

Introducing Simulation Middle Size: A New Soccer League to RoboCup

Mansour Jamzad and Mahdi Asadpour

Sharif University of Technology
Department of Computer Engineering
Azadi Ave. Tehran, Iran.

Abstract. There is a big difference between Simulation League and Middle Size League in RoboCup, that is, a program written for the former can not be easily transferred to the latter. Since the final aim is real robotics, in this paper we suggest a new match for RoboCup Soccer which can fill this gap. We designed and implemented a Modeled Robot Soccer Server most like Middle Size RoboCup that contains a Soccer Field, Ball, Referee, etc. Each participated team should provide not only his robots controller program but also the hardware model of own robots in VRML97 format, using which our system will model a simulation version of them. As an implementation of our ideas, we modeled and simulated two teams of soccer robots and present the set up method for the match between them. According to these setups, two teams will play and the scores are calculated by the automated referee.

1 Introduction

RoboCup soccer competition is divided into two categories: *Real Robots* and *Simulation*. The main goal of any simulation as a testbed is to facilitate the trial and evaluation of ideas that have promise in the real world [1]. But there is a big difference between the style of simulation league and the real robot league, so that a program written for the former can not be easily transferred to the latter.

In this paper, we will introduce a new league for RoboCup that we call *Simulation Middle Size*. In this new league we can use the advantages of the simulation soccer league. In addition it gives many advantages to researchers to model real robots, add new parts and capabilities to them, design advanced software algorithms and actually see the result of implementation of these algorithms on the simulated robots. We believe that this league can fill the big gap between simulation league and real robot league, so that the programs developed for the new league can be easily ported to real robots.

For this new league, each team should provide his robots controller program and also the hardware model of own robots in VRML97 format. Using these data our system enables the simulated robots to play soccer according to middle size league rules. We have modeled and simulated two teams of robots and in this paper we present the set up method for the match.

The rest of this paper is organized as follows: In next section, we present the reasons for introducing a new league, in section three we introduce elements of this new league.

In section four we describe how to set up a competition between two teams of modeled robots. In the conclusion section we discuss the limitations and present our suggestions for the future works.

2 A New League Can Be a Solution

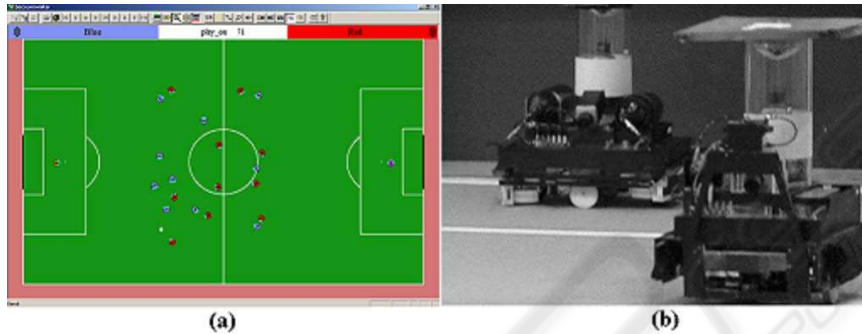


Fig. 1. (a) A typical view of a Simulation League (b) One snapshot of robots in a Real Robots League.

The main differences between Simulation and Middle Size leagues can be listed as follows:

1. An important part of the real robot soccer is how to recognize the environment elements and their positions according to soccer field; participated teams generally use real time machine vision algorithms for this purpose. But in simulation league, there is a SoccerServer which provides the field, ball and players position information [6].
2. In real robot league, a robot may be damaged during the game, impose serious damage to others, or it may not work properly due to hardware or software failures. However, in simulation league such failures are either not considered or might be of limited importance.
3. There has been many advances in designing and implementing new ideas in simulation league that are based on AI methods and multiagent systems. But in real robot league, due to the construction problems and limits in implementation of advanced algorithms, its speed of progress is somehow limited.

However, although the 3D Simulation League provides a 3D shape for robots and the ball movement in 3D space, but the users can not model their own robots in this environment. In addition it can not provide any solution for the above mentioned limitations.

We have a clear goal in RoboCup, that is to make robots that can win the human soccer champion team by 2050. Having a look at the present leagues, we see a great

progress in designing and implementing ideas based on AI, multiagent systems, including cooperative behavior, path planning, team work, etc that can be easily implemented in soccer simulation league. On the other hand there are many limitations in implementing most advanced methods used in simulation league in middle size. The reason is due to the hardware and mechanical limitation of robots. If we continue the leagues as they are, by the time that we could build better robots, we might face some serious limitations, that is we might not be able to apply the advanced software algorithms on those robots.

In the other hand, many research groups who are interested in middle size, but because of the high cost of purchasing real robots, or difficulties in constructing own robots, the number of teams that have been participated in middle size league are limited. The trend in recent years shows that there might not be much change in this regard.

Therefore, in order to speed up many research ideas in related fields, the need for a Simulation Middle Size league seems to be a reality.

In this new league, two teams of modeled robots can play according to middle size league rules. Each team should provide the model of his robot(s) in VRML97¹ format and also its controller program using which the server will model a simulation version of them. At this stage the server will arrange a match between these two teams in an environment similar to that of middle size league.

We highly recommend the reader to look at the movies in the web page of this paper (<http://sina.sharif.edu/~jamzad/robocup.html>). These movies show the matches played in the new league environment. Also, the controller programs source codes (in C/C++ language) are ready to download from this web page.

All our design and implementations were done using Webots², a commercial mobile robot simulation software that provides a rapid prototyping environment for modeling, programming and simulating mobile robots. Using the libraries of this software one can transfer his control programs to several commercially available real mobile robots[3].

The reason that at this stage we have used Webots, was because we were familiar with its usage in mobile robots applications. But, there are other simulations tools such as *Player/Stage* which is an open source, that can be used for this purpose, or even a group of researches might develop a special purpose simulations tool for RoboCup environment.

However, although most middle size teams might have their own simulation software tools, but such tools are limited to test some specific functionality of their robots. In addition due to the hard schedule of most teams in developing practical software and hardware and also mechanical fix ups, there is very limited time left for testing many available methods and new ideas, for example, path planning, cooperative behavior, distributed AI, etc, on real robots.

¹ Virtual Reality Modeling Language, version 97: Most 3D modeling software, like 3D Studio Max, Maya, AutoCAD or Art Of Illusion, include the VRML97 export feature.

² <http://www.cyberbotics.com>

3 Elements of Simulated League

Elements of this new league and its rules are based on Middle Size League. Its most important rules are listed in the web page of this paper, [11]. The complete middle size league rules is available in [2].

3.1 The Field of Play

As shown in figure 2, the color of the field surface is green. Dimensions of the field is rectangular and marked with white lines.

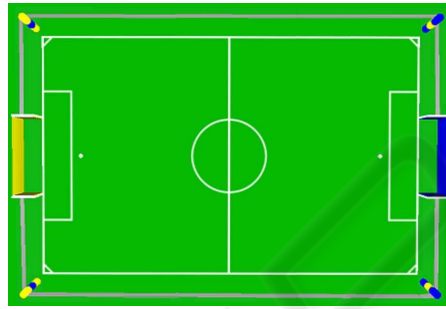


Fig. 2. Top view of the field of play with white lines, yellow and blue goals, goal area, corner flagposts, safety boundaries, etc.

A goal area is defined at each end of the field and cylinders are used instead of flagposts. As you see in the figure 2, the corner cylinders are painted in the same color as the nearest goal. A wall is placed around and behind the goal and the interior of one goal is blue, the other is yellow. The exterior of the goal, the goalposts and the crossbar are painted white.

3.2 The Ball

The ball is spherical and its color is red, as is shown in figure 3 in front of the robots.

3.3 Players

A match is played by two modeled teams, each consisting of not more than six players including a goalkeeper. The maximum difference of number of players between the two teams starting a match is defined as 1, if the team with less robots has 4 or more players.

Each Player/Robot carries the markers, which include Color Markers and Number Markers. The color marker is one of the two predetermined official colors. The official colors are with RGB³(0, 255, 255) for blue robots and with RGB(255, 220, 130) for yellow robots, painted in figure 4.

³ Red, Green and Blue color numbers

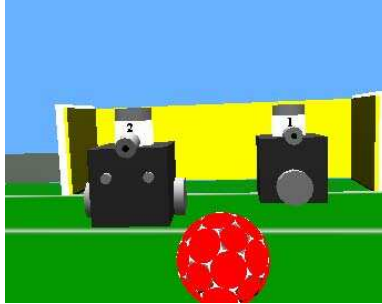


Fig. 3. The yellow team: player and goalkeeper plus the ball and number markers above each robot.

Also each robot carries a number marker. The number 1 is reserved for goalkeepers (as shown on top of robots in figure 3).



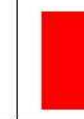



					
Blue Robot	Yellow Robot	Ball	Soccer Field	Yellow Goal	Blue Goal

Fig. 4. The used colors in this match.

3.4 The Referee

In this league, two referees control each match: *Automated Referee* and *Human Referee*.

Rules Judged by The Automated Referee The Automated Referee is a controller/supervisor program that manages the game. It does the following tasks:

1. Starts the game.
2. Assigns blue or yellow color to one team randomly. The blue team starts the match.
3. Imports the model of robots to game and load their controllers programs.
4. Acts as timekeeper and keeps a record of the match.
5. When a team scores, the referee performs a number of tasks. It updates the score (showing in the top of simulation scene), moves the ball to the center mark and moves the players to start predefined positions.
6. When the ball goes out of the field, the referee moves the ball to a proper position.
7. Finishes the game and saves the result of game to a storage file.

Rules Judged by The Human Referee The Human Referee sets up a match firstly. During the match, some errors may occur for the league system, in the other word, there are some fouls like "obstruction" which are difficult to judge automatically. To resolve such situations, the human referee is allowed. Other responsibilities are when intentionally blocking the movement of other players, inappropriate behavior and blocking the goal with too many players [6].

4 How to Setup a Match

In this section, we mention the way of setting up the match from the ground up. Figure 5 shows a scene of match between two teams.



Fig. 5. One scene of the match including images, at the right side, which were captured by each robot camera.

Before setting up a competition, two participated teams should provide their robots models and controlling programs.

We modeled and simulated two teams of robots to show:

1. The ability to construct complex and impressive robots, and
2. To provide a good sample for enthusiastic researchers to use it as it is or modify it as desired.

To run a simulation in Webots, you need three things:

1. Define a 3D virtual world containing one or more 3D virtual robots to be saved as a .wbt file.
2. Write C/C++ or Java programs to control each robot called **Controller Program**, and optionally, a supervisor program to control the simulation.
3. Use Webots to run simulations for your robots and controllers [5].

4.1 An Example of How Teams Participate in The League

For the first team participated in our league, the Arvand robots are selected. *Arvand* is the name of robots designed and constructed by Sharif CE team for playing in the middle size league [9]. Two different types of robots were made, *players* and the *goalkeepers*. These robots showed a high performance in Robocup-99: became champion [7].

Arvand Players We modeled the players as below (see figure 6):

1. *The front camera* captures an image in the field of view of 90 degrees (1.57 radians) with size of 64×64 pixels (used the Camera node of Webots).
2. *The kicker* kicks the ball (used the Servo node of Webots).
3. *Two servo motors for each wheel* for driving left/right and forward/backward (used the Servo node of Webots). The combination of drive units movement and proper settings provides the robot with continuous rotational move around any point [7].
4. *Castor wheel* is used for balancing the robot (used the Solid node of Webots).
5. *Communication* between robots by Emitting and Receiving messages (used the Emitter and Receiver nodes of Webots).

The controller program of this robot is a simple image processing algorithm that recognizes the ball and opponent goal positions. Its C source code is freely available at [11].

We use three copies of this robot plus a goalkeeper for establishing a team.

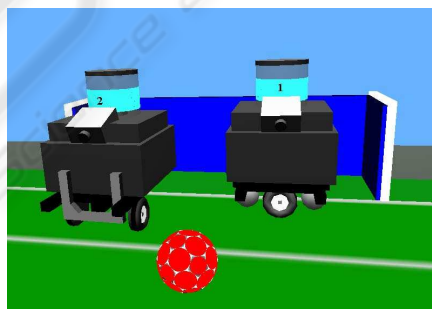


Fig. 6. A simulation model of Arvand player plus its goalkeeper robot and the match ball: team color markers and number markers are included.

Arvand goalkeeper We modeled the goalkeeper as below (see figure 6):

1. *Front camera* is similar to the player one.
2. *Two castor wheels* is similar to the player one.
3. *Two servo motors for each wheel*, similar to the player one but are used for driving according to the goal line.

The controller program of the goalkeeper recognizes the ball position and tries to prevent the ball from passing over the goal line. You can find some sample codes for this goalkeeper in [11].

4.2 The Needed Settings

After you constructed⁴ all the robots, you must observe the following settings before competition:

1. Prepare two copies of your team; one with blue marking, and one with yellow marking.
2. Choose the DEF name⁵ of B_n for robots in blue team, and so on for Y_n .
3. Export⁶ the VRML97 model of your robots to files $b_n.wbt$ for the robots of blue team and $y_n.wbt$ for the robots of yellow team which n indicates the number mark of that robot. Number 1 is reserved for goalkeepers.
4. You should deliver $b_n.wbt$, $y_n.wbt$ and controller programs in a folder with your team name.

4.3 The Match Routine

When two teams deliver the needed files to human referee,

1. He records the names of two teams.
2. He runs the simulation program.
3. The automated referee select one of teams as the starter (blue team) randomly.
4. It imports $b_n.wbt$ files of starter team and $y_n.wbt$ files of other one to the soccer field.
5. Places the players to start predefined positions.
6. Kick off the match.
7. During the match, two teams compete with each other, and
8. Referees do their tasks respectively.
9. After the game finished, the automated referee shows the game result as seen in figure 7, and also saves it in a log file.

⁴ Modeling the hardware and writing controller program

⁵ DEF is a field defined in each node of Webots

⁶ By using "Export" item in Webots software



Fig. 7. At the end of match, the automated referee shows/saves the match result.

5 Conclusion

In this paper we suggested to start a new league called Simulation Middle Size. Not only it can use the advantages of the simulation soccer league, but also it gives many advantages to researchers such as modeling their real robots, adding parts and capabilities to them, designing advance software algorithms and studying how they will be implemented on their modeled robot.

As a result the researches can evaluate the performance of such algorithms in shorter period of time with higher accuracy.

We believe that if the suggested league is presented as a new match in RoboCup, we will see a high improvement in the real robots league. We can design our robots in such a way that certain algorithms that have proven to be efficient in the simulation, could be easily applied on them.

We conclude that our suggestion can be improved as follows:

1. The rules that are not currently applied in Middle Size e.g. Offside, Penalty Kick [2], can be applied here to evaluate its practical effectiveness.
2. Such league can easily be set up for Small Size and Humanoid RoboCup.
3. In order to evaluate the performance of algorithms on real robots, we can program a *fault injection* mechanism to be added to hardware and mechanical parts of the simulated robots. These faults can be injected with the predefined probabilities during the match. At the end of a match, the overall performance evaluation will be informed to each team.
4. The image acquisition of Webots provides noise free images. That is, we can always get the exact predefined fixed colors for corner flagposts, land marks, robots, etc. However, in order to have a realistic simulation of the robot vision system, we can introduce a new option for adding noise and degradation to the images.
5. At this stage the match is setup as a *local* game. Therefore, there are restrictions on programming languages used by teams. To solve this problem, the match can be carried out in a client-server style and using communication via UDP/IP (or TCP/IP).

Acknowledgment

The authors would like to thank Masoud Asadpour from EPFL university (Switzerland), Olivier Michel from Cyberbotics Ltd. and all members of Sharif CE Middle Size RoboCup team for their sharing of knowledge and cooperation.

References

1. P. Stone and M. Veloso, *Multiagent Systems: A Survey from a Machine Learning Perspective*, Journal of Autonomous Robots, Vol. 8, No. 3, pp 345-383, 2000.
2. MSL Technical Committee, *Middle Size Robot League Rules and Regulations for 2004, Draft Version pre-8.2*, May 18, 2004.
3. Olivier Michel, *WebotsTM: Professional Mobile Robot Simulation*, Journal of Advanced Robotics Systems, Vol. 1, No. 1, pp 39-42, 2004.
4. Itsuki Noda, *Soccer server : a simulator of robocup*, Proceedings of AI symposium 95, pp 2934. Japanese Society for Artificial Intelligence, December 1995.
5. Cyberbotics Ltd., *Webots 4.0.5 User Guide*, 2004.
6. Mao Chen et al., *Users Manual RoboCup Soccer Server version 7.07*, 2001.
7. Mansour Jamzad et al., *ARVAND: a Soccer Player Robot*, AI Magazine, Vol. 21, No.3, pp 47-51, Fall 2000.
8. Mansour Jamzad et al., *A Goal Keeper for Middle Size RoboCup*, Lecture Notes in Artificial Intelligence: RoboCup-2000: Robot Soccer World Cup IV, pp 583-586, Springer 2001.
9. Mansour Jamzad et al., *A Fast Vision System for Middle Size RoboCup*, RoboCup-2001: Robot Soccer World Cup V, Peter Stone, et al (Eds), Springer, pp. 76-85, 2001.
10. Mahdi Asadpour, *Simulating ARVAND Middle Size Soccer Player Robot with Webots*, BS thesis, Sharif University of Technology, Department of Computer Engineering, Feb 2005.
11. <http://sina.sharif.edu/~jamzad/robocup.html>
12. <http://www.cyberbotics.com>
13. <http://www.robocup.org>