Suggestions for FIRA’s SimuroSot and its Automatic Referee

Xun Li ¹, YanDuo Zhang¹, Bingrong Hong²

¹ School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan, China
zhangyanduo@hotmail.com

² School of Computer Science and Engineering, Harbin Institute of Technology, Harbin, China
hongbr@hit.edu.cn

Abstract

Popularization as it attracts so many researchers and robot fans. The SimuroSot platform plays a vitally important (core?) role in the game. However, some flaws which cut out of the fun of the game have revealed as time goes on. This paper summarizes part of the flaws and gives some suggestions of how to improve the platform.

1. Introduction

Robot Soccer is base on multi-agent system. SimuroSot platform precludes the influence introduced by difference between robot performances [1]. FIRA Simurosot is a standard software platform for simulated robot soccer game. The platform provides the vision data with full consideration of the practical limitation of the real-time vision and the wireless communication system. Therefore, effort is only needed to focus on the development of the algorithm for path planning, obstacle avoidance and strategy designing. Based on these reasons the SimuroSot (5v5) had been the most popular one in the FIRA games [2].

The platform for 11 vs. 11 SimuroSot game which developed by Harbin Institute of Technology (HIT) was first used officially in the 6th FIRA Robot Soccer World Cup Beijing in August 2001. It adopts the C/S structure, using UDP protocol to communicate with clients. The collision detection module and dynamics model used in the 2-D platform are independently developed by HIT. In 2003, a referee module was added in it. The version broadly used in FIRA games is ver. 2003-09-04.

In March 2002, a 3-D platform for 5 vs. 5 SimuroSot game was released by RSS Development team in Griffith University in Australia. It was developed based on macromedia’s Director 8.5. The collision detection and dynamics models are provided by Havok physical engine which embedded in Director 8.5. The platform exchange data with two clients through their dll file.

2. For Large League SimuroSot

2.1. Collision Model

There are Interpenetration phenomena among robots, ball and the bevel of field. After collided with robot, ball has a ruleless motion in the first cycle and cannot move regularly until separated with robot. Dead-lock may
happen, when a robot collides with others. In this situation, any velocity cannot make them separate. Ball often enters into one of the four corners, and cannot get out until starting free-ball.

The work of modifying the collision model is arduous, for it involves the modification of core code and large workload. A simple and direct scheme is using the collision model of Middle League Simurosot.

2.2. Difference of half-courts

In the game, there is an obvious difference between left half-court and right half-court in the counterbalance strategies or even same strategies-- left half-court can gain an advantage over right half-court.

2.3. Proportion of objects in the Field

The ratio of robots, ball and field is not symmetrical. The current Simurosot platform does not tell us the actual size of robot, ball, and field. By test, we get that the size of field is 1032x818(pixel), the size of robot is 13x13(pixel), and the diameter of ball is 8(pixel). And in large league MiroSot, the size of field is 400x280(cm), the size of robot is 7.5(cm), and the diameter of ball is 4.27(cm) [3]. We compare the ratio of the size of robot and the size of ball to the size of field. For the sake of convenience, we define the ratio of the size of robot to the length of field \(a_{\text{robot\_field\_simu}}\), the ratio of the size of robot to the width of field \(b_{\text{robot\_field\_simu}}\), the ratio of the diameter of ball to the length of field \(a_{\text{ball\_field\_simu}}\), the ratio of the diameter of ball to the width of field \(b_{\text{ball\_field\_simu}}\) in the Simurosot platform, and in the same way, we define the size in the large league MiroSot \(a_{\text{robot\_field\_miro}}, b_{\text{robot\_field\_miro}}, a_{\text{ball\_field\_miro}}, b_{\text{ball\_field\_miro}}\), computing by the data given above, we can get that \(a_{\text{robot\_field\_simu}}\) is 1.59%, \(a_{\text{robot\_field\_miro}}\) is 2.67%, \(b_{\text{robot\_field\_simu}}\) is 1.26%, \(b_{\text{robot\_field\_miro}}\) is 1.88%, \(a_{\text{ball\_field\_simu}}\) is 0.98%, \(a_{\text{ball\_field\_miro}}\) is 1.53%, \(b_{\text{ball\_field\_simu}}\) is 0.78, \(b_{\text{ball\_field\_miro}}\) is 1.07%.

Thus it can be seen that the ratio of size of robot to the size of field in Simurosot is smaller than that in MiroSot.

We advise that we should amend the size in Simurosot follow as that in MiroSot.

3. For Middle League Simurosot

3.1. Attenuation of Robot’s velocity after collision

The stalemate frequently occurs when the ball and robots near the boundary and the robots of each side have to push each other. Velocity of the robot at the direction of perpendicular to the edge of the robot attenuates too quickly when collision occurs between the ball and robots (fig 4). Taking the follow figure for instance, the velocity of the ball before collision is \((v_{x1}, v_{y1}, v_{1}) = (2.40, 1.62, 2.89)\), and it becomes to \((v_{x2}, v_{y2}, v_{2}) = (0.42, 1.31, 1.37)\) after the collision.

\(v_{x1}\) is the speed of the robot in the x axis direction when it at the start point,
\(v_{y1}\) is the speed of the robot in the y axis direction when it at the start point,
\(v_{1}\) is the speed of the robot when the it at the start point,
\(v_{x2}\) is the speed of the robot in the x axis direction when it at the end point,
\(v_{y2}\) is the speed of the robot in the y axis direction when it at the end point,
\(v_{2}\) is the speed of the robot when the it at the end point.

Velocity of the robot at the direction of perpendicular to the edge of the robot attenuates too quickly when collision occurs between the ball and border (fig 4). Taking the follow figure for instance, the velocity of the ball before collision is \((v_{x1}, v_{y1}, v_{1}) = (2.40, 1.62, 2.89)\), and it becomes to \((v_{x2}, v_{y2}, v_{2}) = (0.42, 1.31, 1.37)\) after the collision.

\(v_{x1}\) is the speed of the robot in the x axis direction when it at the start point,
\(v_{y1}\) is the speed of the robot in the y axis direction when it at the start point,
\(v_{1}\) is the speed of the robot when the it at the start point,
\(v_1\) is the speed of the robot when the it at the start point, 
\(v_{x2}\) is the speed of the robot in the x axis direction when it at the end point, 
\(v_{y2}\) is the speed of the robot in the y axis direction when it at the end point, 
\(v_2\) is the speed of the robot when the it at the end point.

3.2. Collision detection

In many situations, the judgment of the collision is necessary, however, the interface of the 5vst platform is 3D and we can come to different conclusion from different angles (eg, fg6). What’s worse, this situation often occurs before the ball goes into the goal and it is hard to avoid controversy when to make a judgment only depend on the referee. We propose to adjudicate this situation by the platform itself.

![Figure 5: track of collision of ball and border](image)

3.3. Position detection

According to the regulation of the simulation match, an object (ball or robot) is considered to in an area only if it is more than 50% inside [4], as judged by the referee. However, the referee is often hard to make a decision at this situation, because the interface of 5v5 platform is 3D and it seems the object is in an area from one angle and it is not from another angle. So we propose to adjudicate this situation by the platform itself.

![Figure 6: puzzled by collision](image)

3.4. Return automatically

At present, the ball and robots must be placed by ourselves on the 5v5 SimuroSot platform, sometimes, it cost a lot time for some players to place the robots or ball accurately, during the time of the match, a lot of status of Freeball occurred frequently in the platform, so there will be a cycle of “place robots –foul–place robots”. Too much time is spent on placing robots in a match.. So we should add a function of returning position automatically by robots itself, we may make the robots return it’s position through our program.

3.5. Automatic referee

Our suggestion is adding an Automatic referee on 5 vs. 5 SimuroSot Platform as the 11 vs. 11 SimuroSot Platform to reduce the referee’s own effect and shorten the time of match. The platform should supply a referee script editor, which is publicized on the official web.

4. CONCLUSION

We summarized some flaws we found according to the experiences gained through development and matches during the last 5 years. Meanwhile some suggestions of how to improve the platform are given in this paper.

5. Acknowledgements

This work is supported by the Ministry of Science and Technology, and the Ministry of Education of Hubei Province, China. And the main opinions had been discussed with some chairs of many chapters under FIRA Branch in China, at Jan. 5, 2007, in Harbin China.

6. REFERENCES


[3] FIRA Large League MiroSot Game Rules

[4] FIRA Middle League SimuroSot Game Rules