

Austras2D Soccer 2D Simulation Team

Description Paper 2021

Sania Ezzati¹, Mozhan Hamavandi¹, Ghazal Sayyadfar¹, and Shayli Shahrad¹

Farzanegan 7 Middle School, South GolAfshan St., West Eyvanak Blvd., Tehran
Province, Tehran, Iran

<http://farzanegan7.sch.ir>

qzl.sayyadfar@gmail.com saniaezzati1384@gmail.com
mozhanhomavandi@gmail.com Shayli.shahrad@gmail.com

Abstract. This paper describes Austras2D latest team activities and strategies for improving the starter-stack base, published by Nader Zare[1]. We will describe our offence strategies such as decision tree, our shoot, pass and unmarking, as well as defense strategies such as ball block, Intersection-marking and shoot block. At the conclusion, the result of changes we made will be compared with the base version.

Keywords: RoboCup · soccer simulation 2D · Decision tree · shoot block · mark · Intersection mark · unmark

1 Introduction

Austras2D is a soccer 2D simulation team from Farzanegan7 high school. All members of the team are pupils. Austras2D team have participated in many competitions before. Continuing last year's work, the team has implemented its offensive decisioning ideas as a tree data structure and used unmarking processes in order to correct players' positioning. In defense, we focused on anticipating opposing team actions to make a move as quickly as possible. We have also classified our defense mechanism into marking, ballblock and shootblock, depending on the ball's position in the pitch. The code we used as our base is a simplified version of Agent2D 3.1.1 base code from Helios team, which is designed for starter teams. The code comes along with a starter version of rcsc library[2], which has omitted some basic actions such as Body Pass, Body dribble, Body clearball, Body advanceball, Body shoot, and the folders related to the formation and neural network. Also the chain action folder and strategy file have been removed from the base code[1].

2 Related Works

In this manual experience, we read some related efforts in other teams' TDPs. Cyrus2D team searched about predicting player's behavior and developed it from 2012. This behavior includes block, mark, and defensive.[3][4] Helios 2018

worked on “share the knowledge”. Knowledge sharing planned to have a prediction model of the player who has the ball by other teammates.[5] Razi team introduced pass, shoot, dribbling, and marking algorithms.[6] Shiraz team worked on multiple tasks at once, such as coach analysis, making the best decision of goalie movement, automatic game, etc[7]. some implementations and ideas are inspired by these TDPs.

3 Offense

3.1 Decision tree

The purpose of this task is to select the best action in offense mode each cycle. To get this we implemented a decision tree to get better predictions. To implement this, the team uses a tree search algorithm in order to select between three main acts (shoot, pass, dribble).

In other words, the team uses a structure of the main three actions (pass, shoot, dribble) and then estimates what would happen if each of these actions is done. (Fig.2.) For instance, in the opponent penalty area, if both pass and shoot are possible simultaneously, the shooting act is almost more logical. The factors used for scoring and choosing between three main actions (shoot-pass-dribble) are:



Fig. 1. for player number 9, kicking is much more sensible than passing to player 11

1-If shoot is possible, it has the highest priority(Fig.1.).

2-If pass and dribble are both possible in a cycle, the player sorts them due to their scores calculated by considering the distances between their targets and the opponent’s goal (the shorter the distance is, the higher the score gets).

3-If pass and dribble are both possible in a cycle, the player adds another factor that is the number of the opponents surrounded by our teammates (the lower the number of opponents are, the higher the score is because the chance of losing the ball will be much lower).

4- Another key fact to remember is that our main concern in every cycle is getting the chance to shoot, on the other hand, sometimes shoot is not possible

but if the players do the right actions, in the next cycles the chance of shooting will be higher. so, that action will get the highest score.

In the end, by assessing, the best action is selected.

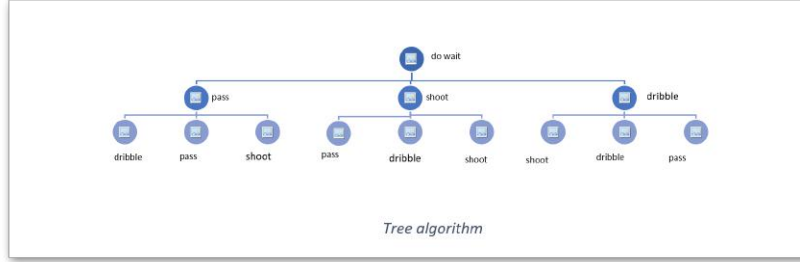


Fig. 2. Decision tree

3.2 Pass

Pass is one of the main three actions with a significant effect on efficiency. The aim of this task is to find the best pass receiver. Factors to find the best receiver:

1-At first the player validates the teammates by Checking whether any opponent is on the way between the player and its teammates.

2-Then the player picks the valid teammate who is nearest to the opponent's goal so the chance of shooting will be raised.

Another thing to remember is that it's not always possible for the player to kick the ball to the best target. In such cases, the player passes the ball to a teammate who has a better situation to do the best action. On the other hand, for the defense players (the teammates with 2,3,4,5 numbers) is not always possible to pass the ball to a point near the opponent's goal. so, there is a limit for them to make sure that opponents can't catch the ball(Fig.3.).



Fig. 3. for player number 6, teammate number 9 is the best choice because no one is between them to catch the ball and 9 is near opponent's goal

3.3 Shoot

To improve the shooting act, the player creates 24 points at the opponent's goal, from left to right. After that, the player estimates the success rate of each shoot candidate by drawing a line from the ball to the point. The success rate is evaluated by the number of opponents near or next to it. In the end, the player picks the point with the highest success rate as the target.

3.4 Unmarking

To improve the shooting act, the player creates 24 points at the opponent's goal, from left to right. After that, the player estimates the success rate of each shoot candidate by drawing a line from the ball to the point. The success rate is evaluated by the number of opponents near or next to it. In the end, the player picks the point with the highest success rate as the target.

We have two methods for "Unmark":

- 1.attack mode: when the ball is in the opponent's penalty area
- 2.simple mode: when the mode is not attack

These two methods may sound the same but there are many differences in players' performance. The main idea to find a new position and maintain the formation's integrity at the same time is generating the points around the player. Afterward, we omit the points that don't end in our team's possession. For the final step, we calculate a value for each point and pick the highest valued point as our action target.



Fig. 4. Generating points around the player and choosing best one (Red point).

4 Defense

4.1 Ballblock

In this task, we are going to get the ball from the opponent's ball holder. To do this, the team needs to specify the best player for ball blocking. After that, the player generates the holder's probable dribble path and goes to a point that is reachable before the holder. So, the player can perform the tackle action and possess the ball.

4.2 Marking

The purpose of marking is to stop opponents from taking control of the ball, usually while they are passing it. To achieve this, we tried two different strategies. Both are used once the defense mode is on.

Regular mark: The idea behind this algorithm is based on the instant behavior of real soccer players. It chooses two opponents that are most likely to catch the ball; one is the closest to the kickable opponent, the other is closest to the line between our goal and the ball (Fig.5.). During our updates, we also consider other factors such as the x parameter of opponents or the last opponent who received the ball. Finally, our two nearest teammates cover the paths between the kickable opponent and the two selected ones. Target opponents get updated every three cycles.



Fig. 5. marker players blocking the nearest opponent (blue) and the opponent with the highest probability of receiving the ball (red)

Intersection mark: By using this algorithm, we find the points (rather than opponent players) that are most likely for the ball to pass. We use this method when the ball is in the center area of the pitch. First, we identify as many opponents as possible in the center area. then we draw virtual lines between pairs of them and store the points of intersection into a vector (Fig.6.). Next, we sort the vector elements based on their proximity to the inertia point of the ball and pick the three best targets to place our marker players into. By using this method we can mark several paths from opponents to the ball holder and increase the chance of catching a ball.

4.3 Shootblock

shoot block aims to directly reduce the scored goals by decreasing the opponent's successful shoots. This idea is divided into two parts for better arrangement: Decision and Act.

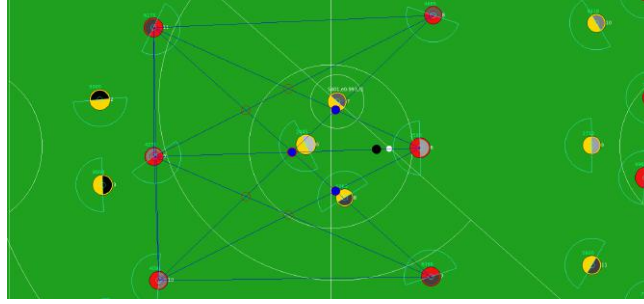


Fig. 6. three players marking the nearest points to the ball inertia point

Decision: In this case, the team decides which point has more probability of scoring a goal and which player is appropriate to block the shoot. so, we generate a few points at the goal, then we evaluate the points based on the distance from the goalkeeper, self and the ball. After choosing the most remarkable point, the player defines a line from the selected point at the goal to the opponent who owns the ball. Then, the player selects a new point as the closest point to itself on the defined line, which is created by verticalizing a line from self to the defined line (Fig.7.).

Act: In this part, the conditions for performing a defensive move are measured and after checking important factors such as the situation of the players and the position of the ball, the decision function is executed and returns the final point, then it ends by catching the ball (Fig.8.).



Fig. 7. opponent (number 9) shoots the ball to the predicted point.



Fig. 8. the blocker (number 2) catches the ball and blocks opponent's shoot (number 9).

5 Future Works

The team is currently working on several algorithms. Some of these algorithms are as followed:

- 1.Improving the penalties
- 2.Finding a way to keep the team's integrity while offense and defense strategies are implemented.
- 3.Improving goalie's decisions while opponents are in an offensive situation.
- 4.Improving goalie's decisions when the ball has been caught.
- 5.modifying the defense mode while the opponent's density in our field is high.
- 6.Making the environment more deterministic by control algorithms

6 Conclusion

This paper describes the previous efforts and current research by using implemented algorithms of Austras2D. The team's main goal is to increase the number of successful kicks and passes, the efficiency of the goalie and defenders using new algorithms and structures to improve the coordination between defense players and offensive players.

Table 1. The results in 200 game against base code by Autoteset[8]

Austras2D	Base Agent	.
432	31	Goals
2.13	0.15	Average Goal
171	4	Won games (25 draw)

References

1. Nader Zare, Starter Agent2D, 2017, GitHub repository, <https://github.com/naderzare/StarterAgent2D>
2. Hidehisa Akiyama, Tomoharu Nakashima, HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation, In Sven Behnke, Manuela Veloso, Arnaud Visser, and Rong Xiong editors, RoboCup2013: Robot World XVII, Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin, 2014.
3. Sarvmeili Keshavarzi, Ashkan Mohsen Sadeghipour, Nader Zare, Mohammad Abolnejad, Arad Firouzkoohi, Reza Aghayari, Sina Elahimanesh, Amin Nikanjam, Amin Akhgari, "CYRUS2D simulation Team Description paper 2018" for the RoboCup Soccer 2D Simulation 2018.
4. Khayami Rauf, Nader Zare, Maryam Karimi, Payman Mahor, Ardavan Afshar, Mohammad Sadegh Najafi, Mahsa Asadi, Fatemeh Tekrar, Ehsan Asali, and Ashkan Keshavarzi. "CYRUS2D simulation Team Description paper 2014" for the RoboCup Soccer 2D Simulation 2014.
5. Tomoharu Nakashima, Hidehisa Akiyama, Yudai Suzuki, An Ohori, and Takuya Fukushima."Helios simulation Team Description paper 2018 " for the RoboCup Soccer 2D Simulation 2018.
6. MortezaNoohpisheh, Maryam Shekariz, Sepehr Barzegar, Danial Borzoo, Ali kariminia, "Razi Team Description 2018" for the RoboCup Soccer 2D Simulation 2018.
7. Ehsan Asali, Mojtaba Valipour, Ardavan Afshar, Omid Asali, MohammadReza Katebzadeh, Saeed Tafazol, Alireza Moravej, Suhair Salehi1, Hosain Karami1, Mahsa Mohammadi, "Shiraz Team Description 2016" for the RoboCup Soccer 2D Simulation 2016.
8. wrighteagle2d team, Autotest2D, 2016, GitHub repository, <https://github.com/wrighteagle2d/autotest2d>