# The CMUnited-97 Small Robot Team

Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{veloso,pstone,kwunh,sorin}@cs.cmu.edu

In *RoboCup-97: Robot Soccer World Cup I*,
H. Kitano (ed.), 1998. Springer Verlag, Berlin.

**Abstract.** Robotic soccer is a challenging research domain which involves multiple agents that need to collaborate in an adversarial environment to achieve specific objectives. In this paper, we describe CMUnited, the team of small robotic agents that we developed to enter the RoboCup-97 competition. We designed and built the robotic agents, devised the appropriate vision algorithm, and developed and implemented algorithms for strategic collaboration between the robots in an uncertain and dynamic environment. The robots can organize themselves in formations, hold specific roles, and pursue their goals. In game situations, they have demonstrated their collaborative behaviors on multiple occasions. The robots can also switch roles to maximize the overall performance of the team. We present an overview of the vision processing algorithm which successfully tracks multiple moving objects and predicts trajectories. The paper then focusses on the agent behaviors ranging from low-level individual behaviors to coordinated, strategic team behaviors. CMUnited won the RoboCup-97 small-robot competition at IJCAI-97 in Nagoya, Japan.

## 1 Introduction

As robots become more adept at operating in the real world, the high-level issues of collaborative and adversarial planning and learning in real-time situations are becoming more important. An interesting emerging domain that is particularly appropriate for studying these issues is Robotic soccer, as first proposed by [9] and actively pursued within the RoboCup initiative [7, 1]. Although realistic simulation environments exist [10, 11] and are useful, it is important to have some physical robotic agents in order to address the full complexity of the task.

Robotic soccer with real robots is a challenging 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.

One of the advantages of Robotic Soccer is that it enables the direct comparison of different systems: they can be matched against each other in competitions. In particular, the system described here was designed specifically for RoboCup97 in which several robotic teams competed on an "even playing field." [6]. The scientific opportunities involved in this effort are enormous. Our particular scientific focus is on multiagent systems coupled with collaborative and adversarial learning in an environment that requires real-time dynamic planning.

This paper describes 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 in this paper is fully implemented as our CMUnited-97 RoboCup team. CMUnited-97 won the RoboCup-97 small-robot competition at IJCAI-97 in Nagoya, Japan. Our team scored a total of thirteen goals and only suffered one. Figure 1 shows a picture of our robotic agents.

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

The specific contributions of the CMUnited-97 robot team, as presented in this paper, include:

- The complete design and development of robots with robust navigation and communication hardware.
- Reliable perception through the use and extension of a Kalman-Bucy filter. Sensing through our vision processing algorithm allows for (i) tracking of multiple moving objects; (ii) and prediction of object movement, particularly the ball, even when inevitable sharp trajectory changes occur.
- Multiagent strategic reasoning. Collaboration between robots is achieved through: (i) a flexible role-based approach by which the task space is decomposed and agents are assigned subtasks; (ii) a flexible team structure by which agents are organized in *formations*, homogeneous agents flexibly switch roles within formations, and agents switch formations dynamically; and (iii) alternative plans allowing for collaboration (e.g. passing or shooting), are controlled by pre-defined metrics for real-time evaluation.

## 2 Overall Architecture

The architecture of our system addresses the combination of high-level and low-level reasoning by viewing the overall system as the combination of the robots, a vision 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.

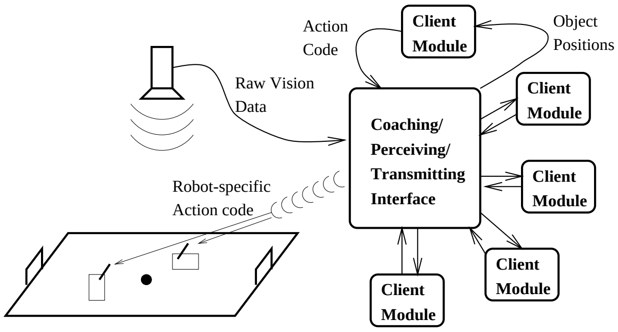


**Fig. 2.** CMUnited-97 Architecture with Global Perception and Distributed Reaction.

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 motion 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.

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 (see Section 3).

The robots are built using two coreless DC motors (differential drive) with built-in magnetic encoders. The various speed and acceleration values are obtained by two specialized motion control processors (PID) which are controlled by a 8-bit micro-controller. The robots are equipped with an on-board two-way radio link, which allows for data exchange with an off-board processing unit. This wireless communication link also enables sending and receiving data to and from other robots at speeds of up to 40 Kbit/s.

One of the interesting characteristics of our robots is that the core electronics is separated from the outside frame. The robot dimensions are as follows:

- Length: limited by the size of the circuit board, which is 9.4cm; actual total length depends on the frame.
- Width: limited by the size of the wheels and motors' axis, which is 9.4cm.
- Frame: we have three sizes of frames, namely a base frame – 12cm $\times$ 12cm, the elongated frame – 18cm $\times$ 10cm, and a wide frame – 15cm $\times$ 12cm.
- Height: 12 cm; Weight: 1.5 lb

The base frame allows for various configurations of the final external frame. The frames are articulated at the edges and made of perforated steel strips and sheets. The flexible frame structure allows for the easy access to the components and easy use of variations of frames, as a function of the purpose of the robots.

Around 60% of the robot weight is made up by the battery packs. The on-board electronics include an 8-bit micro-controller, an on-board memory of 512 bytes RAM, and 2 Kbyte EEPROM, and a half-duplex FM 418 MHz radio.

The dimensions of the circuit boards could be adapted to fit different shapes and sizes of the main frame. More and smaller boards can be stacked inside the main frame, making it possible to incorporate other components if required.

We created a *command server* for handling commands from the individual off-board robot reasoning processes. The radio control processor is connected to the server computer via one serial link. Thus individual "brains" from networked machines must communicate to their robot "bodies" through the server computer. One of the command server's roles is to collect and translate these commands and to send them to the radio control processor.

## 3 Real-Time Perception for Multiple Agents

The vision requirements for robotic soccer have been examined by different researchers [12, 13]. Systems with on-board and off-board types have appeared in recent years. All have found that the reactiveness of soccer robots requires a vision system with a high processing cycle time. However, due to the rich visual input, researchers have found that dedicated processors or even DSPs are often needed [2, 12]. We currently use a frame-grabber with frame-rate transfer from a 3CCD camera. A 166MHz Pentium processor is dedicated to the vision processing.

### 3.1 Color-based Detection

The RoboCup rules specify well defined colors for different objects in the field and these are used as the major cue for object detection. Our vision-processing is therefore color based. Teammates and opponents are identified by blue and yellow circles. We add an additional pink patch for each of our robots to determine teammate orientation. The ball is an orange golf ball (see Figure 1).

Noise is inherent in all vision systems. False detections in the current system are often of a magnitude of 100 spurious detections per frame. The system eliminates false detections via two different methods. First, color patches of size not matching the ones on the robots are discarded. This technique filters off most "salt and pepper" noise. Second, by using a minimum-distance data association mechanism, all false detections are eliminated.

### 3.2 Data Association

The color-based detection algorithm returns an unordered list of robots for each frame. To be able to control the robots, the system must associate each detected robot in the field with a robot identification.

Each of the robots is fitted with the same color tops and no attempts are made to differentiate them via color hue. Experience has shown that, in order to differentiate five different robots by hue, five significantly different hues are needed. However, the rules of the RoboCup game eliminate green (field), white (markings), orange (ball), blue and yellow (team and opponent) from the list of possibilities. Furthermore, inevitable variations in lighting conditions over the area of the field and noise in the sensing system are enough to make a hue-based detection scheme impractical.

With each robot fitted with the same color, visually, all robots on the same team appear identical to the visual system. Data association addresses the problem of retaining robot identification in subsequent frames. We devised an algorithm to retain association based on the spatial locations of the robots.

We assume that the starting positions of all the robots are known. This can be done trivially by specifying the location of the robots at the start time. However, as subsequent frames are processed, the locations of the robots change due to robot movements (due to controlled actions or adversarial pushes). Association can be achieved by making two complementary assumptions: 1) Robot displacements over consecutive frames are local; 2) The vision system can detect objects at a constant frame rate. By measuring the maximum robot velocity, we know that in subsequent frames, the robot is not able to move out of a 3cm radius circular region. This knowledge provides the basis of our association technique.

These assumptions provide the basis to the minimum distance scheme that we devised to retain association between consecutive frames. During consecutive frames, association is maintained by searching for objects within a minimum displacement. Current robot positions are matched with the closest positions from the previous frame. Our greedy algorithm searches through all possible matches, from the smallest distance pair upwards. Whenever a matched pair is found, it greedily accepts it as a matching pair.

Due to noise, it is possible for the detection system to leave a robot undetected. In this case, the number of robots detected on the past frame is larger than the number of robots in the current frame. After the greedy matching process, the remaining unmatched positions from the past frame will be carried over to the current frame. The robots corresponding to the unmatched locations will be assumed to be stationary. Note that if the missing rate of the detection algorithm were high and frequent, this minimum distance association technique would easily lose the robots. However, this greedy algorithm was used in RoboCup-97 successfully, showing its reliability when combined with our accurate and robust vision detection system.

### 3.3 Tracking and Prediction

In the setting of a robot soccer game, the ability to merely detect the locations of objects on the field is often not enough. Like for real soccer players, it is often essential for robots to predict future locations of the ball (or of the other players). We have used an Extended Kalman-Bucy filter (EKF) [5] for ball movement prediction. The EKF is a recursive estimator for a possibly non-linear system. It involves a two-step iterative process, namely *update* and *propagate*. The current best estimate of the system's state and its error covariance is computed on each iteration. During the update step, the current observations are used to refine the current estimate and recompute the covariance. During the propagate step, the state and covariance of the system at the next time step are calculated using the system's equations. The process then repeats, alternating between the update and the propagate steps.

We capture the ball's state with five variables: the ball's $x$ and $y$ location, the ball's velocities in the $x$ and $y$ direction and a friction parameter ($\lambda_k$) for the surface. The system is represented by the following non-linear difference equations:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \dot{x}_k \cdot \Delta t \\ y_k + \dot{y}_k \cdot \Delta t \\ \dot{x}_k \cdot \lambda_k \\ \dot{y}_k \cdot \lambda_k \\ \lambda_k \end{bmatrix} \qquad \begin{aligned} \dot{x}_{k+1} &= \dot{x}_k \cdot \lambda_k \\ \dot{y}_{k+1} &= \dot{y}_k \cdot \lambda_k \\ \lambda_{k+1} &= \lambda_k \end{aligned}$$

The equations model the ball with simple Newtonian dynamics, where $\lambda_k$ is a friction term which discounts the velocity at each time step, and $\Delta t$ is the time-step size.

The prediction equations are:

$$\begin{aligned} x_{k+n} &= x_k + \dot{x}_k \cdot \Delta t \cdot \alpha_{kn} \\ y_{k+n} &= y_k + \dot{y}_k \cdot \Delta t \cdot \alpha_{kn} \\ \alpha_{kn} &= \begin{cases} 1 & \text{if } \lambda_k = 1 \\ (1-(\lambda_k)^n)/(1-\lambda_k) & \text{otherwise} \end{cases} \end{aligned}$$

The prediction equations are derived by solving the recursive equation obtained by substituting the value of $x_{k+i}$ where $i$ decreases from $n$ to 1. We are only interested in the predicted spatial location of the ball thus we do not explicitly calculate the predicted velocity.

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) at 30 frames/s. The prediction provided by the EKF 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 Multiagent Strategy Control

We achieve multiagent 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.

### 4.1 Single-agent Behaviors

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

**Non-holonomic path generation**
The non-holonomic path planning problem has been addressed by many researchers, e.g., [8, 4]. However, most of the algorithms deal with static worlds and generate pre-planned global paths. In the robot soccer domain, this is not possible as the domain is inherently dynamic and response times need to be very high. Furthermore, the world dynamics include also possible interference from other robots (e.g., pushing), making precisely mapped out paths ineffective and unnecessary.

We devised and implemented a reactive controller for our system, which is computationally inexpensive, deals with dynamic environments, and recovers from noisy command execution and possible interferences. A reactive controller also has possible disadvantages, as it may generate sub-optimal paths, due to local minima. We introduced a failure recovery routine to handle such failures.

The navigational movement control for our robots is hence done via reactive control. The control rules described below are inspired by the Braitenburg vehicle [3]. The Braitenburg *love vehicle* defines a reactive control mechanism that directs a differential-drive robot to a certain target. A similar behavior is required in the system; however, the love vehicle's control mechanism is too simplistic and, in some start configurations, tends to converge to the goal very slowly. We devised a modified set of reactive control formulae that allows for effective adjustment of the control trajectory:

$$(v, \dot{\theta}) = \begin{cases} (\alpha \cdot \sin\theta, \beta \cdot \cos\theta) \text{ if } |\theta| < \frac{\pi}{4} \text{ or } |\theta| > \frac{3\pi}{4} \\ (0, \mathrm{sgn}(\theta) \times \beta_0) \text{ otherwise} \end{cases}$$

where $v$ and $\dot{\theta}$ are the desired translational and rotational velocities, respectively, $\theta$ is the direction of the target relative to the robot $(-\pi < \theta < \pi)$, $\beta_0$ is the in-place rotational velocity, and $\alpha$ and $\beta$ are the base translational and rotational velocities, respectively. The translational and rotational velocities can be translated to differential drive parameters via a simple, invertible linear transform. This set of control formulae differs from the love vehicle in that it takes into account the orientation of the robot with respect to the target and explicitly adds rotational control. This set of control rules implicitly allows for heading independence, i.e., the control rules allow for both forward and backward movements, whichever one is most efficient to execute. Figure 3 shows an actual run of the reactive control algorithm described above.

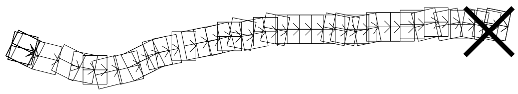

**Fig. 3.** Sample trace of the execution of the reactive control algorithm. The target point is marked with a cross.

**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 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 4(a) illustrates this behavior.

When using the ball interception behavior (Figure 4(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

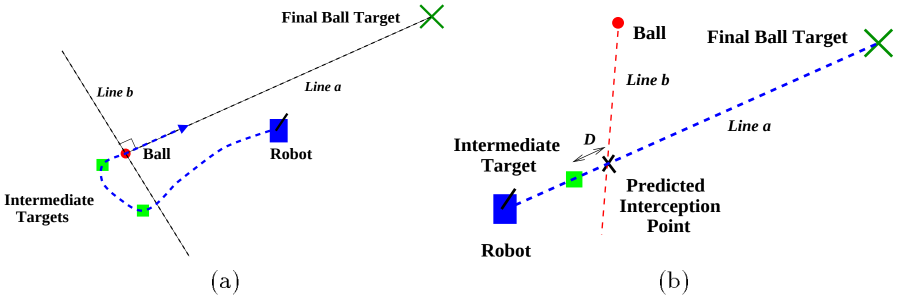


**Fig. 4.** Single-agent behaviors to enable team collaboration (a) Ball collection (aiming for a pass or to the goal); (b) Ball interception (receiving a pass).

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 intercept the ball towards the target.

**Obstacle avoidance**

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. In the event that an obstacle blocks the direct path to the goal location, the robot aims to one side of the obstacle until it is in a position such that it can move directly to its original goal. Rather than planning the entire path to the goal location at once, the robot just looks ahead to the first obstacle in its way under the assumption that other robots are continually moving around. Using the reactive control described above, the robot continually reevaluates its target position. For an illustration, see Figure 5. The robot starts by trying to go straight towards its final target along line a.

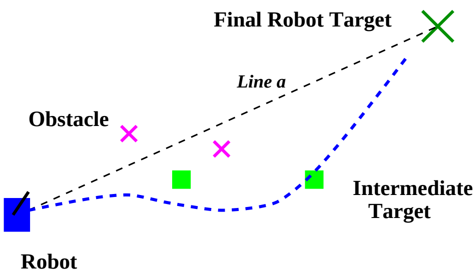


**Fig. 5.** Obstacle avoidance through dynamic generation of intermediate targets.

When it comes across an obstacle within a certain distance of itself and of line a, it aims at an intermediate target to the side, and slightly beyond the obstacle. The robot goes around the obstacle the short way, unless it is at the edge of the field. Using reactive control, the robot continually recomputes line a until the obstacle is no longer in its path. As it comes across further obstacles, it aims at additional intermediate targets until it obtains an unobstructed path to the final target.

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.

### 4.2 Multiagent Behaviors

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 multiagent 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, multiagent 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 [14].

**Positions, Formations, and Active Modes**

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.

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. For illustration, see Figure 6.

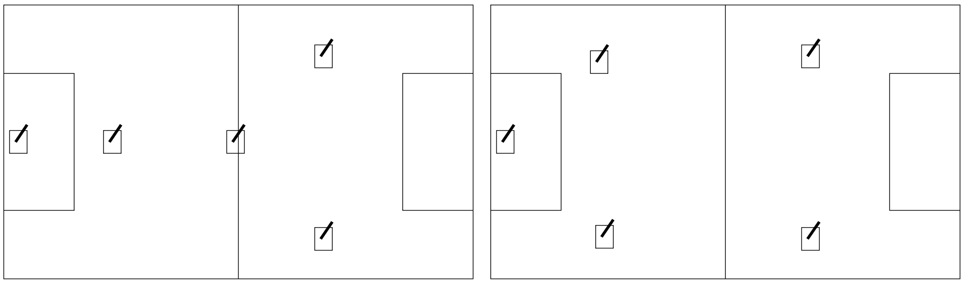

**Fig. 6.** Two different defined formations. Notice that several of the positions are reused between the two formations.

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," [14] 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. 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.

**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 7). 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 7):

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 7).
   - 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 7, 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-demoninator*,$y$).
3. return *obstruction-free-index*.

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.

**The Goalkeeper**

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

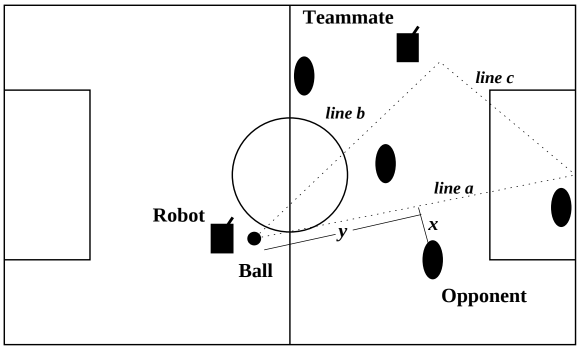


**Fig. 7.** Run-time pass evaluation is based on position of opponents.

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, as illustrated in Figure 8. Our goalkeeper robot currently take into account the predicted velocity and direction of the ball to select its moves.

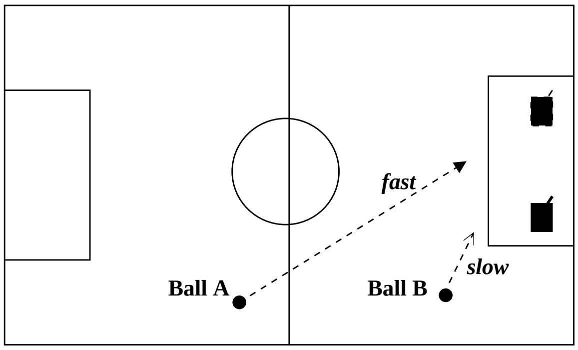


**Fig. 8.** Goalkeeping.

## 5 Discussion and Conclusion

CMUnited-97 successfully demonstrated the feasibility and effectiveness of teams of multiagent 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 multiagent reasoning. CMUnited is an example of a fully implemented multiagent system in which the loop is closed. In addition, we implemented interesting strategic behaviors, including agent collaboration and real-time evaluation of alternative actions.

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 multiagent capabilities of the CMUnited-97 robotic soccer team. Table 1 shows the results of the games at RoboCup-97.

| Opponent | Score |
|---|---|
| NAIST, Institute of Science and Technology, Japan | 5-0 |
| MICROB, University of Paris VI , France | 3-1 |
| University of Girona, Catalonia, Spain | 2-0 |
| NAIST, Japan (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.

In total, CMUnited-97 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. We refined the goalkeeper's goal behavior, as presented in this paper, following the observation of our goalkeeper's error.

As the matches proceeded, spectators noticed many of the team behaviors described in the paper. 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: one robot 1 passed to a second robot 2, which passed back to robot 1; then robot 1 passed to a third robot 3, which shot the ball into the goal.

In general, the robots' behaviors were visually appealing and entertaining to the spectators. Several people attained a first-hand appreciation for the difficulty of the task as we let them try controlling a single robot with a joystick program that we developed. All of these people (several children and a few adults) found it quite difficult to maneuver a single robot well enough to direct a ball into an open goal. These people in particular were impressed with the facility with which the robots were able to autonomously pass, score, and defend.

We are aware that many issues are clearly open for further research and development. We are currently systematically identifying them and addressing them towards our next team version. In particular, we are planning on enhancing the robot's behaviors by using machine learning techniques. We are currently developing techniques to accumulate and analyze real robot data.

## Acknowledgements

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 and in part by the 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 Air Force, of the Department of the Navy, Office of Naval Research or the United States Government.

## References

1. Minoru Asada, Yasuo Kuniyoshi, Alexis Drogoul, Hajime Asama, Maja Mataric, Dominique Duhaut, Peter Stone, and Hiroaki Kitano. The RoboCup physical agent challenge: Phase-i. *Applied Artificial Intelligence*, 12, 1998.
2. Minoru Asada, Shoichi Noda, Sukoya Tawaratumida, and Koh Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.
3. V. Braitenburg. *Vehicles – experiments in synthetic psychology*. MIT Press, 1984.
4. Kikuo Fujimura. *Motion Planning in Dynamic Environments*. Springer-Verlag, 1991.
5. Kwun Han and Manuela Veloso. Physical model based multi-objects tracking and prediction in robosoccer. In *Working Note of the AAAI 1997 Fall Symposium*. AAAI, MIT Press, 1997.
6. 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.
7. 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.
8. Jean-Claude Latombe. *Robot motion planning*. Kluwer Academic Publishers, 1991.
9. 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.
10. Itsuki Noda. Soccer server : a simulator of RoboCup. In *Proceedings of AI symposium '95*, pages 29–34. Japanese Society for AI, December 1995.
11. Michael K. Sahota. Real-time intelligent behaviour in dynamic environments: Soccer-playing robots. Master's thesis, University of British Columbia, 1993.
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. The CMUnited-97 simulator team. In Hiroaki Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 387–397. Springer Verlag, Berlin, 1998.