter Stone, Manuela Veloso, Silvia Coradeschi, Eiichi Osawa, Hitoshi Matsubara, Itsuki Noda, and Minoru Asada. The RoboCup synthetic agent challenge 97. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 24–29, San Francisco, CA, 1997. Morgan Kaufmann.

[6] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer, 1991.

[7] A. K. Mackworth. On seeing robots. In A. Basu and X. Li, editors, *Computer Vision: Systems, Theory, and Applications*, pages 1–13. World Scientific Press, Singapore, 1993.

[8] Maja J. Mataric. Interaction and intelligent behavior. MIT EECS PhD Thesis AITR-1495, MIT AI Lab, August 1994.

[9] Hitoshi Matsubara, Itsuki Noda, and Kazuo Hiraki. Learning of cooperative actions in multi-agent systems: a case study of pass play in soccer. In *Adaptation, Coevolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium*, pages 63–67, Menlo Park,CA, March 1996. AAAI Press. AAAI Technical Report SS-96-01.

[10] Itsuki Noda and Hitoshi Matsubara. Soccer server and researches on multi-agent systems. In *Proceedings of the IROS-96 Workshop on RoboCup*, November 1996. Soccer server is available at http://ci.etl.go.jp/ noda/soccer/server/index.html.

[11] RoboCup web page, 1997. Accessible from http://www.robocup.org.

[12] Michael K. Sahota, Alan K. Mackworth, Rod A. Barman, and Stewart J. Kingdon. Real-time control of soccer-playing robots using off-board vision: the dynamite testbed. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3690–3663, 1995.

[13] Randy Sargent, Bill Bailey, Carl Witty, and Anne Wright. Dynamic object capture using fast vision tracking. *AI Magazine*, 18(1):65–72, Spring 1997.

[14] Peter Stone and Manuela Veloso. A layered approach to learning client behaviors in the RoboCup soccer server. *Applied Artificial Intelligence*, 12:165–188, 1998.

[15] Peter Stone and Manuela Veloso. Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, 48(1):83–104, January 1998.

[16] Peter Stone and Manuela Veloso. Using decision tree confidence factors for multiagent control. In *Proceedings of the Second International Conference on Autonomous Agents*, 1998.

[17] Peter Stone and Manuela Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In J. P. Müller, M. P. Singh, and A. S. Rao, editors, *Intelligent Agents V — Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL-98)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 1999.

[18] Peter Stone, Manuela Veloso, and Patrick Riley. The CMUnited-98 champion simulator team. In Minoru Asada and Hiroaki Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999. To Appear.

[19] Manuela Veloso, Michael Bowling, Sorin Achim, Kwun Han, and Peter Stone. The CMUnited-98 champion small robot team. In Minoru Asada and Hiroaki Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999. To Appear.

[20] Manuela Veloso, Peter Stone, and Kwun Han. The CMUnited-97 robotic soccer team: Perception and multiagent control. In *Proceedings of the Second International Conference on Autonomous Agents*, Minneapolis, MN, May 1998.

[21] Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim. CMUnited-97: RoboCup-97 small-robot world champion team. *AI Magazine*, 19(3):61–69, 1998.

[22] Manuela Veloso, William Uther, Masahiro Fujita, Minoru Asada, and Hiroaki Kitano. Playing soccer with legged robots. In *Proceedings of IROS-98, Intelligent Robots and Systems Conference*, Victoria, Canada, October 1998.