ThunderLeague: Team Description Paper 2022

Caio de Souza Barbosa Costa, Ricardo Tamay Honda, André de Andrade Jabur, Antonio Lago Araújo Seixas, Gabriel Cosme Barbosa, Giovanni Cabral Morales, Heloísa Apulcro do Nascimento, Jonas Gomes de Moraes, and Paulo Guilherme Pinheiro Pereira

University of São Paulo,
São Paulo, São Paulo, Brazil
{caio.costa, ricardo.honda, andre.jabur, antonio.seixas, gabriel.cosme, giovanni.morales, heloisa.apulcro, jonas.gomes, paulo.guilherme}@thunderatz.org

Abstract. This paper details the implementations added to the project in 2021 and early 2022. The new functionalities include a formation evaluator to dynamically change the team's formation, a setplay blocking behavior, a risky passes algorithm to improve through passes, and changes in the evaluation function. After these implementations, the performance of the team was tested against the base team Gliders v2.6 and an improvement was verified when compared to last year's version of this project.

This document also describes some of the ideas that are currently under development, which include the use of reinforcement learning in defensive positioning, improvements in the pass action generation algorithm, the implementation of a new player-to-player communication architecture, as well as a neural network for the risky passes decision-making process.

1 Introduction

ThundeRatz is the robotics team of the University of São Paulo. It was founded in 2001 and, since then, has been participating in many robotics competitions worldwide. Recently, the team is increasingly engaging in autonomous robotics and has developed a Very Small Size Soccer team and a RoboCup 2D Soccer Simulation team. The latter is called ThunderLeague.

ThunderLeague was created in 2018, using the source code of Agent2D [1,2] and, without any modifications, has participated in its first competition: Winter Challenge 2018, the biggest robotics competition in Latin America. In the next year, with big changes in formations, the team was able to win the first prize in that same competition and, since then, has become one of the main Soccer 2D teams in Brazil. Recently, the team has started engaging in international competitions and has already participated in RoboCup 2021, IranOpen 2021, and RoboCup Asia-Pacific 2021.

In the last year, ThunderLeague focused on improving its formations and decision-making, aiming to enhance our offensive and defensive capabilities. Different approaches were used to reach these goals, developing many new features in the process.

In this paper, we describe not only the changes made in the last year, but also some of our current work in progress, such as communication between our players through messages and the usage of Reinforcement Learning to improve the agents' behavior.

2 Related Works

A considerable part of the work presented in this paper is inspired by previous works of other 2D soccer simulation teams.

In 2016, team LeftEagle [3] developed a man-to-man marking technique based on the closest-pair matching algorithm, which tries to pair each opponent to its nearest teammate agent. Therefore, the opponent ball owner has fewer options to pass the ball, constraining offensive options. Those implementations were the base to the current ThunderLeague blocking behavior.

Two years later, in 2018, team Gliders released the source code of a new base team called Gliders2D [4], whose ideas were the foundation of the early development of our team in 2021. Gliders introduced the idea of risky passes by creating a variable that determines the risk of a given pass, allowing the team to take more or less risk depending on the situation. This year, small adjustments were made to that idea and it was merged into ThunderLeague.

During the years of 2019 and 2020, teams Cyrus [5] and RoboCIn [6] implemented a reinforcement learning algorithm to improve players' defensive actions. To do that, both used actor critic, a deep reinforcement learning algorithm, to train neural networks to choose how players will behave given a certain state. Based on these teams' ideas, we started to think of an implementation to choose actions based on reinforcement learning.

Lastly, in 2020, Hidehisa Akiyama et. al. [7] described a way of modeling a soccer 2D simulation team's evaluation function using supervised learning. This inspired us to start using neural networks in our team, with our first implementation aiming to improve the generation of through passes.

3 Implemented Functionalities

3.1 Formations evaluator

In order to have more offensive and defensive options to suit certain situations, other formations were created, which are 442 and 343.

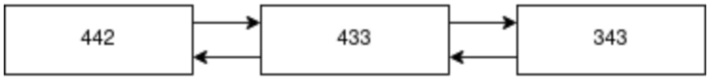

Fig. 1. Available formations

A formation is dynamically chosen among the available ones, based on some parameters, such as mean ball position, absolute ball possession, and relative ball possession. The last one is calculated using an auxiliary parameter (γ), which decreases older ball possessions' weight and helps get a more suitable view of the recent game state. To do this, two auxiliary variables are used to adjust current and last used cycles:

$$adjusted\_last\_cycle = \frac{\gamma^{last\_cycle-1} - 1}{\gamma - 1} \tag{1}$$

$$adjusted\_cycle = \frac{\gamma^{current\_cycle-1} - 1}{\gamma - 1} \tag{2}$$

And then, the relative ball possession can be calculated using (2) and (3) as follows (in case our team has the ball):

$$our\_poss = \frac{our\_poss \cdot adjusted\_last\_cycle \cdot \gamma + 1}{adjusted\_cycle}$$

$$opp\_poss = \frac{opp\_poss \cdot adjusted\_last\_cycle \cdot \gamma}{adjusted\_cycle}$$

Those data help analyze the match's current situation. The coach then evaluates it and transmits a freeform message to the players telling the current formation to use so that they can be positioned correctly.

3.2 Blocking

A new behavior implemented in the team was the setplay blocking. This behavior, inspired by LeftEagle's 2016 marking ideas [3], only occurs during a setplay and makes our players try to block the enemy players. In figure 2, we can see the yellow players blocking the opponents, limiting the blue players' pass possibilities.

Some conditions must be met for a blocking action to happen. First of all, the player will block an opponent if the distance between them is less than a defined distance called "block radius". After this condition is met, the player will check if there's a teammate closer to the opponent (if so, he will stop the action and let the other player block the opponent). It's important to note that the block radius is not a constant number across player roles:

- Midfielders have the biggest block radius since they can act defensively or offensively and a big change in their positions will not influence negatively the player's performance;
- Defense players have the smaller block radius since they can accidentally break the team's defensive line if they leave their ideal position to block an opponent who is far from them;

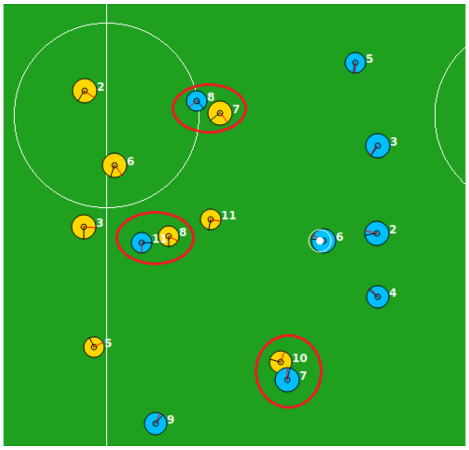

Fig. 2. Yellow players blocking blue players.

- Attackers have a medium block radius (that is bigger than the defense players' and smaller than midfielders') because they should be near their ideal position to be able to attack properly but don't need to follow a strict position like defenders.

During our tests, this blocking behavior has not only enhanced the team's defense but has also increased the team's scored goals, since the opponent team would lose the ball possession and counterattacks happened more often.

3.3 Risky passes

Among the changes to the offensive behavior of the team was the addition of a risk level, based on the Gliders2D [4] algorithm. The risk represents the number of additional cycles "granted" to teammates receiving a pass, in comparison to opponent players that can potentially intercept the pass. This means that even in situations when the opponent would take fewer cycles to get to the ball, the pass is still executed.

The amount of risk that is worth taking depends a lot on the opponent team. In some cases, the risk value is 0, meaning that the original pass behavior is restored. When it's possible to take a risk, other factors are analyzed, related to the proximity of the players to the offside line and the type of pass that is being executed.

3.4 Changes in evaluation

Aiming to improve the quantity and quality of the passes done by the team, two major changes were made in the evaluation function: the implementation of a negative potential field on each enemy and the improvement of the spatial component of the evaluation.

The first one considers the distance between the ball position, on every possible state, and the position of the enemies that the acting player is seeing. Using an expression similar to a negative electrical charge potential field, each enemy negatively influences the viability of the pass according to its distance to the state ball. After testing this change, it was noticed that, besides the increase of passes made, the rating of successful passes increased significantly.

The second one focuses on changing the state viability according to the ball's position on the field. The idea was to increase its viability depending on how much it is closer to the enemy's goal. Therefore the players tend to do more offensive actions. Another change was to make the sidelines less viable to avoid passes that go off field. To visualize these modifications, figure 3 shows a heat map that indicates the state tendency on the field according only by the position.

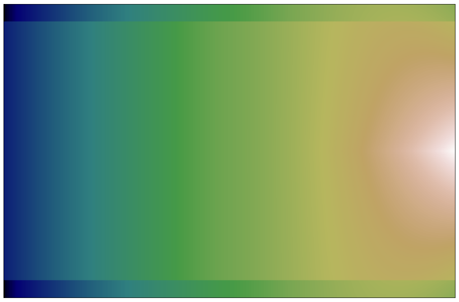

Fig. 3. Heat map indicating the state viability across the field, with blue representing less viability and red more viability

3.5 Results

After these changes, we tested the performance of the team against Gliders v2.6 [8]. The decision of using Gliders as the opponent during the test is because Agent2D (the opponent used in last year's paper) is a relatively simple team so some of the new implementations (especially the new action evaluation) don't improve the team's performance at all.

In table 1, a comparison between the last year's team and the current team is shown. It's interesting to note that ThunderLeague 2021 used to lose the majority of the games against Gliders since the last year's implementations were based on Gliders v1.5 [4], an inferior version. But now, with the addition of some new original changes, the team is able to surpass the opponent.

Team	Points for	Points against	Goals scored	Goals conceded	Goal diff.
ThunderLeague 2021	1.005	1.812	0.737	1.305	-0.568
ThunderLeague 2022	1.859	0.914	1.211	0.683	0.528

Table 1. Performance comparison after 1000 games versus Gliders v2.6

4 Implementations in development

4.1 Defensive positioning using reinforcement learning

Some changes are being developed in the decision-making of which defensive action players should follow. The idea is to use reinforcement learning algorithms instead of hard-coded conditions, which can help cover a wider set of game situations while also choosing the most effective action according to the game state, as shown in previous works done by teams CYRUS[5] and RobôCIn [6]. To do this, we are currently modeling the basic components of a reinforcement learning algorithm (such as the environment, the state, the actions, and the action samples), as well as choosing which algorithm is more suitable.

4.2 Improves on the pass action generation

In order to improve the pass action generation, changes in the estimated pass receiver position are being developed. To do this, not only the player's inertia is being considered, but also its desired placement for a future ball position. This results in a weighted position function based on these factors, whose weights vary depending on ball position and type of pass (direct, leading, or through pass).

Therefore, the created action can be more precise about the field area where the receiver will be, increasing the range of possible promising passes that can be generated.

4.3 Communication

Another area that is being improved is the player communication. Currently, the team communication usually only happens when a player wants to pass the ball to another player so the receiver will not ignore it. By doing so, the two players work collectively to perform the pass action, improving its performance.

But there are many other situations where a collective behavior is desired, so a player-player communication architecture [9] is being developed. In this architecture, every player has a social role, which models the way it interacts with other players. Some roles have authority over others and that grants them the ability to send order messages to players whose roles are submissive to theirs. Those messages contain information about the goals a player must achieve and the set of goals assigned to a player defines its mission in the game. This way, the team works collectively since groups of players have different missions to be accomplished and will have to work together to do so.

4.4 Neural Network for risky passes

The through pass is one of the best ways to make the team's offense efficient. With that in mind, a neural network whose function is to determine whether a through pass will be effective or not is being added to the team. The neural network is still under development but the main idea is that the input data will contain the enemy defense players' position, the team's attackers position, and their respective stamina so that the player will be able to identify good situations for through passes.

The training dataset will contain the necessary input data from all the through passes that happened during thousands of ThunderLeague matches against different teams and the good through passes will be determined as the ones that led the ball to the opponent's goal area. After training using this dataset, the neural network will be able to predict if a through pass will fail or not and decide if it's worth to be performed.

5 Conclusions

This paper describes the implementations created during 2021 and early 2022. Some of them are already fully implemented and tested, while others are still under development. It's interesting to point out the team's performance against Gliders v2.6: by comparing this year's and last year's versions, it's evident that the new implementations, which are the formations evaluator, the blocking, the risky passes, and the changes in evaluation, have a great influence on the team's better performance, being even able to surpass Gliders v2.6.

While developing the new implementations previously mentioned in section 3, we hope that those changes can improve the team's performance even more.

References

1. Akiyama H., & Nakashima T. (2014). HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation. In: Behnke S., Veloso M., Visser A., Xiong R. (eds) RoboCup 2013: Robot World Cup XVII. RoboCup 2013. Lecture Notes in Computer Science, vol 8371. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44468-9
2. Akiyama H. (2012). agent2d-3.1.1 source code. https://osdn.net/projects/rctools/
3. Abdelhafiez, A., Ehab, A., Kamel, A., Breaka, H., Abdelrahman, H., Soliman, N., & Ali, O. (2016). LeftEagle Soccer 2D simulation Description Paper 2016. RoboCup 2016 Symposium and Competitions, Leipzig, Germany. http://2016.robocup.org/web/images/robocup_sim2d_tdp_lefteagle.pdf
4. Prokopenko, M., & Wang P. (2018). Gliders2d: Source Code Base for RoboCup 2D Soccer Simulation League. https://doi.org/10.48550/arXiv.1812.10202
5. Zare, N., Sarvmaili, M., Mehrabian, O., Nikanjam, A., Khasteh, S. H., Sayareh, A., Amini, O., Barahimi, B., Majidi, A., & Mostajeran, A. (2019) Cyrus 2D Simulation 2019 Team Description Paper. RoboCup 2019 Symposium and Competitions, Sydney, Australia. https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2019/CYRUS_SS2D_RC2019_TDP.pdf
6. Oliveira, C. S., Machado, M. G., Assis, M. A. P., Rodrigues, W. M., Araújo, T. S., Costa, V. M. M., Queiroz, L. G. O., Barros, E. N. S., Ren, T. I., & Neto, P. S. G. M. (2020). RobôCIn Team Description Paper 2020. RoboCup 2020 Symposium

and Competitions, Bordeaux, France. https://robocin.com.br/archives/2020_RoboCup_SSL_RoboCInTeamDescriptionPaper.pdf

7. Fukushima, T., Nakashima, T., & Akiyama, H. (2020). Evaluation-function modeling with multi-layered perceptron for RoboCup soccer 2D simulation. Artificial Life and Robotics, 1-6. https://doi.org/10.1007/s10015-020-00602-w
8. Prokopenko M., & Wang P. (2019). Fractals2019: Combinatorial Optimisation with Dynamic Constraint Annealing. In: Chalup S., Niemueller T., Suthakorn J., Williams MA. (eds) RoboCup 2019: Robot World Cup XXIII. RoboCup 2019. Lecture Notes in Computer Science, vol 11531. Springer, Cham. https://doi.org/10.1007/978-3-030-35699-6_50
9. Hübner, J. F., Sichman, J. S., & Boissier, O. (2002). A model for the structural, functional, and deontic specification of organizations in multiagent systems. Brazilian Symposium on Artificial Intelligence (SBIA 2002), Brazil. https://doi.org/10.1007/3-540-36127-8_12