Enigma 2014 Team Description Paper

Hadi Amini Velashani [1](IUST), {Nima Yazdanmehr[2], Alireza Bayat Makou[3], Hesam Rajaei[4]} (Salam High-School of Sadeghie)

Abstract. This Description-paper includes explanations of skills, algorithms, strategies and formations that we have used in our team. Basically this paper is about algorithms of direct pass, through pass, pass strategy, shoot strategy, defensive system and players' roles. This paper will explain about our new debugging program based on neural network algorithms and also our AI (Artificial Intelligence) algorithms and strategies which are used in our team, Enigma. This team is based on agent2d 3.1.1 and all used algorithms are implemented on this base.

1- Introduction

The team started working on 2011. Our first try was not successful and we could not get qualified for IranOpen2012 Competitions. Then we changed our base and started working on agent2d3.1.1. In 2013 we participated in IranOpen2013, SharifCup2013 and we could achieve the 6th place. Also we won the 1st place in SalamCup2014 Junior Competitions. Our focus is on developing skills and actions more than other parts, in this way we can get better results without using complicated strategies. Team name changed in every stage as below.

High Five (IranOpen2012), **D.A.I** (SharifCup2012), **Juggernaut** (IranOpen2013, 2014), **Enigma** (Robocup2014)

2- Formations

There are 2 different parts for formation deciding and each player's actual and suitable home position finding in our team. First part is dynamic graphs generated by fedit2 that we divided them by 2 types, offensive and defensive situations. Main property of these graphs is having optimized relocating and stamina usage for every player in case of switching situations. We considered defensive and offensive moves and skills as second part of our formation decision.

[1] hadiaminiv@gmail.com
[2] nima7r@gmail.com
[3] alireza.shady@yahoo.com
[4] hesamr76@yahoo.com

3- Actions

3-1- Pass

Passing skill is known as the main action for each player in every situation. To obtain the most accurate and suitable pass, we decided to develop several pass methods with different algorithms, in this way we can evaluate all methods and choose the best pass. Besides since we have several skills, we can transmit the ball in dangerous situations to our teammates instead of clearing the ball. Several algorithms development reasons are listed below.

1. Best response in every situation.
2. Less pass failures and ball losing.
3. Being able to execute pass skill fast to confuse opponents defense line.
4. Performing better in Clear-Ball situations and dangerous areas.

3-1-1- Direct Pass

There are 4 different Direct Pass algorithms in our team. The reason for plurality of algorithms is attaining factors such as being uninterrupted, quick, accurate, confident and proper. We have tried to design these algorithms convergent and the main theory is that having a specific action for each parameter would make deciding faster and easier, in this way we can give weights to parameters instead of actions. Every method is described below.

1st method. In first method, accuracy is our highest prioritized parameter. Velocity is in the second priority.
2nd method. This method is a normal configured pass and we spot equal importance for every factor.
3rd method. Used only for forward players and especially for offensive situations.
4th method. Absolutely rapid and quickness has the highest importance.

Figure 1 *Direct Passes Parameters Coverage*

3-1-2- Through Pass

To improve through pass skill in our team, we added communicating between players to the general pass algorithm[5]. In this version (Enigma_rc14) we have divided the whole skill by 2 parts. First part includes normal algorithms (includes static and dynamic algorithms) and second part is based on neural-network algorithms. Algorithms are described separately below.

In this part we have developed 3 different methods, each for specific situations (recognized by applying algorithms), and according to self and team mates positions, decision making class, will decide about executing the best method.

Whole skill plus its decision are generated by our self-developed program[6] based on A.I. algorithms. Main theory of this action is declaring good and bad positions for through pass (for kickable player) in certain situations, to gather enough data to training the network.

[5] Described in Direct Pass section

[6] Described in specific section

3-2- Dribble

To have better technical dribbles, we decided to develop vertical and horizontal (forward and backward) dribbles. These algorithms and dribble methods are designed in a way that can be used in almost every situation. Also to get better results, normal dribbles (not limited to position and role) added to our dribble actions-set. There are two different types of decision making. Both parts are explained below.

To select the best action for each player, according to player's status, static condition implementation considered as the simplest way of decision making. With optimizing the weight of every action in every situation, by using genetic algorithm[7], static decision will be improved. There is another decision class, which is used to make technical dribbles.

A sample of our first dribble decision section is shown below.

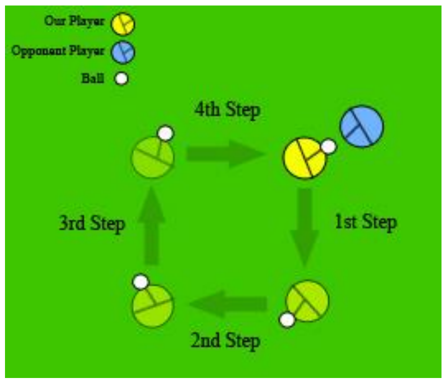

Figure 2 *Technical dribble in offensive situations to save the ball*

Forward: fw, Backward: bw, Vertical: ve

Dribble_Sample = **{fw, ve(contrary to nearest opponent), bw, ve(opposite of the previous vertical move), fw}**

4- A.I. System (Team Trainer and Debugger)

In this part artificial intelligence algorithms are used. The most important uses of them are described below.

4-1- Team Training

Team uses parameters in several aspects like generating actions, scoring actions, decision making, these parameters give a good flexibility on team's play style. The trainer uses these parameters to change the team's play style and it also benefits the software's monitor to analyze and score the team. During the matches the trainer counts the number of cycles that the team owns the ball, succeeded and failed actions, goals and etc. so it can calculate a score for the team. We use this score as the fitness of team in GA. For doing these changes a mix of Genetic Algorithm and Manual Arranging is used. Because the evaluation is time consuming it's better to modify some parameters manually.

4-2- Team Debug

We developed a program to show player internal parameters and thoughts so we can easily find the mistaken action and behavior. It's some kind of logging but it is very easier and more exact. This program facilitates development of the team. We will expand it more and more.

---

[7] Explained in Team Training Part

4-3- Player Training

One of the most challenging domains of 2D Soccer Simulation is composite actions and actions that depend on either strategies or techniques. In this part each player should behave adequate based on position, strategy and other players' potentials. We realized that human observers can decide more efficient and more accurate on the situations that both strategy and technic involved so we added a feature to the program that a human observer can supervise team decisions and suggest better actions.

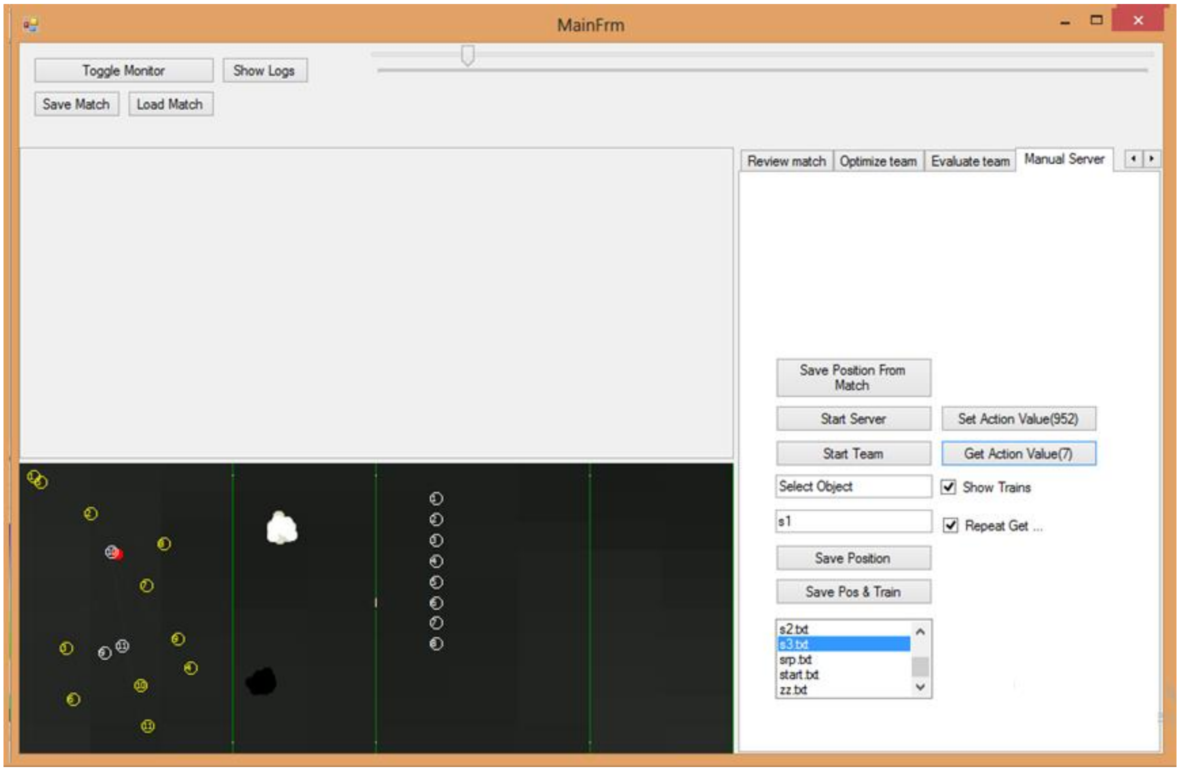

Figure 3 *Player Training Section*

4-4- Evaluating

We faced a problem in these two years of working on team, and it was that we were unable to understanding that is team improving or not? Because when you add a skill to team, after testing you see that team is not changing or it's getting weaker. We found out that our conclusions can be wrong and it's possible that when you say this skill is not working properly, skill is running completely right. Then we add another part to our program and it was evaluating. In this part we run random matches between different versions of our team and we define some opponents to compete with them.

4-5- Reviewing

This part is for team's learning part. In this part you can tell to program the mistake of team. For example you are watching a match and you see a mistake pass or you think that kickable player can do a better action but the agent is unable to understand. Then you can add a critique and specify the wrong action and the action that you want to be the replacement.

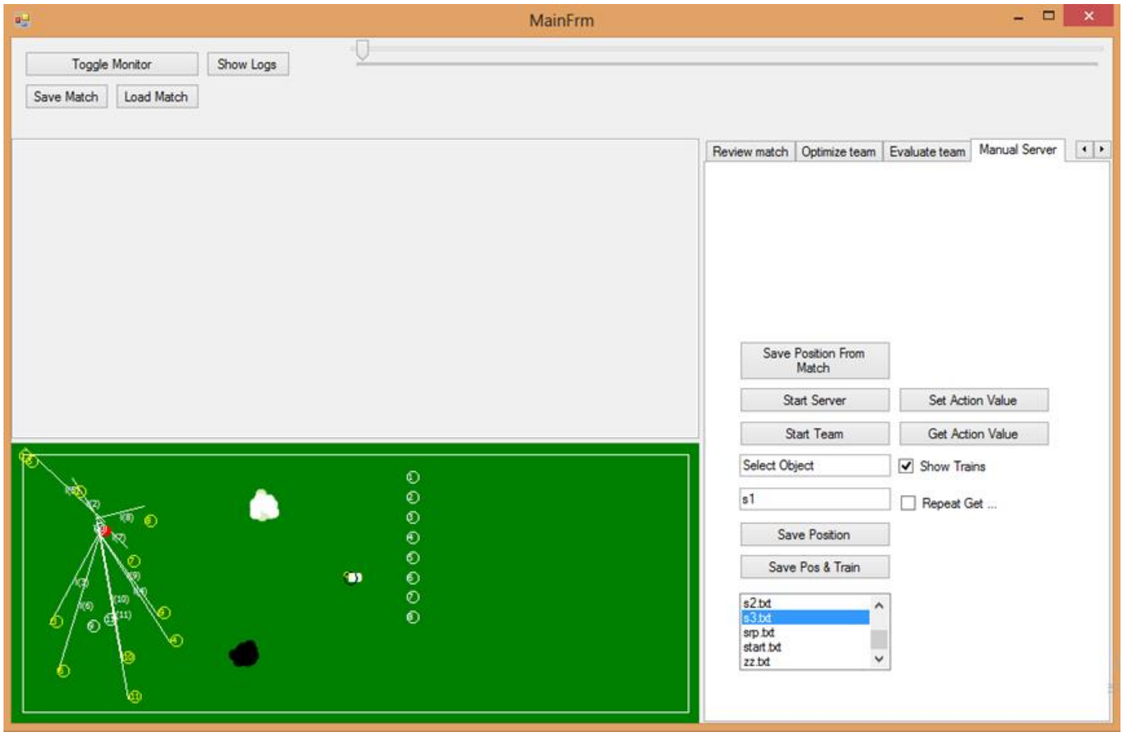

Figure 4 *Explained Program's Screenshot*

5- Future Plans

Further we are going to work on prediction algorithms and then they will be added to our skills and decision making parts. There are many prediction theories; based on our research we found 2 useful algorithms. Both algorithms are described below.

5-1- State Prediction

An agent can be defined as an autonomous entity in an environment with the capacity of taking its own actions in order to achieve a goal. Also, multi-agent systems take a set of agents in order to cooperate and achieve a common goal that cannot be completed without the help of other agents.

Knowing how the opponent is going to behave in a competitive environment such as soccer is a great way to increase a team's effectiveness by being able to anticipate the rival's actions.

In order to create a useful opponent model for Strategy Patterns Prediction Model (SPPM), it is necessary to take into account previous experiences. In this case, records and logs of past games were used. To read data from them easier we convert these binary files to xml files. The original XML file obtained from an RCG files describes not only the server parameters, but also each of the ball's position, each player's position and actions across the game. We use this data as patterns. A pattern is defined as the route that the ball follows inside the game while one team keeps it. Then we need to make a search tree and to have a search tree [Figure 5], first we should make a knowledge base from one or more patterns that we built. This algorithm uses data from all players and decides and predicts based on received data. To sync all data we need a leader in team. Every 15 cycle team chooses one player as leader, itself.

Once all the data is received, it needs to be cleaned and consolidated into a single pattern so that it can be used into the search tree. The leader agent determines the ball's position by calculating an average from the positions received and then a single zone can be assigned. A similar process is followed for each of the opponent players reported and duplicates are eliminated. Then the information is merged in a single list in a format that can be used by the search tree. Once the result is taken from the search tree, we get all the possible patterns' IDs with the best matches. The criteria to decide those matches is based in the distance between the actual field status and the one contained in the pattern, giving more importance to strategies that involve more players. This is called the similarity measure. The similarity measure formula is the following:

$$s = \left(\left(\sqrt{(X_{br} - X_{bp})^2 + (Y_{br} - Y_{bp})^2}\right) \times \alpha\right)$$
$$+ \sum_{i=1}^{n} \frac{\sqrt{(X_{pri} - X_{ppi})^2 + (Y_{pri} - Y_{ppi})^2}}{n}$$

Where, Xbr and Ybr are the X and Y positions of the ball in the actual status and Xbp and Ybp are the X and Y positions of the ball inside the pattern. Xpri and Ypri are the X and Y position of the players in the actual status while the Xppi and Yppi indicate the position of the players inside the pattern analyzed. The symbol α represents a constant value used to give the ball's position more importance in the comparison.

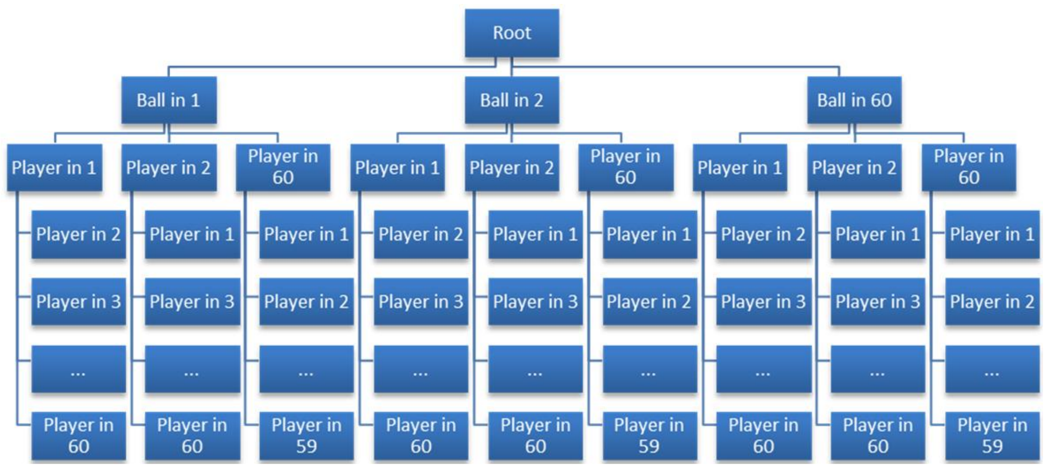

***Figure 5** State Prediction Decision Tree*

6- References:

[1] Yazdanmehr, Nima; 2013; **"Juggernaut 2D Soccer Simulation Team Description Paper"**.

[2] Haupt, Randy L; Haupt, Sue Ellen; 2004, **"Practical Genetic Algorithms"**.

[3] Perez , Aram Baruh Gonzalez; Uresti, Jorge Adolfo Ramirez; 2014, **"Strategy Patterns Prediciton Model".** Available at http://thescipub.com/pdf/10.3844/jcssp.2014.73.84

[4] Kyrylov, Vadim; Razikov, Serguei; 2010; **"Optimal Offensive Player Positioning in the Simulated Robotic Soccer".** Available at http://www.intechopen.com/download/pdf/9360

[5] Akiyama, Hidehisa; Shimora, Hiroki; Nakashima, Tomoharu; Narimoto Yosuke; Yamashita, Katsuhiro; 2012, **"Helios2012 Team Description Paper".** Available at http://www.socsim.robocup.org/files/2D/tdp/RoboCup2012/TDP_HELIOS2012.pdf

[6] Gupta, Madan M; 2003,**"Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory"**