ITAndroids 2D Soccer Simulation Team Description Paper 2026

Diogo C. L. de Paula, Marcos R. O. A. Maximo, João Miguel C. S. S. Mendes, João Pedro P. Sales, Renan A. Farrapo, and Vitor Hugo A. Ferreira

Aeronautics Institute of Technology
São José dos Campos, São Paulo, Brazil
{dcl.depaula, vhaf22}@gmail.com, mmaximo@ita.br
itandroids-soccer2d@googlegroups.com
http://www.itandroids.com.br

Abstract. This paper presents the latest developments in ITAndroids 2D, a RoboCup Soccer Simulation 2D (RCSS2D) team. Building upon previous iterations, we introduced a tuned action chain evaluation to increase offensive aggressiveness, a refactored and spatially simplified goalkeeper positioning model, enhanced defensive aggressiveness parameters, a synchronized dynamic formation switching mechanism based on match context, optimized stamina management with improved dash power allocation, and refined passing heuristics with upgraded opponent interception prediction to enable safe tactical retreats. The performance of the updated team was evaluated against last year's version over a set of 1,000 matches, achieving a win rate of 57.1%, a tie rate of 26.4%, and a loss rate of 16.5%.

Keywords: RoboCup · Soccer Simulation 2D · Multi-agent Systems · ITAndroids.

1 Introduction

ITAndroids is a competitive robotics team from Aeronautics Institute of Technology reestablished in 2011. The group participates in the following leagues: RoboCup Soccer Simulation 2D (RCSS2D), RoboCup Soccer Simulation 3D, RoboCup Humanoid Kid-Size, IEEE Very Small Size, and RoboCup Small Size League.

Our RCSS2D team, ITAndroids 2D, has continuously participated in the Latin American Robotics Competition (LARC) and the Brazilian Robotics Competition (CBR) since 2011. ITAndroids 2D competed in RoboCup from 2012 to 2025, except in 2014. Table 1 shows the placements in these competitions.

2 Previous Works

Our code base uses agent2d [1] as the base team. Since 2011, we have focused on improving the mechanisms already present in the agent2d framework. We improved the action chain evaluator with Particle Swarm Optimization (PSO) [2].

Table 1: Placement of ITAndroids 2D in past RoboCup and CBR competitions.

Year	**RoboCup**	**CBR**
2025	10th	2nd
2024	11th	4th
2023	10th	3rd
2022	11th	3rd
2021	11th	5th
2020	—	4th
2019	13th	1st
2018	9th	2nd
2017	15th	3rd
2016	13th	2nd
2015	13th	1st
2014	—	1st
2013	13th	1st
2012	10th	1st

Furthermore, we developed heuristics [3] to increase attack and defense performance. Many early improvement ideas were inspired by HELIOS [4] and Nemesis [5].

In recent years, we implemented a Finite-State Machine called Possession Automaton [6] and experimented with Deep Reinforcement Learning (DRL) for goalkeeper defense [7] and individual defensive behavior [8]. In 2025, we focused on marking during goal defense, intercept table improvements using trajectory planning, and fixing decision-making loops (the spinning problem) [9].

3 Action Chain Evaluation Tuning

The decision-making process in our agent is based on an action chain evaluator that assigns a "danger" value to potential plays. We observed that the team was often too conservative, opting for passes in situations where a shot on goal was viable. To address this, we adjusted the heuristic evaluation by reducing the danger weight of the 'shoot' action. This change allows the action chain search to prioritize scoring opportunities even under moderate pressure, increasing our team's overall aggressiveness in the final third of the field.

In a specific modification, all the penalties of the shoot action were halved, making it gain a much higher priority in the action chain. Then, we ran 1000 matches against the code before any modifications. The win rate of the new code was 39% compared to 26.25%, thus showing a significant increase in offensive effectiveness.

4 Defensive Aggressiveness and Goalkeeper Refactoring

To minimize vulnerabilities against fast-paced attacks, we increased the proactivity of our defensive units within the penalty area by expanding "chase distance" parameters. Furthermore, the goalkeeper's movement logic was simplified, removing redundant body-neck synchronization checks that caused sub-millisecond delays. These adjustments prevent the goalkeeper from becoming trapped in passive states, ensuring a more decisive response to close-range shots.

Building upon these improvements, we refined the goalkeeper's positioning through a dual-state spatial model. When the ball is near the penalty area, the goalkeeper positions himself along the angle bisector between the ball and the goalposts. Simultaneously, the agent evaluates the attacker's proximity to anticipate imminent one-on-one scenarios and dynamically reduce shooting angles. Conversely, when the ball is far from the danger zone, the goalkeeper maintains a central stance within the goal area to prioritize positional balance and stamina conservation.

This strategy was validated through 1,000 automated matches against the baseline team. As shown in Table 2, the updated version demonstrated higher defensive solidity, conceding 18.4% fewer goals while maintaining competitive consistency.

Table 2: Performance comparison over 1,000 matches

Metric	Updated Version	Baseline Version
Win Percentage (%)	42.00	30.20
Goals Conceded	1028	1260

5 Dynamic Tactical Formation Switching

In previous versions of our code base, the team's structural formation was largely static, determined either at the start of the match or rigidly tied to initial opponent identification. To increase our tactical flexibility and adaptability during a match, we developed a synchronized, in-game formation switching mechanism.

Rather than relying on a single set of positional configurations for the entire duration of a game, our agents now continuously monitor the evolving context of the match. The primary heuristic for this dynamic adaptation is the current score difference. If the match state shifts—for example, transitioning from a draw to a leading state—the agents autonomously trigger a collective tactical reorganization.

To execute this without disrupting the agents' decision-making cycles, the system was designed to dynamically load new formation configuration files on the fly. Because this evaluation is performed synchronously across all agents, the entire team can seamlessly shift its base positioning. For instance, the team can automatically adopt a more conservative, defensively robust structure when protecting a lead, or revert to a standard or aggressive posture when trailing or tied. This dynamic structural response allows the team to better manage game pressure and optimize field control based on real-time needs.

6 Optimized Stamina Management and Dash Power Allocation

In high-paced simulation matches, efficient energy consumption is crucial for maintaining tactical structure over the course of the game. We identified opportunities to refine how agents distribute their energy during standard movements. To improve this, we overhauled the heuristic responsible for calculating normal dash power. The decision-making logic now incorporates specific dynamic thresholds to accurately determine when an agent should enter a stamina recovery state.

Furthermore, this stamina management was integrated with the team's spatial awareness. The updated model specifically evaluates whether a player is correctly positioned along the defensive line to execute or support an advanced forward pass. By factoring in this positional context, defenders can better conserve energy while maintaining the structural line and allocate their dash power more intelligently when transitioning the ball forward.

To empirically validate these behavioral changes, we evaluated the updated stamina management system against our unmodified baseline over 1,000 automated test matches. The updated agents demonstrated a clear performance increase, achieving a 39.40% win rate and a 25.40% draw rate, compared to a 35.20% loss rate. This positive net balance in performance highlights the effectiveness of coupling refined stamina recovery thresholds with tactical defensive positioning.

7 Refined Passing Heuristics and Tactical Retreat Support

In previous iterations, our action generation system for passing was heavily constrained by hardcoded forward-bias rules. The pass generators would predominantly search for direct or leading passes that advanced the ball, often prematurely discarding safer, backward options (retreats) even when the ball handler was under heavy pressure.

To enable a more possession-oriented playstyle, we structurally overhauled the pass evaluation and generation pipelines. First, we removed rigid spatial heuristics that artificially prevented the evaluation of passes to teammates positioned behind or parallel to the passer. To complement this, we rebalanced the scoring function for pass routes. Instead of strictly penalizing negative yardage, we introduced a smoothed penalty curve for backward passes and correspondingly moderated the excessive heuristic bonuses previously awarded to forward passes. This normalization ensures the agent values ball retention over forced, risky forward progressions.

Because backward passes in the defensive half carry a high risk if intercepted, we simultaneously upgraded the opponent interception prediction logic. The refined verification model now calculates an opponent's precise one-cycle reachable radius based on their speed and kickable area. Furthermore, it uses a more rigorous geometric projection to estimate the opponent's time-to-intercept across the planned pass line.

To validate our passing system, we conducted incremental testing against competitive benchmarks, such as CyrusBase [10]. This evaluation allowed us to observe how each heuristic influenced the team's ability to execute tactical retreats and maintain ball possession without increasing turnovers in dangerous areas.

Initially, we observed that removing traditional spatial heuristics, while simplifying the decision-making process, left the team vulnerable to interceptions. The subsequent integration of an advanced interception prediction model proved to be the most effective improvement, significantly enhancing our offensive penetration and defensive reliability. This configuration allowed the agents to identify safer passing lanes and reach the ball more efficiently.

In a later stage, we experimented with a route score smoothing heuristic intended to stabilize possession. However, qualitative analysis revealed that this addition led to an unintended degradation in performance. The agents became overly conservative, prioritizing low-risk horizontal passes and ball retention to the detriment of offensive progression. This excessive caution effectively neutralized the team's ability to create scoring opportunities. Based on these observations, we concluded that the smoothing heuristic negatively impacted the team's net competitiveness. Consequently, we opted to discard this refinement, officially adopting the configuration based on optimized interception prediction as our final solution for the competition.

8 Multi-Agent Reinforcement Learning for One-on-One Situations

A critical scenario in RCSS2D is the one-on-one confrontation between an attacker and the goalkeeper. To address this, we are developing a Reinforcement Learning (RL) task based on asymmetric adversarial training, using a framework developed by a former team member, which trains only the two agents in a half-field scenario [11]. Currently, we have established the final observation and action spaces for both agents and completed their baseline training.

8.1 Attacker and Goalkeeper Modeling

The agents utilize a Rainbow DQN algorithm [12]. A key improvement in this iteration is the refinement of the observation space. Instead of global coordinates, the agents perceive the world through radial and tangential velocity components relative to the line of sight between the attacker and the ball or the goalkeeper. This geometric decomposition enhances the agent's ability to learn precise timing for shots and saves.

The action spaces were also optimized: the attacker manages 37 discrete actions (including targeted shots from $-60°$ to $+60°$), while the goalkeeper's rewards were shaped to prioritize bisector positioning and ball interception (touch reward). In preliminary tests, the baseline attacker reached a goal rate of 76.4%, and the goalkeeper demonstrated a significant reduction in passive behavior through specific rewards for ball approach and bisector positioning.

8.2 Work in Progress: Adversarial Training

Although the individual baseline models have been concluded, the adversarial phase, in which both agents co-evolve by playing against each other—is currently underway. We intend to complete this training and integrate the resulting policies into the official ITAndroids code before RoboCup 2026. The implementation will be triggered specifically in "clear path" situations, where no other players obstruct the trajectory between the attacker and the goal.

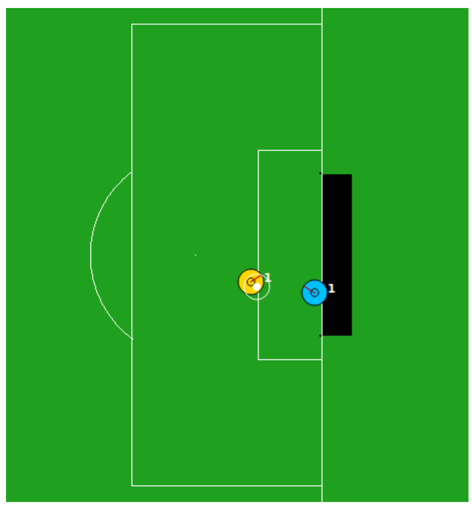

Fig. 1: A training situation of the two agents in the Reinforcement Learning Framework.

9 Integration of Cyrus Open-Source Code

Finally, a significant milestone for ITAndroids 2026 was the partial integration and adaptation of the open-source code published by the Cyrus team [13]. The incorporation of Cyrus' advanced decision-making frameworks and formation management systems has led to a substantial increase in our team's performance. Initial benchmarks show that these implementations significantly improved our win rate, not only against baseline teams but also against internationally relevant opponents. This leap in performance is largely due to more sophisticated field evaluations and enhanced coordination during set plays.

10 Conclusions and Future Work

In this work, we introduced several improvements to ITAndroids 2D, focusing on refining the action chain evaluation, increasing defensive aggressiveness alongside a new goalkeeper positioning logic, implementing a dynamic tactical formation switching system, optimizing stamina management, and refining passing heuristics to support tactical retreats. The results show that these modifications substantially increased the team's overall robustness and tactical flexibility. The new goalkeeper spatial model and optimized dash power allocation consistently reduced the number of goals conceded, while the retreat passing mechanism improved ball retention under opponent pressure. Together, these implementations translated into a tangible increase in win and draw rates during automated tests against reference teams. The compounded difference of these changes can be observed in Table 3. It is worth noting that this comparison does not include the incorporation of CYRUS's open-source code. Only implementations of our own authorship are evaluated here.

Table 3: Performance of ITAndroids2026 against ITAndroids2025 evaluated over a set of 1000 matches.

Outcome	Percentage
Wins	57.1 %
Ties	26.4 %
Losses	16.5 %

In the future, we wish to expand the application of Deep Reinforcement Learning (DRL) algorithms to other collaborative mechanics, refine our action chain to perform more intelligently in high-pressure situations, and continue modernizing our codebase to handle RCSSServer updates.

Regarding our long-term research, we aim to conclude the asymmetric adversarial training cycle for the one-on-one task. The preliminary results from our RL-based attacker and goalkeeper are promising, showing high efficiency in specialized tasks. Our next steps involve the full integration of these neural policies into the agent's decision-making logic for match situations. This will provide ITAndroids with a more robust and adaptive behavior during critical offensive and defensive transitions, potentially increasing our scoring efficiency against high-level opponents.

11 Acknowledgments

We would like to acknowledge the RoboCup 2D Soccer Simulation community for sharing their developments. In particular, we would like to acknowledge Hidehisa Akiyama for agent2d and librcsc [1] and also Nader Zare for CYRUS' code [13]. Finally, we acknowledge our sponsors: Edge Group, Nubank, Altium, CENIC, Field Pro, ITAEx, MathWorks, Rapid, SIATT, SolidWorks, STMicroelectronics, and Visiona Space Technology.

References

1. Hidehisa Akiyama and Tomoharu Nakashima. "HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation". In: *The 17th annual RoboCup International Symposium*. July 2013.
2. Fábio Mello et al. "ITAndroids 2D Soccer Simulation Team Description 2012". (2012).
3. Felipe Coimbra et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2017". (2017).
4. Hidehisa Akiyama and Hiroki Shimora. "HELIOS2010 Team Description". (2010).
5. Mehrab Norouzzilatab et al. "Nemesis Team Description 2010". (2010).
6. Felipe Coimbra and Lucas Lema. "ITAndroids 2D Soccer Simulation Team Description 2018". (2018).
7. Diego Fidalgo et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2020". (2020).
8. Davi Vasconcelos et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2024". (2024).
9. Luiz F. B. Ramos et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2025". (2025).
10. Nader Zare et al. "Cyrus2DBase: A base team for the RoboCup Soccer Simulation". (2022). Available at: https://github.com/Cyrus2D/Cyrus2DBase.
11. Davi Vasconcelos et al. "A Reinforcement Learning Framework for RCSS2D". Available at: https://gitlab.com/davi.vasconcelos/a-reinforcement-learning-framework-for-rcss2d.
12. Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. "Rainbow: Combining Improvements in Deep Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32, No. 1. 2018. doi:10.1609/aaai.v32i1.11796.
13. Nader Zare et al. "CYRUS Soccer Simulation 2D Team". (2022). Available at: https://github.com/Cyrus2D/cyrus-soccer-simulation-team.