CMUnited-98: RoboCup-98 Small-Robot World Champion 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

July 24, 1999

Abstract

The CMUnited small-robot team became the 1998 RoboCup small-robot league champion, repeating its 1997 victory. CMUnited-98 built upon the success of CMUnited-97, and involved a number of improvements. This article gives an overview of the CMUnited-98 team, focusing on this year's improvements. It concludes with the results of the RoboCup - 98 competition.

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 over-looking the playing field, and several clients as the minds of the small-size robot players. The global vision algorithm perceives the 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. Motion is not perfectly executed due to inherent mechanical inaccuracies and unforeseen interventions from other agents. This team competed in and won the 1998 RoboCup (Kitano *et al.* 1997) competition in Paris.

This article gives an overview of the CMUnited-98 robot team, focusing on improvements from the CMUnited-97 team. These improvements include a robust low-level control algorithm that handles a moving target with integrated obstacle avoidance, and active team collaboration at the strategic level. This paper concludes with results from the RoboCup-98 competition.

2 Hardware

The CMUnited-98 robots (shown in Figure 1) are entirely our new constructions built upon our experience in 1997 (Veloso *et al.* 1998). The new robots represent an upgrade of our previously built CMUnited-97 robots. Improvements were made in two major areas: motors and control, and the mechanical chassis, which includes a kicking device.

CMUnited-98 uses two high-torque, 6V DC, geared motors, which are overpowered and use a simple PWM control. This is a simpler design from our CMUnited-97 robots, which made use of motor encoders for hardware feedback.

* An extended version of this article appears in (Veloso *et al.* 1999a) and in (Veloso *et al.* 1999b).

† 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.

Figure 1: The CMUnited-98 robots.

Although our previous team had accurate navigation, it was not easily interruptible, which is necessary for operating in a highly dynamic environment. In the CMUnited-98 robots, the closed-loop motion control is achieved through software using only visual feedback.

In designing the mechanical structure of the CMUnited-98 robots, we focused on modularity and robustness. The final design includes a 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. 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, which is enabled or disabled by the software control. 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.

3 Vision Processing

The CMUnited-98 vision module remains largely the same as the one used in the CMUnited-97 team (Han & Veloso 1998). The algorithm successfully detects and tracks 11 objects (5 teammates, 5 opponents and a ball) at 30 frames/s. The algorithm determines a position and orientation for the robots. In addition a Kalman-Bucy filter (Kalman & Bucy 1961) is used as a predictor of the ball's trajectory. This prediction is an integral factor in our robots' control and strategic decisions.

4 Motion Control

Before developing strategic behaviors, the robots need a general control mechanism. This mechanism must reliably control the robot to a precise position on the field. The goal of our low level motion control mechanism 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 (Veloso *et al.* 1998), CMUnited-98 includes a number of major improvements. The target configuration for the motion planner has been extended to include both (i) the *Cartesian position*, and (ii) the *direction* that the robot is required to be facing when arriving at the target position. The motion controller algorithm drives the two-wheeled robot smoothly while including the following features: (i) obstacle avoidance is integrated into the controller; (ii) 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; and finally (iii) the motion controller returns an estimate of the time that the robot will achieve the desired target configuration.

4.1 Differential Drive Control

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. 2(a))[1].

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 θ and ϕ^*, d is the distance to the target point, and c is a clearance parameter (see Fig. 2(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. 2(b). 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.

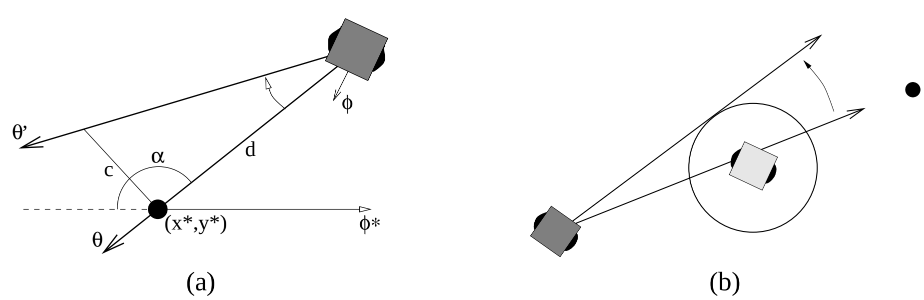


Figure 2: (a) The adjustment of θ to θ' to reach a target configuration of the form (x^*, y^*, ϕ^*); (b) The adjustment to avoid immediate obstacles.

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

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. 3(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,

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

where $t = 0$ corresponds to the present. 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 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 constraint. An example of this is shown in Fig. 3(b), where the goal is to hit the moving ball. 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.

(a) (b)

Figure 3: (a) Control for stationary target. (b) Control for moving target.

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 (Stone & Veloso 1998; Veloso *et al.* 1998). 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. The goaltender's behavior is similar to CMUnited-97's and is described in (Veloso *et al.* 1999a). This article describes the defender's behavior and the collaborative behaviors developed for the attackers.

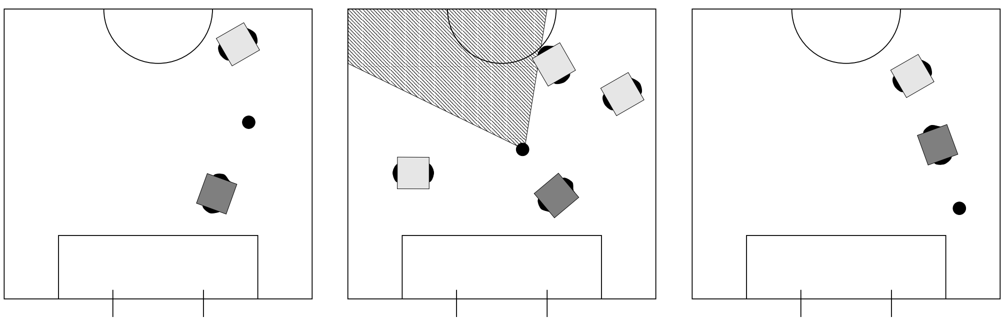
5.1 Defender

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 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. 4(a), is similar to the goalkeeper's behavior except that the defender positions itself further away from the goal line. *Clearing*, illustrated in Fig. 4(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. 4(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

Figure 4: 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. 5 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. Otherwise it either *clears* or *blocks* depending on the value of *Safe to Clear*.

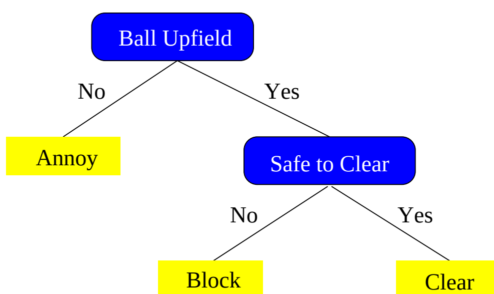


Figure 5: The decision tree heuristic used by the defender to select its behavior.

5.2 Active and Anticipating Attackers

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.

5.2.1 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.

5.2.2 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.

Attacker	Action	Probability of Success: Pass	Probability of Success: 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.

5.2.3 Dynamic Positioning (SPAR)

The selected action determines the behavior for the active agent, but it is unclear what the passive agents should be doing. CMUnited-98 introduced a new technique for the "passive" agents to strategically position themselves to anticipate future opportunities for collaboration. The algorithm for this positioning is called SPAR for *Strategic Positioning with Attraction and Repulsion.* This algorithm was also used successfully in the CMUnited-98 simulator team (Stone *et al.* 1999).

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 position of opponent $i = 1, ..., n$;
- T_i - the position of teammate, $i = 1, ..., n$;
- B - the 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 (Latombe 1991), 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.

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:

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

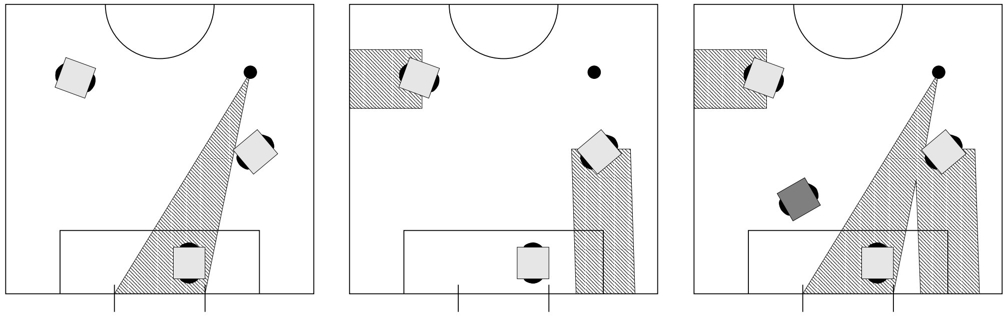
This optimization problem is then solved under a set of constraints:

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

The solution to this optimization problem with constraints gives us a target location for the "passive" agent. Fig. 6(a) and (b) illustrate these two sets of constraints and Fig. 6(c) shows the combination of these constraints and the resulting position of the anticipating passive teammate.

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 its five games, sweeping the playoff competition, with a total of 25 goals scored and only 6 suffered. The individual results of these games are in Table 2.



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

Figure 6: 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.

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 and 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 motion control with 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 goaltender, 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, a role-based team strategy, and in particular an anticipation algorithm to effectively respond to the dynamic environment towards increasing the opportunities for team collaboration. The CMUnited-98 team demonstrated in many occasions its robust motion control and teamwork capabilities. The CMUnited-98 team represents an integrated effort to combine solid research approaches to hardware design, vision processing, and individual and team robot behaviors. Our on-going research includes action policy learning from a crude robot simulator to the real robots, on-line robot recognition of the opponents' team strategy, and dynamic role and formation switching as a function of the opponent team.

References

Han, K., and Veloso, M. (1998). Reactive visual control of multiple non-holonomic robotic agents. In *Proceedings of the International Conference on Robotics and Automation*.

Kalman, R. E., and Bucy, R. S. (1961). New results in linear filter and prediction theory. *Journal of Basic Engineering*, pp. 95–108.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; and Osawa, E. (1997). RoboCup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents*. pp. 340–347.

Latombe, J.-C. (1991). *Robot Motion Planning*. (Boston: Kluwer Academic Publishers).

Stone, P., and Veloso, M. (1998). The CMUnited-97 simulator team. In Kitano, H. (ed.), *RoboCup-97: Robot Soccer World Cup I*. (Berlin: Springer Verlag).

Stone, P.; Veloso, M.; and Riley, P. (1999). The CMUnited-98 champion simulator team. In Asada, M., and Kitano, H. (eds), *RoboCup-98: Robot Soccer World Cup II*. (Berlin: Springer Verlag).

Veloso, M.; Stone, P.; and Han, K. (1998). CMUnited-97: RoboCup-97 small-robot world champion team. *AI Magazine*, **19**(3):61–69.

Veloso, M.; Bowling, M.; Achim, S.; Han, K.; and Stone, P. (1999). The CMUnited-98 champion small robot team. In Asada, M., and Kitano, H. (eds), *RoboCup-98: Robot Soccer World Cup II*. (Berlin: Springer Verlag).

Veloso, M.; Bowling, M.; Achim, S.; Han, K.; and Stone, P. (1999). Cmunited-98: A team of robotic soccer agents. In *Proceedings of IAAI-99*.