

SEU-2D 2006 Team Description

Chunjian Xu, Tsingfa Meng, Yingqiu Xu

Southeast University, Nanjing 210096, P.R.China
xcj1980s@hotmail.com, alphameng@163.com, yingqiuxu@yeah.net

Abstract. This paper describes the main features of SEU-2D robot simulation soccer team. After a brief introduction of SEU_T 2005, the main contributions of SEU-2D are presented, hybrid agent architecture, compound kick, and heterogeneous agent selection. Finally, we describe our future study directions.

1 Introduction

The SEU-2D robot simulation soccer team, previous name was SEU_T, participated in RoboCup 2005 Soccer Simulation competition, and ranked 15th.

Our code is based on UvA Trilearn 2003 source code. To make the code easy to read and modify, we split our code into two independent parts: agent and coach. Most important, we have made several extensions to our team. Firstly, we applied hybrid agent architecture. Secondly, we use transaction, which sends the soccer server newest message based on sense body message and reasoning time, to maintain the consistency of the actions sent to the soccer server with the agent's world model. Thirdly, we implement a new kick skill, which is better than the previous one, and a better dribble ball action to our team. Finally, we introduce heterogeneous agent selection to our team.

2 Agent Architecture

The SEU-2D robot simulation soccer team use hybrid agent architecture. It consists of modeling, communications, skills, domain knowledge, and deliberative reasoning. Since the soccer server's low bandwidth limitation, we only use communication to share world model between agents. The modeling module gathers information from the real world, and abstracts them into an internal form which is used by reasoning. The deliberative reasoning module is our high level decision component. It reads the information abstracted by the modeling, then uses domain knowledge to decide which skill should take. Finally, the skill component decomposes the task into several soccer commands and sends them. Figure 1 depicts our agent architecture.

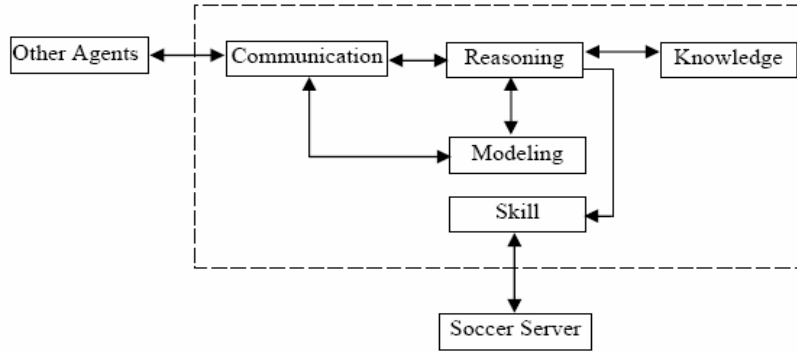


Fig.1. SEU-2D Agent Architecture

3 Compound Kick

Whether to be skillful in completing a kick task, such as pass, shoot, is critical in robocup soccer match. A lot of work have studied the problem. The most promising one is Q-learning combined with adversarial planning introduced by TsinghuAeolus. Generally, we adopt their idea, but made a lot of improvements. Firstly, we optimized for heterogeneous agent's kick rand and kick margin. Secondly, we use a new decision process which is totally different from TsinghuAeolus to map the real continuous space to discrete training state.

A kick task is that given a initial state in current cycle, planning a series of kick to accelerate the ball to desire velocity. The decision process should be efficient, robust, and adversarial. Like TsinghuAeolus, our solution includes two steps. The first one is offline learning. The second one is online planning. In offline learning, the spaces and actions were discretized. The Q-learning method is adopted to evaluate different actions. We use a variable reward method to optimize for heterogeneous agent's kick rand and kick margin. In the second one, we use a mapping method to map the continuous spaces in real math to discrete spaces in offline learning.

Define the player's center as the coordinate origin, the ball's desired velocity direction as the x-axis. In the offline learning phrase, we only consider the ball's relative position and desired speed, all the other things were ignored, i.e. agent body facing, agent global velocity, ball initial speed. The ball was randomly placed in the player's kickable margin area, then the kicker learned how to use a lot of small kicks to accelerate ball to desired speed. In order to optimize for heterogeneous agent's kick rand and kick margin, we use a variable reward method to evaluate the actions agent selected.

If the ball can be accelerated to the desired speed in the next cycle, the reward is $1.0 - \text{cost}(A)$. Otherwise the kicker get the reward $0.05 - \text{cost}(A)$.

$\text{cost}(A)$ is the cost of the action agent take action A .

$$\text{cost}(A) = \alpha \times bvel + \beta \times fabs(bdist) / margin \quad (1)$$

$bvel$ is the absolute speed the ball can get when use A , and $bdist$ is the relative distance the ball travel. In our reward scheme, the higher velocity and further distance the agent kick ball to, the lower the reward he got. Thus we encourage the kicker to kick ball small and short. Use this technique, the influence of the heterogeneous agent's different kick rand and kick margin were decreased to least.

In the online phase, we use a lot of functions to map the continuous spaces to the discrete training one. We consider all other things ignored by offline learning, i.e. agent body facing, player's move, desired velocity direction. For compensating the agent body facing, we use function f_1 to rotate the ball's position to a corresponding training one, following the criteria that in both cases the effective kick power was the same. With the same idea, we use f_2 to compensate the player's move and f_3 to mediate the real desired velocity direction with the offline learning cases. We also use f_4 to avoid side line and f_5 to avoid closing opponent. Finally, we combined the five small mapping functions into one to decide which discrete space should be used. Figure 2 depicts how we compensate agent body facing.

We test our method in 10 real matches, the result is perfect. The agent can accelerate ball to desired speed as fast as possible and with little influences of the kick rand and kick margin. Our implementation was better than TsinghuAeolus, since we take more things into account and we can fully use Q -table, while their implementation can not.

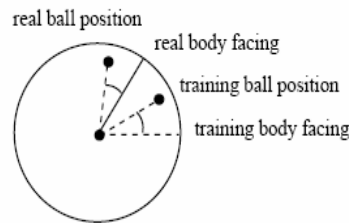


Fig.2. Example for Compensating Agent Body Facing

4 Heterogeneous Agent Selection

We abstract 4 criterions from 11 heterogeneous parameters. Firstly, we calculate the agent's start up cycle. Although the player's max speed was fixed to 1.2 in current soccer server, different agent can reach the speed using different time, the smaller the better. Secondly, the stamina factor is taken into consideration. When the player run in max speed, he will lose stamina, the little the better. We consider the kick rand and kick margin factor finally, because our kick skill was influenced little by the two parameters.

5 Conclusion and Future Work

In this paper, we have discussed our main work done in SEU-2D. For future study, we will concentrate on using the transactions to high level decisions. We are interested in keeping on improving our agent's individual skill using reinforcement learning. We also decide to do more work on cooperative action decision of MAS and online adaptation, to make our team more flexible and more self-adaptable.

References

1. Remco de Boer, Jelle Kok. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team [Master Thesis]. Artificial Intelligence and Computer Science, University of Amsterdam. February 2002.
2. Jinyi Yao, Jiang Chen and Zengqi Sun. An application in RoboCup combining Q-learning with Adversarial Planning. The 4th World Congress on Intelligent Control and Automation.