RobôCIn Soccer Simulation 2D Team Description Paper for RoboCup 2026

Gabriel Souza, Giovanna Mafra, Heitor Souza, Jeferson Araújo, Maria L. Silva, Mariana Freire, Pedro Victor Campello, Tales Cunha, 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 2025, we achieved 4th place at RoboCup, consolidating our position among the top teams in the league. In this paper, we present the work developed over the past year, especially the refinement of free kick behavior and penalty formation, the transition to RCSSServer version 19 and the improvements made based on the Cyrus 2D Open Source code.

Keywords: Soccer Simulation 2D · Gaussian Noise · Bipedal · Penalty formation · Cyrus 2D

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), Small Size since 2018, Flying Robots Trial League since 2022 and Humanoid League since 2025.

RobôCIn used the well-structured base of agent2d 3.1.1, as presented in [1], as the basis for our code development in the first year. Our most recent agent release includes changes to free kick behavior and penalty formation, in addition to the transition to RoboCup Soccer Server version 19 and the integration with the recently released Cyrus 2D Open Source code [8] [9].

2 CYRUS 2D Open Source

CYRUS is a team from Dalhousie University's Institute for Big Data Analytics [7], founded in 2012, that has many considerable achievements in the

RoboCup Soccer Simulation 2D (RCSS2D) league, including first place at the 2021 RoboCup. In August of 2025, they made their 2D Soccer Simulation code open source [8]. For clarity, we will refer to both the code in the referenced repository and its corresponding RCSC library implementation collectively as CYRUS2025.

Considering the relevance of the CYRUS team in the category, we conducted a thorough analysis of CYRUS2025. Based on our findings, we were able to improve our agent's decision-making and overall game performance.

2.1 Code Analysis

We began the study of CYRUS's open source environment by mapping the modules of the RCSC library [9] and their RCSS2D agent in order to identify any differences from our project. The analysis revealed several disparities, including:

- Use of Version 18 of the Soccer Server with modifications for Bipedal use
- Different ball intercept logic
- Use of CyrusAudioMemory class to extend audio memory with support for start set play kick messages.
- Utilization of PointTo action as a means of sharing stamina information among players
- Use of a Deep Neural Network (DNN) for pass prediction [6]
- Different defense and offense behaviors

When the study was completed, we began to carefully integrate the CYRUS' modules that could improve our codebase performance.

To verify such improvement, we conducted several statistical tests utilizing our tester module [4], as well as a few specific manual checks. The tests shown in table 1 were executed with the newly developed versions (RCDev) against our previous binary for the IranOpen 2025 (RCMaster) competition to assure the validity of the changes.

	RCDev	RCMaster
Victories	69.03	14.98
Defeats	14.98	69.03
Draw	15.99	15.99
Score	1044	418
G.B	626	-626
S.P.G	2.11	0.85

Table 1: Presents the results of 1,000 simulated matches comparing our agent with and without the CYRUS2025 integration, including key performance metrics such as win, loss, and draw rates, as well as average goals scored per match (S.P.G) and goal balance (G.B).

2.2 Incorporated Changes

We first began to adjust our library so that the subsequent modifications to the agent would run smoothly. These changes include, but are not limited to, improvements in player state tracking, swapping to CYRUS' intercept structure, adding new files for stamina visualization and logging, as well as adapting our scripts to properly build the DNN.

Some notable alterations to our agent include the addition of CYRUS messages, parsers and formation configuration files, the extension of the FieldAnalyzer class, modifications to the penalty kick behavior, the introduction of CYRUS' denoising system for player localization, as well as the defensive and offensive behaviors.

Along the process, we encountered many challenges in maintaining the organization of our code as well as the stability of our overall performance, as the consecutive adjustments provided an added risk for unknown code behavior and bugs.

2.3 Future plans

In future work, we intend to further refine our agent with the knowledge obtained from CYRUS2025, and implement the necessary adjustments to preserve the advantages of both approaches.

3 Update to RCSSServer V19

The transition of the RoboCup Soccer Simulation Server [3] to version 19 introduced fundamental changes to the environment's physics, most notably the implementation of the bipedal locomotion model and Gaussian error in the vision system.

3.1 Evolution of Movement Models

In versions prior to Version 19, agent movement was based on discrete and atomic commands: *side/backward dash*, *turn*, and *omni dash*. Under this paradigm, agents were unable to rotate and accelerate simultaneously within the same simulation cycle. This limitation necessitated interleaving commands to approximate optimal trajectories, resulting in a loss of kinematic efficiency and increased response time to reach destination points.

The new bipedal model, conversely, allows for the simultaneous execution of translation and rotation, bringing the simulation closer to the physical behavior of real soccer players. However, this flexibility drastically increases control complexity, requiring the coordinated parameterization of four variables: the power and directions applied individually to the left and right legs (power_l, dir_l, power_r, dir_r), while also considering the player type's variables and the current state of the stamina model.

3.2 Implementation of Movement Abstraction

To manage the complexity of the bipedal model, we propose an abstraction function $f(\omega, \theta)$ that maps the agent's tactical intent to the server's low-level commands.

In this approach:

- ω represents the desired rotational momentum.
- θ represents the direction of the desired acceleration vector.

The control strategy prioritizes the rotational component, ensuring directional precision before linear propulsion. The calculation algorithm follows four fundamental stages:

1. **Rotation (ω):** The differential force required in each leg (pivot) is determined to achieve momentum ω.
2. **Longitudinal (x):** A force component is added to each leg to generate acceleration along the agent's longitudinal axis.
3. **Steering (y):** Based on the generated components, the vectors are adjusted to include direction θ, resulting in a pair $(\text{power}, \text{dir})$ that integrates the necessary lateral force component.
4. **Command Output:** The algorithm returns the final action vector (power_l, dir_l, power_r, dir_r).

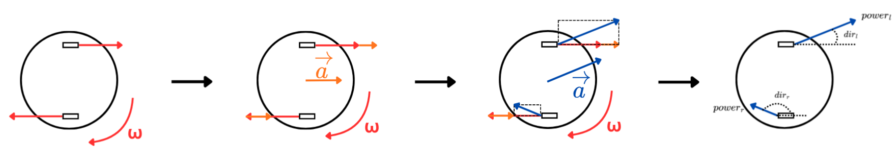

Fig. 1: Bipedal dash calculation approach

This approach is effective because rotation is preserved by the relative acceleration difference between the legs, while the resulting acceleration is determined by the vector sum of the applied forces. As shown in Figure 1, our approach can generate even acceleration and rotation in different directions while being a simple approach.

While other teams, such as FRA-UNIted [2], have recently integrated bipedal motion, their implementation is primarily localized to specific high-level behaviors, most notably the TacklishIntercept2024 skill, which uses bipedal specifically to maximize tackle probability in the final moments of an interception. In contrast, our proposed abstraction $f(\omega, \theta)$ is used in most intercept situations, including long-range interceptions, short-range positioning, and the preparation for kicks or tackles, we achieve a significant reduction in overall intercept times. Furthermore, this systemic adoption provides a competitive advantage, as many existing opponent models still rely on traditional motion predictors that do not account for the trajectories enabled by the bipedal model.

The primary barrier to universal bipedal utilization is its higher stamina cost, unlike the standard turn model, which consumes no stamina, bipedal rotation is energy-intensive. Consequently, the key to our improvement lies in a hybrid approach that leverages the speed of bipedal motion while reverting to the standard model for simple movements to preserve stamina.

3.3 Gaussian Noise Model

Another major improvement in version 19 is the integration of a Gaussian noise model for player vision. Unlike the previous quantization-based model, which lacked realism and complicated error propagation, the Gaussian approach provides a more efficient framework for state estimation. This model not only reduced perception errors but also enabled the effective use of the focus_point action, which had been previously underutilized due to the constraints of the quantization model.

Preliminary tests, conducted by isolating the Gaussian model without full error propagation tuning, already demonstrate its potential. As shown in table 2, the agents achieved an improvement in win rate. These results suggest that the natural noise distribution enhances decision-making even in its early stages of implementation, offering significant opportunities for future optimization.

	RCDev	Bipedal	Gaussian	Both
Victories	28.0	31.40	29.90	32.20
Defeats	26.8	25.80	26.50	25.50
Draw	45.20	42.80	43.60	42.30
Score	521	597	575	604
G.B	-2	75	50	110
S.P.G	0.521	0.597	0.575	0.604

Table 2: Presents the results of 1,000 simulated matches comparing RCDev with Bipedal and Gaussian modifications.

4 Free Kick Correction

Although our test module offers reliable statistical data, a detailed review of past games remains crucial for understanding our agent's behavior. During those reviews, we repeatedly observed an undesirable pattern during defensive free kicks near our goal line.

To investigate the issue thoroughly, we recreated the situation in the trainer module. This scenario, while not frequent, is still significant. By simulating it against various opponents, we could monitor how changes to the agents influenced their behavior. The core problem was clear: players struggled to complete passes near our goal line, particularly when facing teams that applied tight man-to-man marking.

Through empirical testing in the trainer, our initial adjustment was to modify player formations, moving the nearest players closer together just before the free kick, without weakening the defensive structure. Although this step provided useful insight, it alone did not resolve the issue.

After the formation changes proved insufficient, we focused on adapting the specific behavior for defensive free kicks, ensuring other free kick routines remained unaffected. The implemented solution involved having the nearest allies move linearly into the area during the kick, improving both passing and receiving opportunities. Additionally, we found that orienting the free kick taker's vision toward the opponent's goal line slightly enhanced pass quality and subsequent plays.

To measure the impact of these changes, we ran 500 matches against our team's RoboCup IranOpen 2025 version (RCMaster). As shown in Table 3, this adjustment contributed to an increase in match victories

	RCDev	RCMaster
Victories	36.02	33.80
Defeats	33.80	36.02
Draw	30.18	30.18
Score	566	539
G.B	27	-27
S.P.G	1.14	1.08

Table 3: Shows a standard sample RoboCIn-tester output. G.B stands for Goal Balance and S.P.G for score per game.

5 Penalty formation

A different approach from [5] for the penalty situation to be a key improvement involved refining our goalkeeper's behavior during penalty shootouts is the new penalty formation. Previously, the team might have utilized a separate, specialized goalkeeper agent for this mode. We replaced this with our standard, primary game goalkeeper but introduced a critical modification to its strategic positioning.

5.1 General Idea of the New Penalty Formation

Instead of designing an isolated set of actions, we adapted the goalkeeper's standard defensive formation and positioning rules for the penalty scenario. The core concept is to use same goalkeeper's role for the penalty situation.

Firstly, its target position was shifted from the standard, central goal-line position to a more aggressive, forward location. This is calculated based on statistical analysis of opponent penalty trends, positioning the goalkeeper optimally for the most probable shot trajectories without compromising coverage of the goal's extremes.

In addition, by using the main goalkeeper, the agent retains its full suite of calibrated skills, ensuring reliable physical execution. The formation change simply provides a more intelligent starting point and movement bias.

To evaluate the new penalty modification, we ran a test of 500 matches against our team's RoboCup IranOpen 2025 version (RCMaster). As shown in Table 4, this adjustment contributed to an increase in match victories and our penalty's behavior

	RCDev	RCMaster
Victories	90.52	9.48
Defeats	9.48	90.52
Score	1359	544
G.B	815	-815
S.P.G	2.74	1.09

Table 4: Shows a standard sample RoboCIn-tester output. G.B stands for Goal Balance and S.P.G for score per game after the new penalty formation.

6 Conclusions and Future Works

In this work, we introduced key improvements to the RobôCIn Soccer Simulation 2D agents, including significant changes to the code base, updates to support new features available in the latest versions of the RoboCup Soccer Server, and adjustments to SetPlay strategies.

The integration with the Cyrus open-source base code represents a new challenge in the RoboCup 2D League, as we expect that many established and new teams will adopt it as their primary framework. In addition, we report our experience with the latest league updates, particularly the bipedal motion model and the Gaussian noise model. These changes allow the 2D simulation to better approximate real soccer dynamics and provide noticeable improvements in agent behavior and realism. Furthermore, the updated free-kick and penalty mechanisms address important specific game situations, contributing to overall performance improvements. The results of all adjustments are presented, as we believe that the gains achieved in each aspect may serve as useful references for other teams.

For future work, we plan to further improve our integration with the Gaussian noise model, especially by developing and refining Kalman filter approaches to enhance position estimation accuracy. We also intend to explore the Change Focus Model, available since version 18 of the server. This model was not previously utilized due to limitations associated with the older Quantization Noise Model; however, the introduction of the new Gaussian Noise Model now enables a more effective adoption of the mechanism. Finally, we are continuing the process of fully integrating our code with the Cyrus open-source framework, and we expect additional improvements once this process is completed.

7 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, Microsoft, CESAR, Incognia, HSBS, SENAI, Neurotech and STMicroelectronics.

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. Gabel, T., Eren, B., Vanegas, J., Godehardt, E.: FRA-UNIted — Team Description 2025 (2025)
3. Noda, I.: Soccer server: A simulator of robocup. In: Proceedings of the AI Symposium 1995. pp. 29–34. Japanese Society for Artificial Intelligence (1995)
4. 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`
5. Ramos, L.F.B., Maximo, M.R.O.A., de Paula, D.C.L., Tizon, J.A.F., Tedesco, G.S., da Silva, P.A.C., Gonzaga, J.G.M.: ITAndroids 2D Soccer Simulation Team Description Paper 2025. RoboCup 2D Soccer Simulation League (2025)

6. Zare, N., Firouzkouhi, A., Amini, O., Sarvmaili, M., Sayareh, A., Rad, S.R., Matwin, S., Soares, A.: CYRUS Soccer Simulation 2D Team Description Paper 2022 (2022), archive.robocup.info/Soccer/Simulation/2D/TDPs/RoboCup/2022/CYRUS_SS2D_RC2022_TDP.pdf
7. Zare, N., Sayareh, A., Khanjari, S., Firouzkouhi, A.: Observation Denoising in CYRUS Soccer Simulation 2D Team for RoboCup 2024. arXiv preprint arXiv:2406.05623 (2024)
8. Zare, N., Sayareh, A., Khanjari, S., Firouzkouhi, A., Amini, O.: CYRUS Soccer Simulation 2D Team. https://github.com/Cyrus2D/cyrus-soccer-simulation-team (2025), accessed: 2026-02-19
9. Zare, N., Sayareh, A., Khanjari, S., Firouzkouhi, A., Amini, O.: Librcsc. https://github.com/Cyrus2D/cyrus-soccer-simulation-lib (2025), accessed: 2026-02-19