

## Pardis, Our team in RoboCup Simulator League

### Pardis

*Shahriar Pourazin, Ali Ajdari Rad, Houman Atashbar*

---

---

SP, AA, HA: AmirKabir University of Technology

## 1 Introduction

The team consists of a coach and a number of players, running as separate communicating programs. Each player has three major software components, dealing with (1) Communication, (2) Movement, and (3) Decision respectively. The former component is the medium between the player and the server. It initially checks the server to hunt the most appropriate time slice to send messages to the server, not to lose any simulation cycle without being too noisy. After that, its duty will be reading/writing from/to two queues, one for messages out to the server and the other one for incoming messages from the server. The second component handles the abstract macro instructions of the third component. If the player should go to a location (as a macro instruction determined by Decision component), the action will be done by a sequence of movements collected by Movement manager according to the situation using simple rules of thumb. The (third) Decision manager is really the brain of the player. It controls the player. It should have the ability to select a mode. Each mode corresponds to a particular algorithm. Each algorithm can handle some sort of plan. And a plan, may require some partner among the players in our team. All of these are handled by seeing the situation or hearing some hints from the coach.

In next two sections, some major characteristics of our team have been discussed. The next part describes the design of multi-method players. Many obvious things in all algorithms, like not to play foul or do something illegal have been eliminated to make the paper as short as possible. The last section of this paper describes the way our coach coordinates the players.

## 2 How each player acts?

The main controlling element of the program simulating the player is, Decision component (here sometimes called Decision manager). As the player is a code written in C++, it is easy to replace the Decision component even dynamically. What we have done, is: designing some high performance and

simple algorithms and testing them as the main engine of the players. In this way we selected some better algorithms.

The idea of merging the algorithms all in one player end permitting the player to select the method of thinking, enabled us to design a better and more powerful player.

All the algorithms were categorized in two classes: (1) greedy and (2) cooperative. Greedy algorithms are those which enable the player not to see anything except the ball and the goal. These algorithms led the players to come up with other players even in their own team to control the ball. The other category is also to get the control of the ball, but not when it is under the control of the player in our own team. Our players use both types of algorithms and some others which fall in between of the above categories.

### 3 How the coach helps the players?

The coach has a bunch of algorithms which are all applicable for the Decision manager of any player. It records and replays the game. So it selects all the players in the opposing team one after the other. For each of them, replaces the selected player with an imaginary player using coach's known algorithms. This way it tries to use all the algorithms as an imaginary player instead of the players in the other team. By permitting the false player to regenerate the logged actions, the coach may discover their algorithm. This requires using all the algorithms one by one. By discovering the algorithm they use, the best anti-algorithm arrangement will be determined for the next half. In the time between two halves [1] this process could be handled.

It is very likely not to achieve an entirely matching algorithm. In such a case, The most matching one will be selected. The number indicating the algorithm will be sent to inform the players. After a few seconds of the second half, another message will be sent to players to let them continue their job. This is necessary because all the process may fail when the competitor team changes the code of its players.

Let's see how we have found the algorithms. During the design and implementation of our players we backed up all previous versions of the programs. Next, we enhanced each of them in some various manner. To determine the best of those algorithms, we had to let them really compete. During this process we learned that some algorithms are very likely to succeed against some others. It was very interesting that some of those intelligent and complicated algorithms [2] have been defeated in struggle with some other crazy ones. We have not yet completed all the classes of algorithms. And also have not enough time to compare all with each other. But we hope to be ready until the deadline.

The coach will do its best not to let the other coach do the same decoding to discover the algorithm of its own team. To do so, it changes the preferred algorithm of all players. This may lead to get less goals but will guarantee the success by being inevitable.

## 4 Bibliography

[1] David Andre, Emiel Corten, Klaus Dorer, Pascal Gugenberger, Marius Joldos, John Kummeneje, Paul Arthur Navratil, Itsuki Noda, Patrick Riley, Peter Stone, Tomoichi Takahashi, Tralvex Yeap. Soccer Server Manual, version 4.0 Rev. 02, Jan. 22, 1999.

[2] Peter Stone, Manuela Veloso, Patrick Riley. The CMUnited-98 Champion Simulator Team, in “RoboCup-98: Robot Soccer Wrold Cup II”, M. Asada H. Kitano (Editors) 1999, Springer Verlag, Berlin.