A Layered Approach for an Autonomous Robotic Soccer System

Manuela Veloso Peter Stone Sorin Achim
Computer Science Department
Carnegie Mellon University
Pittsburgh PA 15213-3891
{mmv,pstone,sorin}@cs.cmu.edu
http://www.cs.cmu.edu/˜{pstone,mmv,sorin}

Introduction

As robots become more adept at operating in the real world, it becomes more important to build teams of robots, capable of high-level collaborative and adversarial planning and learning in real-time situations. Robotic Soccer is an interesting emerging domain that is particularly appropriate for studying these issues (Kitano *et al.* 1995; Stone & Veloso 1996a; Kim 1996). We have been pursuing research in this domain using both a simulator and real physical agents. This paper is a very brief introduction to our work and the reader is referred to the references provided.

We present the architecture of the physical system and introduce how actions are layered building upon each other to create strategic reasoning. We decompose the system's capabilities in different layers, namely behavioral, perceptual, and strategic. We view the strategic layer itself consisting of different levels. We have been using realistic simulation environments (Noda 1995; Sahota 1993) to learn basic collaborative strategic procedures. Our on-going research consists of extending and applying these robust strategic templates to the physical agents.

A ground-breaking system for Robotic Soccer, and the one that served as the inspiration and basis for our work, is the Dynamo System developed at the University of British Columbia (Sahota 1993). This system introduces a decision making strategy called reactive deliberation which is used to choose from among seven hard-wired behaviors. In our approach, we structure the deliberation and reaction as a layered learning architecture. Other efforts have been pursued on applying learning to acquire specific behaviors in different setups (e.g., (Asada *et al.* 1994)). We have chosen to focus on producing a simple, robust design that will enable us to concentrate our efforts on learning low-level behaviors and high-level strategies.

Our current mini-robotic system is certainly usable for tasks other than Robotic Soccer, but since our main purpose in building the system was to work in the Robotic Soccer domain, we made most of our design decisions with this domain primarily in mind.

Overall Architecture

The architecture of our physical RoboSoccer system addresses the combination of high-level and low-level reasoning by viewing the overall system as the combination of the mini-robots, a vision camera over-looking the playing field connected to a centralized interface computer, and several clients as the minds of the mini-robot players. Figure 1 sketches the building blocks of the architecture.

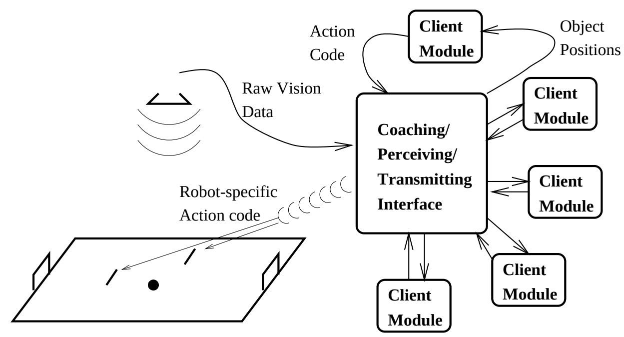

Figure 1: Our Robotic Soccer Architecture as a Distributed Deliberation and Reacting System.

Our architecture implements the overall robotic soccer system as a set of different platforms with different processing features. The current system can use either radio (RF) or infrared (IR) communication. 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 using the robot-specific action codes. Each robot has a self identification binary code that is used in the 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.

Layer	Entity	Input	Ouput
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.

The complete hardware details of the physical agents can be found in (Achim, Stone, & Veloso 1996). We now briefly introduce our work on learning behaviors towards collaborative strategies.

Layered Behavioral Learning

We have been applying machine learning techniques to acquire strategic knowledge initially using a simulator based on the Dynamo system and currently the Soccer Server. Further details of our learning work can be found in (Stone & Veloso 1996c; 1996b).

We have currently applied machine learning to learn the following two layers of behaviors:

- A robust ball interception and shooting behavior;
- The choice of which teamplayer to pass the ball to.

We use supervised neural network to learn the ball intercept behavior. Our setup consists of two agents: a *passer* accelerates as fast as possible towards a stationary ball in order to propel it between a *shooter* and the goal. The resulting speed of the ball is determined by the distance that the passer started from the ball. The shooter's task is to time its acceleration so that it intercepts the ball's path and redirects it into the goal. We constrain the shooter to accelerate at a fixed constant rate, once it has decided to begin its approach. Thus, precisely, the behavior to be learned consists of the decision of when to begin moving. At each action opportunity the shooter either starts or waits. The decision needs to be made based on the observed field through the noisy observation available: the ball's and the shooter's (x, y, θ) coordinates reported at a (simulated) rate of $60Hz$.

We selected the inputs features that enabled the learning to generalize to different field positions and were easily computable from the observations. Given that the shooter is able to estimating the point at which it hoped to strike the ball, or the *Contact Point*, we used the following features:

- *Ball's Distance* to the Contact Point;
- *Agent's Distance* to the Contact Point; and
- *Heading Offset*: the difference between the agent's initial heading and its desired heading.

These inputs (and one hidden layer) proved to be sufficient for learning the task at hand. Furthermore, since they contained no coordinate-specific information, they enabled training in a narrow setting to apply much more widely as shown at the end of this section.

We trained using a random shooting policy. Using the learned 3-input NN shooting policy, the shooter scored **96.5%** of the time at different field locations.

To account for varying ball speeds, we added the estimate of the ball's speed to the inputs of the NN. We achieved equivalent successful interception and scoring performance with this new NN for a variety of generalization situations, including different ball's speeds, ball's trajectories, and locations of the goal.

All our client teamplayers are now equipped with the learned NN for their low-level behavior to intercept and shoot the ball. At the low individual level, each teamplayer is also provided with a probabilistic predictive memory to account for the inaccessibility of the environment (Bowling, Stone, & Veloso 1996).

We have been incrementally teaching other basic collaborative strategy behaviors to our robots. In particular, we are using decision-tree learning algorithms for the selection of which teamplayer to pass the ball to and reinforcement learning approaches to learn to handle adversarial strategies. These higher learning levels use and build upon the low-level behavioral neural networks. Our system drives along the line of learning a layered reasoning system, to handle the individual, collaborative and adversarial characteristics of multiagent domains, in particular robotic soccer.

References

Achim, S.; Stone, P.; and Veloso, M. 1996. Building a dedicated robotic soccer system. In *Proceedings of the IROS-96 Workshop on RoboCup*.

Asada, M.; Uchibe, E.; Noda, S.; Tawaratsumida, S.; and Hosoda, K. 1994. Coordination of multiple behaviors acquired by vision-based reinforcement learning. In *Proceedings of IROS-94*.

Bowling, M.; Stone, P.; and Veloso, M. 1996. Predictive memory for an inaccessible environment. In *Proceedings of the IROS-96 Workshop on RoboCup*.

Kim, J.-H. 1996. Call for participation: Micro-robot world cup soccer tournament (MIROSOT'96), KAIST.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; and Osawa, E. 1995. Robocup: The robot world cup initiative. In *IJCAI-95 Workshop on Entertainment and AI/Alife*.

Noda, I. 1995. Soccer server : A simulator of RoboCup. In *Proceedings of Japanese AI symposium '95*.

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

Stone, P., and Veloso, M. 1996a. Multiagent systems: A survey from a machine learning perspective. *Submitted to IEEE Transactions on Knowledge and Data Engineering*.

Stone, P., and Veloso, M. 1996b. A layered approach to learning client behaviors in the robocup soccer server. *Submitted to Applied Artificial Intelligence (AAI) Journal*.

Stone, P., and Veloso, M. M. 1996c. Towards collaborative and adversarial learning: A case study in robotic soccer. *To appear in International Journal of Human-Computer Systems*.