Wright Eagle 3-D Simulation Team

Bai Peng, Chen Xiaoping, Li Yang, Liu Jinsu, Miao Qiankun,
Wang Guibing, Yang Zhenyu, Zhang Xiuwu

AI Center of Department of Computer Science and Technology,
University of Science and Technology of China,
Hefei,China
baipeng@ustc.edu
WWW home page: http://mas.ai.ustc.edu.cn

Abstract. This paper introduces the implementation of Wright Eagle 3-D Simulation team. Simulation 3-D is a new domain of RoboCup Soccer-Simulation, which is more realistic than the old 2-D. We have tried several different approaches to resolve problems in the extension to 3D. In our application, we built a simple model to ensure both robustness and efficiency of agents' action.

1 Overview

In this paper, we mainly talk on basic actions of the 3-D world agent. When it comes to high level models, we employ some old 2-D code. Since the 3-D soccer server has not been complete yet, we use several simple models for the agent's movement and kick action. The purpose on the implementation is to test whether the agent also can act rightly in the new world. This Description is separated into two parts, the overview of major actions and detail implementations of agents' basic actions and the decision model.

1.1 Basic Actions

Snatch ball and run to the strategy position are important parts of the basic actions in robot soccer. We mainly focus on how the player intercepts the ball and how the player goes to the right position.

In the 2-D world, one agent is able to calculate the best intercept point accurately. When it comes to 3-D world, the situation is totally different. The ball moves inside the 3-D space, and the agent have to know where and when to get the ball and to ensure that action occurs before the opponent's. Although the intercept model will be clearly described, there is still a big problem that the time consuming is beyond the decision time limit. It is the same with kick action. The current kick action can only kick the ball into the front direction of the agent body, and the kick angle of the latitude is limited so that it is hard to control the kick-ball action and the complex kick action is impossible here.

1.2 Decision Making

In this part, we employ several functions to evaluate the state of the agent and to estimate the risk factor for agent's choice of actions. In a word, it is still a MDP model.[1]

2 Implementation

For the limited time from the stable server came up to the qualification deadline, we can only achieve some simplest modules for our team. Whatever, This paper will refer to some unfinished modules.

2.1 Run With the Ball

This is a practical action. The main purpose is to let the agent kick the ball all the way to a requested position while running. During the process, we have to ensure the control of the ball (grabbed by opponents is not considered here), not to lose the ball by kicking too far, e.g. As the agent could only kick the ball in the direction ahead, and the server doesn't provide a single turn action, some actions are needed when want to kick the ball to a specified direction. For the implementation, we dynamicaly changes the drive-force of the agent to face to the target direction. If the agent have an initial speed which isn't in the needed direction and it is too high (greater than a certain value), we have to firstly reduce this speed and adjust our direction and power of driving to accomplish our task. This could approximately make the agent arrive at the specific position with right direction and proper speed.

By this action, we complete the $RunWithBall$ action as following: When the agent is far away from the ball, try to reach the ball and turn to the target direction; When the ball is close enough for kick, a kick action with a force to make the ball move a small distance will be executed; at last the agent follows the ball and get ready to continue to kick again. When close to the destination, the process is accomplished.

2.2 Handle Ball

Stop the Ball This skill is to freeze a moving ball. It returns a kick command that stops the ball at its current position. Since ball movement in the soccer server is implemented as a vector addition , the ball will stop in the next cycle when kicked in such a way that the resulting acceleration vector has the same length but opposite direction of current ball's velocity. The desired speed that should be given to the ball on the kick thus equals the current ball speed $vBall$ and the Power that must be supplied to the kick command to achieve this.

Kick Ball This skill is to kick the ball from its current position to target position $posTarget$ in a way that it has a remaining speed equal to $dEndSpeed$ when it reaches this position.

2.3 Positioning

The behavior of run to the correct position is very important for a player. If some impediments in his path, we must find a path planning for the him. We use the APF method here.[2] As for the strategic positioning, we use the old SBSP method.[3]

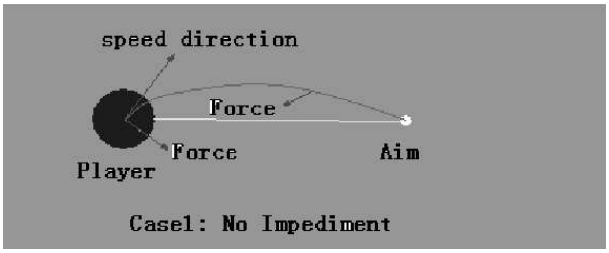

Fig. 1. Case 1

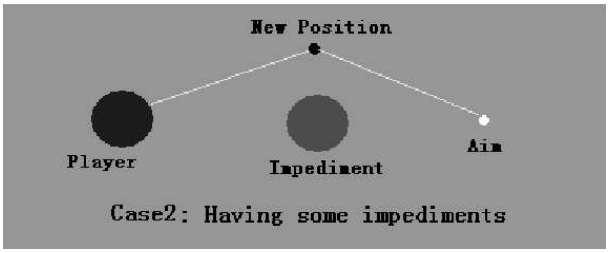

Fig. 2. Case 2

2.4 Intercept

Snatching ball from the opponent players is very complex. We must try our best to snatch the ball controlled by opponent players.

If the player wants to snatch the ball from the opponent players, he must have some information at first, such as the ball's speed, position, the resistance force of the ground, the opponent player position, speed. If he got enough parameters, he can calculate the track of the ball, and compute the best position to catch the ball.

An Example: If we know the ball's speed (V_x, V_y, V_z), position (P_x, P_y, P_z), the resistance force of the ground(f_x, f_y, f_z), we can calculate the track of the ball as follows: After t cycles, the ball's position is:
$(Pos_x, Pos_y, Pos_z) = (P_x, P_y, P_z) + (V_x, V_y, V_z) \times t + 0.5 \times (f_x, f_y, f_z) \times t^2$
Then, the player can decide to which is the best position to run.

2.5 Decision Making

Currently, we do not have much new ideas about this problem, we are planning to use the MDP[1] decision method derived from the old team.

There is a simple model of the decision for our qulifacation team. Firstly, we define x as whether there is a ball is in the controller's area.If x equals 1 the agent is with ball, and x equals 0 means not.We define y as whether there is an opponent close to the controller. If y equals 1, there are some opponents, and y=0 meams not. We define z as the types of the agent, which means

$z = 4$, Forward; $z = 2$, Midfield; $z = -2$,Defenders; $z = -4$,Goalie.

We use the evalutate function

$$value = (1 - x) \times (1 - y) + x \times (2x + (1 - y)) \times z$$

to calculate the decision value. if the $value$ belongs to $(8, 12)$, the agent will be specialized to attack.When the $value$ belongs to $(4, 6)$, to midfield organizing. When the $value$ belongs to $(-6, -4)$, to defend. When the $value$ belongs to $(-8, -12)$, to goal keeping. Note that if $value = 1$,there is no ball and no other agent, the agent will do nothing. If $value = 0$,there is another agent in its close area, it will make a further decisions. If the agent is a forward or a midfiledplayer, it will breakthrough the loose opponents. If the agent is a defender,it will chase. If the agent is a goalie, it will keep the goal.

Attack When the distance to goal is far, the agent make a decision to run with ball, when the distance is close enough, the agent will shot.

Midfield The midfield players will pass the ball to the forwards,or execute the runwithball.

Defence The defender will intercept the ball,and then pass the ball.

Goalie The goalie will intercept the ball,and kick the ball to the other side of field.

3 Future Work

We only refer to some rough ideas here, and there is too much work to do in the future. The agent is expected to act more freely in the 3-d space, and we have a lot to do for the basic level of the agent. The intercept modules will be one of the most important parts in the future work. The kick action requires more complected calculation and more CPU-time. The agent also need some body action such as juste. Above all, we need to test whether the old decision module can still work in the completely new environment.

References

1. Sutton, R.S., Precup, D., Singh, S. *Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning*, Arificial Intelligence 112:181-211. 1999
2. P.Vadakkepat, V.C.Tan and W.Ming-Ling, *Evolutionary Artificial Potential Fields and Their Application in Real Time Robot Path Planning*, from the URL `http://www.ee.nus.edu.sg/stfpage/elepv/publication/cec2000.pdf`
3. Luís Paulo Reis, Nuno Lau and Eugénio Costa Oliveira, *Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents*, Robocup 2000