The CMUnited-98 Champion Small-Robot Team

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

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{veloso,mhb,sorin,kwunh,pstone}@cs.cmu.edu
http://www.cs.cmu.edu/˜robosoccer

Abstract. In this chapter, we present the main research contributions of our champion CMUnited-98 small robot team. The team is a multi-agent robotic system with global perception, and distributed cognition and action. We describe the main features of the hardware design of the physical robots, including differential drive, robust mechanical structure, and a kicking device. We briefly review the CMUnited-98 global vision processing algorithm, which is the same as the one used by the previous champion CMUnited-97 team. We introduce our new robot motion algorithm which reactively generates motion control to account for the target point, the desired robot orientation, and obstacle avoidance. Our robots exhibit successful collision-free motion in the highly dynamic robotic soccer environment. At the strategic and decision-making level, we present the role-based behaviors of the CMUnited-98 robotic agents. Team collaboration is remarkably achieved through a new algorithm that allows for team agents to anticipate possible collaboration opportunities. Robots position themselves strategically in open positions that increase passing opportunities. The chapter terminates with a summary of the results of the RoboCup-98 games in which the CMUnited-98 small robot team scored a total of 25 goals and suffered 6 goals in the 5 games that it played.

1 Introduction

The CMUnited-98 small-size robot team is a complete, autonomous architecture composed of the physical robotic agents, a global vision processing camera overlooking the playing field, and several clients as the minds of the small-size robot players. Fig. 1 sketches the building blocks of the architecture.

[0] This research is sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement numbers F30602-97-2-0250 and F30602-98-2-0135, and in part by the Department of the Navy, Office of Naval Research under contract number N00014-95-1-0591. The 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.

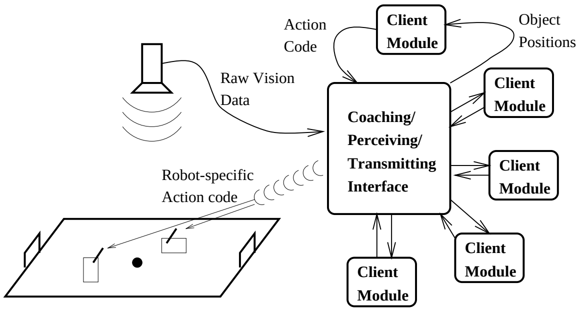


Fig. 1. The CMUnited architecture with global perception and distributed reaction.

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 make decisions. Actions are motion commands that are sent by the off-board controller through radio frequency 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. Motion is not perfectly executed due to inherent mechanical inaccuracies and unforeseen interventions from other agents. The effects of the actions are therefore uncertain.

The physical robots themselves are of size 15cm $\times$ 12cm $\times$ 10cm. Fig. 2 shows our robots. A differential drive mechanism is used in all of the robots. Two motors with integrated gear boxes are used for the two wheels. Differential drive was chosen due to its simplicity and due to the size constraints. The size of our robots conforms to RoboCup Competition rules[1]. Employing the differential drive mechanism means that the robot is non-holonomic, which makes the robot control problem considerably more challenging.

Although it may be possible to fit an on-board vision system onto robots of small size, in the interest of being able to quickly move on to strategic multiagent issues, the CMUnited-98 teams uses a global vision system. The fact that perception is achieved by a video camera overlooking the complete field offers an opportunity to get a global view of the world state. This setup may simplify the sharing of information among multiple agents, but it also presents a challenge for reliable and real-time processing of the movement of multiple mobile objects, namely the ball, five robots on our team, and five robots on the opponent's team.

This chapter presents the main technical contributions of our CMUnited-98 small robot team. It is organized as follows. Section 2 briefly overviews the hardware design of the robots. Section 3 describes the vision processing algorithm. Section 4 presents the motion planning approach for our robots including path planning to intercept moving targets and obstacle avoidance. Section 5 in-

[1] see http://www.robocup.org/RoboCup/

Fig. 2. The CMUnited-98 robots.

troduces the individual and team behaviors of the CMUnited-98 robots. We introduce the novel concept of *anticipation* which allows for the robots to effectively receive passes from teammates. Section 6 summarizes the results of the RoboCup-98 games and Section 7 draws conclusions.

2 Hardware

The CMUnited-98 robots are entirely new constructions built upon our experience in 1997. The new robots represent an upgrade of our own-built CMUnited-97 robots [8]. In this section, we present some of the 'features of our robot design and refer to the differences with the CMUnited-97 robots when appropriate.

The robots of the CMUnited-98 team were designed according to the following principles: simple but robust, reliable electronics, and modularized mechanics. These goals were partially achieved in CMUnited-97 and some of the design was retained. We continue to use the Motorola HC11 8-bit micro-controller running at 8 MHz and the 40 KBd Radiometrix radio module. Improvements were made in two major areas: motors and control, and the mechanical chassis.

2.1 Motors and Control

CMUnited-98 uses two high-torque, 6V DC, geared motors, which are overpowered and use a simple PWM control. It is interesting to note that this represents a simpler design than our CMUnited-97 robots which used hardwired motion controllers where the control loop was implemented in hardware as a Proportional Integral Derivative (PID) filter and associated motor encoders. We realized that, although this design provided accurate navigation, it was not easily interruptible. We found that these interrupts were often needed in the highly dynamic robotic soccer environment. Hence, in the CMUnited-98 robots, the closed-loop motion control is achieved through software using visual feedback.

The new implementation only makes use of a "WRITE" operation to the corresponding 8-bit register for each requested change in the speed of the motors. Additionally, a "safety mechanism" guarantees that a "STOP" command is issued locally by on-board resources in case of a radio link malfunction (i.e., when the next valid command is not received within a preset time interval).

The main advantages obtained by implementing these changes are the simplification of the low level motion control, the simplification of the electronic circuitry design, and the reduction of the space needed by the motion control functions in the micro-controller's 2Kb EEPROM.

2.2 Mechanical Chassis

In designing the mechanical structure of the CMUnited-98 robots, we focused on modularity and robustness. The final design was a robot with a very compact and powerful mobile base. It includes a single battery module supplying three independent power paths (for the main-board, motors, and radio modules.) It also includes a single board containing all the required electronic circuitry, with multiple add-on capabilities. This was all combined in a layered design within an aluminum and plastic Frame. In addition, each of the modules within this design is completely interchangeable.

The mobile base module includes a kicking device driven by a DC motor. This motor is hardware activated by an array of four infrared sensors with adjustable sensing range which can be enabled or disabled by the software control. The circular motion of the kicker motor was transformed to linear motion using a rack-pinion system. The push-pull effect of the kicking device was implemented by alternating the voltage polarity at the DC motor. The push, stop and pull stages of the kicking action were all timed, which eliminated the need for control sensors. Other implementations for the kicking device were considered, like a dual front/back kicker driven by flexible strings, or a push-pull DC solenoid. Neither of these met the size, power and precision requirements for a reliable kicking system.

The hardware design proved to be challenging. By building upon our experience with CMUnited-97, CMUnited-98 successfully achieves a robust team of physical robots.

3 Vision Processing

The CMUnited-98 vision module is the same as the one used in the CMUnited-97 team. It successfully implements a very reliable and fast image processing algorithm. In this chapter, we focus mainly our detection and association algorithm which we present in more detail than in our description of the CMUnited-97 team [8]. In particular, in the setting of a robot soccer game, the ability to detect merely the locations of objects on the field is often not enough. We use a Kalman-Bucy filter to successfully predict the movement of the ball[8], which is very suitable since the detection of the ball's location is noisy.

3.1 Color-based Detection

The vision requirements for robotic soccer have been examined by different researchers [3, 4]. Both on-board and off-board systems have appeared in recent years. All have found that the response necessary for soccer robots requires a vision system with a fast processing cycle. However, due to the rich visual input, researchers have found that dedicated processors or even DSPs are often needed [1, 3]. Our current system uses a frame-grabber with frame-rate transfer from a 3CCD camera.

The RoboCup rules specify colors for different objects in the field and these are used as the major cue for object detection. The RoboCup rules specify a green color field with specific white markings. The ball is an orange golf ball. It also specifies a yellow or blue colored circle on the top of the robots, one color for each team (see Fig. 2.) A single color patch on the robot is not enough to provide orientation information. Thus, an additional pink color patch was added to each of our robots. These colors can be differentiated reliably in color-space.

The set of detected patches are initially unordered. The detected color patches on the tops of the robots are then matched by their distance. Using the constant distance between the team-color (blue or yellow) and the pink orientation patch, our detection algorithm matches patches that are this distance apart. Two distance-matched patches are detected as a robot.

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 consistent with the ones on our robots are discarded. This technique filters off most "salt and pepper" noise. Second, by adding the distance matching mechanism described above, false detections are practically eliminated.

3.2 Data Association

The detection scheme described above 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 5 different robots by hue, 5 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 our team look identical to the visual system. With visually homogeneous objects distinguishing between them in any given frame is not possible. Data association addresses this problem by retaining robot identification between 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 start time. However, problems arise when subsequent frames are processed, the locations of the robots have changed due to robot movements. 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 constant frame rate. By measuring the maximum robot velocity, we can know that in subsequent frames, the robot is not able to move out of a 5cm radius circular region. This provides the basis of our association technique.

3.3 Greedy Association

With these assumptions in mind, a minimum distance scheme can be used to retain association between consecutive frames. For each frame, association is maintained by searching for objects with a minimum displacement. Current robot positions are matched with the closest positions from the previous frame.

The following is the pseudo-code of a greedy association procedure:

let $prev[1..n]$ be the array of robot locations from the previous frame
let $cur[1..m]$ be the array of robot locations from the current frame
let ma be triangular array of size $n-1$ s.t.
$ma[i][j] = \text{dist}(prev[i], cur[j])$
for $i := 1$ to m **do**
    find smallest element $ma[i][j]$
    save (i, j) as a matched pair
    set all elements in row i and column j to be ∞
end
if $m < n$ **then**
    forall $prev[i]$ unmatched, save $(prev[i], prev[i])$
return the set of saved pairs as the set of matchings.

This 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 or two undetected (i.e.. in the pseudo-code $m < n$). In this case, some locations will be left unmatched. The unmatched location will then be carried over to the current frame, and the robots corresponding to this location will be assumed to be stationary for this one frame.

This algorithm was implemented and was used in RoboCup-97 and RoboCup-98. Although the implementation was very robust, we present an improvement that allows for a globally optimal association.

3.4 Globally Optimal Association

The greedy association algorithm, as described above, fails in some cases. An example is the one illustrated in Fig. 3. In the figure, a greedy algorithm incorrectly matches the closest square and circular objects.

An improved algorithm was devised to handle the situation depicted above. The new algorithm generates all possible sets of matching and calculates the

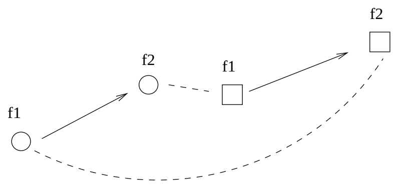


Fig. 3. A case in which greedy association fails but global optimal association performs well. The arrow indicates the actual object displacement over subsequent frames f1 and f2. The dotted lines indicate the wrong matches returned by greedy association.

fitness of each of the complete sets according to the following least square criteria:

$$\sum_{i=1}^{N}(dist(prev_i, cur_i))^2,$$

where $(prev_i, cur_i)$ are the i^{th} matching pair. And the function $dist(x, y)$ is the Euclidean distance. The set of matches that minimizes the above criteria is selected.

While these algorithms do not theoretically guarantee perfect associations, in particular with noisy perception and cluttered environments, the implementation has proven to be very robust. Our vision processing approach worked perfectly during the RoboCup-98 games. We were able to detect and track 11 objects (5 teammates, 5 opponents and a ball) at 30 frames/s. Also the prediction provided by the Kalman-Bucy filter is an integral factor in our robots' strategic decisions.

4 Motion Control

The goal of our low level motion control is to be as fast as possible while remaining accurate and reliable. This is challenging due to the lack of feedback from the motors, forcing all control to be done using only visual feedback. Our motion control algorithm is robust. It addresses stationary and moving targets with integrated obstacle avoidance. The algorithm makes effective use of the prediction of the ball's trajectory provided by the Kalman-Bucy filter.

We achieve this motion control functionality by a reactive control mechanism that directs a differential drive robot to a target configuration. Though based on the CMUnited-97's motion control [8], CMUnited-98 includes a number of major improvements. The target configuration for the motion planner has been extended. The target configuration includes: (i) the *Cartesian position*; and (ii) the *direction* that the robot is required to be facing when arriving at the target position. Obstacle avoidance is integrated into this controller. Also, the target configuration can be given as a function of time to allow for the controller to reason about intercepting the trajectory of a moving target.

4.1 Differential Drive Control for Position and Direction

CMUnited- 98's basic control rules were improved from those used in CMUnited-97. The rules are a set of reactive equations for deriving the left and right wheel velocities, v_l and v_r, in order to reach a target position, (x^*, y^*):

$$\begin{aligned} \Delta &= \theta - \phi \\ (t, r) &= (\cos^2 \Delta \cdot \mathrm{sgn}(\cos \Delta), \sin^2 \Delta \cdot \mathrm{sgn}(\sin \Delta)) \\ v_l &= v(t - r) \\ v_r &= v(t + r), \end{aligned} \tag{1}$$

where θ is the direction of the target point (x^*, y^*), ϕ is the robot's orientation, and v is the desired speed (see Fig. 4(a))[2].

We extend these equations for target configurations of the form (x^*, y^*, ϕ^*), where the goal is for the robot to reach the specified target point (x^*, y^*) while facing the direction ϕ^*. This is achieved with the following adjustment:

$$\theta' = \theta + \min\left(\alpha, \tan^{-1}\left(\frac{c}{d}\right)\right),$$

where θ' is the new target direction, α is the difference between our angle to the target point and ϕ^*, d is the distance to the target point, and c is a clearance parameter (see Fig. 4(a).) This will keep the robot a distance c from the target point while it is circling to line up with the target direction, ϕ^*. This new target direction, θ', is now substituted into equation 1 to derive wheel velocities.

In addition to our motion controller computing the desired wheel velocities, it also returns an estimate of the time to reach the target configuration, $\hat{T}(x^*, y^*, \phi^*)$. This estimate is a crucial component in our robot's strategy. It is used both in high-level decision making, and for low-level ball interception, which is described later in this section. For CMUnited-98, $\hat{T}(x^*, y^*, \phi^*)$ is computed using a hand-tuned linear function of d, α, and Δ.

4.2 Obstacle Avoidance

Obstacle avoidance was also integrated into the motion control. This is done by adjusting the target direction of the robot based on any immediate obstacles in its path. This adjustment can be seen in Fig. 4(b).

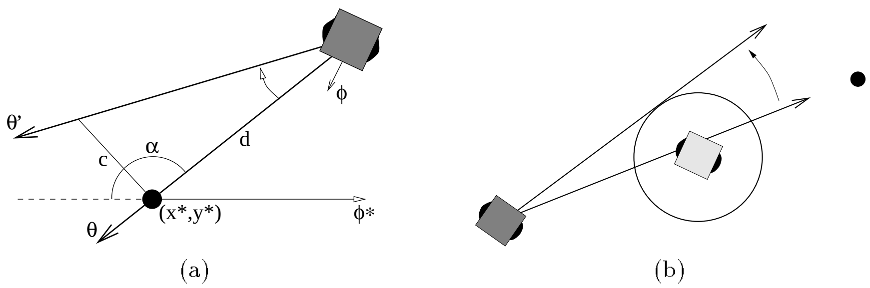


Fig. 4. (a) The adjustment of θ to θ' to reach a target configuration of the form (x^*, y^*, ϕ^*); (b) The adjustment to avoid immediate obstacles.

[2] All angles are measured with respect to a fixed coordinate system.

If a target direction passes too close to an obstacle, the direction is adjusted to run tangent to the a preset allowed clearance for obstacles. Since the motion control mechanism is running continuously, the obstacle analysis is constantly replanning obstacle-free paths. This continuous replanning allows for the robot to handle the highly dynamic environment and immediately take advantage of short lived opportunities.

4.3 Moving Targets

One of the real challenges in robotic soccer is to be able to control the robots to intercept a moving ball. This capability is essential for a high-level ball passing behavior. CMUnited-98's robots successfully intercept a moving ball and several of their goals in RoboCup-98 were scored using this capability.

This interception capability is achieved as an extension of the control algorithm to aim at a stationary target. Fig. 5(a) illustrates the control path to reach a stationary target with a specific direction, using the control mechanism described above. Our extension allows for the target configuration to be given as a function of time, where $t = 0$ corresponds to the present,

$$f(t) = (x^*, y^*, \phi^*).$$

At some point in the future, t_0, we can compute the target configuration, $f(t_0)$. We can also use our control rules for a stationary point to find the wheel velocities and estimated time to reach this hypothetical target as if it were stationary. The time estimate to reach the target then informs us whether it is possible to reach it within the allotted time. Our goal is to find the nearest point in the future where the target can be reached. Formally, we want to find,

$$t^* = \min\{t > 0 : \hat{T}(f(t)) \leq t\}.$$

After finding t^*, we can use our stationary control rules to reach $f(t^*)$. In addition we scale the robot speed so to cross the target point at exactly t^*.

Unfortunately, t^*, cannot be easily computed within a reasonable time-frame. We approximate this value, t^*, by discretizing time with a small time-step. We then find the smallest of these discretized time points that satisfies our estimate constraint. An example of this is shown in Fig. 5(b), where the goal is to hit the moving ball.

(a) (b)

Fig. 5. (a) Control for stationary target. (b) Control for moving target.

The target configuration as a function of time is computed using the ball's predicted trajectory. Our control algorithm for stationary points is then used to find a path and time estimate for each discretized point along this trajectory, and the appropriate target point is selected.

5 Strategy

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

CMUnited-97 introduced 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 [5, 8]. CMUnited-98 builds upon this team architecture by defining a set of roles for the agents. It also introduces improvements within this architecture to help address the highly dynamic environment.

CMUnited-98 uses the following roles: goalkeeper, defender, and attacker. The formation used throughout RoboCup-98 involved a single goalkeeper and defender, and three attackers.

5.1 Goalkeeper

The ideal goalie behavior is to reach the expected entry point of the ball in the goal *before* the ball reaches it. Assuming that the prediction of the ball trajectory is correct and the robot has a uniform movement, we can state the ideal goalie behavior: given the predicted v_g and v_b as the velocities of the goalie and of the ball respectively, and d_g and d_b as the distances from the goalie and the ball to the predicted entry point, then, we want $\frac{d_g}{v_g} = \frac{d_b}{v_b} - \epsilon$, where ϵ is a small positive value to account for the goalie reaching the entry point slightly before the ball.

Unfortunately, the ball easily changes velocity and the movement of the robot is not uniform and is uncertain. Therefore we have followed a switching behavior for the goalie based on a threshold of the ball's estimated trajectory.

If the ball's estimated speed is higher than a preset threshold, the goalie moves directly to the ball's predicted entry goal point. Otherwise, the goalie selects the position that minimizes the largest portion of unobstructed goal area. This is done by finding the location that bisects the angles of the ball and the goal posts as is illustrated in Fig. 6.

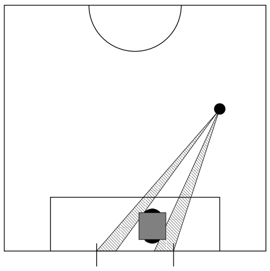


Fig. 6. The goalkeeper positions itself to minimize the unobstructed goal area.

The use of the predicted ball's velocity for the goalie's behavior was shown to be very effective in the RoboCup-98 games. It was particularly appropriate for defending a penalty shot, due to the accuracy of the predicted ball's trajectory when only one robot is pushing the ball.

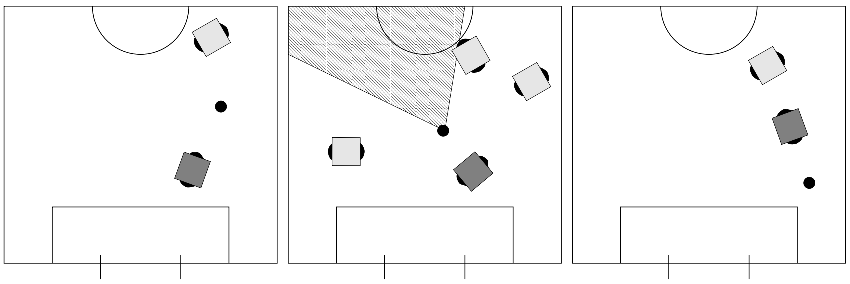
5.2 Defender

The CMUnited-97's team did not have a well-specified defender's role, but our experience at RoboCup-97 made us understand that the purpose of a defending behavior is two-fold:

1. to stop the opponents from scoring in our goal; and
2. to not endanger our own goal.

The first goal is clearly a defender's role. The second goal comes as the result of the uncertain ball handling by the robots. The robots can easily push (or touch) the ball unexpectedly in the wrong direction when performing a difficult maneuver.

To achieve the two goals, we implemented three behaviors for the defender. *Blocking*, illustrated in Fig. 7(a), is similar to the goalkeeper's behavior except that the defender positions itself further away from the goal line. *Clearing*, illustrated in Fig. 7(b), pushes the ball out of the defending area. It does this by finding the largest angular direction free of obstacles (opponents and teammates) that the robot can push the ball towards. *Annoying*, illustrated in Fig. 7(c), is somewhat similar to the goalkeeping behavior except that the robot tries to position itself between the ball and the opponent nearest to it. This is an effort to keep the opponent from reaching the ball.



(a) Blocking (b) Clearing (c) Annoying

Fig. 7. The defender's behaviors. The dark and light robots represent the defender and the opponents respectively.

Selecting when each of these behaviors is used is very important to the effectiveness of the defender. For example, clearing the ball when it is close to our own goal or when it can bounce back off another robot, can lead to scoring in our own goal. We used the decision tree in Fig. 8 to select which action to perform based on the current state.

The two attributes in the tree, namely *Ball Upfield* and *Safe to Clear*, are binary. *Ball Upfield* tests whether the ball is upfield (towards the opponent's goal) of the defender. *Safe to Clear* tests whether the open area is larger than a preset angle threshold. If *Ball Upfield* is false then the ball is closer to the goal than the defender and the robot *annoys* the attacking robot. The CMUnited-98's annoying behavior needs to select one particular opponent robot to annoy. For example, when two opponent robots attack simultaneously, the current annoying behavior is able to annoy only one of them. We are planning on further improving this behavior for RoboCup-99.

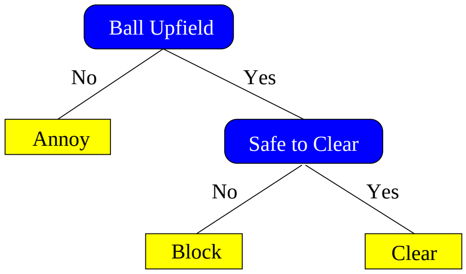


Fig. 8. The decision tree heuristic used by the defender to select its behavior.

If *Ball Upfield* is true, the defender clears or blocks, depending on the value of *Safe to Clear*. Clearing was shown to be very useful at RoboCup-98, with even a couple of our goals scored directly by a clearing action of the defender.

5.3 Attackers - Active Teammate and Anticipation

Attacking involves one of the best opportunities for collaboration, and much of the innovation of CMUnited-98 has been developing techniques for finding and exploiting these opportunities.

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 another task or assist the active agent(s). This simplistic distinction between active and passive agents to capture teamwork was realized in CMUnited-97. The agent that goes to the ball is viewed as the active agent, while the other teammates are passive.

CMUnited-98 significantly extends this simplistic view in two ways: (i) we use a decision theoretic algorithm to select the active agent; and (ii) we use a technique for passive agents *to anticipate* future collaboration. Passive agents are therefore not actually "passive;" instead, they actively *anticipate* opportunities for collaboration. In CMUnited-98 this collaboration is built on robust individual behaviors.

Individual Behaviors. We first developed individual behaviors for passing and shooting. Passing and shooting in CMUnited-98 is handled effectively by the motion controller. The target configuration is specified to be the ball (using its estimated trajectory) and the target direction is either towards the goal or another teammate. This gives us robust and accurate individual behaviors that can handle obstacles as well as intercepting a moving ball.

Decision Theoretic Action Selection. Given the individual behaviors, we must select an active agent and appropriate behavior. This is done by a decision theoretic analysis using a single step look-ahead. With n agents this amounts to n^2 choices of actions involving shooting or a pass to another agent followed by that agent shooting. An estimated probability of success for each pass and shot is computed along with the time estimate to complete the action, which is provided by the motion controller. A value for each action is computed,

$$\text{Value} = \frac{\text{Pr}_{\text{pass}}\text{Pr}_{\text{shoot}}}{\text{time}}.$$

The action with the largest value is selected, which determines both the active agent and its behavior. Table 1 illustrates an example of the values for the selection considering two attackers, 1 and 2.

		Probability of Success			
Attacker	Action	Pass	Shoot	Time(s)	Value
1	Shoot	–	60%	2.0	0.30
1*	Pass to 2	60%	90%	1.0	0.54
2	Shoot	–	80%	1.5	0.53
2	Pass to 1	50%	40%	0.8	0.25

Table 1. Action choices and computed values are based on the probability of success and estimate of time. The largest-valued action (marked with an *) is selected.

It is important to note that this action selection is occurring on each iteration of control, i.e., approximately 30 times per second. The probabilities of success, estimates of time, and values of actions, are being continuously recomputed. This allows for quick changes of actions if shooting opportunities become available or collaboration with another agent appears more useful.

Dynamic Positioning (SPAR). Although there is a clear action to be taken by the active agent, it is unclear what the passive agents should be doing. Although, in a team multiagent system such as robotic soccer, success and goal achievement often depends upon collaboration; so, we introduce in CMUnited-98, the concept that team agents should not actually be "passive."

CMUnited-97's 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. This is a first-level of single-agent anticipation towards a better individual goal achievement [7].

However, for CMUnited-98, we introduce a team-based notion of *anticipation*, which goes beyond individual single-agent anticipation. The passive team members 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 that of the ball, they are able to position themselves more usefully: they *anticipate* their future contributions to the team.

This strategic position takes into account the position of the other robots (teammates and opponents), the ball, and the opponent's goal. The position is found as the solution to a multiple-objective function with repulsion and attraction points. Let's introduce the following variables:

- 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;
- P - the desired position for the passive agent in anticipation of a pass.

Given these defined variables, we can then formalize our algorithm for strategic position, which we call SPAR for *Strategic Positioning with Attraction and Repulsion.* This extends similar approaches using potential fields [2], to our highly dynamic, multi-agent domain. The probability of collaboration is directly related to how "open" a position is to allow for a successful pass. SPAR maximizes the repulsion from other robots and minimizes attraction to the ball and to the goal, namely:

- *Repulsion* from opponents. Maximize the distance to each opponent: $\forall i, \max dist(P, O_i)$.
- *Repulsion* from teammates. Maximize the distance to other passive teammates: $\forall i, \max dist(P, T_i)$.
- *Attraction* to the 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. This approach has also been applied to the CMUnited-98 simulator team [6].

As each term in the multiple-objective function 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 different functions of each term. In our CMUnited-98 team, we weight the terms differently, namely w_{O_i}, w_{T_i}, w_B, and w_G, for the weights for opponents, teammates, the ball, and the goal, respectively. For CMUnited-98, these weights were hand tuned to create a proper balance. This gives us a weighted single-objective function:

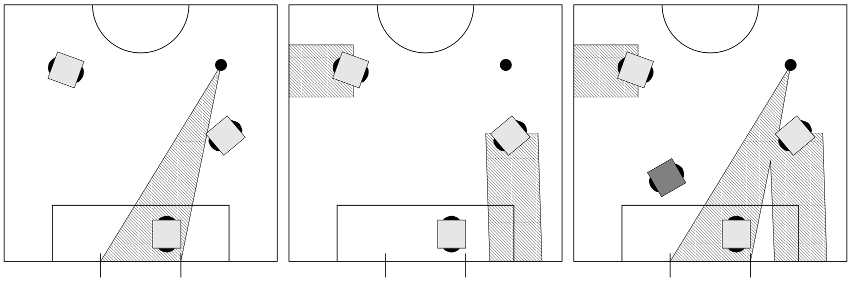
$$\max \left(\sum_{i=1}^{n} w_{O_i} dist(P, O_i) + \sum_{i=1}^{n} w_{T_i} dist(P, T_i) - w_B dist(P, B) - w_G dist(P, G) \right).$$

This optimization problem is then solved under a set of constraints:

- Do not block a possible direct shot from active teammate.
- Do not stand behind other robots, because these are difficult positions to receive passes from the active teammate.

The solution to this optimization problem under constraints gives us a target location for the "passive" agent. Fig. 9(a) and (b) illustrate these two sets of constraints and Fig. 9(c) shows the combination of these constraints and the resulting position of the anticipating passive teammate.

This positioning was very effective for CMUnited-98. The attacking robots very effectively and dynamically adapted to the positioning of the other robots.



(a) Don't block goal shot. (b) Avoid difficult collaboration. (c) Anticipate optimal position for collaboration.

Fig. 9. Constraints for the dynamic anticipation algorithm are represented as shaded regions; (a) and (b) show three opponents and the current position of the ball; (c) illustrates the position of the passive agent - dark square - as returned by SPAR.

The SPAR anticipation algorithm created a number of opportunities for passes and rebounds that often led to goals and other scoring chances.

In general, we believe that our approach represents a major step in team multiagent systems in terms of incorporating *anticipation* as a key aspect of teamwork.

6 Results

CMUnited-98 successfully defended our title of the Small Robot Champion at RoboCup-98 in Paris. The competition involved 11 teams from 7 different countries. It consisted of a preliminary round of two games, followed by the 8 advancing teams playing a 3-round playoff. CMUnited-98 won four of five games, sweeping the playoff competition, scoring a total of 25 goals scored and only 6 suffered. The individual results of these games are in Table 2.

Phase	Opponent	Affiliation	Score (CMU - Opp.)
round-robin	iXS	iXs Inc., Japan	16 – 2
round-robin	5DPO*	University of Porto, Portugal	0 – 3
quarter-final	Paris-8*	University of Paris-8, France	3 – 0
semi-final	Cambridge	University of Cambridge, UK	3 – 0
final	Roboroos	University of Queensland, Australia	3 – 1

Table 2. The scores of CMUnited-98's games at RoboCup-98. The games marked with an * were forfeited at half time.

There were a number of technical problems during the preliminary rounds, including outside interference with our radio communication. This problem was the worst during our game against 5DPO, in which our robots were often responding to outside commands just spinning in circles. This led to our forfeit at half time and a clear loss against 5DPO, a very good team which ended in third place at RoboCup-98. Fortunately, the communication problems were isolated and dealt with prior to the playoff rounds.

The three playoff games were very competitive and showcased the strengths of our team. Paris-8 had a strong defense with a lot of traffic in front of the goal. Our team's obstacle avoidance still managed to find paths and to create scoring chances around their defenders. The final two games were very close against very good opponents. Our interception was tested against Cambridge, and included blocking a powerful shot by their goalie, which was deflected back into their goal. The final game against Roboroos demonstrated the dynamic positioning, especially during the final goal, which involved a pass to a strategically positioned teammate.

7 Conclusion

The success of CMUnited-98 at RoboCup-98 was due to several technical innovations, including robust hardware design, effective vision processing, reliable time-prediction based robot motion with obstacle avoidance, and a dynamic role-based team approach. The CMUnited-98 team demonstrated in many occasions its collaboration capabilities which resulted from the robots' behaviors. Most remarkably, CMUnited-98 introduces the concept of *anticipation*, in which passive robots (not going to the ball) strategically position themselves using attraction and repulsion (SPAR) to maximize the chances of a successful pass.

The CMUnited-98 team represents an integrated effort to combine solid research approaches to hardware design, vision processing, and individual and team robot behaviors.

References

1. 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.
2. Jean-Claude Latombe. *Robot Motion Planning.* Kluwer, 1991.
3. 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.
4. Randy Sargent, Bill Bailey, Carl Witty, and Anne Wright. Dynamic object capture using fast vision tracking. *AI Magazine*, 18(1):65–72, Spring 1997.
5. 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.
6. 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.
7. Manuela Veloso, Peter Stone, and Kwun Han. CMUnited-97: RoboCup-97 small-robot world champion team. *AI Magazine*, 19(3):61–69, 1998.
8. Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim. The CMUnited-97 small-robot team. In Hiroaki Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 242–256. Springer Verlag, Berlin, 1998.

This article was processed using the LaTeX macro package with LLNCS style