Totally Autonomous Soccer Robots

Wei-Min Shen, Jafar Adibi, Rogelio Adobbati, Ali Erdem,
Hadi Moradi, Behnam Salemi, Sheila Tejada
University of Southern California / Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
email: {shen, robot-fans}@isi.edu

Abstract

The Robocup 97 competition provides an opportunity to demonstrate the techniques and methods of artificial intelligence, autonomous agents and computer vision. In this competition each of the robots (or agents) must know how to dribble, shoot, pass, and recover the ball from an opponent. Each agent must also be able to evaluate its position with respect to its teammates and opponents, and then decide whether to wait for a pass, run for the ball, cover an opponent's attack, or go to help a teammate; while at the same time following the rules of the game. The most important feature of the USC/ISI robot soccer team is that every robot is autonomous and self-contained with all of its essential capabilities on-board. There are three types of soccer playing position that a robot can be: goal keeper, defender and forward. The role that a robot plays on the field influences the robot's perception of the environment and its playing strategy on the field. While each robot has different motivations and goals, they all share the same general architecture and basic hardware.

Introduction

The Robocup task is for a team of multiple fast-moving robots to cooperatively play soccer in a dynamic environment [6]. Since teamwork and individual skills are fundamental factors in the performance of a soccer team, Robocup is an excellent test-bed for autonomous agents. For this project we define an autonomous agent as an active physical entity intelligently maneuvering and performing in realistic and challenging surroundings [4]. On a soccer field the core capabilities a player must have are to navigate the soccer field, track the ball and other agents, recognize the difference between agents, collaborate with other agents, and hit the ball in the correct direction. In this paper we describe our general architecture for a team of autonomous robot agents, as well as discuss some of the key issues and challenges in creating agents of this type.

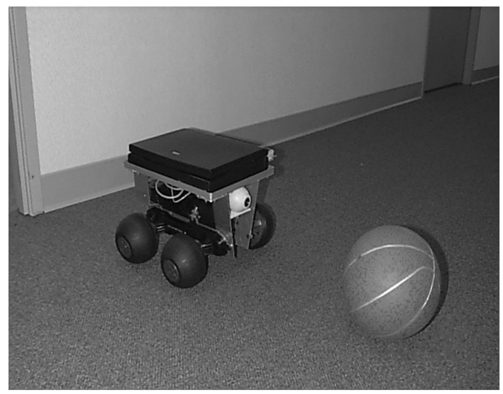

Figure 1

Totally Autonomous Soccer Robot

General Architecture

The critical design feature of our agent architecture is for every agent to have all of the its essential capabilities on-board, so that each robot is self-contained. Such systems need to be physically strong, computationally fast, and behaviorally accurate to survive the challenging environment of a soccer field. Our RoboCup team can be described as a group of mobile autonomous agents (the players) collaborating in a rapidly changing environment (the field and opponent team). In our general architecture, great importance is given to an individual robot's ability to perform on its own, without outside guidance or help of any kind. Each robot agent

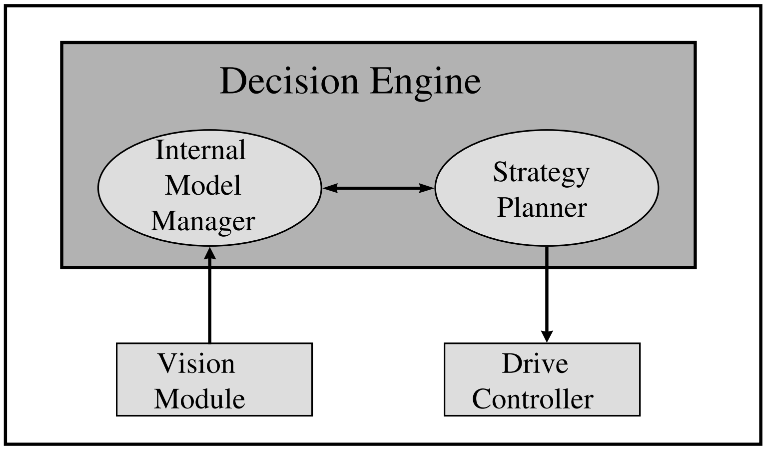

Figure 2

Diagram of the interactions between the main software components of the robot architecture

bases its behavior in its own sensor data, decision-making software, and eventually communication from other teammates.

There are many techniques in agent modeling and design [1,4]; however, much of the work in this area has focused on agents performing under supervision. We believe that further work should be done in the area of autonomous agents [4, 5], with the capability of autonomous learning and adaptation to the environment. Moreover, agents have to be intelligent enough to cooperate among themselves.

A robot consists of a single PC laptop board on a 30cm x 50cm, 4-wheel drive, DC model car (Figure 1). The inputs from the on-board camera, and tactile sensors are collected through parallel and serial ports. Every agent uses the input from the sensors as well as an internal model of the environment and itself to decide its next action. After determining the proper action the robot is steered by commands sent through a serial port to the motor controller.

The main software components of a robot agent are the vision module, decision engine, and drive controller. The interactions between these components are shown in Figure 2. The vision module outputs a vector for every frame taken by the agent's on-board camera. Each vector contains the positions of the objects in the frame, such as the ball, players and the goal. This information is then processed by the decision engine. The decision engine is composed of two processing units - the internal model manager and the strategy planner. The model manager takes the vision module's output vectors and maintains an internal representation of the key objects in the soccer field. The strategy planner combines the information that is contained in the internal model with its own strategy knowledge, in order to decide the robot's next action. Once the action has been decided, the command is sent to the drive controller which is in charge of properly executing the command. In the following sections we explain each component in further detail.

1. Vision Module

Eyesight plays a fundamental role in a human player's ability to perform these tasks (though complemented with tactile and sound cues). Making a good pass or tackle requires a knowledge of the position of the ball and other players; making a good shot at goal involves a precise notion of where the goal is. The navigation, ball tracking, and agent recognition are implemented on-board based on the images received from the on-board camera. Our vision system needed to be flexible, robust, and efficient in order to succeed in such a rapidly changing environment as a soccer field.

We have chosen to use a commercial on-board digital video camera (the popular QuickCam [7]) as the main sensor to implement positioning, collision avoidance, and ball tracking due to its simplicity, robustness, and software availability. To provide a faster reaction to changes in the environment, we use a simple color shape scan of the input frames fed by the camera. This scan uses heuristics to decide which objects are present in the frame. These objects can be stored as a vector which contains information about their location and size in the frame. The vector is then sent to the decision engine where this information can be used to determine relative direction and distance of objects in the field, as well as to predict future positions of relevant objects and collision tracks.

2. Decision Engine

This component makes decisions about the actions of an agent. It receives input from the sensor modules and sends move commands to the drive controller. The decision engine bases its decisions on a combination of the received sensor input, the agent's internal model of its environment, and knowledge about the agent's strategies and goals. The agent's internal model and strategies are influenced by the role the agent plays on the soccer field. There are three types of agent roles or playing positions: goal keeper, defender, and forward. Depending on the role type, an agent can be more concerned about a particular area or object on the soccer field, e.g. a goal keeper is more concerned about its own goal, while the forward is interested in the opponent's goal. These differences are encoded into the modules that deal with the internal model and the agent's strategies.

Together, the internal model manager and the strategy planner, form the decision engine. These sub components of the decision engine communicate with each other to formulate the best decision for the agent's next action. The model manager converts the vision module's frame vectors into a map of the agent's current environment, as well as generating a set of object movement predictions. It calculates the salient features and then communicates them to the strategy planner. To calculate the best action, the strategy planner uses both the information from the model manager and the strategy knowledge that it has about the agent's role on the field. It then sends this information to the drive controller and back to the model manager, so that the internal model can be properly updated.

Internal Model Manager

Each agent has a model of itself and the environment. Modeling helps an agent to predict, interpret, and interact robustly. The internal model contains a map of the soccer field marked with the agent's current location, the locations of its opponents and teammates, and most importantly the location of the ball. The model manager updates the known object positions with every incoming frame vector. It also calculates the distances to the goal, ball, other players, and the walls. This information is extremely important for collision avoidance, hitting the ball, and navigation through the soccer field which are tasks performed by the strategy planner. Not only does the model manager determines the distances of relevant objects, but it also keeps a history of the object movements; in order to predict the future speed and direction of an object. This is also valuable for planning the agent's next action.

The model manager tailors the distance and prediction information sent to the strategy planner, according to the agent's role on the field. After the strategy planner has made its decision for the next action, it sends this information back to the model manager to update the model. At this stage the model manager attempts to reconcile its predictions and the expectations from performing its new action with the input that it is receiving from the vision module. The discrepancies are sent to the strategy planner, so that it can decide the best course of action.

Strategy Planner

Each agent knows its purpose and role on the field. Player roles are different from agent to agent. Each agent's strategy is stored as a hierarchy of tasks and functions. These hierarchies help agents to make decisions about their future actions. On this team there are three soccer player positions: goal keeper, defender and forward. The differences in the responsibility of players may cause differences in their physical structures and equipment. Each agent knows about other players and their responsibilities as well. Though this team does not communicate with each other, they know about each other movements and strategies. The agent's current environment and predictions about object movements are sent by the model manager. The strategy planner combines this information with the knowledge that it has about the agent's strategies. These are the main strategies defined by the agent's playing position:

***Goal Keeper*:**
The goal keeper's strategy requires that it is at all times between the ball and the goal. If it can not see the ball after an acceptable search it moves to the best position based on its internal model. The goal keeper is aware of the 4 other players in the field, but it is mostly concerned with the movement of the ball and its opponents. The goal keeper will not move away from the goal. Keeping the goal keeper close to its goal is a challenge which we have addressed by having a fast robot which can quickly turn 360 degrees to adjust its position regarding the goal. The turning time for this robot for a 360 degree turn is 1 second.

Defender
Like in a real soccer team there are two robots playing defenders. A defender is more concerned about keeping the ball far from its own goal than attempting to score. The biggest challenge for a defender is to avoid scoring by mistake on its own goal. Two key responsibilities or challenges of a defender are to help pass the ball to a forward and ensuring that it is always in the best defending position.

Forward
The forward position is the most important role in terms of scoring on the opponent's goal. A forward has to be able to reason about the weaknesses and strengths of the opponent goal keeper and defenders in order to shoot as accurately as possible at the goal. In our team a forward may do not move back to help the defenders. Each player is assigned to play in a specific area of the soccer field.

3. Drive Controller

The drive controller takes commands from the decision engine, and is a composed of a 4-wheel-drive DC model car. The mobile robot can quickly turn left and right, and move forward and in reverse. It also has the ability to increase and decrease in its height. The drive controller does not have any special breaking mechanism. It stops by disconnecting the move order. The car has the capability to spin in place. There is motor for each wheel which can be controlled independently to enable its quick-turn ability. The control module is a combination of production rules and behavior-based [2] systems with each behavior represented as a set of rules [2,3]. The most important problem in this module is counterbalancing the different sources of uncertainty which come to the system from environment and from the system equipment.

Conclusions

In this project we would next like to address the problem of creating a robust communication scheme between a team of collaborative autonomous agents [5]. A team cannot achieve its optimal performance without correct coordination between members of a team. Therefore, it is necessary to power up each agent with proper communication tools to negotiate the future plans with its teammates. The challenge of inter-agent communication is to implement a robust, noise-free system which can transfer the desire information without any interference. Achieving a high quality teamwork needs a high level of communication [6]. We hope to eventually implement a multi-agent communication architecture to provide our robot team with tools to real-time autonomous team collaboration.

References

[1] Arbib,M. 1981. Perceptual Structures and Distributed Motor Control. I Handbook of Physiology- The Nervous System, II, ed. V. B. Brooks, 1449-1465. American Physiological So

[2] Arki, R.C. 1987. Motor Schema-Based Mobile Robot Navigation. International Journal of Robotics Research, 92-112

[3] Brooks, R. A. 1986. A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation 2(1).

[4] Shen, W.M. 1994. Autonomous Learning From Environment. New York: W. H. F.

[5] Shen, W. M. 1991. LIVE: An Architecture for Autonomous Learning from the Environment ACM SIGART Bulletin (Special Issue on Integrated Cognitive Architectures) 2(4): 151-155.

[6] Kitano H., Asada M. , Kuniyoshi Y., Noda I., Osawa E. Robocup: The Robot World Cup Initiative, Proceeding of the first International Conference on Autonomous Agents, Marina del Rey, CA, 1997, 340:347.

[7] Connectix Corporation, Connectix Color QuickCam, May 1996.