

Team Description for Robocup Soccer Simulation 3D

Robogulf Simulation Team 2006

Moein Alinaghian, Seyed Salim Tabatabaei, Jawad Arif Al-Shaikh
[Http://robocup.linux-dubai.com](http://robocup.linux-dubai.com) - Robogulf (at) Gmail.com
United Arabic Emirates

1. Introduction

In this paper you can read about the ideas we had for designing intelligent agents for a multi-agent system, and about the techniques and algorithms which we used.

Also this paper shows some details about our work, our aim, and even the football strategy that we used.

2. Basic Strategy

Our base strategy in the game is anti-attacking. We defend with full power in order to avoid receiving any goal from opponents, and of course we don't risk, especially when ball is near our goal.

Then as soon as we get opportunity to attack, we start attack with forwards and midfielders, hardly !

3. Type of Agents

There are four basic type of agents:

- Golie
- Defender
- Midfielder
- Attacker

Each agent has its own specific property and character, and by our design, they will use these strategies:

1. **Golie:** to avoid wasting battery, the golie starts processing and moving when the ball is closer than a specific distance, also it tries to keep its position stable and comes to a suitable place even after being thrown by the ball. The idea that golie uses to avoid

getting goal is to check the angle between the opponent kicker and the ball, then the goalie moves in front of the ball, also for decreasing the error, it uses the velocity vector of the ball after the ball is kicked. In the cases that there is no own defenders between ball and goal, then the goalie starts curve motion to the ball in order to decrease the probability of receiving goal.

2. **Midfielders:** these agents try to detect what is the status of the game, if we are attacking, they try to participate in attacking, if we are defending, they also participate in defending and the ball is in the middle of the field, they try to pass it to forwards. Their territory is dynamic, but they always try to shape like a triangle and move forward or backward or mark attackers of opponent.
3. **Forwards:** these agents start the game and their territory is more dynamic than midfielders in order to spread in opponent's field to be available for passing. They usually don't approach own goal too much and they do defend in middle of the field by trying to kick the ball in opponent direction. If they reach a specific distance from the opponent goal, they synchronize them
4. **Defenders:** these agents are somehow different from others, their aim is to undo an attack and try to interrupt the opponent passes. Their territory is dynamic but also limited to some specifications, which means if there is no need of them in the middle of the field, they stay in their positions nearby own goal. They try to have a zig-zag shape to be able to do their task faster and easier (interrupting passes and also marking the opponents).

4. Algorithms

4.1. Dribble Algorithm

When there is an opportunity to dribble with no interrupt from the opponents, the player starts dribbling towards the opponent goal.

Depending on the position of the player, the player chooses a flag and synchronizes its direction with it (with a small error) and kick the ball with a relatively small angle and power in order to shift the ball forward.

There are several conditions which will cause dribbling to switch to passing. One of them is when an opponent is approaching to the ball, depending on his velocity and his distance the agent decides either to continue dribbling or pass the ball to the best case possible and these conditions changes depending on the battery and position of the ball in the field, also it will learn what to do depending on the action of the nearest opponent player after number of cycles.

4.2. Passing Algorithm

The passing algorithm is divided into three other algorithms: finding the best case for passing, synchronizing the direction with that best case, approaching and kicking the ball.

4.3. Finding the best case to pass

There are some points to be considered which have different weights and priorities. To accomplish the task as fast and as good as possible, there is a combination of fuzzy logic and neural networks. Some of the points that affect the agent decision are as follows:

1. Difference in angle between agent and ball, and agent to the case of pass.
2. Distance of the case to pass from self.
3. Distance of the nearest opponent to the case of pass.
4. Distance of the case to pass to the opponent goal.

...

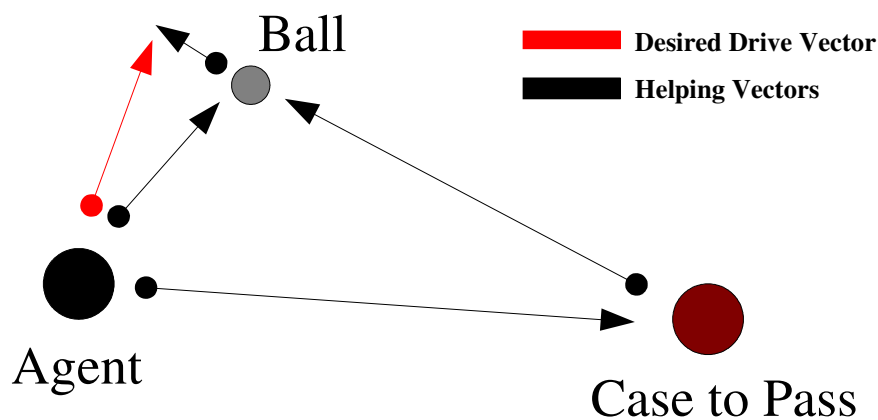
4.4. Synchronizing Algorithm

There are two synchronizing algorithms for different situations during the game (e.g. depending on the ball speed):

- Vector Based Synchronization
- Angular Based Synchronization

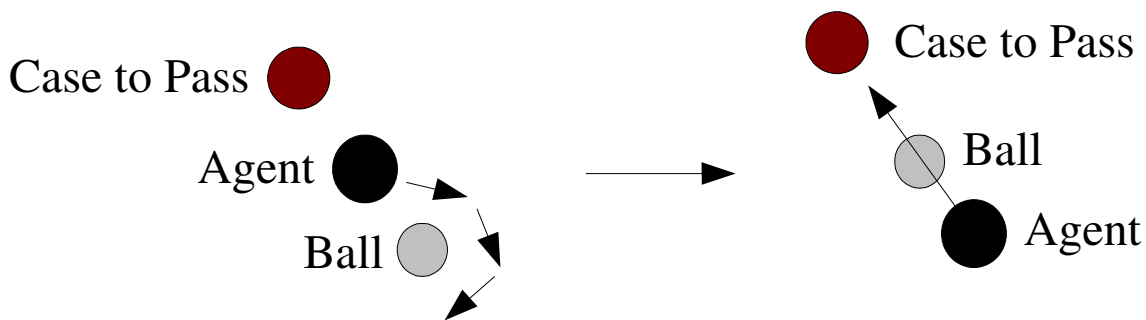
4.4.1. Vector Based Synchronization

In this algorithm depending on the ball speed, player speed, case to pass speed, difference in angle and the distance of the agent to the ball, there is a drive vector and also a power calculated and used to drive to the desired location with a relatively small distance to the ball.



4.4.2. Angular Based Synchronization

In this algorithm, the agent calculates and finds a drive line vector to move circular around an object (e.g. ball), until the difference in the angle becomes small enough. This algorithm finds the agent angle by a tangent function, then calculate the drive line vector by using some mathematical functions, depending on sinesoidal function. It is used when we want to pass specially when ball has a little speed, also it is used to defend against the opponent agents not to let them kick the ball towards our goal.



5. AI Parser

Our Agents in every step, depending on the situation, need to do some calculation for every another agents. For example calculate the distance of another agents from the ball, calculate the distance of two agents, find the nearest agent to an another specific agent, etc. if we want to do these works for all the agents, we must do these calculations for 21 agents in every step, and it takes big deal of process and time, so it will decrease our speed. because of this reason, we made the parser intelligent.

At the first, the parser parse everything in two array that one of them is for the opponent team and the another one is for our team.

After a while of playing, the parser detect the agents that are important for us, and we need to do calculations for them. so it will save their uniform number, and in next steps it will parse these special agents in another arrays. These array elements numbers vary from 2 to 6 depend on the type and position of our agent.

This technique will increase our speed and even decrease the processes approximately to half in each step.

6. Future

We are willing to improve the team that we have by adding a learning algorithm which learns from the monitor logfiles and also add multithreading in order to waste less time occasionally (e.g kicking takes 10 cycles which can be spent for thinking in parallel).

The other plan that we are having is to have another team for developing the 3D Simulation Server itself and we are gaining much experience and getting familiar with the problems and bugs of the server and we have some ideas to solve them. therefore we are going to implement those ideas while developing the server itself.

7. About us

Our team is the first Robocup team from UAE and contains three computer engineering students in middle years of study who are interested in Programming, AI and Robotics. We had some another robotics experience, but it's the first time that we participate in Robocup.

Moein Alinaghian
Seyed Salim Tabatabaei
Jawad Arif Al-Shaikh

8. References

1. Introduction to Algorithms (2nd Edition), by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
2. Artificial Intelligence: A Modern Approach (2nd Edition), by Stuart J. Russell, Peter Norvig.
3. 3D Math Primer for Graphics and Game Development, by Fletcher Dunn.