

MT2016 Robocup Simulation 2D Team Description

Lipeng Zhang, Benchu Yao, Shengbing Chen, Gang Lv

Department of computer science and technology,

Hefei University,

Anhui Province, P.R.China, 230601

hfuumt@163.com

Abstract. MT2016 is a team of 2D soccer simulation league which is consisted of the students who are coming from Hefei University and all of them are with strong robot enthusiasm . Since 2012, the MT2016 team has participated in RoboCup China open tournament every year and has achieved many good results. This paper briefly describes the background of MT2016 and the main works of our team since the 2015 RoboCup World Cup. Through these works we have greatly improved the overall capacity of our team.

Keywords: 2D soccer simulation; agent2d; attacking model; defense model

1、 Introduction

As one of oldest leagues in RoboCup, the 2D soccer simulation league has achieved great success and inspired many researchers all over the world to engage in this game each year [1, 2].

Attracted by the magic goal of the RoboCup, some students who strongly love the robot control in Hefei University established in May 2012. Since then, we have study a lot of literatures and worked hard in optimizing the code to improve the overall capabilities of offensive and defensive [3-8]. We take an active part in annual competitions of RoboCup, and there are some achievements, in 2012 and 2013 we have won the second prize; and in 2014 we won the grand prize; last year, unfortunately we got ninth because of the errors of model parameter. By the communication with other teams, we found some deficiencies, and then proposed improvement measures. We hope to verify the effect of improved code in this year's competition, and improve the team's level gradually. For this competition, we spare no efforts to do it. We hope we can get remarkable achievements, make more friends, and learn more things in this year.

2、 The underlying of the MT2016

We use agent2d-3.1.1 as the underlying code, the download address is: <http://en.sourceforge.jp/projects/rctools/>. Using librcsc as the underlying database, the team's underlying is action-chained style. Based on the basic underlying, we added some class files as follows: *bhv_basic_offensive_kick*, *body_offensive_block* and *Fuzzy_Logic_Rhythm*. *Bhv_basic_offensive_move* is to realize active cooperation and offensive moves when we handle the ball; *body_offensive_block* is to realize our players quickly blocked and then look for opportunities to challenge for the ball when the other players handle the ball; *Fuzzy_Logic_Rhythm* is to deal with fuzzy logic. We try to use fuzzy logic to identify the situation on the ground, and decide the offensive rhythm.

In addition, on the basis of MT2015, we restructured the related code of heterogeneous model, because the `role_***.cpp/h` file was rewritten many times and became bigger and bigger, which make it is difficult to improve. So, we added the new file `bhv_** (role).cpp/h` in the code of MT2016. We created a new CPP file to save the code of the actions except `execute ()`, `do Kick ()`, `do Move ()`, reduced the redundancy of evaluation criteria in the debugging, and optimized the actions codes.

3、 The main strategies of MT2016

According to the need of offensive and defensive strategy, we added some new technologies and improved part of underlying action of Agent2D based on the code of MT2015. In MT2016, we established a set of independent models for attack and defense, in order to deal with the different situation of the front court and the back court. At the same time, we obtained the optimal steering time and intercept distance for the penalty of offensive and defensive through times of testing data.

Besides, we optimized the tackle model to increase the rate of fault tolerance. In original tackle model, the rate of tackling error is higher and the rate of losing ball is not low. Now we optimized numerical in `chain_action` and `tackle_generator`, increased the success rate of the tackles.

In addition, more detailed area is divided for the action assessment to capture the optimal solution as far as possible. MT2016 divides the whole court into 5*6 rectangles; each rectangle is assigned a value (7, 8, 9, 10, or 11) according to the distance to the goal and the shooting angle, and then adjusts the value according to the number of opponents in this rectangle. The ball-holder will pass the ball to a higher value teammate (shown in figure 3-1 and figure 3-2).

3.1 Attacking model

In the past, the primary attacking member of our team was mainly based on the agent numbered 11, and the agents numbered 7, 8, 9, 10 acted as the supporting roles and served for agent numbered 11 in different locations with an unset models. But now, MT2016 changes this model and establishes a new attacking model which does not attack opponents by just depending on a heterogeneous player but changes attacking members according to the actual situation of the game. In attacking half, assuming that the value of a regional is y , the number of the opposing agents is x , then, the agent in this regional has a value $z = f(x, y)$.

As shown in figure 3-1 (In the following figures, the black agent is our teammate, and the white agent represents the opposite player), assuming that the $f(x, y)$ is defined as $z = y - 1.5x$, then, the ball-holder may pass the ball to other two teammates whose values are 8.5 ($10 - 1.5$) and 7.5 ($9 - 1.5$). According to our new attacking strategy, the agent will pass the ball to the teammate who has a higher value (8.5).

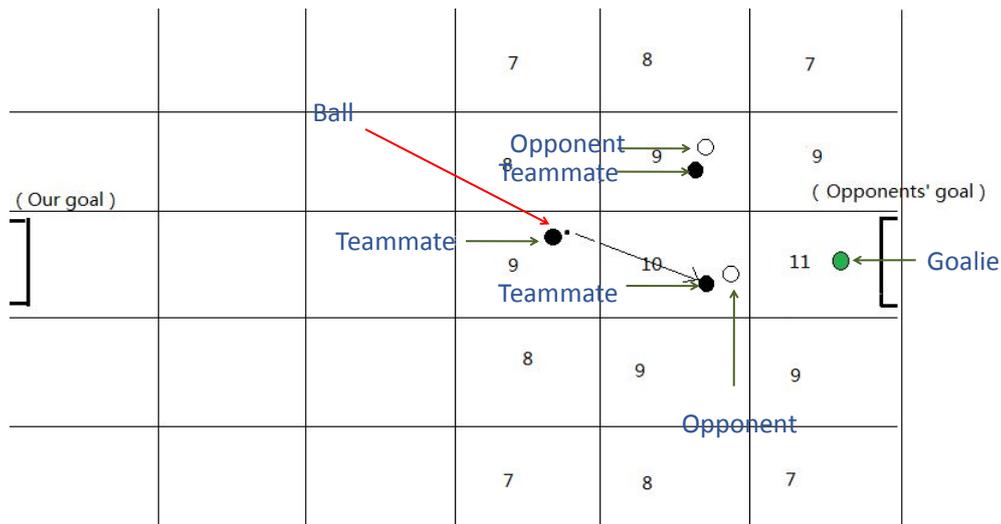


Figure 3-1 the agent 's value is based on the position and the number of the opponent

Figure 3-2 shows another application scenario when the application opponent emphasizes the intensive and defensive strategy. If the opponents retreat to the penalty area, the success rate of attacking is quite low. Therefore, the value of agent who stands in the penalty area becomes quite low, besides, the ball-holder will pass the ball back to the teammate who isn't marked by the opponents, and it will evacuate opponents to create conditions for the next attacking. As shown in figure 3-2, the region of 11th agent has three opposite agents, the 11th agent's value is 6.5 ($11 - 1.5 * 3$), and the other two opposite agents whose values are 1 and 2, and both of them are in the area of 9. The agent's values are 6 ($9 - 1.5 * 2$) and 7.5 ($9 - 1.5$), and then it will pass the ball back to the largest value of our agents.



Figure 3-2 throwback diagram based on the number of offensive regional rivals

3.2、 Defense model

In the defense model of MT2016, goalkeeper and defender build the attacking model through the coordination of our own and the other agents and then to determine the next movement, and to complete the defense by the mutual cooperation between the goalkeeper and defender.

As shown in figure 3-3, it is in the situation that an opposite agent holds the ball which is close to our goal. The defender computes dynamically the opposite shooting angle and the success rates according to our own and the other agents' coordinations. Then to estimate the opponent's attacking action and to make the defensive action.

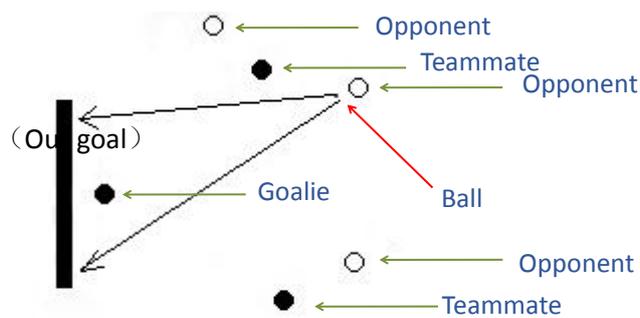


Figure 3-3 the shot angle of the opponent

If the shot distance is quite long, we will limit the opposite agents' shot angle to force it to pass the ball (as shown in figure 3-4①) or to carry further with the ball (as shown in figure 3-4②). If the distance is quite short, the defender will intercept the ball directly. If the opposite agents choose to pass the ball which is caused by angle limited (as shown in figure 3-4③), the defender will judge which player is the next ball personnel according to the position, and then to adjust the position with predicting. In addition, we are looking for the statistics about the opponents' major offensive

area and attacking routes and we will regard the statistical information as an important factor for predicating. The next ,we will make the best prediction with the consideration of the current situation of the pitch.

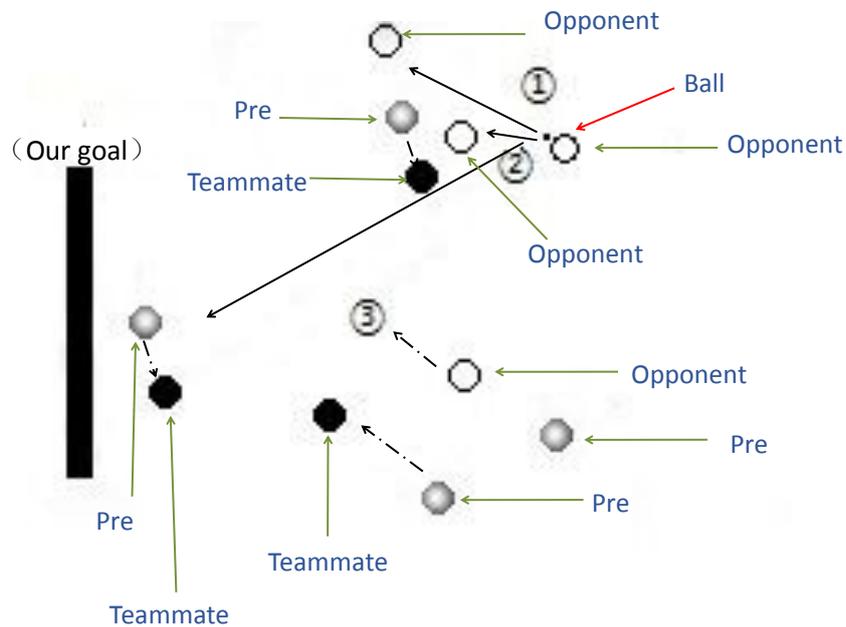


Figure 3-4 The schemes of the movement prediction of the ball

3.3 The optimization of based actions

Except in the design of high-level strategy, we also try to deal with the the based actions' problems of the team, to strengthen the based code and enhance the overall level of the team.

- 1) The optimization of formation marking. It is hard for us to serve when my teammates are pegged by the other agents when serve .This situation has changed in bhv_set_play.cpp which makes our agents flexible to prevent the peg.
- 2) To prevent the peg. We have noticed that if the opponent agent tackles the ball to our agents, the ball will touch our agent and get out, then the opponent will get the right to serve. Therefore, we optimize it to make the agents become more "smart",and to avoid the similar situation as much as possible .
- 3) Keeper will foul if he receives the ball. In based action , our goalkeeper in all clubs which are near the goal will save or kick the ball according to the judgment. Actually, catching the ball that passed by our own may lead to foul. Therefore, we will build a touch of judgment for preventing foul.
- 4) Individual agents may trap in the situation that they are with full-court running but having no effective actions. We optimize that situation and make a solution to the problem of bugs in the agent.

3.4 Code refactoring

In MT2016 code, all actions called by every doMove function of role are put into cpp file separately. With the continuous perfect and supplement, this form greatly increases the code coupling and reduces the independence. It is so inconvenient to continue that all the role_***CPP files are reduced to the original form only with execute (), doKick (), doMove function. However, all the concrete by callback function will be put into another CPP file to expand and perfect. In addition, the same code called by every different roles will be put into bhv_basic_move.cpp for all roles to call.

3.5 Summary and Outlook

Our team actively works hard to catch up with the world top teams. Although we have some problem, we believe that our team could deal with the problems and have our unique strategy. We will continue to communicate with other teams to study in the following competitions. We will learn the advanced parts of the other teams, and also let them find our strength at the same time. We hope that we can work together with other world Robocup teams and become better in developing artificial intelligence.

Reference Material:

- [1] Kalyanakrishnan, S., Liu, Y., Stone, P. Half field offense in RoboCup Soccer: A multi-agent reinforcement learning case study. RoboCup 2006: Robot Soccer World Cup X pp.72-85(2007).
- [2] Akiyama, H., Shimora, H., Nakashima, T., Narimoto, Y., Yamashita, K. HELIOS 2D simulation team description 2012. 2012 RoboCupWorld, Mexico. 2012.06.
- [3] <http://www.wrighteagle.org/2d/>.
- [4] <http://sourceforge.jp/projects/rctools/>.
- [5] Robot soccer simulation design and implementation of China University of Science and Technology.
- [6] Akiyama, H., Nakashima, T. HELIOS base. An open source package for the robocup soccer 2d simulation. RoboCup 2013: Robot World Cup XVII (2014)
- [7] Zhou Hui. soccer Agent Collaborative Research on RoboCup2D simulation. Nanjing University of Posts and Telecommunications. (2013)
- [8] Budden, D., Prokopenko, M. Improved particle filtering for pseudo-uniform belief distributions in robot localisation. In: RoboCup 2013: Robot Soccer World Cup XVII, Springer (2013)