Multi-Agent Systems: The UvA Trilearn 2001 Robotic Soccer Simulation Team

Remco de Boer, remdboer@science.uva.nl
Jelle Kok, jellekok@science.uva.nl
Frans Groen, groen@science.uva.nl

August 27, 2001

Abstract

This article describes the Robot World Cup (RoboCup) Initiative and the RoboCup Simulation League in particular. The authors have spent the last year working in this domain on the *UvA Trilearn 2001* robotic soccer simulation team. This team finished 5th out of 14 teams in the German Open 2001 (the unofficial European Championship) and reached 4th place out of 87 teams at the World Championship in Seattle (USA). The main features of the *UvA Trilearn 2001* robotic soccer simulation team will be discussed throughout this article.

1 Introduction

The Robot World Cup (RoboCup) Initiative is an attempt to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined ([3], [5]). In order for a robot team to actually perform a soccer game, various technologies must be incorporated. These include design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics and sensor fusion. The goal of RoboCup is to have a team of fully autonomous humanoid robot players, which by 2050 can win a soccer game against the winner of the most recent World Cup for human players. In order to achieve this goal, the RoboCup organization has introduced several leagues which each focus on a different abstraction level of the problem.

The RoboCup simulation league is based on a physical soccer simulation system called the *RoboCup Soccer Server* ([2]). This system enables teams of autonomous software agents to play a game of soccer against each other. Games are visualized by the *soccer monitor*, which displays the simulated soccer field on a computer screen. The *soccer server* provides a multi-agent environment in which everything happens in real time and where sensing and acting are asynchronous. Various forms of uncertainty are added into the simulation such as sensor and actuator noise, noise in object movement, limited perception, unreliable low-bandwidth communication and limited physical ability. One of the advantages of the *soccer server* is the abstraction made, which relieves researchers from having to handle robot problems such as object recognition and movement. This abstraction makes it possible to focus on higher level concepts such as learning, opponent modeling and strategic reasoning.

2 UvA Trilearn 2001

The *UvA Trilearn 2001* robotic soccer simulation team has been built by two masters students from the Intelligent Autonomous Systems Group at the University of Amsterdam for their graduation project. *UvA*

Trilearn 2001 is a new team that participated for the first time in RoboCup 2001. The team has been built from scratch and does not contain any code copied from other RoboCup teams from previous years. Much of the initial effort has therefore gone into getting the lower level aspects to work. The higher level team strategy is mainly focused on playing fast. In this section the main features of the *UvA Trilearn 2001* team will be separately discussed.

2.1 World Model

Each agent only has a limited view cone giving him a partial view of the world with lots of hidden state. It is therefore a big challenge to create a complete and accurate world state representation for the agent. Each agent keeps a world model that contains information about all the objects on the soccer field. This world model is a probabilistic representation of the world state based on past perceptions. For each object an estimation of its position and velocity is stored (among other things) together with a confidence value which indicates the accuracy of the estimation. We have experimentally compared several different methods for estimating positions and velocities of dynamic objects and used the best of these methods based on statistical results. The world model is updated as soon as new information about the world state is received by the agent and thus always contains the last known information. Objects which are not seen are also updated based on their last observed velocity. The confidence in the estimate decreases however for each cycle in which the object is not seen.

The *UvA Trilearn 2001* agents also use communication to improve the accuracy of their world model. The *soccer server* communication paradigm models a crowded, low-bandwidth environment in which the agents from both teams use a single, unreliable communication channel ([6]). Depending on the position of the ball, the agent that has the best view of the field communicates his world model to nearby teammates. The teammates that hear this message then update their world model using the communicated information about the parts of the field that are currently not visible to them. Experiments have shown that on average the world model of an *UvA Trilearn 2001* agent contains information about 17 other players on the field (out of a possible 21). Despite the fact that each agent only has a limited view cone from which he receives visual information, agents thus still have a good global representation of the environment. In addition, the world model also contains various methods which use the low-level world state information to derive higher-level conclusions. The agent can use these methods to reason about the best possible action.

2.2 Architecture

UvA Trilearn 2001 agents are capable of perception, reasoning and acting. We use a multi-threaded architecture which allows each agent to use a separate thread for each of these three tasks. In this way we minimize the delay caused by I/O to and from the server so that the agent can spend the majority of his time thinking about his next action. The different aspects of the decision making process are thus clearly separated. We use a 3-layer architecture for each agent. The threads used for perception and acting can be seen as the bottom layer which hides the soccer server details from the other layers. On top of this we have the *Skills Layer* which uses the functionality offered by the layer below to implement the different skills of each player (ball interception, player marking, etc.). The *Control Layer* then selects the best possible action from the *Skills Layer* depending on the current world state.

2.3 Synchronisation

One of the most complex features of *soccer server* is that sensing and acting are asynchronous. Agents can send action requests to the server once every 100ms, but they only receive visual information at 150ms intervals. Since it is crucial for each agent to perform an action whenever it has the opportunity, this means that in some simulator cycles agents must act without receiving new visual information. This feature is very challenging for agents to handle, since it requires the ability to make a prediction about the current world state based on past perceptions. Since actions that need to be executed in a given cycle must arrive at the server during the right interval, it is important to have a good synchronisation scheme for sending actions to the server. We use an advanced synchronisation method which for each simulator cycle determines the optimal moment to send an action to the server. This moment depends on the arrival times of the different server messages and thus changes from each cycle to the next. Furthermore our method guarantees that the action sent is always based on the latest information from the server. After sending an action it is checked whether the server has actually performed it. When this is not the case (i.e. the action did not arrive in time) no action is sent in the next cycle to avoid a clash[1]. Experiments have shown that our synchronisation method clearly outperforms several others used in the past. During a full-length match consisting of 6000 simulator cycles the number of holes[2] for each agent is less than 0.2% of the total number of cycles and the number of clashes is less than 0.01%.

2.4 Optimal Scoring

The *UvA Trilearn 2001* agents use an optimal scoring policy which determines the probability of scoring when the ball is shot to a specific point in the goal in a given situation. Based on this probability the agent can then decide whether he should attempt to shoot to this point. The optimal scoring policy was learned by decomposing the problem into two subproblems:

1. Determining the probability that the ball will enter the goal when shot to a specific point from a given position.
2. Determining the probability of passing the goalkeeper in a given situation.

A solution to the first problem was learned by performing an experiment in which a player was placed straight in front of the goal and shot the ball to the middle of the goal from different distances. From the resulting data set it was possible to determine the average deviation of the ball given the distance that the ball had traveled. Using this function we could calculate the likelihood that the ball would move left of the right pole or right of the left pole (i.e. the likelihood that the ball would miss the goal) and thus the probability that the ball would enter the goal when shot at a certain point from a given position on the field. A solution to the second problem was learned by performing an experiment in which a player shot the ball in a straight line while a goalkeeper would be placed in different positions relative to this player. In this experiment we used the goalkeeper of FC Portugal 2000, the world champion team in the year 2000. A data set was formed by recording several values for each situation together with a boolean indicating whether the goalkeeper had intercepted the ball or not. From this data it was possible to determine a discriminant function which separated the data using only the angle and the distance to the goalkeeper. Combining the solutions to both subproblems led to the optimal scoring policy used by the agents.

[1] When two actions are sent during one simulator cycle this is called a clash. This situation is undesirable since the server then randomly chooses one for execution.

[2] A hole occurs when no action is sent during a cycle. This is also undesirable since missing out on an action opportunity may lead to the opponents gaining an advantage.

2.5 Team Strategy

The main philosophy of the *UvA Trilearn 2001* soccer simulation team is to keep the ball moving quickly from each player to the next and preferably in a forward direction. In addition, our agents often try to pass the ball into the depth in front of the wing attackers at the side of the field thereby cutting through the opponent defense. In this way we often succeed in moving the ball rapidly to the enemy half of the field which disorganizes the opponent team. This tactic has even more success due to the fact that we use heterogeneous players on the wings. The latest version of the *soccer server* gives each team a choice of seven different player types each with their own strengths and weaknesses. The heterogeneous players are always faster, but also become tired more quickly. We have developed a method which determines the quality of each player type using a formula which returns a utility value based on the player parameters. Using this method it is possible to determine which player types are best suited for which positions on the field. By using fast players on the wings, the effectiveness of the *UvA Trilearn* fast-play strategy is greatly enhanced. The standard formation used by the team is a 4-3-3 formation. Inside this formation each player chooses a strategic position determined by a weighted sum of its home position in the formation and the current position of the ball which serves as an attraction factor.

3 Results

Over the course of the year, the *UvA Trilearn 2001* simulation soccer team has participated in two international competitions. The first one was the German Open held in Paderborn (Germany) in june of this year. In this competition *UvA Trilearn 2001* finished in 5th place (results can be found at *http://ais.gmd.de/german-open/*). Although this was a very promising result for a rookie team, it became obvious during this competition that our high-level team strategy was not flexible enough to play well against different types of opponents. Especially teams which played very defensively proved difficult to beat for us. In the next two months much effort therefore went into improving our high-level strategy as well as the implementation of the optimal scoring policy. In the beginning of august *UvA Trilearn 2001* then participated in the RoboCup World Championship in Seattle (USA). Here the team finished 4th out of a total of 87 entries, which is a fantastic result when you consider the fact that we started from scratch less than a year ago and only have two people to do the work (most teams have many more). During the championship we proved that we could now also play well against defensive opponents and that our flexibility on a higher level had thus improved. Our most significant result during this competition came in the second group stage where we beat the 2000 winner FC Portugal by 6-0. This was a special performance since it was their first ever defeat in an official competition. The results from the world cup can be found at *http://www.cs.cmu.edu/%7Erobocup2001/AllSchedules.html*

References

[1] Marc Butler, Mikhail Prokopenko, Thomas Howard. *Flexible Synchronisation within the RoboCup Environment: a Comparative Analysis.* In Proceedings of the RoboCup-2000 Workshop.

[2] Ehsan Foroughi, Fredrik Heintz, Spiros Kapetanakis, Kostas Kostiadis, Johan Kummeneje, Itsuki Noda, Oliver Obst, Pat Riley, Timo Steffens, USTC9811 Group. *RoboCup Soccer Server User Manual; for Soccer Server version 7.06 and later.*

[3] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa. *RoboCup: The Robot World Cup Initiative.* In Proceedings of the IJCAI-95 Workshop on Entertainment and AI/Alife, 1995.

[4] Kostas Kostiadis, Huosheng Hu. *A Multi-Threaded Approach to Simulated Soccer Agents for the RoboCup Competition.* In Manuela Veloso, Enrico Pagello, Hiroaki Kitano, editors, RoboCup-99: Robot Soccer World Cup III. Springer Verlag, Berlin, 2000.

[5] The official RoboCup website at *http://www.robocup.org.*

[6] Peter Stone. *Layered Learning in Multi-Agent Systems.* PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, December 15, 1998. Available as technical report CMU-CS-98-187.

[7] The official *UvA Trilearn 2001* website at *http://gene.wins.uva.nl/~jellekok/robocup.*