

Individual and Collaborative Behaviors in a Team of Homogeneous Robotic Soccer Agents*

Manuela Veloso and Peter Stone
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{veloso,pstone}@cs.cmu.edu
<http://www.cs.cmu.edu/~mmv,~pstone>

Abstract

Robotic soccer is a new challenging multi-agent domain, in which agents need to collaborate in an adversarial environment to achieve specific objectives. In this paper, we describe CMUnited-97, our team of robotic agents that we developed to enter the RoboCup-97 competition. We first discuss the challenges underlying the robotic soccer domain as a multi-agent system. We then introduce our team architecture, briefly describe the system's perception, and present the robots' actions ranging from low-level individual behaviors to coordinated, strategic team behaviors. The robots can organize themselves in formations, hold specific roles, and pursue their goals. In game situations, they extensively used their role-based behaviors, and demonstrated collaboration on multiple occasions. As homogeneous agents, the robots can also switch roles to maximize the overall performance of the team. CMUnited-97 won the RoboCup-97 small-robot competition at IJCAI in Nagoya, Japan.

1 Introduction

A complete intelligent agent should exhibit cognition, perception, and action capabilities. For many years now, Artificial Intelligence and Robotics researchers have been

*We thank Sorin Achim for designing and building the robots and Kwun Han for developing the perception algorithm. This research is sponsored in part by the Defense Advanced Research Projects Agency (DARPA), and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-95-1-0018 in part by the National Science Foundation under grant No. IRI-9502548, and in part by Department of the Navy, Office of Naval Research under contract number N00014-95-1-0591. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Rome Laboratory, Department of the Navy, Office of Naval Research or the United States Government.

working towards the goal of building physical (i.e. robotic) intelligent agents. This goal has proven to be quite challenging, but great progress has been made.

As individual robots can operate in the real world with increasing reliability, researchers have recently been focusing on building systems with *multiple* robotic agents. CMUnited, as presented in this paper, represents an effort to build a multi-agent robotic system in which the emphasis is on designing mobile robots not only capable of navigation and perception, but most importantly, capable of robust individual and collaborative high-level behaviors.

CMUnited is developed within the context of robotic soccer, an interesting emerging domain that is particularly appropriate for studying these multi-agent issues. Robotic soccer is being developed both in simulation environments and with different kinds of robots. We have pursued research in both simulation and with real robots. Although the simulation test-bed is not meant to mimic the real robots, we have successfully applied our multi-agent team structure across the two frameworks.

This paper is organized as follows: Section 2 discusses the multi-agent characteristics and challenges of the robotic soccer domain. Section 3 introduces the CMUnited robotic architecture. Section 4 describes the underlying algorithmic team architecture. Section 5 presents some of the specific single-agent and multi-agent behaviors implemented. Section 6 draws conclusions.

This paper contributes the overall architecture of our robotic soccer system. The combination of robust hardware, real-time vision, and intelligent control represented a significant challenge which we were able to successfully meet. The work described is fully implemented using the robotic agents pictured in Figure 1.



Figure 1. The CMUnited-97 robot team that competed in RoboCup-97.

2 Robotic Soccer

Robotic soccer with real robots is an exciting domain for many reasons. The fast-paced nature of the domain necessitates real-time sensing coupled with quick behaving and decision making. Furthermore, the behaviors and decision making processes can range from the most simple reactive behaviors, such as moving directly towards the ball, to arbitrarily complex reasoning procedures that take into account the actions and perceived strategies of teammates and opponents. Opportunities, and indeed demands, for innovative and novel techniques abound.

Previously, multi-agent systems have been applied to several domains, including design, planning, entertainment, games, air-traffic control, air combat, personal assistants, load-balancing, and robotic leg control (as surveyed in [10]).

Robotic soccer is a particularly good domain for studying multi-agent systems. Originated by Alan Mackworth [9], it has been gaining popularity in recent years, in particular in the RoboCup international competition [4]. It is also the subject of an official IJCAI-97 Challenge [5]. It can be used to evaluate different multi-agent systems techniques in a direct manner: teams implemented with different techniques can play against each other.

The pursuit domain is another example of a test-bed domain. Robotic soccer is much more complex and interesting as a general test-bed. Even with many predators and several prey, the pursuit domain is not complex enough to simulate the real world. Although robotic soccer is a game, most real-world complexities are retained. A key aspect of soccer's complexity is the need for agents not only to control themselves, but also to control the ball which is a passive part of the environment.

Robotic soccer can be played either with real robots or in a simulator [7, 8]. Although more costly and time consuming to develop, a number of groups have developed real robotic systems. The first robotic soccer system was the Dynamo system [9]. Sahota et al. built a 1 vs. 1 version of the game. Asada et al. have used vision-based RL with their soccer playing robots [2].

Although there are many possible ways to divide multi-agent systems, we identify two main dimensions: agent heterogeneity and amount of communication among agents [10]. Robotic soccer offers different multi-agent scenarios. Homogeneous non-communicating agents can be studied in robotic soccer by fixing the behavior of the opposing team and populating the team being studied with identical, mute players. To keep within the homogeneous agent scenario, the opponents must not be modeled as agents. In this context, the players can be reactive or deliberative to any degree. The extremely reactive agent might simply look for the ball and move straight at it, shooting whenever possible. At this extreme, the players may or may not have any knowledge that they are part of a team. On the other hand, players might model each other, thus enabling deliberative reasoning about whether to approach the ball or whether to move to a different part of the field in order to defend or to receive a pass. With players modeling each other, they may also reason about how to affect each other's behaviors in this inherently dynamic environment. Finally it is possible to study the relative merits of local and global perspectives on the world. Robots can be given global views with the help of an overhead camera; however, robotic soccer is often approached as a problem requiring local sensing.

Robotic soccer is also useful for studying the issues associated with heterogeneous non-communicating agents. Since each player has several teammates with the same global goal and several opponents with the diametrically opposed goal, each player is both benevolent and competitive at the same time. This possibility for combination of collaborative and adversarial reasoning is a major feature of the domain. When trying to collaborate, players' actions are usually interdependent: to execute a successful pass, both the passer and the receiver must execute the appropriate actions. Thus modeling each other for the purpose of coordination is helpful. Social conventions, such as programmed notions of when a given agent will pass or which agents should play defense, can also help coordination. Since communication is still not allowed, the players must have a reliable method for filling the different team roles needed on a soccer team (defense, offense, goalie). Ideally, the players are able to switch roles during the course of a game when appropriate.

Robotic soccer is perhaps best suited for the study of the most complex multi-agent scenario: heterogeneous communicating agents. Since the agents can indeed communicate,

the full potential of the domain is realized in this scenario. With players posting messages to the blackboard, they must have a language in order to understand each other. Protocols are also needed for commitment to team plays: the passer and receiver in a pass play must both agree to execute the pass. For more complex team plays, several players may need to commit to participate. But then the issue arises of how single-mindedly they must adhere to the committed play: when may they react to more pressing situations and ignore the commitment? For any team play, including a simple pass, timing is very important in such a real-time scenario. Thus, players must coordinate their actions very carefully. Finally, speech acts are particularly interesting in the environment that is both collaborative and adversarial. If the opponents can understand the same language, a planned utterance can affect the knowledge of both teammates and opponents. The utility of communication must be carefully considered and the possibility of lying in order to fool the opponent arises.

While our research in simulation considers communicating agents [11], our real robot team described here implements homogeneous non-communicating agents.

3 The CMUnited Architecture

Our robotic architecture merges high-level and low-level reasoning by viewing the overall system as the combination of the robots, a camera over-looking the playing field connected to a centralized interface computer, and several clients as the minds of the small-size robot players. Figure 2 sketches the building blocks of the architecture.

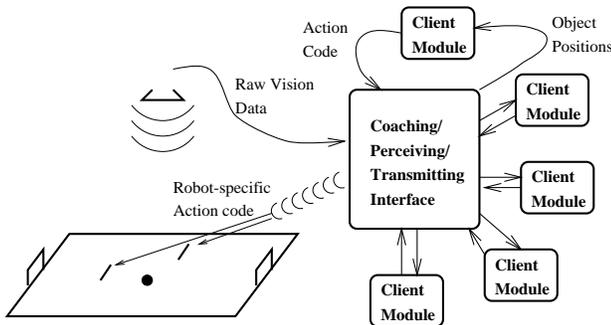


Figure 2. CMUnited architecture with global perception and multiple agents.

The complete system is fully autonomous consisting of a well-defined and challenging processing cycle. The global vision algorithm perceives the dynamic environment and processes the images, giving the positions of each robot and the ball. This information is sent to an off-board controller and distributed to the different agent algorithms. Each agent

evaluates the world state and uses its strategic knowledge to decide what to do next. Actions are motor commands that are sent by the off-board controller through radio communication. Commands can be broadcast or sent directly to individual agents. Each robot has an identification binary code that is used on-board to detect commands intended for that robot. This complete system is fully implemented.

The fact that perception is achieved by a video camera that over-looks the complete field offers an opportunity to get a global view of the world state. Although this setup may simplify the sharing of information among multiple agents, it presents a challenge for reliable and real-time processing of the movement of multiple moving objects—in our case, the ball, five agents on our team, and five agents on the opposing team.

The RoboCup small robot league limits the size of each robot to approximately the equivalent area of a 15cm diameter circle. The reactivity of the soccer robots requires real-time vision processing. However, due to rich visual input, researchers have found that dedicated processors or even DSPs are often needed [9, 1]. We use a color-based vision processing algorithm that allows the use of fixed color space thresholds to segment the different colors into regions.

Each robot is fitted with two colors to differentiate the team and the orientation. We developed a reliable detection algorithm that uses a minimum distance principle between frames to retain association [3]. In the setting of a robot soccer game, the sole ability to detect the location of objects in the field is often not enough. Like real soccer players, it is often useful and necessary to have the ability to predict future locations of the ball and of the players. We have utilized an extended Kalman-Bucy filter for such a purpose. Through a careful adjustment of the filter parameters modelling the system, we were able to achieve successful tracking and, in particular prediction of the ball trajectory, even when sharp bounces occur.

Our vision processing approach worked perfectly during the RoboCup-97 games. We were able to detect and track 11 objects (5 teammates, 5 opponents and a ball). The prediction provided by the filter allowed the goalkeeper to look ahead in time and predict the best defending position. During the game, no goals were suffered due to miscalculation of the predicted ball position.

4 Team Architecture

Our new teamwork structure is situated within a team member architecture suitable for *periodic time synchronization* domains [12] in which individual agents can capture locker-room agreements and respond to the environment, while acting autonomously. Based on a standard agent architecture, our team member architecture allows agents to sense the environment, to reason about and select their actions, and

to act in the real world. The locker-room agreement captures for example, the triggers for specific individual and collaborative actions, and role changing rules. Figure 3 shows the functional input/output model of the architecture.

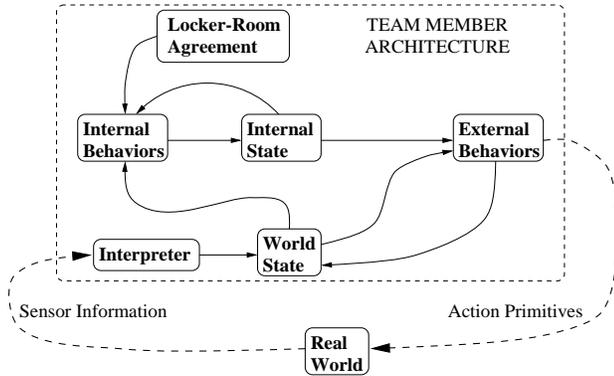


Figure 3. The CMUnited team member architecture.

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*.

The World State reflects the agent’s conception of the real world, both via its sensors and via the predicted effects of its actions. It is updated as a result of processed sensory information. It may also be updated according to the predicted effects of the external behavior module’s chosen actions. The world state is directly accessible to both internal and external behaviors.

The Locker-Room Agreement is set by the team when it is able to privately synchronize. It defines the flexible teamwork structure as presented below. The locker-room agreement may change periodically when the team is able to re-synchronize; however, it generally remains unchanged. The locker-room agreement is accessible only to internal behaviors.

The Internal State stores the agent’s internal variables. It may reflect previous and current world states, possibly as specified by the locker-room agreement. For example, the agent’s role within a team behavior could be stored as part of the internal state, as could a distribution of past world states. The agent updates its internal state via its internal behaviors.

The Internal Behaviors update the agent’s internal state based on its current internal state, the world state, and the team’s locker-room agreement.

The External Behaviors reference the world and internal states, sending commands to the actuators. The actions affect the real world, thus altering the agent’s future percepts. External behaviors consider only the world and internal states, without direct access to the locker-room agreement.

Our notion of behavior is consistent with that laid out in [6]. 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.

Our teamwork structure involves *flexible roles* that are organized into *formations*.

4.1 Role

A *role, r*, consists of a specification of an agent’s internal and external behaviors. The conditions and arguments of any behavior can depend on the agent’s current role, which is a function of its internal state. At the extreme, a top-level behavior could be a switch, calling an entirely different behavior graph for each possible role. However, the role can affect the agent’s overall behavior at any level of its complete behavior graph. Notice that roles need not be rigid: by specifying ranges of parameters or behavior options, the agent filling role *r* can be given an arbitrary amount of flexibility.

For example, a role in the robotic soccer domain, can be a position such as a midfielder. In the hospital maintenance domain, a role could specify the wing of the hospital whose floors the appropriate agent should keep clean, while in the web search domain, it could specify a server to search.

4.2 Formation

We achieve collaboration between agents through the introduction of *formations* as a team structure. A formation decomposes the task space defining a set of roles. Formations include as many roles as there are agents in the team, so that each role is filled by one agent. In addition, formations can specify sub-formations, or *units*, that do not involve the whole team. A unit consists of a subset of roles from the formation, a *captain*, and intra-unit interactions among the roles.

For a team of n agents $A = \{a_1, a_2, \dots, a_n\}$, any formation is of the form $F = \{R, \{U_1, U_2, \dots, U_k\}\}$ where R is a set of roles $R = \{r_1, r_2, \dots, r_n\}$ such that

$i \neq j \Rightarrow r_i \neq r_j$. Note that there are the same number of roles as there are agents. Each unit U_i is a subset of R : $U_i = \{r_{i1}, r_{i2}, \dots, r_{ik}\}$ such that $r_{ia} \in R$, $a \neq b \Rightarrow r_{ia} \neq r_{ib}$ and r_{i1} is the captain. The map $A \mapsto R$ is not pre-specified: roles can be filled by different homogeneous agents. A single role may be a part of any number of units and formations.

Formations can affect the agent’s external behaviors by specifying inter-role interactions. Since roles can be re-used among formations, their formation-specific interactions cannot be included in the role definitions. Instead these interactions are part of the formation specification.

Units are used to deal with local problem solving issues. Rather than involving the entire team in a sub-problem, the roles that address it are organized into a unit.

Roles and formations are introduced independently from the agents that are to fill them. The locker-room agreement specifies an initial formation, a map from agents to roles, and run-time triggers for dynamic changing of formations. At any given time, each agent should know what formation the team is currently using. Agents keep mappings $A \mapsto R$ from teammates to roles in the current formation. All this team structuring information is stored in the agent’s internal state. It can be altered via the agent’s internal behaviors.

In the real robotic soccer domain, the agents are autonomous and do not communicate. Hence, there is no guarantee that they will all think that the team is using the same formation, nor that they have accurate maps $A \mapsto R$. In fact, the only guarantee is that each agent knows its own current role. However, we create robust behaviors for team agents which ensure that the behaviors never absolutely depend upon having correct, up-to-date knowledge of teammates’ internal states: they must degrade gracefully.

We achieve multi-agent strategy through the combination of accurate individual and collaborative behaviors. Agents reason through the use of persistent reactive behaviors that are developed to aim at reaching team objectives.

Figure 4 summarizes the CMUnited overall team structure.

5 Individual and Team Behaviors

In order to be able to successfully collaborate, agents require robust basic skills. These skills include the ability to go to a given place on the field, the ability to direct the ball in a given direction, and the ability to intercept a moving ball. All of these skills must be executed while avoiding obstacles such as the walls and other robots.

5.1 Ball Handling

If a robot is to accurately direct the ball towards a target position, it must be able to approach the ball from a specified

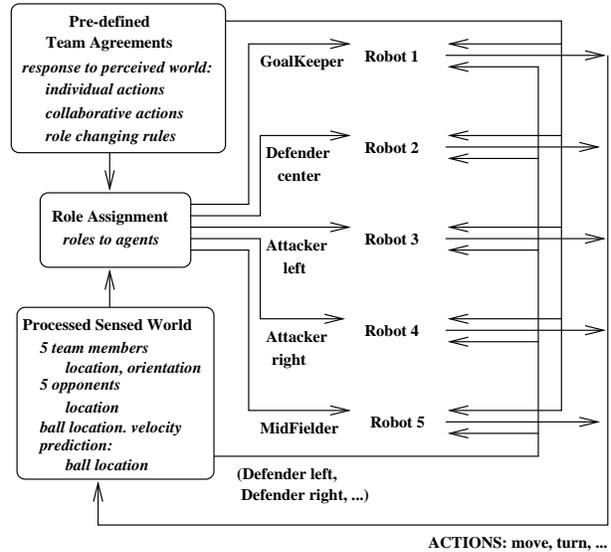


Figure 4. Team agreements, behaviors, and roles in the CMUnited-97 team.

direction. Using the ball prediction from the vision system, the robot aims at a point on the far side of the target position. The robots are equipped with two methods of doing so:

- **Ball collection:** Moving behind a ball and knocking it towards the target.
- **Ball interception:** Waiting for the ball to cross its path and then intercepting the moving ball towards the target.

When using the ball collection behavior, the robot considers a line from the target position to the ball’s current or predicted position, depending on whether or not the ball is moving. The robot then plans a path to a point on the line and behind the ball such that it does not hit the ball on the way and such that it ends up facing the target position. Finally, the robot accelerates to the target. Figure 5(a) illustrates this behavior.

When using the ball interception behavior (Figure 5(b)), on the other hand, the robot considers a line from *itself* to the target position and determines where the ball’s path will intersect this line. The robot then positions itself along this line so that it will be able to accelerate to the point of intersection at the same time that the ball arrives.

In practice, the robot chooses from between its two ball handling routines based on whether the ball will eventually cross its path at a point such that the robot could intercept it towards the goal. Thus, the robot gives precedence to the ball interception routine, only using ball collection when necessary. When using ball collection, it actually aims at the ball’s predicted location a fixed time in the future so as to eventually position itself in a place from which it can

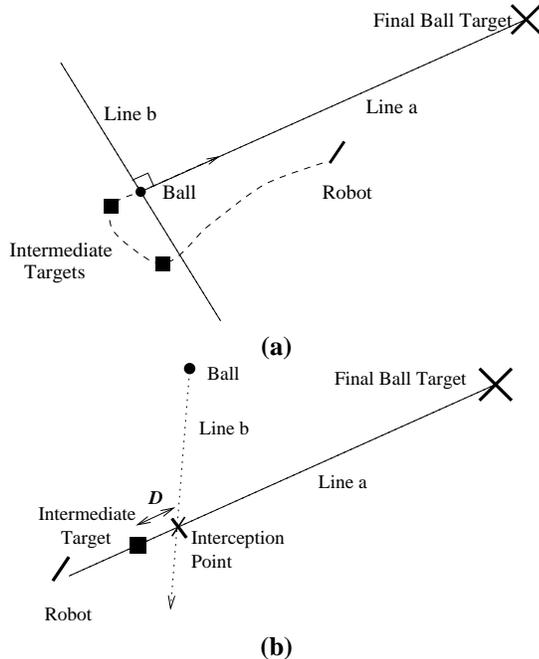


Figure 5. Single-agent behaviors to enable team collaboration: (a) Ball collection; (b) Ball interception.

intercept the ball towards the target.

In the robotic soccer field, there are often obstacles between the robot and its goal location. Our robots try to avoid collisions by planning a path around the obstacles. Due to the highly dynamic nature of this domain, our obstacle avoidance algorithm uses closed-loop control by which the robots continually replan their goal positions around obstacles. Even with obstacle avoidance in place, the robots can occasionally get stuck against other robots or against the wall. Particularly if opponent robots do not use obstacle avoidance, collisions are inevitable. When unable to move, our robots identify the source of the problem as the closest obstacle and “unstick” themselves by moving away. Once free, normal control resumes.

5.2 Positions, Formations, and Active Modes

Although the single-agent behaviors are very effective when just a single robot is on the field, if all five robots were simultaneously chasing the ball and trying to shoot it at the goal, chaos would result. In order to achieve coordinated multi-agent behavior, we organize the five robots into a flexible team structure.

The team structure, or *formation*, defines a set of roles, or *positions* with associated behaviors. The robots are then dynamically mapped into the positions. Each robot is equipped

with the knowledge required to play any position in each of several formations.

The positions indicate the areas of the field which the robots should move to in the default situation. There are also different *active modes* which determine when a given robot should move to the ball or do something else instead. Finally, the robot with the ball chooses whether to shoot or pass to a teammate using a passing evaluation function.

These high-level, multi-agent behaviors were originally developed in simulation and then transferred over to the robot-control code. Only the run-time passing evaluation function was redefined. Further details, particularly about the flexible team structures, are available in [12].

Positions are defined as flexible regions within which the player attempts to move towards the ball. For example, a robot playing the “right-wing” (or “right forward”) position remains on the right side of the field near the opponents’ goal until the ball comes towards it. Positions are classified as defender/midfielder/forward based on the locations of these regions. They are also given behavior specifications in terms of which other positions should be considered as potential pass-receivers (see Section 5.3).

At any given time each of the robots plays a particular position on the field. However, each robot has all of the knowledge necessary to play any position. Therefore the robots can—and do—switch positions on the fly. For example, robots A and B switch positions when robot A chases the ball into the region of robot B. Then robot A continues chasing the ball, and robot B moves to the position vacated by A.

The pre-defined positions known to all players are collected into formations, which are also commonly known. An example of a formation is the collection of positions consisting of the goalkeeper, one defender, one midfielder, and two attackers. Another possible formation consists of the goalkeeper, two defenders and two attackers.

As is the case for position-switches, the robots switch formations based on pre-determined conditions. For example, if the team is losing with very not much time left in the game, the robots would switch to a more offensive formation. On the other hand, if winning, they might choose a defensive formation. The precise conditions for switching positions and formations are decided upon in advance, in what we call a “locker-room agreement,” [12] in order to eliminate the need for complex on-line negotiation protocols.

Although the default action of each robot is to go to its position and face the ball, there are three *active modes* from which the robot must choose. The default position-holding behavior occurs when the robot is in an *inactive* state. However, when the ball is nearby, the robot changes into an *active* state. In the active state, the robot moves towards the ball, attempting either to pass it to a teammate or to shoot it towards the goal based on an evaluation function

that takes into account teammate and opponent positions (see Section 5.3). A robot that is the intended receiver of a pass moves into the *auxiliary* state in which it tries to intercept a moving ball towards the goal. Our current decision function sets the robot that is closest to the ball into the active state; the intended receiver robot (if any) into the auxiliary state; and all other robots into the inactive state.

5.3 Run-time Evaluation of Collaborative Opportunities

One of CMUnited-97's main features is the robots' ability to collaborate by passing the ball. When in active mode, the robots use an evaluation function that takes into account teammate and opponent positions to determine whether to pass the ball or whether to shoot. In particular, as part of the formation definition, each position has a set of positions to which it considers passing. For example, a defender might consider passing to any forward or midfielder, while a forward would consider passing to other forwards, but not backwards to a midfielder or defender.

For each such position that is occupied by a teammate, the robot evaluates the pass to that position as well as evaluating its own shot. To evaluate each possible pass, the robot computes the *obstruction-free-index* of the two line segments that the ball must traverse if the receiver is to shoot the ball (lines b and c in Figure 6). In the case of a shot, only one line segment must be considered (line a). The *value* of each possible pass or shot is the product of the relevant obstruction-free-indices. Robots can be biased towards passing or shooting by further multiplying the values by a factor determined by the relative proximities of the active robot and the potential receivers to the goal. The robot chooses the pass or shot with the maximum value. The obstruction-free-index of line segment l is computed by the following algorithm (variable names correspond to those in Figure 6):

1. $obstruction-free-index = 1$.
2. For each opponent O :
 - Compute the distance x from O to l and the distance y along l to l 's origin, i.e. the end at which the ball will be kicked by the robot (See Figure 6).
 - Define constants $min-dist$ and $max-denominator$. Opponents farther than $min-dist$ from l are not considered. When discounting $obstruction-free-index$ in the next step, the y distance is never considered to be larger than $max-denominator$. For example, in Figure 6, the opponent near the goal would be evaluated with $y = max-denominator$, rather than its actual distance from the ball. The reasoning is that beyond distance $max-$

denominator, the opponent has enough time to block the ball: the extra distance is no longer useful.

- if $x < min-dist$ and $x < y$,
 $obstruction-free-index *= x / MIN(max-denominator, y)$.

3. return $obstruction-free-index$.

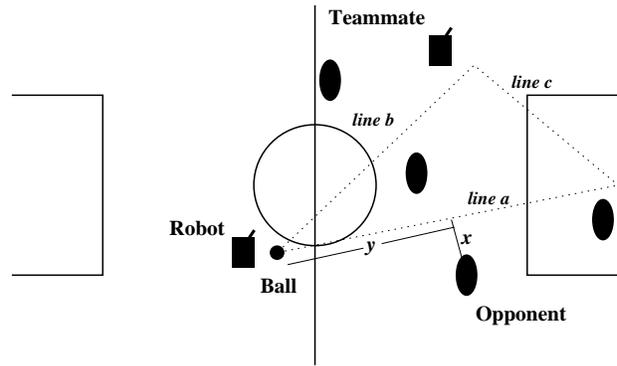


Figure 6. Pass evaluation is based on path position of opponents.

Thus the obstruction-free-index reflects how easily an opponent could intercept the pass or the subsequent shot. The closer the opponent is to the line and the farther it is from the ball, the better chance it has of intercepting the ball.

5.4 A Special-Purpose Agent: The Goalkeeper

To this point, we have assumed that the entire team consists of homogeneous agents. Thus they are able to freely interchange roles as the game proceeds. However, there is one exception: the goalkeeper robot has both special hardware and special software. Thus, it does not switch positions or active modes like the others. The goalkeeper's physical frame is distinct from that of the other robots in that it is as long as allowed under the RoboCup-97 rules (18cm) so as to block as much of the goal as possible. The goalkeeper's role is to prevent the ball from entering the goal. It stays parallel to and close to the goal, aiming always to be directly even with the ball's lateral coordinate on the field.

Ideally, simply staying even with the ball would guarantee that the ball would never get past the goalkeeper. However, since the robots cannot accelerate as fast as the ball can, it would be possible to defeat such a behavior. Therefore, the goalkeeper continually monitors the ball's trajectory. In some cases it moves to the ball's predicted destination point ahead of time. The decision of when to move to the predicted ball position is both crucial and difficult. Our goalie

robot currently take into account the predicted velocity and direction of the ball to select its moves.

6 Discussion and Conclusion

CMUnited-97 successfully demonstrated the feasibility and effectiveness of teams of multi-agent robotic systems. Within this paradigm, one of the major challenges was to “close the loop,” i.e., to integrate all the different modules, ranging from perception to strategic multi-agent reasoning. CMUnited is an example of a fully implemented multi-agent system in which the loop is closed. In addition, we implemented interesting strategic behaviors, including agent collaboration and real-time evaluation of alternative actions, and we developed a new multi-agent team structure.

It is generally very difficult to accumulate significant scientific results to test teams of robots. Realistically, extended runs are prohibited by battery limitations and the difficulty of keeping many robots operational concurrently. Furthermore, we only had the resources to build a single team of five robots, with one spare so far. Therefore, we offer a restricted evaluation of CMUnited based on the results of four effective 10-minute games that were played at RoboCup-97. We also include anecdotal evidence of the multi-agent capabilities of the CMUnited-97 robotic soccer team.

The CMUnited-97 robot team played games against robot teams from Nara Institute of Science and Technology (NAIST), Japan; University of Paris VI, France (team name “MICROB”); and University of Girona, Spain. The results of the games are given in Table 1. In total, CMUnited-97

Opponent	Score
NAIST	5-0
MICROB	3-1
U. of Girona	2-0
NAIST (finals)	3-0
TOTAL	13-1

Table 1. The scores of CMUnited’s games in the small robot league of RoboCup-97. CMUnited-97 won all four games.

scored thirteen goals, allowing only one against. The one goal against was scored by the CMUnited goalkeeper against itself, though under an attacking situation from France.

As the matches proceeded, spectators noticed many of the team behaviors. The robots switched positions during the games, and there were several successful passes. The most impressive goal of the tournament was the result of a 4-way passing play that ended with a successful shot into the goal.

We are aware that many issues are open for further research and development. We are systematically identifying them and including them in our on-going research agenda.

References

- [1] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.
- [2] M. Asada, E. Uchibe, S. Noda, S. Tawaratsumida, and K. 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.
- [3] K. Han and M. Veloso. Reactive visual control of multiple non-holonomic robotic agents. In *Proceedings of the International Conference on Robotics and Automation*, 1998.
- [4] H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):73–85, Spring 1997.
- [5] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. 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] M. J. Mataric. Interaction and intelligent behavior. MIT EECS PhD Thesis AITR-1495, MIT AI Lab, August 1994.
- [7] I. Noda. Soccer server: a simulator of RoboCup. In *Proceedings of AI symposium '95*, pages 29–34. Japanese Society for AI, December 1995.
- [8] M. K. Sahota. Real-time intelligent behaviour in dynamic environments: Soccer-playing robots. Master’s thesis, University of British Columbia, 1993.
- [9] M. K. Sahota, A. K. Mackworth, R. A. Barman, and S. 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.
- [10] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Under revision, available at <http://www.cs.cmu.edu/pstone/>*.
- [11] P. Stone and M. Veloso. Communication in domains with unreliable, single-channel, low-bandwidth communication. In *1st International Workshop on Collective Robotics*, July 1998.
- [12] P. Stone and M. Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In *5th International Workshop on Agent Theories, Architectures, and Languages*, July 1998.