

CSU_Yunlu 2D Soccer Simulation Team Description Paper 2013

Yili Zhou, Xiangcen Bo, Shuai Liu, Zhenyi Lin, Li Liu,
Jun Peng, Xiaoyong Zhang and Fu Jiang

School of Information Science and Engineering,
Central South University, Changsha,
Hunan Province, China

zhouyili92@gmail.com, jiangfu0912@csu.edu.cn

Abstract. CSU_Yunlu is a 2D soccer simulation team which has been participating in the RoboCup competition since 2002. In these past years, our team has got some achievements in this field and it was growing rapidly more than our expectation. In this paper, we briefly describe the scientific focus of CSU_Yunlu, our current research efforts and the recent improvements since the late competition.

1 Introduction

CSU_Yunlu 2D soccer simulation team, which was established in 2002 by School of Information Science and Engineering of Central South University, is a simulated soccer team for the RoboCup soccer 2D simulator. The team participated in the RoboCup China Open 2002 and ranked the 3rd place.

In 2004, CSU_Yunlu started to adopt UVA_base_2003 as the base code. In these past years, when we built our team on the base of UVA_base_2003, we used a dynamic role-based cooperation model and a decision algorithm based on the behavior of Multi-Agent System (MAS) in our team, so the team was good at cooperation. In the RoboCup China Open 2010, CSU_Yunlu got the 7th place of 2D soccer simulation. In the next year, we decided to change the base code from UVA_base_2003 to agent2d.

In 2011, CSU_Yunlu was based on agent2d-3.1.0, a well-know base code developed by Akiyama et al. Having our continuous efforts, CSU_Yunlu gained 2nd place in the RoboCup China Open 2011, which was a big breakthrough for us. In the Robocup China Open 2012, we ranked 5th place after we made a lot of improvements on the research of last year.

We hope we can participate in the RoboCup 2013 Soccer Simulation League, 2D competition to improve ourselves and test the development since the late competition.

2 A multi-agent cooperation model based on action prediction

The agent can string the suitable actions that it will do in next a few cycles into a chain instead of only deciding what to do for the next one cycle through the Chain_Action system which is raised in agent2d. The system has proved to be efficient. However, after a lot of experiments, we found the Chain_Action system cannot always make the right decisions according to the whole situation.

Considering that action selection based on multi-agent cooperation is more reliable, we adopt the multi-agent cooperation model based on action prediction to supply the Chain_Action system. This cooperation model consists of several parts, as shown below.

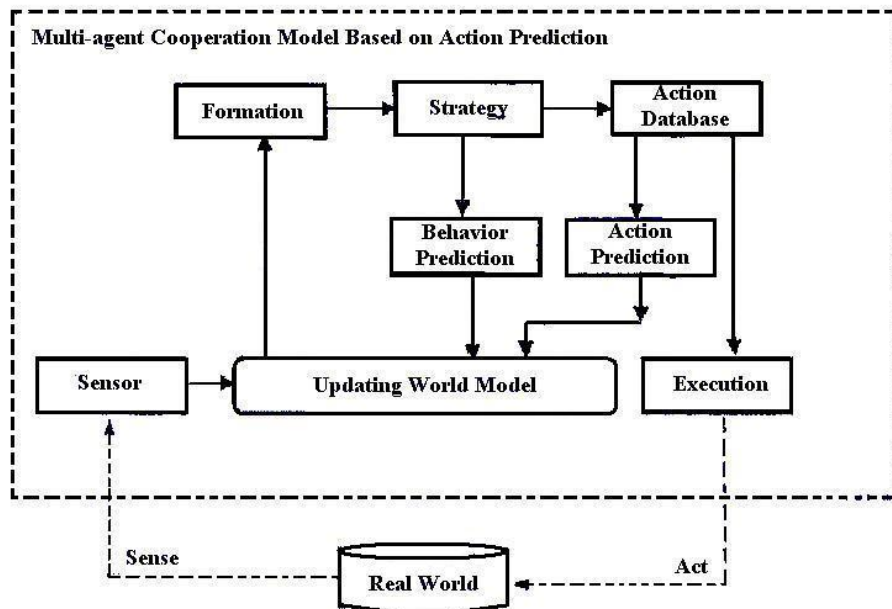


Fig. 1. Multi-agent cooperation model based on action prediction

The kernel in Fig. 1 is the behavior prediction and the action prediction. According to the current world model and known formation, the agent can predict the other agents' behavior through the part of behavior prediction. According to the current world model, known formation and strategy, the agent can predict the other agents' action through the part of action prediction. By updating world model through the behavior and action prediction, the agent can make the appropriate action decision to achieve the multi-agent cooperation and the joint goal--scoring. The part of behavior prediction adopts the prediction technology of IMBBOP (idea-model-based behavior outcome prediction) and the part of action prediction adopts the prediction technology of MMTB (mutual modeling of teammate behavior). The part of behavior prediction

decides the agent's behavior in multi-agent cooperation and the part of action prediction decide the specific actions to certain behavior.

The strategy improved the flexibility and cooperation of players and has proved to be effective when combined with the Chain_Action system.

3 Pass strategy

The pass skill is of critical importance in a soccer match. Through large amount of experiments, we found the old offensive strategy mainly focused on dribble and the accuracy of pass was weak. So we adopt a better pass strategy which can increase the possession percentage of CSU_Yunlu.

We can research the pass strategy by dividing the problem into two parts which are deciding the pass path and calculating the kick power. The selection of pass path is based on the position of the teammates without ball and the path should not be exposure in opponents' kickable area.

The main factor we should consider about the kicker is the power. It determines the initial velocity of the ball. Low velocity may lead to an easy interception by the opponents. Similarly, high velocity isn't always good for the pass. The receiver may intercept the ball with excessive high velocity difficultly. To find out the most appropriate velocity, we adopt the following algorithm.

The algorithm of pass is briefly introduced as the following.

1. For the purpose of decreasing the time of calculation and improving the accuracy of pass, we exclude the teammates who are too far, too near or around by too many opponents as receiver at first.
2. We draw a line from the kicker to each teammate who has been chosen by the step 1. Then we create some pass paths for each teammate as the following picture shows.

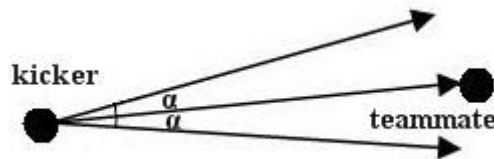


Fig. 2. Pass paths for each teammate chosen by step 1

3. Because we can't predict the intercept point of opponents, we decide to discretize the pass path into discrete points. To check whether the path is safe or not, we set n points on the path (the last point is the receiver's intercept point). The values of the distance from the pass point to n points set on the path are s_1, s_2, \dots, s_n . For each point, we calculate the reaching cycles of the nearest opponent at the maximum speed. We assume that the values of the cycles for m points are c_1, c_2, \dots, c_n .

4. In the model that the ball is kicked from point A to point B in N cycles, we can figure out the initial velocity V_0 of the ball in point A by the equation (1) in which S represents the distance from the point A to the point B and ball_decay represents the natural attenuation value of the ball.

$$V_0 = S / \left(1 + \frac{\text{ball_decay} \times (1 - \text{ball_decay}^N)}{1 - \text{ball_decay}} \right) \quad (1)$$

5. For $S = s_1, s_2, \dots, s_n$ and $N = c_1, c_2, \dots, c_n$, we can calculate the $V_0 = v_1, v_2, \dots, v_n$ by the equation (1). So we set the threshold of the velocity v_{thr} by the equation (2). Obviously, when the initial velocity of the ball is not more than v_{thr} , the ball will be intercepted by the opponent easily.

$$v_{\text{thr}} = \max\{v_1, v_2, \dots, v_n\} \quad (2)$$

6. Importantly, we should calculate the value of the cycles that the receiver needs to reach its intercept point at the maximum speed as c_0 . By equation (1), we can also calculate the initial velocity v_0 in the condition that the ball reaches the intercept point of the receiver in c_0 cycles. Only the initial velocity of the ball is less than v_0 , the receiver can intercept the ball. If $v_0 < v_{\text{thr}}$, the pass path should be abandoned obviously.
7. Finally, if $v_0 > v_{\text{thr}}$, v_{thr} is the lowest initial velocity of the ball and v_0 is the highest initial velocity of the ball. So we should choose an appropriate velocity between the two values as the safe initial velocity. In this way, we will get the safe velocity. The velocity can not only increase the success rate of pass but also save the kicker's stamina.

After executing the algorithm mentioned above, a number of practical paths will be created. We score each of them considering parameters such as the position of the intercept point, the receiver's stamina and so on. We choose the path with the highest score to pass. Through a lot of experiments, this pass strategy proves to be effective to enhance the possession percentage of our team.

4 Evaluation based on dynamic position

We inherit the evaluation mechanism of last year which is described in TDP_CSU_Yunlu2012. To adapt our team to the dynamic feature of the match, we add a new evaluation mechanism based on dynamic position. The field is divided into three parts as Fig.2 shows. We draw a circle with specific radius around each opponent and the radius varies with which division the opponent is in. The score of the area inside the circle is negative and it is different based on the field partition which the opponent is in.

The new evaluation mechanism weakens the success rate of being intercepted by opponents effectively.

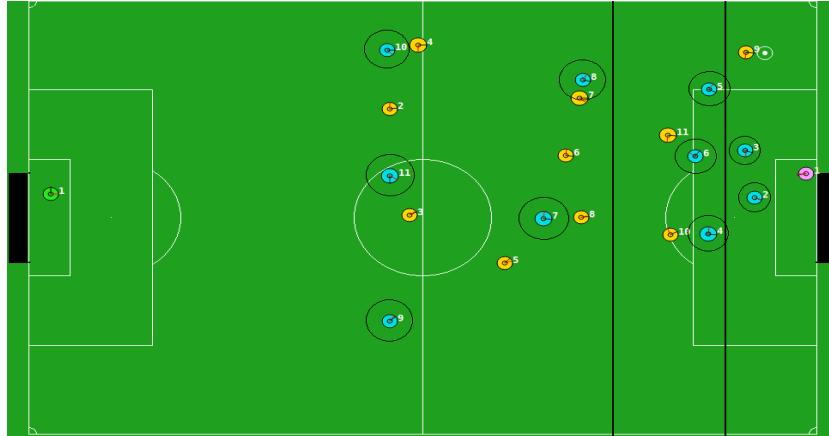


Fig. 3. Field partition in evaluation mechanism

5 Defensive strategy based on the field partition and role

In the released source code agent2d-3.1.0, the field partition has existed but it isn't used in the defensive part. In the base code, the pitch is divided into 12 parts as the following picture. When we improve the defensive part, we adopt the strategy of field partition.

The agent without ball chooses an appropriate action to make, which is depended on the position of the ball and the agent's role. This strategy is just like the reality. For different roles we write different functions when the ball is in the different zone. Finally, the agent makes the action selection considering the allocation of roles and the region where the ball is.



Fig. 4. Field partition in defense

6 Formation and positioning

Our formation inherits from CSU_Yunlu2011 and we make some improvements. When the agent is marked by the opponent, we use the equation (3) to update the agent's strategic position.

$$\begin{aligned} \text{posStraN}_x &= \text{posStra}_x + (\text{posStra}_x - \text{posOpp}_x) \times \text{attr}_x \\ \text{posStraN}_y &= \text{posStra}_y + (\text{posStra}_y - \text{posOpp}_y) \times \text{attr}_y \end{aligned} \quad (3)$$

($\text{posStra}_x, \text{posStra}_y$) which is the agent's strategic position based on the position of the ball has defined in the formation files. With the software Fedit2, we can directly modify the agent's position according to the position of the ball. ($\text{posOpp}_x, \text{posOpp}_y$) is the position of the opponent who marks the agent. The value of attr is based on the agent's role and the field partition which the ball is in. The field partition is like the Fig.4. ($\text{posStraN}_x, \text{posStraN}_y$) is a new strategic position to avoid being marked after calculating.

7 Conclusion and future work

In this paper, we have briefly presented our current work to CSU_Yunlu. Although we have made a lot of effort, the team still has plenty of room to develop. In the next period, we will focus on the reinforcement learning method and the intelligent cooperative-defense strategy.

References

1. Peng, J., Wu, M., Cao, W.: Some Key Techniques in RoboCup Simulation League [J]. *Computer Engineering*, 2004, 30(4): 49-51
2. Peng, J., Wu, M.: Multi-agent Cooperation Model Based on Action Prediction [J]. *Computer Engineering and Applications*, 2005, 41(9): 23-25
3. Zhang, X., Peng, J.: Realization of Pass Ball Strategy in RoboCup [J]. *Computer Engineering*, 2004, 30(23): 123, 149
4. Li, S., Chen, J., Sun, Z.: Structural design and implementation of Tsinghua Robot Soccer Team [J]. *J Tsinghua Univ (Sci & Tech)*, 2001, 41(7): 94-98
5. Yun, J., Zhang, X., Wei, X., Wang, C.: Key tactics of intercepting the ball, possessing or dribbling the ball and running in RoboCup [J]. *Journal of Inner Mongolia University of Science and Technology*, 2009, 28(2):136-142
6. Peng, J., Ding, C.: Realization of formation strategy in RoboCup. In: *The 2006 International Conference on Artificial Intelligence*, 2006/6/26.
7. Akiyama, H., Shimora, H.: HELIOS2010 Team Description. In: *RoboCup 2010: Robot Soccer World Cup XIV*. Volume 6556 of *Lecture Notes in Computer Science*, Springer (2011)