HERMES
Soccer 2D Simulation Team
Description Paper 2015

Seyed Emad Mohamadi [1], Amirabbas Pashaee [1], Hooman Shahrokhi [2], Alimohammad Foroutannezhad [3]

{emadolsky , aaph.1377 , hooman.shaah , alimohammad1995}@gmail.com

1 Department of Computer Science and Information Technology,
Allameh Helli High School, Tehran, Iran

2 Sharif University of Technology, Tehran, Iran

3 Electrical and Computer Engineering Faculty, University of Tehran, Tehran, Iran

Abstract. This paper describes the important facts of Hermes 2D Soccer Simulation Team, such as some of our ideas, important actions, current situation and our future plans. We tried to introduce our new works and ideas and explain them as well. We now explain our activities and researches.

1 Introduction

HERMES 2D simulation team is a sequent to Mersad, HelliBASH and Eskilas teams by Allameh Helli High School. Allameh Helli High School is competing in RoboCup Soccer 2D competitions since 2003.

It is the second year that HERMES is the primary team of Allameh Helli High school. In the first year we nearly completed all basic defense and offense actions so then we could start working on plans. This year we spent most of our time working on defensive and offensive plans for scoring goals as much as possible and do the opposite as less as we can.

Furthermore, we modified some of our old actions and also defined some new actions which would advance the result of basic actions for better plan making and would raise the quality of game play when the ball is ours and when it is not.

2 Basic Actions

Our basic structure of team is not quite changed from last year until now. But still there were some changes in the actions like Pass, SRP, Dribble, Mark, etc. Some of them were modified like mark and Pass but some actions were completely rewritten for better performance in the team such as SRP. Most of our current basic actions are described in HERMES 2014 TDP and ESKILAS 2010 TDP.

3 Opponents' Block Pattern Recognition

3.1 Defining Problem

One of the main reasons for losing ball is opponents' intercepting before us which in some situations is not possible for the agent to predict opponents' intercepting (with FastIC implement) and get correct data. These situations are those which in the cycle that we are calculating FastIC, opponent moves to a good position so intercepting would be possible for him. It is obvious that calculating this move is not possible for us because we almost have no idea about opponents' decision making in every cycle.
However, from our observation in these situations, we found that most of the times that the problem occurs, it is the BlockerOpponent who moves and gets the ball. So we thought that in case we know where BlockerOpponent will be in next cycle (even approximately), it would lower the probability of losing the ball to BlockerOpponent. Afterwards we started to think how it can be possible for us to predict BlockerOpponent's next move. We knew that blocking pattern is something different in every team so we deducted that predicting such pattern needs blocking data from the same team. It was quite obvious that a single player agent cannot be a very good observer because:

1. His view is not always set to blocker.
2. His seeData is not accurate enough in most of times.
3. Not all of our agents would be able to gather enough information for predicting the action.

Coach did not have the problems mentioned above because he could see everywhere accurately and then he could say the data that everyone could hear and use.
If the coach is able to tell our agents where the BlockerOpponent intends to go, it has two benefits for our decision making. First, we would not pass the ball to opponent because of his move in his block so the ball will stay ours. Second, it might help us to generate new passes and dribbles which we couldn't generate before because of the body angle or the position of the BlockerOpponent in the deciding cycle.
Now the question is how to predict opponent's block.

3.2 Learning Opponents' block action with limited data

An important point in finding opponents' next move is that we did not see every single state so our data is limited for predicting the action. For that, we thought it might be a good idea to learn the general blocking pattern with the limited information we had. We checked couple of algorithms to see what the results will be.
Let r be the distance between the ball ,which is own by our team-mates and relative, and θ be the BlockerOpponent's relative angle from our agent. Our intention is to find a formula which is a function of these two arguments and gives us the correct angle (α) that is the opponent's block angle. (At first, we quantize r and θ so learning will be possible. accuracy(r) = 1.5 units , accuracy(θ) = 20 degrees)
Here are the algorithms we checked for this idea:

1. Raw Genetic Algorithm for getting a general formula for opponents' block action

First we defined a simple function for our genetic evaluating of all of the (r, θ) states we had: $f(r, \theta) = \mathrm{A} + \mathrm{B}\theta + \mathrm{C}r + D\,{}^{1}\!/_{r}$. After completing the idea and the code, the result we gathered from the algorithm was not as promising as we expected. The average of difference between the first data (the correct data that the coach gathers in the game time) and the average of calculated $f(r, \theta)$, was in order of 65 degrees. This number was larger than the standard we needed for predicting opponent's block. So we concluded that genetic algorithm with this formula and these assumptions will not give us a very good result. So we decided to check other ideas instead of raw Genetic algorithm for predicting BlockerOpponent's action.

2. Completing the (r, θ, α) table with first data

(a) Algorithm for filling up the table

After gathering data from different matches we saw that there is a relation between one cell and its neighbor cells in the (r, θ) table. Thus we began to think about a way to predict the value of one cell with the help of values in other cells. We needed a formula that would return the value of the opponent move angle (α) by looking to the values in neighbor cells. We understood that near neighbor cells has more effects on the angle we are looking for. So we concluded that we need an evaluating function which is function of distance between cells in the table. After checking different formulas for this intention, we chose this formula: $val(r, \theta, \alpha) = \sum k^{dist((r,\theta,\alpha),(r_i,\theta_i,\alpha_i))}$.

In this formula k is a factor which might be different in different matches. At first we defined k equal to 0.5.

Now for every quantized α, we calculate $val\alpha$. then we find the maximum of these calculated variables. The angle (α), which makes the maximum value, is the angle of the unknown cell.

(b) Improving learning data by finding k factor by Genetic Algorithm

As we mentioned above, k is factor which makes the value of the different α angles and with different kinds of blocking methods this factor is different as well. Therefore defining a static value for this variable is not a perfect idea because if the number is not suitable for even one kind of block, our final data for that block will be useless. As a result we decided to use Genetic Algorithm as a tool for finding k factor. And the results were quite promising and there was a great improvement on the results after applying this algorithm. (For instance in Table 2 the values are calculated by $k = 0.952$)

r θ	3	4.5	6	7.5
-140	30.0	30.0	40.0	30.0
-120	60.0	40.0	20.0	40.0
-100	80.0	50.0	-10.0	70.0
-80	70.0	70.0	110.0	50.0
-60	120.0	110.0	100.0	100.0

r θ	3	4.5	6	7.5
-140	30.0	40.0	40.0	30.0
-120	60.0	60.0	20.0	40.0
-100	50.0	50.0	50.0	70.0
-80	100.0	70.0	100.0	70.0
-60	120.0	100.0	100.0	100.0

Table 1 & 2. Predicting values of cells(gathered from one match against YuShan2014)
1. White cells are first table data.
2. Orange cells in first table are correct data (got from logfile).
3. Green cells in second table are the predicted values by first table data.

(c) Checking results of the algorithm and evaluate its usefulness

Unlike our Raw Genetic Algorithm idea, this idea worked pretty well and after running the code against different teams and thorough calculating, we concluded that our table completing algorithm is accurate enough for our needs in team. The average difference between first table data and calculated data (with about %35 of first table data _ that means we kept %35 of gathered data from a match and then tried to learn the other %65 with those data) is a number in range of 20.0 (2 α-unit after quantization). This accuracy is good enough to have an approximate prediction about opponent's next move which will help the agents to have a better action than before.

4 Advanced FastIC

FastIC (fast intercept calculator) is an implement used to calculate all of the players intercept as fast as possible. FastIC is completely described in Eskilas2010 TDP and HERMES2014 TDP.

Almost every ball owning action has a relation with FastIC. For example, SRP and pass actions, both need FastIC to calculate reach cycles and points to see if they are capable of intercepting the ball in a suitable time or not. Therefore this implement is the most used implement in our team. So improving this implement will improve almost every ball owning action we do and also some other actions such as blocking.

Like every other calculations in soccer2D matches, FastIC has some errors, too. Most of these errors are because of ball and player random. Thus in case we could calculate random of intercepting, then we would be able to avoid these errors to happen.

4.1 Basic calculations

The goal of Advanced FastIC is to calculate the probability of reaching the ball for a player before the others. In the other word, we want to see how much is the probability of reaching the ball for one player when no one else would be able to reach the ball in that time period. First of all, we want to calculate the probability of reaching the ball in t cycles. Function f will give us the answer:

$$f(x,t) = 0.5 + \frac{x}{rt}$$

$$x \rightarrow distance\ from\ ball\ in\ the\ t - th\ cycle$$

$$r \rightarrow (BallRand * BallSpeedMax) + (PlayerRand * PlayerSpeedMax)$$

But we also know that the probability of an object in the circle with the radios of r is not uniform. That means the probability of being in the center of the circle is more than probability of being in the edges. Because of that, we need another function for bigger cycle difference. For that, we used central limit theorem and we presumed that the function is a normal distribution function (we used numeral approximation for reaching the function). And finally, we reached to this function for $t > 5$:

$$f(x,t) = 0.5 * (1 + \mathrm{erf}(\frac{x}{\sqrt{2}\sigma}))$$

And also we presume that: $\sigma = {}^{rt}/_{3}$

But in the end, what we wanted was a function that did not calculate probabilities with time argument because we wanted to know the probability of reaching the ball generally and not for a specific cycle.

4.2 Final probability calculating

For calculating final probability of reaching the ball, we should have a function which represents the probability without time as an argument in it.

From probability axioms, we know that probability of reaching the ball for player i is P_i :

$$s(x,t) = \prod_i 1 - f_i(x,t) \qquad P_i = \sum_t \left(\prod_c^t s(x_c, c) * \frac{1-f_i(x_t,t)}{f_i(x_t,t)}\right)$$

So now we have a formula which returns the probability of reaching the ball before the others for player i. After finding the formula we started to check the results of it.

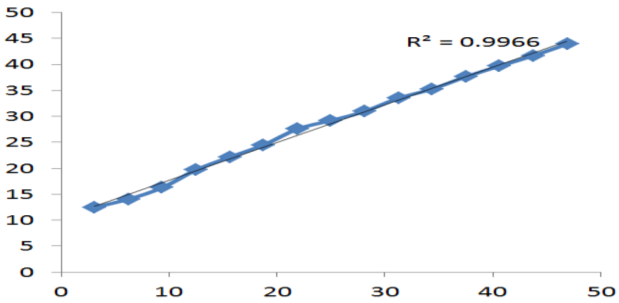

Fig. 1. Plot of pass results and FastIC calculated probabilities
x axis shows “probability ranges*100 - 50”
y axis shows “$\frac{passes\ received}{\text{all passes}}$„

5 Future Plans

5.1 Using Block Pattern Recognition online

Our final goal in using block pattern recognition is to complete the blocking table when the match is running and bit by bit telling the results to players. We believe that with further works on the idea and the algorithms, we will be able to fill up the table in the first half. So for the second half we would be able to predict opponents’ block and this would improve the result of all ball owning actions and bring it to a new higher level.

6 References

[1] Amini Zanjani M. Saharkhiz S. Bakhtiari M. Montazeri M. Vosoughpour M. Kaviani P.
Eskilas soccer 2D simulation team description paper, RoboCup 2010 Singapore
[2] Mohamadi E. Pashaee A. Kaviani P. Foroutannezhad A.
HERMES soccer 2D simulation team description paper, Robocup 2014 Brazil
[3] Hungarian Matching Algorithm. Kuhn H.W. The Hungarian method for the assignment problem, Naval research logistics (NRL)
[4] Abramovitz M. Handbook of mathematical functions
[5] Ross S.M. Introduction to Probability Models, Tenth Edition
[6] Brownlee J. Clever Algorithms: Nature-Inspired Programming recipe