# RobôCIn Soccer Simulation 2D
# Team Description Paper for RoboCup 2025

Gabriel Souza, Heitor Souza, Jeferson Araújo, Maria L. Silva, Pedro V. Cunha, Rafael Labio, Tales Cunha, Walber M. Rodrigues, and Edna Barros

Centro de Informática, Universidade Federal de Pernambuco.
Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - Pernambuco, Brazil.
`robocin@cin.ufpe.br`
`https://robocin.com.br/`

**Abstract.** RobôCIn Soccer Simulation 2D team, based at the Universidade Federal de Pernambuco, was founded in 2018. In our debut competition at the Latin American Robotics Competition (LARC) in João Pessoa, Paraíba, Brazil, we secured fourth place against other Latin American teams. The following year, we competed in the RoboCup for the first time and achieved a ninth-place finish. In 2024, we placed in the 6th position on RoboCup and first place at LARC. In this paper, we present the work developed over the past year, especially the refinement of the agent's behaviors, changes made to the field evaluator and the improvement of our testing infrastructure.

**Keywords:** Soccer Simulation 2D · Swarm Optimization · Statistics · Behavior · Field Evaluator · Action Classification

## 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 Flying Robots Trial League since 2022.

RobôCIn used the well-structured base of agent2d 3.1.1, as presented in [2], as the basis for our code development in the first year. Our most recent agent release includes changes in field evaluator, goalie kick prediction, improvement of test infrastructure, an action classifier, and an action generation optimization.

## 2 Field Evaluator Optimization

The FieldEvaluator module plays a crucial role in analyzing possible game states, attributing rewards or penalties based on several criteria. One of the most significant aspects evaluated is the ball's position in the field. In previous work, we introduced a Potential Evaluation Function[3] parameterized by the ball's position, as presented in Equation 1.

$$
\begin{aligned}
&E_x = \lambda_x^{dist_x} \\
&E_y = \begin{cases} 2*^{-\lambda_y * dist_y}, & \text{if } x > 0 \\ ^{-\lambda_y * dist_y}, & \text{otherwise} \end{cases} \\
&Evaluation = E_x + E_y,
\end{aligned} \tag{1}
$$

Where $\lambda_{x,y}$ are the weights for each function and $dist_{x,y}$ are the distances on each axis to our goal center $(-52, 0)$. Initially, the weight values $\lambda_{x,y}$ were arbitrarily chosen, but in the past year we conducted a study to refine the definition of those weights.

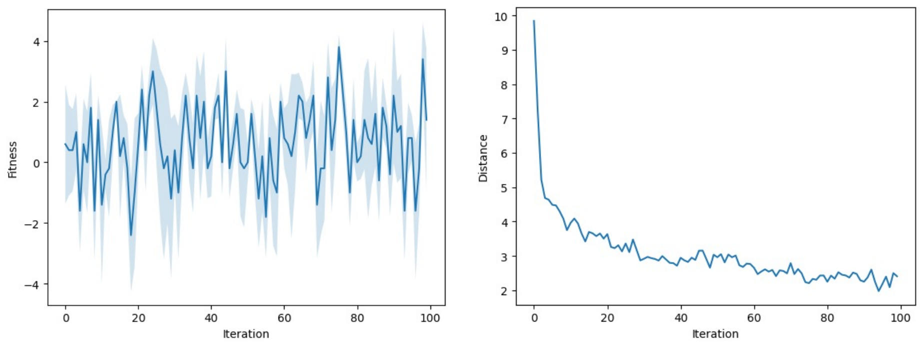
We decided to establish different potential fields for each player in the field, setting a different $\lambda_{x,y}$ for each of them. To obtain the 22 weights, we used Particle Swarm Optimization (PSO)[4]. For the execution of the algorithm, a fitness function and stopping criteria need to be defined. In our case, the fitness function was set as follows:

$$\sum_{i=1}^{N} GB_i \tag{2}$$

Where $GB$ is the goal balance after game played against the LARC2023's version of our agent. We play $N = 2$ games to mitigate the influence of the inherent randomness of the 2D environment on the results.

Since evaluating each particle involved executing a game, it was necessary to develop an infrastructure to run these evaluations in parallel, as without this, a single execution of the algorithm could take weeks. To address this, we used the structure of the RoboCIn Testing Module [6] as a basis, which we use to perform statistical tests. The containerized setup was adapted so that each container performs the evaluation of an individual, parallelizing the games' execution in a flexible way to be scalable depending on the machine on which it is executed.

In order to obtain the best results, we conducted a series of experiments, executing the algorithm under various population configurations. The Fig. 1a shows the fitness evolution of a population of 40 individuals over 100 iterations of the algorithm, and the Fig. 1b shows the mean distance between the best 20 individuals through the iterations.




(a) Best 5 individuals mean fitness and standard deviation

(b) Mean distance between the best 20 individuals

Fig. 1: Comparison of fitness and mean distance of top individuals

## 3 Kick Prediction

Due to the fact that the SS2D environment is highly dynamic, making kick prediction and interception is essential for improving goalkeeper performance and reducing conceded goals. In this section, we explore the importance of predicting kick targets and defining the sequence of actions which a goalkeeper must take to intercept these kicks effectively, combining heuristic analysis and advanced pathfinding algorithms.

Analyzing scenarios in which kick predictions can significantly reduce goal-scoring opportunities highlights the risks involved when goalkeepers do not engage in predictive actions. By employing a heuristic approach, we estimate the most probable kick target when a player gains possession of the ball, exploring various potential kick targets and factors influencing their likelihood.

To improve interception efficiency, we introduce the “Goalie Catch Generator” module, designed to generate an optimal sequence of actions to intercept the predicted kick. This module uses the A-Star algorithm (A*)[8] to determine the best combination of actions, including dash, turn, tackle, and catch, based on heuristic evaluations. After generating the optimal sequence, the goalkeeper predicatively executes the first action, reducing the probability of conceding a goal. The Fig. 2 illustrates both the kick prediction and the corresponding optimal interception sequence.

To evaluate the effectiveness of this approach, we conducted a series of 500 games played against RobôCIn, YuShan and ThunderLeague binaries from RoboCup IranOpen 2024. Table 1 demonstrates how predictive actions improve defensive strategies, reducing the scores per game of opponent teams, and consequently increasing game wins.

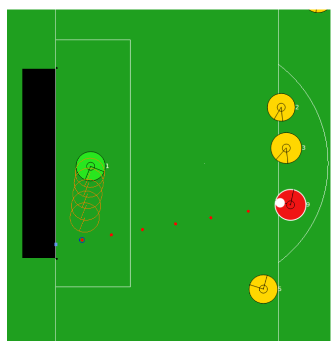


Fig. 2: Prediction of kick and optimal interception.

| | **RCPrediction** | **RCMaster** |
|---|---|---|
| **RCMaster** | 1.304 | 1.812 |
| **Yushan** | 2.006 | 2.630 |
| **ThunderLeague** | 0.677 | 0.973 |

Table 1: Statistical analysis of average score per game.

## 4 Stamina Analyzer

Beyond the dynamic nature of the environment, the 2D Simulation category also deals with observation uncertainty and the evaluation of adversarial actions. One of the challenges is to estimate the hidden state of the stamina model of other players. Our main objective is to enable both the coach and team players to infer the stamina of others during the game (teammates and opponents), allowing us to generate new features for both cooperation and exploiting opponents' conditions.

The key idea for stamina inference lies in using the displacement of the player between two consecutive observed states $O_t$ and $O_{t+1}$. With the inferred used stamina, we can estimate the current stamina model $E_t$ of a player, given that we have the initial state of the model $E_0$. The first approach to minimize inference errors is using the coach's vision, as it captures the positions of all players at every step without direct noise. Our dedicated module, "Stamina Analyzer" estimates the stamina of every player throughout the game, with the primary goal of communicating this information to teammates.

As shown in Fig. 3, the coach's estimation (yellow) exhibits discrepancies compared to the real value (blue), particularly in the final moments of the game. This occurs because player type information is necessary to avoid underestimat-

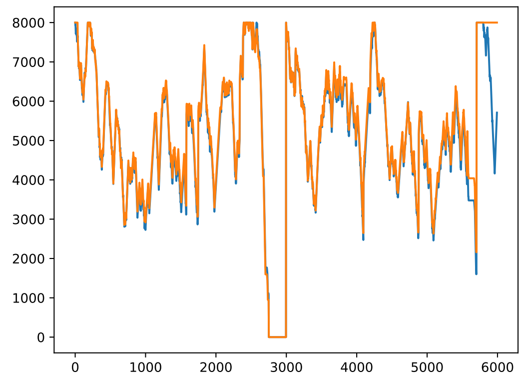



Fig. 3: Coach estimating opponent player stamina over the game.

ing an opponent's stamina. So, no estimations are made when a player substitution occurs until the player's type is identified. Additional deviations from real values result from errors introduced by the RoboCup Soccer Simulator Server [1] over players' actions. Moreover, large estimation errors also occur when players collide with objects or when movement is restricted, leading to stamina loss without a corresponding position change.

However, due to limited communication between the coach and players, relying solely on this approach is inefficient. Consequently, the usefulness of such predictions is reduced or even insufficient.

These issues led us to develop a secondary structure for players that estimates stamina by combining its own calculations with coach-provided stamina data. Since players have partial observations (leading to gaps), this structure calculates the minimum possible stamina consumption (a lower bound) between non-consecutive observations like $O_t$ and $O_{t+n}$ $(n > 2)$. The coach's data supplements estimations to reduce the accumulated error caused by observation noise and unobserved intervals. When observation pairs are unavailable, it assumes the player is recovering stamina to avoid underestimating the stamina value.

The module under development has already shown promising results but is still being improved to handle high-noise situations in players, as the current mean error is five times larger compared to the coach's error. Additionally, we plan to integrate the module with the localization class used by RobôCIn, allowing better noise handling with the Gaussian Model that is included in version 19 of the RoboCup Soccer Simulator Server.

## 5 Testing Infrastructure Improvements

Validation is one of the most essential aspects of our development cycle. Since the SS2D environment is dynamic, the best way to validate a new change is to run many matches between two distinct teams. After that, we can compare the test statistics (number of victories, number of defeats, goals scored, and other metrics) to compare the testing version with our main version. If a change produces a better result, it is integrated into our main version.

Our former test infrastructure [6] was enough for running the tests and doing the statistical analysis; however, there were some problems with test results reports and usability. Therefore, we worked on an automatic report of test results and built a GUI to improve the usability of the test module.

For automatic test reports, we created a Discord bot that sends the test metrics and results to a specific thread on a server. Besides the statistical metrics, the bot also sends information that helps us identify the version that is being tested.

For usability improvements, the created GUI makes it easier to adjust the test configuration. With this interface, we can easily select the branch to be tested, choose the opponent binary, and specify the number of games. In addition, adding a new opponent is straightforward. The GUI also allows to add comments to a specific test and view partial results of the current test.

## 6 Action Classifier

The agents' action selection plays a crucial role in the team's overall performance. During matches, we observed that RobôCIn agents often made suboptimal decisions, executing ineffective actions that frequently resulted in losing ball possession to the opposing team. This issue led us to explore ways to improve action selection and improve decision-making effectiveness.

Our solution was inspired by the work presented in "Space evaluation in football games via field weighting based on tracking data" [5]. The study introduces a classification of field areas based on two key factors: sparsity and safety. Sparsity refers to the density of players in a given region: sparse areas are those with fewer players, offering more space for long passes or advancing the ball further, while dense areas are crowded with players, making it harder to find open passing areas. The concept of safety is related to the team of the player in the region. Safe regions are those where there are more teammates in proximity, which can facilitate the execution of a successful pass. In contrast, risky regions are areas with more opponents nearby, increasing the chance that the pass will be intercepted by an opposite player. Based on these two concepts, the entire field can be classified into one of four categories: safe dense, safe sparse, risky dense, or risky sparse.

To improve the effectiveness of the actions performed by the agent, we classified the field into five strategic regions, each with a set of action priorities based on the four possible categories The goal was to prioritize or avoid certain types of actions depending on the region of the field where they happened.

To achieve this, all possible actions go through a series of evaluations in our field evaluation module. Then, each action is classified according to the four categories, and a bonus is assigned based on the priority order for the corresponding region. For example, in our defensive area, we decided to prioritize safe sparse actions and avoid risky dense ones, as sending the ball to a region crowded with opponents near our goal could be highly risky, so the bonus assigned to safe sparse actions is higher than the one assigned to risky dense actions.

The implementation of this feature has not resulted in significant improvements in our agent's performance yet, but we identified several areas for further refinement. The thresholds for the risky and sparse regions were defined based on an analysis of player data; however, those values can be adjusted and refined through deeper analysis. One possible solution could be to incorporate data from additional matches on the data analysis, resulting in more precise values for the thresholds. Also, during development, several adjustments were made to bonus values and action priorities. These adjustments were based on initial testing and could benefit from a more refined approach. Future work could involve more experimentation or the use of advanced techniques, such as parameter optimization using swarm algorithms.

## 7 Single Pass Optimization

In our ongoing efforts to enhance the decision-making capabilities of our agents in the RoboCup SS2D league, we have developed and implemented the Single Pass Optimization (SPO) model, as detailed in our recent publication [7]. The SPO model addresses the computational challenges associated with traditional action generation and evaluation methods, which often require multiple evaluations of heuristic functions, leading to increased latency and suboptimal real-time performance.

It introduces a machine learning-based architecture that generates a single optimal pass in one step, significantly reducing computational overhead. The model consists of two key neural networks: the Target Point Generator (TPG), which generates the optimal target point for a pass based on the current game state, and is trained by the Learnable Field Evaluator (LFE), which tries to learn and approximate the heuristic field evaluation function. The Best Player Selector [9] model is also used to return the best player choice inside the Player Selection Features.

The results of the experiments demonstrate the effectiveness of the SPO model. The model achieves a Root Mean Square Error (RMSE) of 309.2768, a Kendall-$\tau$ of 0.9902, and a Spearman-$\rho$ of 0.9998, indicating strong agreement with the heuristic evaluator. Furthermore, the model is up to 3.07 times faster than the current pass evaluation architecture, with a 99.98% selection rate for the generated passes.

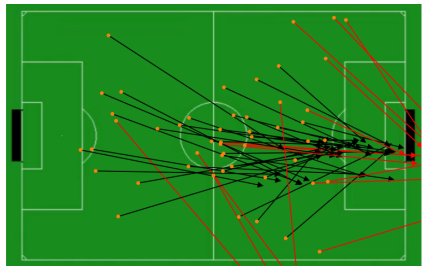


Fig. 4: Example of generated passes. The ball positions are marked in orange, while the arrows represent the direction and target point of each pass. Passes out of pitch are indicated in red.

## 8 Conclusions and Future Works

In this work, we introduced key improvements to the RobôCIn Soccer Simulation 2D agents, focusing on decision-making and infrastructure. The Field Evaluator was upgraded using bioinspired algorithms and action classification based on sparsity and safety, while new modules like the Goalie Catch Generator and Stamina Analyzer improved agents' decision-making through the execution of more precise actions. Additionally, the Single Pass Optimization (SPO) model refined pass generation using a machine learning-based architecture to produce optimal passes in a single step, reducing computational overhead while maintaining accuracy. Testing infrastructure upgrades included automated testing reports and a GUI for better experiment management.

For future work, we will refine the stamina estimation by integrating it with localization techniques for better noise handling and optimize the action classifier's priority weights using advanced methods. Our SPO model will be extended with temporal-conditioned heuristic functions and will explore additional machine learning techniques to improve pass accuracy and risk assessment. Additionally, we plan to explore RL-based policies to improve the agents' decision-making process, making them more adaptable to different opponent play styles.

## 9 Acknowledgement

First, we would like to thank our advisors and the Centro de Informática (CIn) - UFPE for all the support and knowledge during these years of project and development. We also would like to thank all our sponsors: Moura Batteries, ITEMM Institute, Microsoft, CESAR, Neurotech, Incognia, HSBS, Embraer, Instituto Nacional de Engenharia de Software (INES), Mathworks, Tractian and STMicroelectronics.

## References

1. Robocup soccer simulation server. https://github.com/rcsoccersim/rcssserver, accessed: 2025-01-01
2. 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)
3. Borsoi, B., Pereira, F., Souza, G., Souza, H., Araújo, J., Silva, M.L., Coelho, M., Machado, M., Soares, M., Cunha, P.V., Labio, R., Rodrigues, W.M., Barros, E.: RoboCIn2D Team Description Paper (2024)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948 vol.4 (1995). https://doi.org/10.1109/ICNN.1995.488968
5. Narizuka, T., Yamazaki, Y., Takizawa, K.: Space evaluation in football games via field weighting based on tracking data. Scientific Reports (2021). https://doi.org/10.1038/s41598-021-84939-7
6. de Oliveira, C.S., Machado, M.G., de Macedo Rodrigues, W., da Silva Araújo, T., Cunha, P.V., dos Reis de Labio, R., de Almeida Pereira, F.N., Soares, M.F.B., de Souza, G.L., dos Santos Silva, M.L., da Silva Barros, E.N., de Mattos Neto, P.S.G., , Ren, T.I.: RobôCIn team description paper 2023 (2023), https://archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2023/RoboCIn_SS2D_RC2023_TDP.pdf
7. Rodrigues, W.M., Cunha, P.V., De Labio, R.R., Machado, M.G., De Mattos Neto, P.S.G., Barros, E.N.S.: Spo: Single pass optimization for soccer simulation 2d. In: 2024 Latin American Robotics Symposium (LARS). pp. 1–6 (2024). https://doi.org/10.1109/LARS64411.2024.10786402
8. Stuart J. Russell, P.N.: Artificial intelligence : a modern approach. Pearson, Boston, fourth edn. (2020)
9. Zare, N., Sarvmaili, M., Sayareh, A., Amini, O., Matwin, S., Soares, A.: Engineering features to improve pass prediction in soccer simulation 2d games. In: Robot World Cup. pp. 140–152. Springer (2021)