CMUnited: A Team of Robotic Soccer Agents Collaborating in an Adversarial Environment *

Manuela Veloso Peter Stone Kwun Han Sorin Achim
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{veloso,pstone,kwunh,sorin}@cs.cmu.edu
http://www.cs.cmu.edu/{˜mmv,˜pstone,˜kwunh}

Abstract

Robotic soccer involves multiple agents that need to collaborate in an adversarial environment to achieve specific objectives. In this paper, we describe the robotic agents and architecture that we have developed to enter RoboCup-97. The framework integrates high-level and low-level reasoning as an overall wireless communication system of small-size robots, an overhead vision camera for perception, a centralized interface computer, and several clients as the minds of the robot players. We present the mobile robot platform specifications, the different communication servers and links, the vision processing algorithm, and the control code that enables strategic collaboration between teammates.

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. Although realistic simulation environments exist [Noda, 1995; Sahota, 1993] and are useful, it is important to have some actual physical agents in order to address the full complexity of the task.

Robotic Soccer is an exciting domain for Intelligent Robotics 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 the Robotic Soccer domain is that it enables the direct comparison of different systems: they can be matched against each other in competitive tournaments. In particular, the system described here was designed specifically for RoboCup97 in which several robotic teams from around the world are competing on an “even playing field.” [Kitano *et al.*, 1997]. 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.

Along with the real robot competition, RoboCup97 will also include a simulator-based tournament using the Soccer Server system designed by Noda [Noda, 1995]. While we continue working on our real-world system, we have been concurrently developing learning techniques in simulation [Stone and Veloso, 1997; 1996]. We eventually hope to transfer these learning techniques to the real system as we develop a complete Robotic Soccer architecture.

This paper describes the overall architecture of our robotic soccer system. The combination of robust hardware, real-time vision, and intelligent control code represented a significant challenge which we were able to successfully meet. While the hardware and vision systems have been fixed for the current version of the robots, we continue to improve the robot control code.

2 Overview of the Architecture

The architecture of our system addresses the combination of high-level and low-level reasoning by viewing the

*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.

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 1 sketches the building blocks of the architecture.

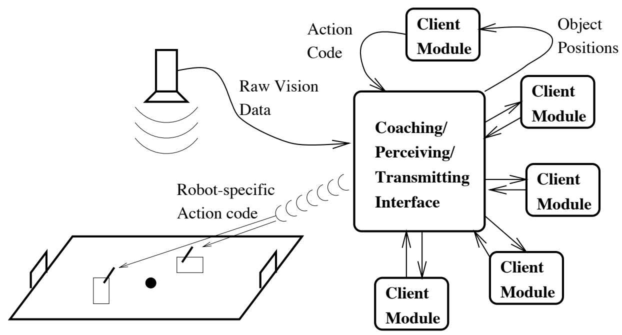

Figure 1: Our Robotic Soccer Architecture as a Distributed Deliberation and Reacting System.

Our architecture implements the overall robotic soccer system as a set of different platforms with different processing features. The small-size robots perform the physical navigation actions, decode commands, and can respond to positioning requests. Off-board computers perceive the environment through a camera, perform the high-level decision making and send commands to the small-size robots. Communication in our current system is done by radio frequency (RF). The complete system is fully autonomous consisting of the following processing cycle: the vision system perceives the dynamic environment, namely the positioning of the robots and the ball; the image is processed and transferred to the host computer that makes the perception available to the client modules; based on the perceived positioning of the agents and any other needed information about the state of the game (e.g. winning, losing, attacking), each client uses its strategic knowledge to decide what to do next; the client selects navigational commands to send to its corresponding robot agent; these commands are sent by the main computer to the robots through RF communication using the robot-specific action codes. Commands can be broadcast or sent directly to individual agents. Each robot has a self identification binary code that is used in the RF communication. This complete system is now fully implemented.

Figure 2 shows the architecture as a layered functional system. The protocols of communication between the layers are specified in terms of the modular inputs and outputs at each level.

Commands include positioning requests and navigation primitives, such as forward, backward, and turning moves at specific speeds. We may provide the small-size robots with transmitting ability for direct communication between agents and some local sensing capabilities. We will consider these alternatives based on the empirical experience that we are currently gathering using the vision camera. We illustrate now our work along the different aspects of the architecture. Section 3 describes the low level hardware including everything from the actual robots to the control code interface. Section 4 details the method by which the robots obtain their sensory inputs. Section 5 presents the strategic robot behaviors.

Layer	Entity	Input	Output
Behavioral	Robots	Commands	Actual Moves
Perceptual	Vision Camera	View of Field	Robots & Ball Coordinates
Strategic	Computer	Robots & Ball Coordinates	Commands

(a)

Layered Strategic Level	Examples
Robot–ball	intercept
One-to-one player	pass, aim
One-to-many player	pass to teamplayer
Action selection	pass or dribble
Team collaboration	strategic positioning

(b)

Figure 2: (a) The functional layers of the architecture, and (b) strategic level decomposition.

3 The Hardware

We define the hardware component of the system as the actual robots coupled with the systems that directly manage the radio link between the off-board computer and the robots. These systems include an off-board microcontroller and a command server.

3.1 The Robots

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.

Dimensions

- Length: limited by the size of the circuit board, which is 9.4cm; actual 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 × 12cm, the elongated frame – 18cm × 10cm, and a wide frame – 15cm × 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 development 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. We are currently contemplating the possibility of using very low power Surface Mount Technology (SMT) devices, in order to get smaller printed circuit boards and lighter battery packs. Decreasing the weight of the robots would alleviate the load on the motors, allowing them to move faster.

Electronics

- CPU: 8-bit micro-controller
- Run Time: 2.5 hours, with full-charged batteries
- On-board memory: 512 bytes RAM, 2 Kbyte EEPROM
- Radio: half-duplex FM 418 MHz

The current electronics are mounted on two separate circuit boards: one of the dimensions of the main frame includes the micro-controller, the motion processors, the motor drivers, and the power supply circuitry; the other one, of smaller dimensions, includes only the radio communication circuitry.

The dimensions of the boards for the electronics 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. In addition, each robot has a cover top with color code that enables detection by an external camera (see Section 4).

On-board Power Source

The power supplies consists of three sets of batteries that provide power to the different modules:

- Motors: powered by one 7.2V/1Ah NiCd Rechargeable battery pack;
- Main Circuit Board: powered by one 7.2V/1Ah NiCd Rechargeable battery pack;
- Radio: powered by one 7.2V/120mAh NiCd rechargeable battery.

3.2 Off-board Microcontroller

Between the off-board computer, which handles the robot control code, and the RF transmitter is a microcontroller programmed to convert directional movement commands (such as turn right 45 degrees or move forward at a given speed) into the actual bytes that are understood by the motor drivers. These bytes are then encoded and padded with stabilizing bytes and a checksum to ensure maximum reliability in radio communication. Ultimately, if a particular command fails to execute due to noise in the radio link, it is the job of the control code to notice based on the sequential visual frames and to re-send the command. But the protocol observed by the off-board microcontroller allows for near 100% success rates in command transmission. The conversion of directional commands to transmissible bytes on a distinct off-board microcontroller servers the dual purpose of freeing resources on the off-board computer and ensuring that the time-critical RF transmission proceeds without interruption.

3.3 Command Server

We created a *Command Server* for handling commands from individual robot brains. 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.

The broadcast robot control mechanism used by the radio control processor supports a 15 commands/sec bandwidth to *all* the robots. We anticipate the command loop of individual brains to be much faster than that. Therefore, another of the command server's roles is to queue and filter out outdated commands. The brains are the most complex part of the system. When they use reactive control, this mechanism simplifies the programming task.

4 The Vision System

The RoboCup small robot league limits the size of each robot to the equivalent area of a 15cm diameter circle. Although it is possible to fit an on-board vision system into such a small volume, due to time limits, we have opted for a global vision system instead. We hope to incorporate an on-board vision system in a future revision of our robots.

The vision requirements for robot soccer has been examined by many researchers [Sahota *et al.*, 1995; Sargent *et al.*, 1997]. Systems with on-board and off-board types have appeared in recent years. All have found that the reactiveness of soccer robots requires real-time vision processing. However, due to rich visual input, researchers have found that dedicated processors or even DSPs are often needed [Sahota *et al.*, 1995; Asada *et al.*, 1996]. Our approach to such a problem is more simplistic. We have acquired a fast framegrabber board that can perform visual captures at frame rate.

By engineering the input scene in an appropriate manner, we found that a fast general purpose processor (a 166MHz Pentium processor) is adequate for the task.

4.1 Detection and Association

Following the techniques used by many teams in MIROSOT [Stone *et al.*, 1996], we have decided to use color as a cue for the vision system to detect. This color-based system allows the use of fixed color space thresholds to segment the different colors into regions.

Each robot is fitted with two colors to differentiate the team and the orientation. We considered using different colors for each robot. However, we found that our detection is reliable enough that we are able to use the minimum distance from the previous frame to retain association.

4.2 Tracking and Prediction

In the setting of a robot soccer game, the sole ability to detect the location of objects in the field is often not enough. Like real soccer players, it is often useful and necessary to have the ability to predict future locations of the ball (or even the players). We have utilized a Kalman filter for such a purpose[Han and Veloso, 1997]. The Kalman filter is very suitable for such a purpose since the detection of ball's location is noisy. The Kalman filter takes into account the existence of such noise and gives a best estimate.

5 The Robot Control Code

The robot control code itself consists of several different behavior levels as summarized in Figure 2 [Stone and Veloso, 1997]. First, there is reactive control code which enables the robot to move to a goal location—either a fixed position on the field or a moving target such as the ball. This reactive control is embedded within an obstacle avoidance routine which allows the robots to avoid both teammates and opponents on the way to their target positions. More complex than moving to a particular point is the ability to approach a target from a specified direction as required if the robots are to accurately control the ball. Finally, we describe the robots' mechanisms for action selection and strategic coordination.

5.1 Reactive Control

We use close loop reactive control for the low level movement of our robots. The control strategy follows a modified version of a simple Braitenburg vehicle [Braitenburg, 1984]. The Braitenburg *love vehicle* defines a reactive control mechanism that directs a differentially driven robot to a certain destination point (goal). We require a similar behavior for our robots. However, the love vehicle's control mechanism is too simplistic and, in some start configurations, tends to converge to the goal very slowly. For a robot with two distance-to-goal sensors (which we simulate using vision), we have designed the following reactive control formulae:

$$\begin{aligned}
\textit{translational velocity} &= \alpha \cdot \sin(\theta) \\
\textit{rotational velocity} &= \beta \cdot \cos(\theta) \\
\textit{where } \theta &= \textit{direction of target} \\
& \textit{relative to robot} \\
\alpha &= \textit{base translational velocity} \\
\beta &= \textit{base rotational velocity}
\end{aligned}$$

This set of control formulae differs from the love vehicle in that it takes into account the orientation of the robot from the goal and explicitly adds rotational control. The rotation and translation parameters can be transformed to differential parameters by a simple linear transformation.

5.2 Obstacle Avoidance

If there is an obstacle between the robot and its goal location, the robot must find an alternative path to its goal. Due to the highly dynamic nature of this domain, our approach to path planning is close 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 one step under the assumption that other robots are continually moving around.

5.3 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 stationary 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 routine, the robot draws a line from the ball's current position (or a predicted future position in some cases) to the target position. 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.

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

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

5.4 Formations

At any given time, each of the robots plays a particular *position* on the field. 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.

The pre-defined positions are known to all players and collected into *formations*, which are also commonly known. An example of a formation is the collection of positions consisting of the goaltender, one defender, one midfielder, and two attackers. Although, the formations and positions are known to all robots, neither are static. Robots can switch positions during the course of the game and they can also switch formations entirely as a team. The conditions for switching positions and formations are decided upon in advance, in what we call a "locker-room agreement," in order to eliminate the need for complex on-line negotiation protocols.

5.5 Active modes

Although the default action of each robot is to go to its position and face the ball, there are several *active modes* from which the robot must choose. The default position-holding behavior occurs when the robot is in an *inactive* state. However, when the ball is nearby, the robot changes into an *active* state. In the active state, the robot moves towards the ball, attempting either to pass it to a teammate or to shoot it towards the goal based on an evaluation function that takes into account teammate and opponent positions (see Section 5.7). 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. Finally, the robots in the goalie and defender positions remain in special *goaltend* and *defend* states. Figure 3 summarizes the different active modes and their associated behaviors.

Active Mode	Behavior
inactive	Go to position
active	pass or shoot
auxiliary	receive pass
goaltend	defend the goal
defend	defend and clear

Figure 3: The different active modes available to the robots.

5.6 Goalie and Defender

The goalkeeper's and defender's main role is to prevent the ball from entering the goal. We have designed a pair of position behaviors that complement each other. The defender constantly tries to stay in between the ball and the goal's center line. Whenever it detects a ball moving directly towards it, it turn to clear the ball forward to the attackers. Behind the defender, the goalie aims to always 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 goalie. However, since the robots cannot accelerate as fast as the ball can, it would be possible to defeat such a behavior. Therefore, the goalie continually monitors the ball's trajectory, moving to its destination point ahead of time in the event that the ball is moving towards the goal.

5.7 Attackers

As mentioned above, when in active mode, the robots use an evaluation function that takes into account teammate and opponent positions when determining 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 midfielder might consider passing to both forwards, while a forward would consider passing to other forwards, but not backwards to a midfielder.

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. The algorithm is as follows:

1. Initialize the value of the pass as the ratio of the passer's distance to the (opponents') goal to the receiver's distance to the goal. Thus passes to players closer to the goal are preferred.

2. Draw the line segments from the ball to the receiver and from the receiver to the goal. For each opponent that is within a short distance from these segments, discount the value of the pass in inverse proportion to the opponent's distance from the line segment and in direct proportion to the opponent's distance from the ball. Thus the pass is discounted according

to how easily an opponent would intercept the pass or the subsequent shot.

3. Return the discounted value.

Shots are evaluated using exactly the same algorithm with the passer and the receiver considered as the same robot. Thus step 1 initializes the value of all shots to 1, and step 2 only considers opponents along the line segment from the ball directly to the goal.

Attackers pass or shoot based on the maximum of the values returned for their different potential receivers and for their own shots.

6 Conclusion

The engineering present within this system represents a significant improvement over our previous implementation in many respects [Achim *et al.*, 1996]. The vision processing is at least 5 times faster; the robots move faster and more smoothly; and the behaviors are more complex and effective. Although several 3-robot teams have been implemented in the past [Stone *et al.*, 1996], the system described in the paper is among the first implemented 5-robot teams (along with several of the other RoboCup-97 entries). Having the two additional robots enables several more interesting and effective multiagent behaviors than were previously possible. Our current system represents a significant step forward in the world of cooperative robotics situated in real-time, dynamic, adversarial environments.

References

[Achim *et al.*, 1996] Sorin Achim, Peter Stone, and Manuela Veloso. Building a dedicated robotic soccer system. In *Proceedings of the IROS-96 Workshop on RoboCup*, November 1996.

[Asada *et al.*, 1996] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[Braitenburg, 1984] V. Braitenburg. *Vehicles – experiments in synthetic psychology.* MIT Press, 1984.

[Han and Veloso, 1997] 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.

[Kitano *et al.*, 1997] Hiroaki Kitano, Yasuo Kuniyoshi, Itsuki Noda, Minoru Asada, Hitoshi Matsubara, and Ei-Ichi Osawa. Robocup: A challenge problem for ai. *AI Magazine*, 18(1):73–85, Spring 1997.

[Noda, 1995] Itsuki Noda. Soccer server : a simulator of robocup. In *Proceedings of AI symposium '95*, pages 29–34. Japanese Society for Artificial Intelligence, December 1995.

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

[Sahota, 1993] Michael K. Sahota. Real-time intelligent behaviour in dynamic environments: Soccer-playing robots. Master's thesis, University of British Columbia, August 1993.

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

[Stone and Veloso, 1996] Peter Stone and Manuela Veloso. Beating a defender in robotic soccer: Memory-based learning of a continuous function. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 896–902, Cambridge, MA, 1996. MIT press.

[Stone and Veloso, 1997] Peter Stone and Manuela Veloso. A layered approach to learning client behaviors in the robocup soccer server. *To appear in Applied Artificial Intelligence (AAI) Journal*, 1997.

[Stone *et al.*, 1996] Peter Stone, Manuela M. Veloso, and Sorin Achim. Collaboration and learning in robotic soccer. In *Proceedings of the Micro-Robot World Cup Soccer Tournament*, Taejon, Korea, November 1996. IEEE Robotics and Automation Society.