LEAKIN' DROPS 2009

Soccer 2D Simulation Team Description

Meisam Vosoughpour[1], Amir Tavafi[2], Sasan Dalir Arznaghi[2], Nima Nozari[2], Sina Sheikholeslami[2]

[1] University of Tehran, Department of Electronics and Computer Engineering,
[2]Mirza Koochak Khan High School, National Organization for Development of Exceptional Talents (NODET)

Abstract. *LEAKIN' DROPS* is a soccer 2D simulation team based on Mersad05 released code and placed 12th after participating in the RoboCup 2008 competitions at SUZHOU China. In this paper we describe our works in 2008 and the beginning of 2009 including both the algorithms extending imperfect already existing modules such as blocking system and some newly added modules such as collaborative marking system.

Keywords: RoboCup 2D Soccer Simulation, LEAKIN'DROPS, Mersad, multi-agent systems, Marking System, Artificial Intelligence.

1 Introduction

LEAKIN' DROPS soccer 2D simulation team, based on Mersad 2005 released source code, and participated and placed 12th in the RoboCup 2008 SUZHOU China, started to work in the extension of our previous team, Marlik, which was placed 2nd in the first Khwarizmi robotic competitions and the 2nd in the first national high school robotics competitions. According to all incomplete modules implemented in last stages of Mersad project, we found it promising to complete these remaining parts, and also add some new ideas which were come up during two past years of soccer 2D simulation, using the experience of our team leader, a member of Mersad 2004 and 2005.

In this paper, we first describe a big picture of Mersad [1][2] with all its major ideas and implemented modules, and then we sketch a general idea for defense system that is no longer based only on perfect blocking skill but a mixture of marking and blocking skills where a synchronization algorithm between them can be tactfully applied.

2 Mersad's Architecture

Mersad's Architecture consists of five layers [2], four executive and one data layer. The only data layer, "Information" (also called as "WorldModel" in the source code), consists of all information about the surrounding world the agent is living in and reacting with.

The lowest executive layer, "Connection and Synchronization", has two major tasks; the first task is sending all commands the above layer, "Basic Actions", wants it to send to the soccer server and also updating the Information Center through whatever received from the server as the natural information of the environment; the second task, as it can be understood out of its name, is to synchronize the agent's activities and decision timing with that of soccer server.

The first layer above the "Connection and Synchronization" is called "Basic Actions" layer. This layer is implemented in some classes each which support one of the primitive and basic skills of the agent that is needed to act in the environment in the lowest possible level of complexity.

The next executive layer is "Advanced Actions" layer. This layer fulfills more complicated tasks that can neither be counted as a Basic Action - because they are actually a mixture of application of different basic actions in a high level procedure - nor be classified as high level multi-cycle skills that were mainly added to Mersad architecture in 2005 as "Plan" layer.

Above all three executive layers, explained in the above lines, "Advanced Decision" layer is situated. This layer determines which of the actions proposed by the "Advanced Actions" layer is better to be done as the only action the agent can do in the current cycle.

The highest layer in this architecture is "Coach" layer that is concerned with everything an agent can interchange with the coach. This interchange is actually containing some information or advices that coach has extracted from the past cycles of the game. These received advices and information are mainly affecting "Advanced Decision" and "Plan" layers [1][2].

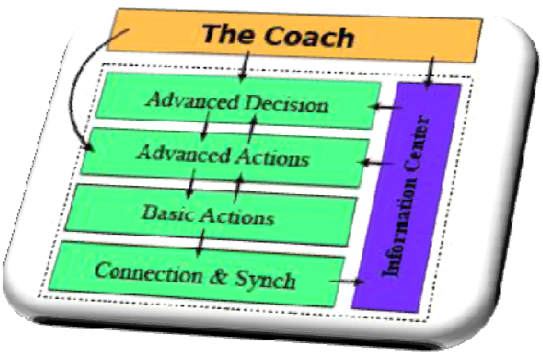

Fig. 1. Mersad's Architecture

3 Defense

In spite of all soccer 2D released codes, Mersad released code is not limited to its basic modules and contains all advanced modules in Mersad 2005 with detailed implementations. A strong point in Mersad 2005 is its defense system. Defense systems in soccer 2D simulation are mainly based on two skills, blocking and marking.

In RoboCup 2004 and before, teams used to have some limited algorithms in dribbling, passing, and offensive positioning in such a way that a team with a defense system with the power of Mersad's defense could easily overwhelm any existing offense systems. As soccer 2D teams are improving their algorithms and applying new methods, the other teams will have to change and improve their own methods in turn not to allow the rivals to be adopted against their style of playing in the game.

Mersad 2005's defense power was due to its perfect blocking skill, but marking skill was not much powerful; because in those years Mersad could handle any offensive attacks only using its blocking skill -an assumption that doesn't work for today's RoboCup soccer 2D teams. So we found it very effective to enhance Mersad's marking algorithms in our team.

3.1 Marking

In order to achieve an efficient marking system, we first needed to have a good pattern of collaborative marking algorithm. A good typical marking algorithm has some obvious features. As the first feature, assigning the marking points, the players involved in marking process should be able to map different opponent players supposed to be marked, onto them in a good manner to satisfy two factors, evenly distribution and passing the minimum total distance to get the marking point.

The second feature comes in when the assigning process is done. It means as the second feature, knowing which opponent to mark and current situation of the environment and the position of the ball and etc., a complete marking algorithm should be able to physically mark the calculated best opponent player in an efficient and smooth way. We call this part of the marking skill "physical marking".

In previous methods of simple marking in Mersad05 and Mersad04, players in order to know whether to do the marking or not and which opponent player to mark, were following a simple and imperfect algorithm in which first the player was counting all opponents in the confines of two horizontal and two vertical borders. Among all these opponent players, the best choice was the one with the smallest distance from the home point and also from the current position of the agent. As a result of this algorithm that can be understood at the first look the teammate players do not care no two players should mark one opponent player at the same time. This problem leads us to design algorithms to respect evenly distribution of opponent players' mapping.

Fig. 2. LEAKIN'DROPS marking system in defense

3.1.1 Assigning the marking points

In order for the groups of teammate players to mark the opponents in an evenly distributed way and also achieving the minimum total distance to get the marking points every teammate player should be able to have an image of which teammate player is currently going to mark which opponent player. For this purpose, there are logically two ways, distributed and centralized task assigning. Each of these choices has its own advantages and disadvantages.

From the hardness of the implementation point of view, centralized approach has some fatal problems, but its ability to have a unit choral decision for all teammates led us to pick "centralized" choice. What we mean by the term "centralized" in this case needs some elaborations. In centralized mark assigning, the coach having a perfect model of the world employs algorithms to extract the formation of the opponent and the area in which each of the opponents is frequently supposed to be.

After passing reasonable cycles of the game, the coach having the information of the previous cycles especially the positions of the opponents in each cycle and using some simple averaging algorithms extracts the necessary information about the opponent's formation. Then having the current formation of the opponent and teammate team, the coach is to assign each opponent position to one teammate. When one teammate is assigned to one opponent, it means the assigned teammate is the best one to mark the corresponding opponent.

The main factor contributing to determine which state of player matching is better is simply towards minimizing the total distance of each teammate's home position to its corresponding opponent's calculated home position. To achieve this, we have used graph mapping algorithms. After all these stages are done, the coach informs each teammate of its corresponding opponent to mark. But this information is not actually a strict command to be followed in any cases; In other words, it's an advice rather than a command. Teammates' receiving these advises in 300 cycles intervals, decide which opponent to mark using a mixture of this information and the algorithm they used to employ before this centralized system was provided.

3.1.2 Physical Marking

After the first stage of the marking algorithm is done, the agent knows which opponent he should mark. If the agent superficially tries to mark the target opponent, regardless of the situation of the game or the position of the ball, it won't work efficiently.

After we looked through the most efficient existing marking systems, Brainstormers' in particular, we found that the strong point making their marking skill perfect is not actually their ability to determine which opponent to mark, but their expertness to physically mark the target opponent.

For an instance of the importance of physical marking skill, we can consider the situation in which the agent is one of the two wing defenders and the opponent offenders have the possession of the ball in the middle of the field, on the back of the teammates' defense line. In this situation, it's likely for wing defenders to mark the opponent wing offenders. If it's not done in a fine manner, the likely strong passing system of the opponent using fine though passes for wing offenders can break the offside line and due to the defenders' lack of information about the exact cycle the ball is passed through breaking the offside line, in comparison to that of offenders, the defenders will be left behind by the offenders and will not be able to block the offenders and the defense line will be easily broken. This difficulty is due to the improper body angle of the defenders disabling them to run to block once the pass is sent by the opponents towards the wing offenders.

Briefly speaking, we have a precisely designed module in our marking algorithm that is intended to make out two things about the physical situation of the agent. The first thing to be made out is how, at which speed, in which direction, and in what relative position from the target opponent the marking action should be done. The second thing is to determine the optimum body angle for the agent when very close to the target marking point.

4 Conclusions and Future Work

In this paper we have quickly addressed some improvements in our soccer simulation team LEAKIN'DROPS 2009. For future directions, we are interested in studying reinforcement-learning techniques and applying fuzzy control & expert control to our team strategy. Our current team strategy is mostly depending on the expert rules in each divided area. So, how to fuzzicate partitions and regulate rules using theory of expert control is our future work.

References

1. Boorghany, A., Rokooey, M., Salehe, M., Vosoughpour, M.: Mersad 2004 Team Description. In: RoboCup 2004 Symposium and Competitions
2. Boorghany, A., Rokooey, M., Salehe, M., Vosoughpour, M., Jalali Nasab, D., Haradj, S.: Mersad 2005 Team Description. In: RoboCup2005 Symposium and competitions

3. R. de Boer and J. Kok. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. Master's thesis, University of Amsterdam, The Netherlands, Feb. 2002.
4. M. Riedmiller, T. Gabel: Brainstormers 2D Team Description 2008: RoboCup 2008 Simulation 2D League Champion.
5. A. K. Agogino, K. Turner.Team formation in partially observable multi-agent systems.