

# RobôCIn Team Description Paper 2022

Cristiano Santos de Oliveira<sup>1</sup>, Mateus Gonçalves Machado<sup>1</sup>, Walber de Macedo Rodrigues<sup>1</sup>, Thiago da Silva Araújo<sup>1</sup>, Pedro Vitor Cunha<sup>1</sup>, Rafael dos Reis de Labio<sup>1</sup>, Felipe Nunes de Almeida Pereira<sup>1</sup>, Mateus Ferreira Borges Soares,<sup>1</sup>, Edna Natividade da Silva Barros<sup>1</sup>, Paulo Salgado Gomes de Mattos Neto<sup>1</sup>, and Tsang Ing Ren<sup>1</sup>

Universidade Federal de Pernambuco, Centro de Informática, Recife PE, Brazil  
robocin@cin.ufpe.br  
<https://www.robocin.com.br/>

**Abstract.** RobôCIn Soccer Simulation 2D team started in 2018 at the Universidade Federal de Pernambuco. Our first competition was at João Pessoa, Paraíba, Brazil, in Latin American Robotics Competition (LARC) 2018, where we obtained fourth place against teams from Latin America. In 2019 we participated for the first time at the RoboCup obtained ninth place, and we also participated at the Brazil RoboCup Open 2019 (LARC), where we obtained second place. In 2020 we participated at the Brazil RoboCup Open 2020 (LARC), where we obtained third place. In 2021 we participated at the Robocup obtained a seventh place, and we also participated at the Brazil Robocup Open 2021 (LARC), where we obtained first place. This paper describes the evolution of the approaches developed last year, the new improvements, and our research for the simulation 2D.

**Keywords:** Data Science · Data Analysis · Graphic User Interfaces · Web Development · Statistics · Classification · Formation · Adaptive Formations · Behaviour

## 1 Introduction

RobôCIn is a robotics research team from the Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), created in 2015 to participate in competitions and research subjects related to robotics. We are currently working in four categories: Soccer Simulation 2D (SS2D), Very Small Size (VSS) since 2015, Small Size since 2018, and Line Follow since 2019.

We based our code on agent2d 3.1.1 [1] at the first year since it is a well-structured base, and from it, we could develop our studies and approaches more quickly. To the new release, we integrated the gliders2d-v1.6 [6] to our code, since it demonstrates a better performance and it is an evolution of the agent2d-3.1.1; we integrated the MarlikBlock of the Marlik2011 [9] to our moving behavior and made some improvements to the function that controls the behavior selection. We also made hard-coded improvements to some game situations in order to make

our attack more aggressive, creating new attack behaviors and goalie behaviors formations, and improving the action evaluation algorithm.

Our log analyzer, revealed in the last TDP, has been refactored and split into a soccer analysis python library and a full-stack web application, both currently closed-source but soon to be shared with the soccer simulation 2d community. The soccer analysis library is fully-fledged and can be installed and imported as a python package on any python program. It provides types, operations, and other kinds of abstractions that help the user extract useful information from log files in the form of statistics and data insights in an organized and standardized manner.

We discussed the development of new resources focused on team formation throughout the year. We are looking to apply real football team formations in the 2D simulation environment and test them, looking for statistics to see if the training had a positive or negative impact. We are developing behaviors for game situations, using parameters such as elapsed time and team scores. The coach will identify those behaviors and forward to the team the changed behaviors.

## 2 Adaptive Formations

We believe that formations are essential regarding football strategies, and every formation has its purpose in a game [3]. Both in real-life and simulated football, we see teams changing formations and player's positions as the game occurs to adapt themselves to a current situation [4]; depending on the situation the team seeks to hold the ball, be more offensive to score a goal or be the most defensive as possible.

Our team was playing in a 4-3-3 formation for every game situation, which is a good formation when we pursue side attacks, and we are currently setting this formation to be our main one. However, if the game asks us to be more conservative or even hold the ball, it is not good as our players are wide open in the field.

Thus, we are currently developing and testing adaptable formations that better suit in-match situations. The coach analyzes the current game situation and changes the team strategy to find a better action plan based on the opponent's team pressure, score, and game time.

### 2.1 Change strategy based on the opponent's team pressure

We want our team to learn how to deal with the opponent's team strategy, to be able to overcome an attack from the opponent situation efficiently, as we want to implement more functions to our coach. We are implementing a strategy similar to Oxy's since their coach can adapt their defense line based on the opponent's attacking behavior [7]. As the coach has a noise-free vision of movable object [2], we keep an average value of where is our defensive line for the last period of cycles, and if it is close to our goal means we are under high opponent's pressure. Therefore, we might want to change our current formation to one more defensive or more offensive.

## 2.2 Change strategy based on the score and game time

We think the score and game time are walking together when analyzing if we should change to another formation or not. As the game goes by our players lose stamina, then depending on the score, we do not want them playing and positioning as at the beginning of the game. Therefore, if we reach an important amount of elapsed cycles and we are winning the game, we want the players to focus on the defense and the team to be more compact.

## 2.3 Rehearsed plays

We were looking to make our game style more offensive and with some rehearsed plays. To implement those news strategies, we started to look after formations from real soccer teams and created news formations for our team. Also, we tried a new rehearsed play using a semicircle formation for passing the ball and making the game play more dynamic.

# 3 Goalie behavior

The goalie is the player whose actions can have a bigger impact on a game. A mistake it makes can easily lead to an opponent's score. Because our goalie was still the same as the agent2d-3.1.1 [1], we implemented two essential features for a trustworthy goalie: identify a teammate's backpass and improve the clear ball skill.

## 3.1 Identifying backpass

When reviewing our games, we noticed that our goalie was committing many fouls when our players back passed. To adjust this mistake, we now check if the last player who touched the ball was the opponent or not; if it was, we can't catch; otherwise, we can.

## 3.2 Improvement on goalie's clear ball skill

We were delivering many balls to the opponent's players because our goalie was kicking the ball away randomly, so we tried to make our goalie smarter when cleaning the ball. To accomplish this behavior now, it is looking for a kick where the opponent can't reach or where our players can reach first than the opponent's players. In the future, we intend to choose the best possible kick.

# 4 Log Analyzer

In our past works[5] to better understand the behaviors and Key Points of Interest (KPI) of our team in the SIM2D environment, we created a Desktop GUI application that used single-purpose functions to collect information in the logs

generated from the server. We learned that our previous approach was limiting what we could extract from the logs due to the code complexity and unreliability, so we decided to enhance that application to widen its usage and benefits for the team.

Thus, our new approach of analyzing the information with a reusable, reliable, and maintainable code that can be quickly shared and used for the team members. Also, we improve data analysis using better-suited data visualization technologies such as React and Apache ECharts.

#### 4.1 Architecture

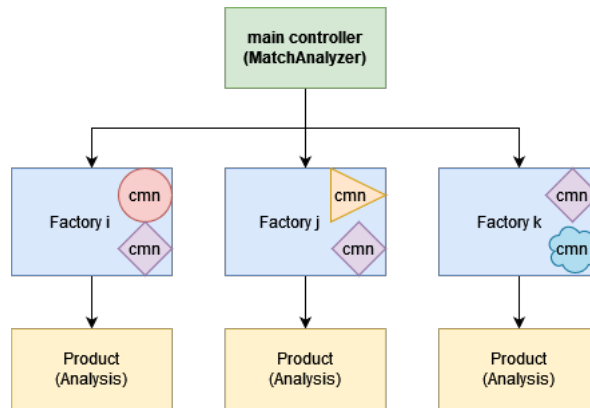


Fig. 1: SoccerAnalyzer factory design

We created a platform in the domain of data analysis and web development at the same time, and by doing so, we wanted two things: Correctly create analysis in a modular way, and display and interact with graphs and pictures generated from our analysis. Code that had different purposes was set apart, so the analysis handler became a package in Python called SoccerAnalyzer, and the GUI became the Analyzer Platform.

#### 4.2 Python Package

The SoccerAnalyzer was created in an object-oriented programming fashion with modules that generated analysis from a given log input, enabling sharing results across the entire application that can be reused elsewhere in the code.

We used the Factory design pattern to create classes that produced the desired output. A global controller manages the factories that have access to all the application information and can share data between other classes. All classes that generate analysis were designed to be agnostic of their content, meaning

that a class that produces an analysis X for the SIM2D category can also be applied to generate the same analysis X for Very Small Size Soccer or Small Size League category, for instance.

As the architecture is modular, the SoccerAnalyzer can reuse parts of its code in various modules. These parts are mostly from algorithms created to collect some specific information or calculations needed to obtain the final complete analysis, as shown in Figure 1

Currently, this is a private project used internally by the team members, which has proven its usefulness as it enabled scalation modularization and made our previous analysis easier to understand. Our objective is to open its development to the community in the short term.

### 4.3 Web Application

The web application we developed is intended to be a graphical user interface accessible from the browser, from which the user can interact with the soccer match analysis library and work with 3 main ideas:

1. Team Profiles: each team has its own profile with past victories and defeats, contact, prizes and place in competitions, and most importantly, a hub of general statistics about the team, such as win/lose rate, ball possession, and team formations as can be briefly seen in Fig.2

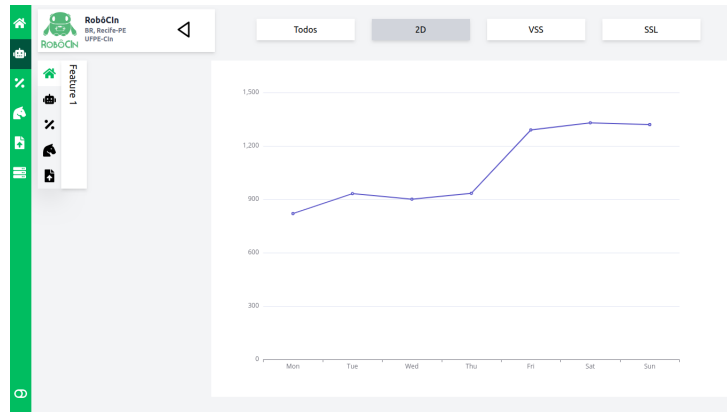


Fig. 2: Analyzer Team Profile Early Implementation

2. Whiteboard: instead of having all of the possible analysis laid out on a list or a more rigid way, we introduced an infinite whiteboard where the user can draw or pin, for example, images, analysis plots, text blocks, or a pdf. The user can invoke a floating panel and use command-line-like commands to interact with the system, or, if desirable, a graphical option can be used to search through possible analysis and actions.

3. Team Timeline: each team has a timeline composed of whiteboard sessions. Each whiteboard session is a full whiteboard created for a specific purpose (study the behavior of a specific team, analyze why a team has low ball possession, or justify the introduction of a particular feature on the code of a specific team) and has metadata about the date of the creation of the whiteboard and what category it is. These features enable the system to display a timeline with all of the insights that the user and its team had about its team or other teams.

## 5 Web Soccer Monitor (WSM)

Another tool we have been developing is the Web Soccer Monitor. It is a web tool that aims to be a novel, simple and yet powerful open-source way of visualizing robot soccer matches directly on the browser without any setup or installation, and it's available as a GitHub repository[8]. It is currently able of replaying past matches of the 2D Soccer Simulation League given a log file, and we plan to expand the support to other categories such as VSS and SSL, and also enable the system to show matches occurring in real-time running locally on the user's computer or on the cloud, similarly to a live-streaming service. Figure 2 shows the GUI in which the user can upload a log file, watch the match replay, change the current playing time, and tweak other viewing options such as playback speed and if the player view area should be shown.

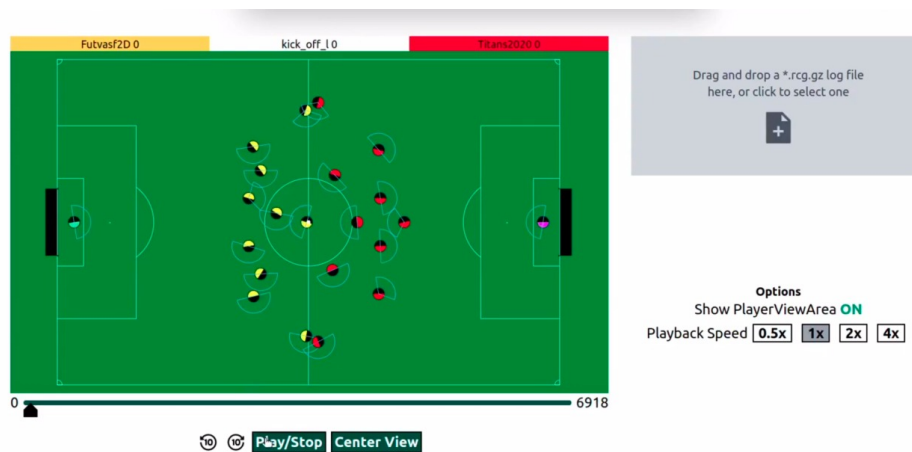


Fig. 3: Web Soccer Monitor GUI

## 6 Experiments and Results

We created a training setup to analyze our modifications, it contains four binaries that are referenced as:

1. **Control team:** RobôCIn team from RoboCup 2021
2. **RBCN2022:** same as Control team but with current development for the next Robo Cup.
3. **RBCN2019:** RobôCIn team from RoboCup 2019
4. In development team(**Adapted**): a derived team from RBCN2022 that implements the formation modifications.

### 6.1 Formation experiment

We carried tests using a branch (Adapted) with formations implemented based on real soccer teams and rehearsed formation plays. Adapted played against RBCN2022 and Control team to generate the statistics data studied.

Formation test: complete match			
VERSION	S.P.G	Balance	Games
RBCN2022	0.8616	196	506
Adapted	0.6272	91	502

Table 1: Shows the statistical results of both teams

However, the changes made in the formations were not as effective. This can be seen in Table 1. The statistics related to scoring and the balance of goals had no significant improvement to the Adapted and rehearsed plays. We think this is related to the plays not being predominantly executed during the games tested due to being overlapped by other behaviors.

Finally, from these results, it became possible to verify whether the modifications made to the formation would be attributed to the binary of RobôCIn 2022. The parameters used were: S.P.G: Score per game, Game balance, and the number of games. It became possible to perform a statistical analysis and define whether the results were positive or not.

### 6.2 Goalie behavior experiments

We produced an analysis to identify when our player passes the ball to our goalie and if the goalie catches the ball or not. Table 2 shows an experiment where we played against the control team, looking to minimize the number of wrong retreats per game (W.R.P.G), which was computed by the division of the total number of wrong retreats (T.W.R) divided by the total number of games.

Then table 3 shows the complete training session statistics of the version with the new features against our control version. So we can verify if there were some considerable decreases in our statistics.

We considered these results satisfactory as they fulfilled our expectations over the goalie back pass and had good general statistics against our older self.

Goalie retreat test			
VERSION	T.W.R	W.R.P.G	Games
RBCN2019	133	0.266	500
Adapted	58	0.113	510

Table 2: Shows the fouls statics before the change and after it

Control Team Vs Adapted Version			
Wins	Draws	Loses	Games
0.31	0.44	0.28	500

Table 3: Shows the complete statistics of a training session after the goalie's behaviors implementation

## References

1. Akiyama, H., Nakashima, T.: 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. pp. 528–535. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
2. Chen, M., Dorer, K., Foroughi, E., Heintz, F., Huang, Z., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Murray, J., Noda, I., Obst, O., Riley, P., Steffens, T., Wang, Y., Yin, X.: Robocup soccer server for soccer server version 7.07 and later: User manual (2003), <https://rcsoccersim.github.io/rcssserver-manual-20030211.pdf>
3. Clemente, F.M., Martins, F.M.L., Mendes, R.S., Figueiredo, A.J.: A systemic overview of football game: the principles behind the game. *Journal of Human Sport and Exercise* **9**(2), 656–667 (2014)
4. Nakabayashi, Y., Hanyu, D., Masuda, M.: NCL08 team description paper (2008), [https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2008/NCL\\_SS2D\\_RC2008\\_TDP.pdf](https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2008/NCL_SS2D_RC2008_TDP.pdf)
5. Pereira, F.N.A., Soares, M.F.B., Barros, E.N.S.: A data analysis graphical user interface for robocup 2d soccer simulation league pp. 1–6 (2020). <https://doi.org/10.1109/LARS/SBR/WRE51543.2020.9307104>
6. Prokopenko, M., Wang, P.: Gliders2d: Source code base for robocup 2d soccer simulation league (2018)
7. Sebastian Marian, Dorin Luca, B.S., Cotarlea, O.: Oxsy 2021 team description (2021), [https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2021/Oxxy\\_SS2D\\_RC2021\\_TDP.pdf](https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2021/Oxxy_SS2D_RC2021_TDP.pdf)
8. Soares, M.: Websoccermonitor (2021), <https://github.com/robocin/WebSoccerMonitor>
9. Tavafi, A., Nozari, N., Vatani, R., Yousefi, M.R., Rahmatinia, S., Pird-eyr, P.: MarliK 2011 Soccer 2D Simulation team description paper (2011), [http://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2011/MarliK\\_SS2D\\_RC2011\\_TDP.pdf](http://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2011/MarliK_SS2D_RC2011_TDP.pdf)