OXSY 2012 Team Description

Sebastian Marian, Dorin Luca, Bogdan Sarac, Ovidiu Cotarlea

OXygen-SYstems laboratory, Str. Constantin Noica,
Bl.5, Sc.C, Ap.36, C.P. 550169, Sibiu, ROMANIA
Email: (sebastian.marian@oxsy.ro) Web: (http://oxsy.ro)

Abstract. Oxsy team was founded in July 2002 for a graduation project of one student, Sebastian Marian, in the field of Multi-Agent Systems, at the Department of Computer Science of Lucian Blaga University (Sibiu - Romania). After graduation he continued the work on this project and so was born Oxsy team. As we started from scratch, our ideas, concepts and beliefs, was implemented year by year and today, we are happy to see that we gone on the right way, as our team was growing in these years, more than we expected from the beginning. If we'll qualify to the competition, this year we'll reach at the 10th consecutive participation in RoboCup Soccer Simulation League.

1 Introduction

In July 2003 at RoboCup competition, which was held in Padua - Italy, we won the first round and for us it was a good surprise for first year of participation. Then, next year, we participated in Lisbon - Portugal for the second time and, again, we obtained a good result (the 11th place). In 2005 in Osaka – Japan, we participated for the third time and finally we entered in the first 8 teams in the world, of soccer simulation league, as we won (the 8th place). In 2006 the competition was held in Bremen – Germany and we won (the 7th place). In 2007 we went to Atlanta – Georgia (U.S.A), where we obtained (the 5th place), the same result which we achieved in 2008 in Suzhou – China. Finally, in 2009 in Graz we entered in the first 3 teams in the soccer simulation league, as we won (the 3rd place), the same result which we achieved in 2010 in Singapore. Last year we came back from Istanbul - Turkey with 4th place in this amazing competition.

This year the competition will be held in Mexico City - Mexico. As we already have a very good experience in 2D Soccer Simulation league, we hope that our new ideas and improvements for this year will be reflected in the competition, where we will also test other tactical elements developed.

2 Increase ball possession through accurate kick ball decision

Having a good world model of the opponent player's position, in the moment when our team has control of the ball, is very important if we want to take a good decision. Generally, not only in soccer, teams that have a better possession of the ball, have more chances to score. In addition, opponents will run all the time to regain possession of the ball, to cover spaces and so on. In fact they will have much less situations to score against us. To achieve a better world model of our opponent, we implemented the following steps:

- Creating a player map's positions, since his last trustable known position, which was obtained through "see" or "hear" message.
- Adjusting the player map's positions through "see" messages, received in meantime, which prove that player can't be in certain places.
- Reshaping the player map's positions after a clearly pattern of moving which has observed in most of cases.

2.1 Creating a player map's positions

Basically we created a map of positions, which tell us where could be a player after a period of time, in which we didn't receive any information about him. This is simply done, by starting from last known position of this player, simulating that he could go in any direction, in every cycle, that passed, without any kind of information about him. Of course, in the next cycle we'll generate next possible locations of the player, starting from every position from his map that was recently generated. We set a maximum map size (10 cycles) and also a limited number of moving directions (45 degree) for every player, due the fact that we must give real time response to the server (100 milliseconds) and secondly because we don't really need such a granulated map, as we determinate through tests, where eight directions and ten cycles, were more than sufficient for what we need. In figure 1 you can see an example of what really mean this map of positions.

In cycle 2048 our player with number 7, observed the opponent with number 7, through visual information which was received. He initializes his opponent map, with current location and current time. In next cycle because he used wide angle he didn't received "see" information. So, he updates his opponent map with four new possible locations. The same thing is happens also in next cycle and again he updates his opponent's map. In cycle 2051 he receives "see" information, but unfortunately, he not receives any information about his opponent, so, again he must updates his opponent's map. Then in next two cycles, he updates the same map due the lack of information about his opponent. So, we can see that after only five cycles, our opponent can be in many positions in the pitch.

Fig. 1. Our teammate with number 7, creates map's positions for opponent's with number 7.

2.2 Adjusting player map's positions

Adjusting player map's positions is essential to create a reasonable map for every player from the pitch. If we look again at figure 1, we observe, that in reality, player with number 7 didn't move so much, from cycle 2048 to cycle 2053, but in his map he could be anywhere in a range of five dashes in any direction. If we always consider the whole map, as it is, our players will have very few opportunities to kick ball somewhere. If we always consider the whole map, as it is, our players will have very

few opportunities to kick ball somewhere, due the fact, that they will believe, that opponents could be almost everywhere. So, is very important to use, every “see” information that will be received, even if they didn’t contain information about some players, but they helps us to remove some map’s positions of some players, which certainly can not be in some places. In figure 2 you can see an example.

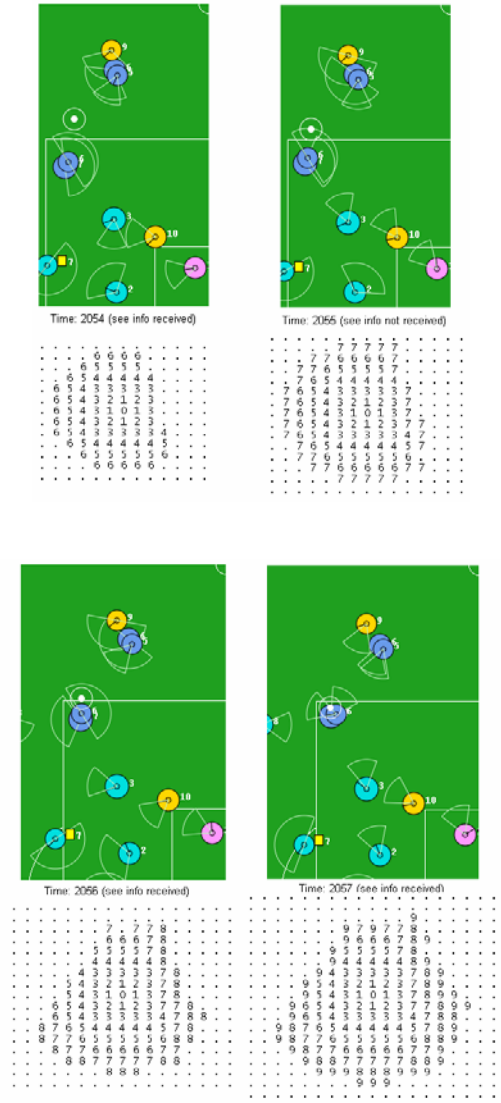
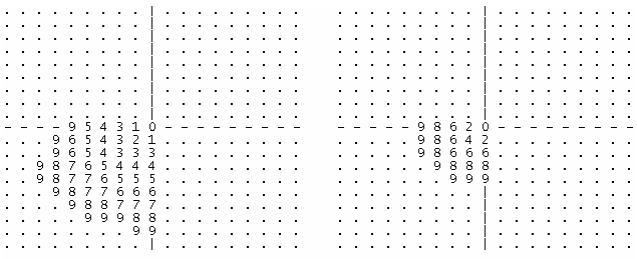


Fig. 2. Our teammate with number 7, adjusts map’s positions of opponent with number 7, using "see" information received.

In figure 2, we see how our player with number 7, updates map's positions of the opponent with number 7, thanks to "see" information received in cycle 2054. Because of his view cone, which covers some of the opponent with number 7 map's positions, we know certainly that he can not be in those positions and we will remove them from the map. Also, all the positions that are in visible distance (3.0 m) from our teammate, even if they are not in his view cone, will be removed from the map. Of course, all these actions will be taken only in the case when the opponent will not appear in this "see" information received. Else, we must reinitialize his map's positions with his new current position, and all this process will be started from the beginning. In this way, we will obtain a better and more real map's positions for all players in the pitch.

2.3 Reshaping the player map's positions

After we have analyzed real movements of some players in the pitch, and comparing them with theirs map's positions, we saw that there are some clear patterns of moving, depending of the action type (defensive or offensive) and of ball position. Also we saw that some players have a better reactivity than others. For these reasons we decide to introduce some new parameters in this equation. First, we tried to divide every player map's positions in four quadrants. In this way we will update map's positions with according to ball speed direction, as we saw that player's movement in the pitch is closely related to the ball's position and ball's speed direction. In this case we will have positions, sometimes only in three, two and also just in one quadrant, depending of the situation of the game. Also, we introduced a variable named "refresh map cycle rate", variable which helps in adjusting the player map's positions. For example, if some players have a weak reactivity, we can set for them this variable with a value of 2, which mean, that for these players, we will update map's positions with a period of 2 cycles. Let's try now to reconsider the situation explained above in figure 1 and figure 2. If we assume that the opponent with number 7 is a player with a weak reactivity, and if we also take into account the ball's direction, the map generated for this player will be radically changed. You can see this map in figure 3.



Opponent's player 7 map positions after 10 cycles when we are using ball speed direction.

Opponent's player 7 map positions after 10 cycles when we are using ball speed direction and when we also update map positions with a period of 2 cycles.

Fig. 3. Opponent with number 7 map's positions after applying some of reshaping factors.

3 Future work

We must accept that right now, many teams involved in this competition, adapts theirs strategy before starting the game, instead to adapt while the game is running. A team will be more powerful, when it can adapt his strategy correctly, depending of the opponent behaviors and not by his name, and also during the game and not only before it.

In the same way, when we used this concept of "player map's positions" to generate a better world model of our opponent, we tried to adapt to some unexpected situations, which are generated by an incomplete world model of our opponents. We also think that is better to determine these factors of reshaping, more exactly and differentiated, for every opponent player from every opponent team. Only in this case we can consider that we completed this task. Probably, we must adapt a neural network, which can give us back these factors of reshaping, depending of how some players were moving according to theirs map's positions. If we can dynamically adjust these factors during the game, we will have for sure a better world model. And if we have a good world model for all our opponents, every action that will be made, it should be more accurate and it has more chances to be completed successfully.

References

1. Peter S. Layered Learning in Multi-agent System [D]. Pittsburgh: school of computer science, Carnegie Mellon University, 1998.
2. Jinyi Yao, Jiang Chen, Zengqi Sun. An application in RoboCup combining Q-learning with Adversarial Planning. Available in http://robocup.lits.tsinghua.edu.cn/download/document/tsinghuaeolus_kick.zip.
3. M. Riedmiller and Artur Merke, " Using machine learning techniques in complex multiagent domains," In I. Stamatescu, W. Menzel, M. Richter and U. Ratsch, editors, Perspectiveson Adaptivity and Learning, LNCS, Springer, 2002.
4. Itsuki Noda et al, Soccer Server Manual, RoboCupFederation. http://www.robocup.org.
5. Y. Jinyi, C. Jiang, and S. Zengqi. An application in RoboCup combining Q-learning with Adversarial Planning. In the 4th World Congress on Intelligent Control and Automation (WCICA'02), Shanghai, China, June 2002.
6. Peter Stone and David McAllester. An Architecture for Action Selection in Robotic Soccer. In Proceedings of the Fifth International Conference on Autonomous Agents, pages Pages: 316–323, 2001.
7. Takashi Suzuki. Team YowAI Description. RoboCup 99: Robot Soccer World Cup III, 2000.
8. Soccer Tactics. An analysis o attack and defense, by Lucchesi Massimo, originally printed in Italy – 1999 by Edizioni Nuova Prhomos Via O.Bettaccini.
9. Coaching the 4-4-2. Zone Play, The Flat Back Four Defense, Attacking Schemes, by Floriano Marziali and Vincenzo Mora, originally published in Italy by Edizioni Nuova Prhomos.