Research and Implementation of an Autonomous Agent using Adaptive and Real-time Learning in a 3-D Soccer Environment

A Proposal for Independent Study in Computer Science, Yale University

Submitted by: Brian Bergstrom and Robert Rodi
Course: CPSC 490
Advisor: Prof. Julie Dorsey
Date: 7 February 2003
To be completed: May 2003

Topic:

Robocup, an ongoing international project related to artificial intelligence and robotics, desires to, “by the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.” While the achievement of this goal lies some distance in the future, a competition is held each year pitting design teams against each other in a robotic soccer tournament. Several categories exist in the Robocup challenge, and the one most relevant to our work is the simulation division, which places two software AI soccer teams into a simulated environment and allows them to compete in a mock soccer game. For our project, we propose to delve into the established literature on this simulation league and build an adaptive learning soccer AI agent to be integrated with a graphics engine being developed separately.

The soccer simulator consists of three main components:

1. Server – This controls the state of the world, and all of the simulation aspects of the program are taken care of by the server. The server returns play state, visual and auditory information to the client as well as information about the client itself (such as remaining stamina, etc.). It also sends out complete data as to the state of the field to the monitor portion of the soccer environment.

2. Client – This controls the “mental state” of the player. The client essentially receives appropriate visual and auditory data from the server, and then, using this data and the players “mind” and “memory,” makes a decision what to do at the particular time step and relays this information to the server. The clients are completely autonomous of each other, and their only communication occurs through the server, which may or may not allow us to easily enable a multiplayer version of the game. The client sends action commands to the server and receives imperfect sensory data.

3. Graphics Engine – This, to be developed separately from the CS 490 project, renders the game world to the screen, is responsible for handling inputted user control, and will present the human player with an attractive and functional user interface. The graphics engine will communicate directly with the server, and will incorporate a physics model, although this may also become part of the server.
Our project will concentrate in the areas of artificial intelligence and machine learning, incorporating elements from genetic algorithms, memetic learning, and traditional artificial intelligence, among other sources. We will use these methods to attempt to implement both simple behavioral responses (passing, dribbling, shooting, goalie ball blocking) and more complex strategic elements, such as communication through a simple vocabulary. We are also interested in the efficacy of each learning method in this environment.

The program will operate in three modes:

1. Player vs. computer – Here the human player will control a single player on a team, while the remaining 20 players will be computer controlled. In addition to having physical control over a single character, the player can also set simple coaching strategies to be followed by the AI players on his team.
2. Computer vs. Computer – This mode places the human player in the role of observer, or, optionally, as a coach over one of two fully AI controlled teams.
3. Player vs. Player – Each human player would control a single character, either on the same team or opposing teams.

Rather than creating the server and clients from scratch, we will be using a modified version of the standard issue Robocup server, along with the skeleton client created by Itsuki Noda. Since the server is open source, we will be able to modify any of the game physics/rules that we see necessary for both an optimal learning environment for the clients and for an enjoyable game for the human. The Noda client is a program that has the basic structures and communications with which a programmer can easily add intelligence, without the high overhead of developing the entire server interface. It is with this framework that we will develop the various intelligence/learning systems and the human control of the players using the data structures and communication mechanisms Noda has created.

We intend to use a client-server-monitor interface similar to that used in Robocup competition, which will allow easy communication with the incorporated graphics engine. The current scene representation used in Robocup is a 2-D representation of the field, with players characterized by dots with vectors. This representation will be heavily overhauled with advanced physics and changed to a 3-D model. Additionally, the Robocup graphics display will be wholly replaced, although we hope to salvage the core of the communications module for communication between the graphics engine and the server.

**Goals:**

1. Implement effective adaptive learning algorithms that will allow an essentially mindless group of players to learn the basic and essential aspects of soccer.
2. Merge the results and other aspects from the above with traditional AI techniques to create an effective simulated soccer player which relies on a union of hard-coded responses and real-time learning.
3. Implement efficient interaction and communication between the three main program modules (server, client, graphics engine).
4. Adapt the current 2-D Robocup format into a more robust 3-D environment, complete with rudimentary physics set sufficient for a soccer simulation. This will involve creating an efficient and effective representational model of the game area.
5. Merge the AI constructs and simulated environment with a 3D graphics engine to be developed separately from CS 490.

Of our goals, Rob will be primarily concerned with developing the adaptive learning aspects of the autonomous agents, while Brian will be working on traditional AI strategies; the real-time learning implementation; and the communication, physics, and environmental representational problems. Together, we will combine our knowledge to create the final agent and to interface the AI with the graphics engine.

Meetings:

Weekly meetings will be held with Prof. Dorsey, and we will be coordinating regularly with members of the AI faculty.

Readings:

The readings will be comprised mainly of various academic papers concerning machine learning, genetic and memetic algorithms, game playing artificial intelligence, and game-based physics models.

Deliverables:

1. Conference-style paper documenting achievements and conclusions.
2. Set of regularly updated web pages containing information on the progress of the project and detailing the theory, implementation, and outcome behind the project.
3. Reading list containing all sources used throughout the course of the project.
4. Any code, executables, and resulting data from the project implementation.

---