R3CESBU Soccer Simulation 2D Team Description Paper 2023

Mohammad Hesam Nasiri[1], Seyed Hassan Majid Zonouzi[1], Arya Parvizi[1],
Seyed Mostafa Atyabi[1], Seyedeh Rana Rokni[1], Sanaz Moosapour[1], Mahan Veisi[1],
Kiarah Kowsari[1], and Farbod Saghfi[1]

[1] Shahid Beheshti University, Tehran, Tehran, The Islamic Republic of Iran
Mohammadh.nasiri@mail.sbu.ac.ir, s.majidzonouzi@mail.sbu.ac.ir,
se.atyabi@mail.sbu.ac.ir, a.parvizi@mail.sbu.ac.ir,
s.rokni@mail.sbu.ac.ir, s.moosapour@mail.sbu.ac.ir,
mah.veisi@mail.sbu.ac.ir, k.kowsari@mail.sbu.ac.ir,
fa.saghafi@mail.sbu.ac.ir

Abstract. This paper contains some explanations about the algorithms implemented by R3CESBU team members. We will study algorithms used for the goalie, blocking, shooting, and marking. The base code used by R3CESBU is agent-2d. This year, the team will participate in its first competition in the 2D soccer simulation RoboCup 2023 league.

Keywords: RoboCup, soccer simulation, chain actions

1 Introduction

We will be using agent-2d base code [1] to develop our ideas and strategies for the 2D soccer simulation league. We explored various sections to improve our agents' behaviors and looked for the weak spots that we can fix with more emphasis on the defense. To be more specific, we studied and optimized the formation, blocking, shooting, tackling, goalkeeping, and defense. In some of the sections, we will be discussing how we developed ideas and applied various techniques to boost our agents' performance, consequently changing the outcome of the matches by a significant margin. In the following paragraphs, we will discuss in detail how we made these improvements and elaborate on the ideas, which our team proposed.

2 Goalkeeper

The goalie's performance in decision-making in the base code has poor quality. So, our first effort was to improve its positioning. By dividing the field into 5 parts (Figure 1), we create a simpler form to analyze the field. Since these sections have independent logic regarding how the goalie should act, it makes a good modular logic structure that handles the complex field in simpler terms. The logic and behavior of the goalie in each section are as follows; If the ball is in the first part (BA_Safe), the goalie doesn't move so as not to lose stamina. If the ball is in the second part (BA_DribbleBlock), the goalie position is the bisector between the ball and the opponent possessing the ball. In the third position (BA_DefMidField), the goalie stands in the bisector between the goal-mid and the nearest opponent. If the ball is in the fourth position (BA_CrossBlock), the goalie sticks to the nearest opponent. Finally, in the last state (BA_Danger), the goalie chases the ball and one of the center-back defenders follows to support the goalie according to the position of the ball.

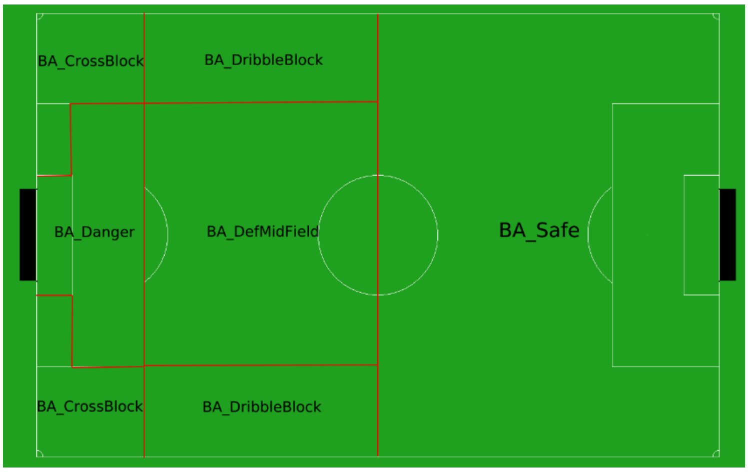

Fig. 1. Dividing pitch for goalie decision-making

3 Shoot

We have found that when we are on offense, the position of the kicker can greatly influence whether we can score. We noticed that in the primitive base code of Agent2D, the kicker prefers shooting the ball rather than passing to a teammate who has a potentially better position for scoring. To increase the likelihood of success, we introduced a threshold to change the previous policy and gain significantly better results. [2][3][4]

4 Tackle

In the base agent-2d code regarding tackles, the base code’s approach is to prevent agents from making tackles when the agent has a card, so we modified this policy and made the game a bit aggressive, especially when counterattacking. In the base code, the agent doesn’t usually make tackles in the penalty area. To preserve the results when we are ahead, we modified the threshold of this strict policy only in the final moments of the game. Further, we try to find the best distances for tackling by using artificial intelligence algorithms.

5 Strategy

Since the base code strategy is very primitive, we designed a new strategy with a Delaunay Triangulation algorithm, and we used the Fedit2 tool[5] to make our new formation. To achieve this, R3CESBU changed the strategy pattern in Agent2d by designing a new strategy maker file and adding new formation configs. In addition, we implemented the first phase of dynamic positioning same as real soccer. For example, when we are on offense, two of our defender's roles change to the halfback or attacking winger to surge our attack strength. Not only this strategy can be used in the offense but we claim that it can also influence pivotal moments in the defense as well.

6 Basic Block

For implementation, we use a function that returns us teammates who can block the opponent player having the ball. For this purpose, first, we leave the goalie and those who have special conditions (e.g., they are frozen or have very little energy, etc.) and those who are far away from the ball. Then, for the output of the mentioned function, we also check the following two conditions:

- The cycles that reach the ball should be the least possible.
- Our player reaches the ball earlier than the opponent.

According to the above conditions, we choose our best teammate for the block.[6][7]

7 Dockerizing RCSS

One of the major issues which almost all teams encounter at the start of the 2D simulation league is its incompatibility across all platforms, dependencies, and prerequisite packages. After overcoming all of them only, to finally reached the point to run the first match game and the first step to begin the development of soccer simulation 2D agents. We dockerized the listed files below to bring a complete and compatible package of soccer simulation 2D tools which are runnable regardless of hardware and whether the prerequisites are installed. Not only is it platform-independent, but it’s also showing better performance and less capacity consumption. The links to the docker images are also available.

- RCSSServer [1]
- RCSSMonitor [2]
- Soccerwindow2 [3]

[1] https://hub.docker.com/r/itsfarbod/rcssserver

[2] https://hub.docker.com/r/itsfarbod/rcssmonitor

[3] https://hub.docker.com/r/itsfarbod/soccerwindow2

- Fedit2 [4]

8 Calculate Opponent Danger Score

In the new agent code (i.e., R3CESBU), a function has been implemented which evaluates the danger level based on the opponent players in the surroundings of the agent (i.e., the possessor of the ball). This danger score depends on ten parameters, including the nearest opponent's magnitude and direction of velocity, its body angle, the number of cycles elapsed since the last time we received the information, and finally, a series of complex parameters based on the teammates' role and position on the pitch. Each of these parameters has a different level of importance and is assigned different weight values accordingly. Combining these parameters results in a scalar value representing the danger score.

9 Future Work

9.1 Improve positioning and Making Teamwork Actions:

Since the initial code for chain action is flawed, we have made modifications to it and, as previously mentioned, by predicting the level of danger around the teammate who owns the ball, we produce safer kicks to the ball. Further, to improve the overall teamwork of the agents we will improve the positioning of each agent while our team has the ball. Therefore, we gain potentially better situations that will likely lead to a goal, while preserving our defense strength, blocking probable counterattacks and potential risks. Our candidate solution highly revolves around predicting the position of the ball in the next cycles.

9.2 Data mining Logs:

After each game, two log files are generated in RCG and RCL formats, which we converted into readable and analyzable datasets. With these, we train our artificial intelligence models to improve the goalkeeper's movements and forward's kicks. By examining the heat map of the position of the successful kicks across various teams, we suggest that we can find key areas in the field, where the agents can make their decisive kicks. Furthermore, we plan to tune the previously mentioned parameters and weights using statistical learning techniques.[8][9]

[4] https://hub.docker.com/r/itsfarbod/fedit2

References

1. Akiyama, H., Nakashima, T: Helios base: An open-source package for the RoboCup soccer 2d simulation. In Robot Soccer World Cup 2013 Jun 24 (pp. 528-535). Springer, Berlin, Heidelberg.
2. Zekai Cheng, Feng Zhang, Bolun Guang and Liting Wang ”YuShan2021 Team Description Paper for RoboCup2021,”In RoboCup 2021 Symposium and Competitions, Worldwide. 2021.
3. Zekai Cheng, Ningyu Xie, Feng Zhang, Bolun Guang, Qin Zhang, Chen Sun, Xuecheng Wang and Liting Wang,” YuShan2019 Team Description Paper for RoboCup2019,” In RoboCup 2019 Symposium and Competitions, worldwide. 2019
4. ZeKai-Cheng, NingYu-Xie, Chen-Sun, ChengLong-Tan, KeWei-Zhang, LiTing-Wang, GenShen-Zhang, XingXing-She, Xiao-Zheng,” YuShan2018 Team Description Paper for RoboCup2021,” In RoboCup 2018 Symposium and Competitions, worldwide. 2018.
5. Hidehisa Akiyama, Tomoharu Nakashima, HELIOS Base: An Open-Source Package for the RoboCup Soccer 2D Simulation, In Sven Behnke, Manuela Veloso, Arnoud Visser, and Rong Xiong editors, RoboCup2013: Robot World XVII, Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin, 2014.
6. Akhondi, F., Esmaelifar, S., Esmaelifar, S., Rokni, S. R., Rajabi, A, Hasanpour, G.: Hades2D soccer2D simulation Team Description paper. In: RoboCup 2021 Symposium and Competitions: Team Description Papers. Worldwide (2021).
7. Sana Esmaelifar, Seyedeh Raha Rokni, Sama Esmaelifar, Fatemeh Akhondi, Adrina Rajabi and Rozhina Pourmoghaddam G.: Hades2D soccer2D simulation Team Description paper. In: RoboCup 2022 Symposium and Competitions: Team Description Papers. Worldwide (2022).
8. Namira Soccer 2D Simulation Team Description Paper 2018. [PDF](https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2018/Namira_SS2D_RC2018_TDP.pdf)
9. Persian Gulf Soccer 2D Simulation Team Description Paper 2017. In The 21th annual RoboCup International Symposium, Japan, Nagoya. [PDF](https://www.robocup2017.org/file/symposium/soccer_sim_2D/TDP_PersianGulf.pdf)