RobôCIn 2026 Large Robot Division Extended Team Description Paper

Eduardo M. Melo, Erick V. R. Cruz, Fernanda M. Neves, Giovanna M. de C. Bardi, Isabela M. de M. Nascimento, Jefferson P. de O. Júnior, Lara V. Luchi, Pedro H. A. da Silva, Thiago R. Magalhães, 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://www.robocin.com.br/

Abstract. This article presents an overview of the RobôCIn team, a newcomer to the Humanoid League, and describes the main development efforts and preparations for RoboCup 2026. Initially, the hardware platform adopted by the team, Booster T1 robots, is presented, followed by an overview of the software architecture, which is under continuous development and refinement until June. The paper discusses the main challenges encountered so far, as well as the team's short- and long-term plans, which include the expected developments and future directions of the project.

Keywords: RoboCup · Humanoid League · Autonomous Soccer · ROS 2 · Computer Vision · Behavior Trees

1 Introduction

RobôCIn is a Robotics Research and Development group from the Federal University of Pernambuco, Brazil, which has participated in robotics competitions for the past ten years. Initially active in a single category, the team has since expanded its participation to six categories, with the Humanoid League being the most recent. The team entered this category in 2025, following the acquisition of two Booster T1 robots after RoboCup 2025. Since then, RobôCIn has been focused on learning and development in this new domain, aiming to be fully prepared for RoboCup 2026. The following sections describe the work completed so far, as well as the objectives and planned developments leading up to RoboCup 2026.

2 Hardware Description

Two Booster T1 humanoid robots are used as the hardware platform of the team. Each robot consists of a head, torso, arms, and legs, totaling 23 degrees of freedom (DoFs), which allow flexible motion and accurate posture control

[1]. The Booster T1 provides key capabilities such as omnidirectional walking, disturbance resistance during locomotion, and autonomous operation as a soccer-playing agent, serving as a foundation for research and development in humanoid robot soccer.

Communication with the robots is typically established via Ethernet or Wi-Fi. During matches, the robots connect via Wi-Fi to the GameController, enabling automatic synchronization of game states and fast match initialization. Each robot is equipped with an NVIDIA Jetson AGX Orin (32 GB) for onboard processing and a ZED 2i depth camera, which enables environmental perception through the detection of field lines and the ball, supporting self-localization during gameplay. In addition, a built-in microphone reports the current operating mode, which is useful for development and debugging. Overall, this hardware configuration provides an appropriate basis for the development, testing, and validation of the solutions adopted by the team.

3 Software Description

The team's software is partially based on open-source code. Our development uses the official `robocup_demo` codebase [2] provided by Booster Robotics as a foundation, on which we implement incremental modifications and improvements. Additionally, we received technical support from other teams in the Brazilian Humanoid League, and some fixes were incorporated based on shared solutions, including adaptations derived from the RoboFEI team's codebase.

The software architecture is built upon ROS 2 Humble using C++ and Python, providing a modular framework that facilitates distributed processing between the Motion Control and Perception boards via Fast DDS. The system consists of three primary high-level packages: `vision`, `brain` and `game_controller`, along with manufacturer interface packages.

The perception system (vision node) utilizes a YOLOv8 model on the NVIDIA Jetson AGX Orin. By employing optimized heuristic methods, the system detects key game elements and transforms their positions from image space to the robot's Cartesian frame, based on a flat-ground assumption and intrinsic/extrinsic camera calibration.

For decision-making, the team utilizes the BehaviorTree.CPP library within the `brain` node. This hierarchical structure manages the robot's logic, handling transitions between GameController states (Initial, Ready, Set, Playing) and executing role-specific behaviors. Currently, the system supports distinct strategies for the Striker (focused on ball tracking and kicking) and the Goalkeeper (focused on positioning and blocking), defined in modular XML subtrees.

Finally, motion control is abstracted through a high-level API provided by the robot manufacturer, while development is supported by Isaac Sim for simulation and Rerun for real-time data visualization and debugging.

4 Major Problems

Throughout the project, the team encountered a variety of challenges, ranging from connectivity issues to the complexity of the interface tools. The main ones that we faced are detailed below:

Simulation environment: The installation of the simulation environment, Isaac-Sim, required complex configurations. Additionally, the processing power of our laboratory computers is insufficient for the software to run smoothly, which slows down the development process. The team is currently adapting to the simulation workflow, specifically regarding the implementation and testing of new behaviors on the robot.

Rerun visualization tool: Initially, the team attempted to use a newer version of Rerun, which did not support the Transmission Control Protocol (TCP) that was implemented on the Humanoid for communication. The solution to this issue was to revert to an older version of Rerun that met our technical requirements.

Camera Calibration: To ensure the robot functioned properly, camera calibration was required. However, we encountered an issue with the libsceres.so and libsceres.so.4 files. These were intended to be symbolic links but were generated in an incorrect format, which caused the calibration to fail. The team resolved this by deleting the files and manually fixing the library dependency chain.

Booster T1 software stack: The software provided by the manufacturer is complex, featuring multiple communicating layers. Consequently, fully understanding the system's architecture and how its components interact remains a current challenge for the team.

5 Short- and Long-Term Goals

5.1 Short-Term Goals

Improve our in-game strategy: During the exhibition matches that we participated in October 2025 in São Paulo, we observed that although our robot is already capable of performing several actions from a hardware perspective, the software lacks a robust behavior tree that fully exploits these predefined movements. Therefore, we plan to improve the behavior tree in order to better leverage the robot's existing capabilities and enhance its in-game performance.

5.2 Long-Term Goals

Research the use of reinforcement learning to improve motion primitives: Although our robot is able to operate using the currently trained behaviors, their performance is far from optimal. We aim to improve the robot's gait, decision making, kicking actions, and turning behavior, the latter of which is particularly slow at present. Enhancing these motion primitives will result in faster and more precise movement, enabling overall improvements in the robot's performance. In particular, improved turning speed and stability will facilitate the development of other system components, such as localization, which is currently hindered by the robot's slow rotational movements.

Research and develop algorithms for autonomous humanoid robotic systems. We aim to develop and refine algorithms across all core areas of autonomous humanoid robotics, including vision systems, perception through machine learning, localization, and path planning and execution. These improvements will contribute to more robust, efficient, and adaptable autonomous behavior.

References

1. T1 Instruction Manual, https://booster.feishu.cn/wiki/XAS3wv4lwiSiXXkDbMrceE6UnHc, last accessed 2026/01/14
2. RoboCup Demo Codebase, https://github.com/BoosterRobotics/robocup_demo, last accessed 2026/01/26