Team Description of Apollo

Jie Ma, Mao Li, Guoxia Qiu, and Zhiyong Zhang

Department of Computer Science
Nanjing University of Posts and Telecommunications, Nanjing, 210003, P.R.China
me@goodma.com, limao516@yahoo.com.cn, Y031004@njupt.edu.cn, zyzh@njupt.edu.cn

Abstract. This paper describes the main features of the Apollo soccer simulation team, which won the China RoboCup 2004 Simulation league champion in Guangzhou. We explore two major themes with this team: online-adaptable coordination and machine learning**.**

1 Introduction

Apollo is the 2D simulation league from Nanjing University of Posts and Telecommunications which is built on TsinghuAeolus 2002 [1]. We have made three extensions to our team. Firstly, we fixed many programming bugs of the low decision level, including Newton Method to calculate interception points [1] and the function to get offside line. Secondly, we have improved some basic skills such as interception, pass and dribble. Finally, we have made use of several methods to establish the online-adaptable coordination between agents.

2 Dribble Skill

The traditional purpose of dribble skill is to keep the ball in the kickable area every cycle and out of the kickable areas of all opponents [2]. According to the experiences of RoboCup2003 and RoboCup2004, we feel that rapid attacks can often bring good chances to shoot. But if we pay too much attention to dribbling security, we are difficult to breakthrough the opponent defense line in 2D competitions.

Our purpose of dribble skill is to balance between attack rapidity and security. One successful dribbling procedure can be simply described as follows:

Step 1. Stop the ball (if necessary).
Step 2. Select the dribbling direction, the angle to kick and the dribbling cycle.
Step 3. Turn the body to the dribbling direction (if necessary).
Step 4. Kick the ball.
Step 5. Dash in the following several cycles.

Step 6. The ball is in the kickable area again.

In addition, we have to avoid collision induced by random noise every cycle. In order to decide the dribbling direction and the angle to kick, we have defined a dribbling channel model and a dribbling state model. We made use of a back propagation artificial neural network to get the best receiving point. In our current program, we use hand-code to get the dribbling cycle. And we are exploring a more available way to evaluate dribbling cycle.

3 Online Adaptable Coordination

Online adaptation is an important part of a multi-agent system. We have tried some interesting approaches to meet the needs of the competition.

3.1 Improving collaboration on non-PlayOn playing mode

Since the RoboCup simulation league is a distributed multi-agent system, the cooperation among the team players sometimes becomes difficult, which is especially apparent on non-PlayOn playing mode. For the appropriate collaboration, we have introduced some refinements that establish some pieces of inner common rules to specify the proper actions amid given circumstances. Each player of our team must abide these common rules. Experiments show that these refinements reduce conflicts between agents and improve our team's scoring ability dramatically.

3.2 Opponents Modeling

The opponent modeling is one of our most interesting problems, but there is still a long way to build an effectual model [3, 4]. At the present time, we can record and analyze some opponents' behaviors, such as kick in, corner kick, and goal kick. Then, we can adjust our strategy accordingly.

4 Conclusion and Future Directions

In this paper we have presented a brief description of the Apollo RoboCup simulation league's improvement. We haven't systemically test the performance of the program modified by us, but the outcome of the three regional tournaments in which we participated does encourage our team to explore optimization of our program as a whole.

In order to make our team more flexible and more self-adaptable we are currently working on the other two themes: online position adjustment [5] and online tactic advice

[6]. For future directions, we are interested in applying an efficient tracking mechanism [7], which provides an available method to say more information in limited length of characters per cycle.

References

1. Jinyi Yao, Ni Lao, Fan Yang, Yunpeng Cai, and Zengqi Sun: Technical solutions of TsinghuAeolus: in RoboCup Symposium 2002
2. Jinyi Yao, Jiang Chen, Yunpeng Cai, and Shi Li: Architecture of TsinghuAeolus: Available in http://www.lits.tsinghua.edu.cn/RoboCup/RoboCup/Docs/Tsinghuaeolus.ps
3. Stone, P., Riley, P., Veloso, M.: Defining and using ideal teammate and opponent models, in: IAAI-2000. (2000) 1040-1045
4. Riley, P., Veloso, M., Kaminka, G.: An empirical study of coaching. In Asama, H., Arai, T., Fukuda, T., Hasegawa, T., eds.: Distributed Autonomous Robotic Systems
5. Timo Steffens: Feature-based Declarative Opponent-Modeling
6. Gregory Kuhlmann, Peter Stone, and Justin Lallinger: The Champion UT Austin Villa 2003 Si-mulator Online Coach Team
7. Lao Ni, Yao Jinyi, Cai Yunpeng, Yang Fan, and Sun Zengqi: Efficient Tracking of Planar Moving Objects from Local Moving Scope