WrightEagle 2D Soccer Simulation Team Description 2013

Haochong Zhang, Miao Jiang, Haibo Dai, Aijun Bai and Xiaoping Chen

Department of Computer Science,
University of Science and Technology of China,
solomonz@mail.ustc.edu.cn, xpchen@ustc.edu.cn

Abstract. WrightEagle 2D soccer simulation team is a 2D soccer simulation team which has been participating in annual RoboCup competitions since 1999 and won 3 champions and 5 runners-up in the past 8 years. In this paper, we briefly present our current research efforts and some newly introduced techniques since the last competition.

1 Introduction

WrightEagle 2D soccer simulation team, which was established in 1998 as the first branch of WrightEagle RoboCup team developed by Multi-agent Systems Lab. of USTC, has been participating in annual competitions of RoboCup since 1999. In recent years, we have won the champions of RoboCup 2011, 2009 and 2006, the runners-up of RoboCup 2012, 2010, 2008, 2007 and 2005.

We take RoboCup soccer simulation 2D as a typical problem of multi-agent systems, and we concentrate on planning algorithm research and other challenging problems in artificial intelligence [2]. This year, we developed some new techniques for both the fundamental tools and the high level decision-making model in our new team WE2013, based on our research efforts[3,6,7,5,4,10,9,1,8].

In this paper, we present a brief description of some of our progress mentioned above.

In Jan 2013, we also released the latest version (4.0.0) of our team's base code WrightEagleBASE to the public as an open-source software which can be freely accessed from our team's website.[1] After this update, we gave WrightEagleBASE a new module called "Trainer". We believe that will greatly expand the scope of the use of WrightEagleBASE. We hope that our released software will be helpful to a team who wants to participate in the RoboCup event and/or start a research of multi-agent systems.

The reminder of this paper is organized as follows. Section 2 introduces the trainer system, especially how it works and what can it do. Section 3 presents the method we use to improve shot action. Finally, the paper is concluded in Section 4.

[1] `http://www.wrighteagle.org/2d/`

2 Trainer

Modular testing is one of the most effective and focused way of testing, which is crucial for software development. Generally speaking, a new feature is proposed because of bad performance in some special scenes. After completing the patches, we need to do some test in order to ensure the new feature works well. However, this test is hard to do before the train system is developed. On the one hand, due to the complexity of the simulation 2D problem, we cannot test the module alone. On the other hand, the scene, in which the new feature could work, is rare in one match. The test procedure is time-consuming and ineffective if we play multiple matches to test new features. As a result, the trainer is a good way to resolve this dilemma.

The trainer, in other words, the offline-coach, can reproduce scenes one needs very easily. Under the environment provided by rcssserver-15.1.0, the trainer can do the following actions besides the actions which the online coach can do such as say or change player type command:

1. Move player
   Move any given player to a given point with a certain status (only including his velocity and body direction).
2. Move ball
   Move the ball to a given point with a given velocity.
3. Recover
   Recover all the players's stamina and stamina capacity.
4. Change play mode
   Change current play mode to a given play mode, such as free kick left and play on.

As a result, the specific scene, called training scene, can be reproduced to some degree by using these functions. For the convenience, it has been designed to be able to parse the rcg file. In practice, the user can tell trainer the training scene by giving a cycle and a rcg file.

In addition, sometimes we needn't wait to start another training session until the end of the game. For example, a new feature about interception of goalie is developed. What we care about is only the chance that the goalie catches the ball. The goal kick after that is another problem which needn't be tested here. Therefore, the "end condition" system has been developed. The user can specify the end condition of a train process, such as the distance between the ball and the goalie shorter than 1m, to enhance pertinence of training. Nevertheless, the trainer will generate a training report, which contains some statistical information of training, after the end of all the training episode.

It should be noted that,the visual information in the current model provided by rcssserver-15.1.0 is limited. It results in the great inaccuracy of the players' world state. Most basic reason is that a player cannot see all his teammates and opponents at the same time, he doesn't know their position has changed greatly. In this case, the scene in the player's mind is not the training scene. In order to reduce its effect, there should be some time, which we called initial time, for him

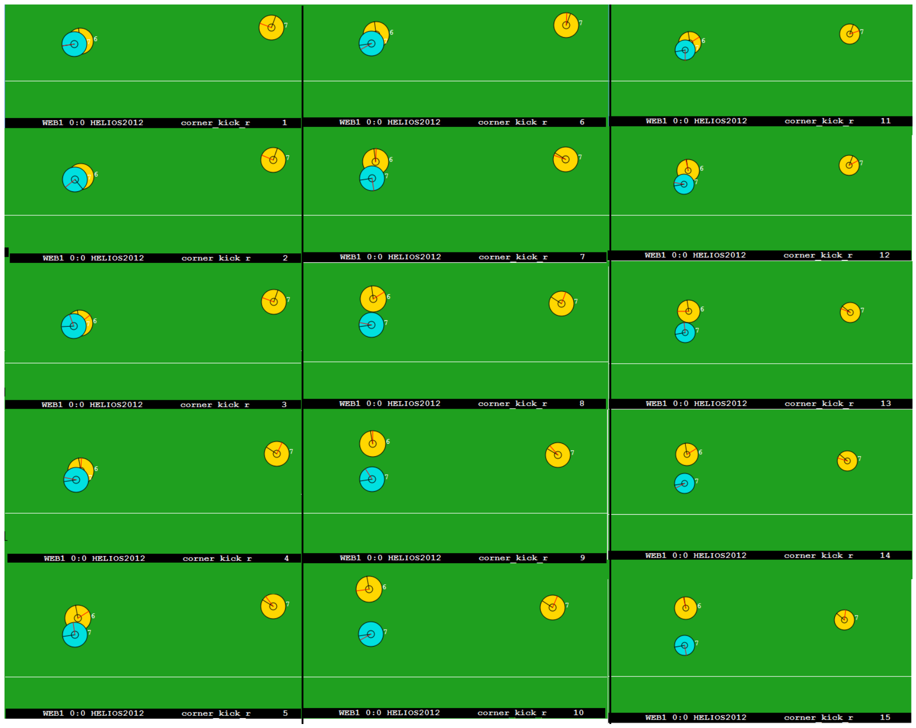

Fig. 1. The trainer has been triggered if the end condition is satisfied. In this case, the condition is "Time = 5" which means that one training process will not last for more than 5 cycles. Of course, you can set a longer time limit in practice. This is just for example. You can see that all the players, whether the opponents or teammates will be back to the initial state of the training process.

to acquaint himself with the surroundings. During the initial time, the trainer will reproduce the scene before the training scene sequentially. The player could have a precise world state when the training starts.

In summary, the trainer enables us to test specified module in a more targeted and effective way.

3 Higher precision shot decision-making system

A shot decision-making system aims at seeking the best shot solution ,when the player faces the goal and defenders. We can calculate goal probability without defenders and intercept probability by defenders as mutually independent events.

In previous WrightEagle, we encountered problems caused by calculation error of shot success probability. In this year, we improved our algorithms to reach higher precision.

To advance the precision of goal probability without defenders, we replace Newton-Coates algorithm by normal-distribution-table based algorithm. And we also enhance our algorithm using Linear Interpolation Algorithm. With all these efforts, the error of algorithm is reduced by five orders of magnitude, and the accuracy almost reaches the true probability by simulation algorithm.

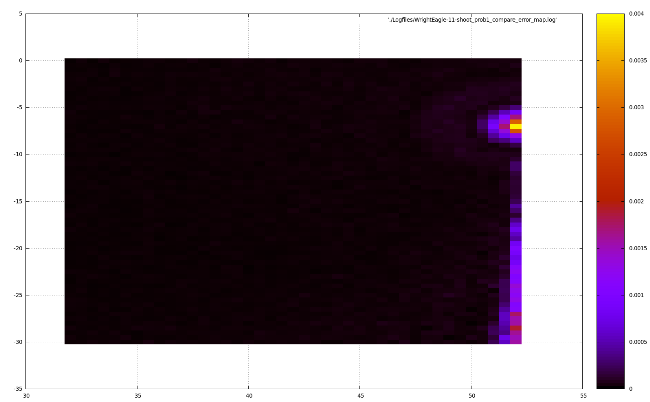

Fig. 2. shoot probability error in the field

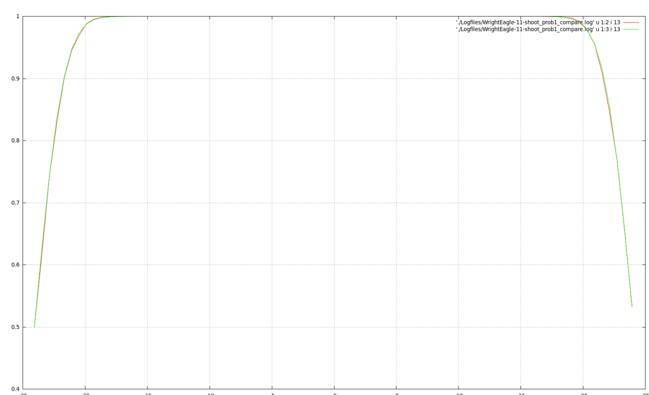

Fig. 3. shoot probability curve comparison between new method and offline simulation result

Considering the impact caused by defenders, we first assume that the defenders all have their best condition to intercept the ball in the nearest position. We can calculate a probability in this hypothesis. Then as every opponent team can not reach the intercept success probability by ideal defender, we should add a factor to weaken the defender. Though different opponent team has different defender, to simplify, we take helios' defender as the most minatory defender.

Our future work focuses on how to manage defenders, such as opponent team modeling in shot decision.

4 Conclusions

This paper introduces our RoboCup soccer simulation 2D team, WrightEagle, and described our current research efforts and some newly introduced techniques since last competition. The new trainer system would be a powerful tool in modular testing in team development and learnning algorithm implementation for all the teams which use WrightEagleBASE.

5 Acknowledgments

The authors would like to express their gratitude to Guanghui Lu, Xiao Li, Rongya Chen for their contributions to WrightEagle2D simulation team and Mengya Hu for her helpful comments and remarks for this paper. This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150 and the Natural Science Foundation of China under grants 60745002 and 61175057, as well as the USTC 985 Project.

References

1. Bai, A., Wu, F., Chen, X.: Online planning for large mdps with maxq decomposition (extended abstract). In: Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems. Valencia, Spain (June 2012)
2. Chen, X.: Challenges in research on autonomous robots. Communications of China Computer Federation 3(12) (2007)
3. Fan, C., Chen, X.: Bounded incremental real-time dynamic programming. In: Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007. pp. 637–644. IEEE (2007)
4. Shi, K., Chen, X.: Action-driven markov decision process and the application in robocup. Journal of Chinese Computer Systems (2009)
5. Wu, F., Chen, X.: Solving large-scale and sparse-reward dec-pomdps with correlation-mdps. RoboCup 2007: Robot Soccer World Cup XI pp. 208–219 (2008)
6. Wu, F., Zilberstein, S., Chen, X.: Multi-agent online planning with communication. In: Proc. of the 19th Int. Conf. on Automated Planning and Scheduling. pp. 321–328 (2009)
7. Wu, F., Zilberstein, S., Chen, X.: Point-based policy generation for decentralized pomdps. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1. pp. 1307–1314. International Foundation for Autonomous Agents and Multiagent Systems (2010)
8. Wu, F., Zilberstein, S., Chen, X.: Online planning for ad hoc autonomous agent teams. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)

9. Wu, F., Zilberstein, S., Chen, X.: Online planning for multi-agent systems with bounded communication. Artificial Intelligence 175(2), 487–511 (2011)
10. Zhang, Z., Chen, X.: Accelerating point-based pomdp algorithms via greedy strategies. Simulation, Modeling, and Programming for Autonomous Robots pp. 545–556 (2010)