

Software Survey 2026

Team name

ITAndroids

Which division(s) are you applying for? If your used software differs between divisions, please fill out the survey once per division.

Small Size (height < 110 cm, weight < 15 kg)

Is your software fully or partially OpenSource? If so, where can it be found?

Some parts of our code, especially those associated with publications, can be found at: <https://gitlab.com/itandroids/open-projects>

Are you using any software developed by other teams? If so, list every component that you are reusing and the team that originally developed it.

The mechanical design of the Chape robot was strongly inspired by the Darwin platform. In addition, part of our linear algebra codebase (including vector, matrix, and transformation operations) was initially based on the implementation made publicly available by the UT Austin Villa team. The remaining software and hardware components were developed by the team.

Are you using any datasets in your research? If you are using your own datasets, are they public?

We use Bit-Bot's TORSO-21 Dataset for training and testing our vision system.

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).

Arthur Costa Stevenson Mota and Marcos R. O. A. Máximo. Minimum Time Footstep Planning for Simulated Robot Soccer Kicks using Model Predictive Control. Brazilian Symposium on Robotics (SBR), 2024.

Are there any other contributions you would like to share with the RoboCup community?

We have a partially open source code and a full open source Webots model that can be

downloaded at the Humanoid League website.

We have taught a course on humanoid walking (in Portuguese) in two Brazilian events: the Brazilian Humanoid Robot Workshop (BRAHUR) 2019 and the

Brazilian Congress of Automatics (CBA) 2020. The CBA version is publicly available online (in Portuguese): <https://www.youtube.com/watch?v=v2cxIF8oExE>.

Some teams in Brazil have reported that this course has helped them to understand more about humanoid walking.

Moreover, we have a YouTube channel with research videos and courses:

<https://www.youtube.com/@ITAndroids>.

We have participated in many competitions, as shown below, sharing our experience with other teams and the whole league:

- 2st place Brazilian Robotics Competition (CBR) 2025;
- Top 8 - Participated in RoboCup 2025;
- 1st place Brazilian Robotics Competition (CBR) 2024;
- Participated in RoboCup 2024;
- 1st place Latin America Robotics Competition (LARC) 2023;
- Top 8 - Participated in RoboCup 2023;
- 2nd place in Humanoid League Virtual Season 2022/23
- 1st place Latin America Robotics Competition (LARC) 2022;
- Participated in RoboCup 2022;
- 4th place in Humanoid League Virtual Season 2021/22
- 2nd place Latin America Robotics Competition (LARC) 2021;
- Participated in RoboCup 2021;
- 1st place Latin America Robotics Competition (LARC) 2020;
- Top 8 - RoboCup 2019;
- 1st place - Latin America Robotics Competition (LARC) 2019;
- Top 8 - RoboCup 2018;
- 1st place - Latin America Robotics Competition (LARC) 2018;
- Top 16 - RoboCup 2017.

Moreover, we have participated in Bit-Bot's Humanoid League Survey:

<https://robocup.informatik.uni-hamburg.de/en/2022/07/results-of-humanoid-league-survey-2/>

Which approach are you using to generate the robot walking motion?

For walking, we use the ZMP-based algorithms described in [1]. We augment these algorithms with gravity compensation and a torso stabilization controller. Since these techniques strongly rely on dynamics models, we use accurate information from the CAD files to determine the mass properties of each robot's piece. In addition, the team is experimenting with reinforcement learning approaches for walking control, with preliminary results, although these methods are not yet used in the competition codebase.

[1] Marcos R. O. A. Maximo. Omnidirectional ZMP-Based Walking for a Humanoid Robot. Master's thesis, Aeronautics Institute of Technology, 2015.

Which approach are you using to generate other motions of the robot (e.g. kicking, standing up)?

For the motions for standing up, we use keyframes and interpolate the positions linearly in the joint space.

For kicking, we use an approach similar to the one used for walking. We use the ZMP-based algorithms described in [1]. We augment these algorithms with gravity compensation and a torso stabilization controller. Since these techniques strongly rely on dynamics models, we use accurate information from the CAD files to determine the mass properties of each robot's piece. The kicking algorithm is based on cubic splines.

[1] Marcos R. O. A. Maximo. Omnidirectional ZMP-Based Walking for a Humanoid Robot. Master's thesis, Aeronautics Institute of Technology, 2015.

Do you have a kinematic or dynamic model of your robot? If so, how did you create it (e.g. measure physical robot, export from CAD model)?

We have both kinematic and dynamic models. Most of the parameters were obtained from the CAD files through Solidworks. However, some elements were configured in Solidworks based on experiments, e.g. the center of mass of the NUC computer was estimated by experiments. Moreover, some kinematic parameters were initially measured directly on the physical robots,

but there was previously a discrepancy between the physical robot and the CAD model, which has now been corrected, and the CAD is currently synchronized. We have done work on estimating dynamic parameters from experiments using pressure sensors on the feet for the Darwin robot [1], but we still need to replicate this using our current robot (Chape).

[1] Silva, C., Vacarini de Faria, D., Herculano Vasconcelos Barroso, D., Maximo, M.R., & Sandoval Góes, L.C. (2019). THREE-DIMENSIONAL IDENTIFICATION OF A HUMANOID ROBOT. Proceedings of the 25th International Congress of Mechanical Engineering.

What approaches are you using in your robot's visual perception?

Our approach to computer vision is based on a convolutional neural network (CNN) algorithm, that is responsible for detecting the ball, goalposts, and the line intersections. We use a custom YOLOv8-nano model to do those detections. Furthermore, the information that our team uses is made accessible by the team Bit-Bots to train the model. The data is available in [1].

[1] M. Bestmann, et al. TORSO-21 Dataset: Typical Objects in RoboCup Soccer 2021, 2021, https://github.com/bit-bots/TORSO_21_dataset

Are you planning with objects in Cartesian or image space? If you are using Cartesian space, how do you transform between the image space and cartesian space?

We plan with objects in Cartesian space using Inverse Perspective Mapping.

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?

After finding the ball, the robot camera moves to focus on it for a couple of seconds. Then, it scans around to gather information about its position and goes back to focus on the ball. Besides that, the scan pattern is fixed, alternating between some scan heights.

What approach are you using to localize your robot?

To solve the global localization problem, we use a standard particle filter, i. e. Monte Carlo Localization, as described in [1, 2]. The landmark ambiguity makes initializing the filter using a uniform distribution risky, since the filter may converge to the wrong side of the field. Therefore, at the beginning, our algorithm distributes the particles in the field border, as dictated by the rules. Then, resampling is disabled while the head does a 180° scan,

accumulating information from the whole scan in the particles' weights before the first resampling. Parameters were tuned for it to work better, especially the resetting policy using Game Controller states.

[1] W Burgard S., et al. Probabilistic Robotics. MIT Press, 2005.

[2] Alexandre Muzio, et al. Monte Carlo Localization with Field Lines Observations for Simulated Humanoid Robotic Soccer. Latin American Robotics Symposium (LARS), 2016.

Is your team performing team communication? Which communication protocol are you using?

We do not have any form of team communication for the moment.

What approach are you using for navigation? Are you avoiding obstacles?

We use potential fields for path planning. To execute the path, we use some control loops and heuristics.

How is the behavior of your robots structured? (e.g. Behaviour Trees)

Our decision-making is based on a tree of behaviors. The execution begins at a very high-level behavior depending on the robot's role, such as Attack, and goes to lower-level behaviors, such as Position to Kick, until arriving at the behaviors which will request actions to the control module. Some behaviors are modeled as finite-state machines. Our head policy switches between scanning the field for localization features and tracking the ball (when it has been seen). For navigation, we are using potential fields [1]. Since we have been focusing on developing the low skills of our robots in the last years, we still lack some basic mechanisms, such as positioning. We are refactoring our behaviors into a formal open-source framework for behavior trees we developed [2], to maximize the code modularity, organization, and efficiency.

[1] O. Khatib. Real-Time Obstacle Avoidance for Robots.

[2] G. L. Silva, et al. A minimalist open source behavior tree framework in C++. LARS 2021

Are you simulating your robot? If so, which simulator are you using and for what purpose do you use simulations?

We simulate our robot in the simulators Gazebo, Webots, and one that is house-made. Gazebo is mostly used for simulating motion and perception, while Webots are used for simulating matchlike situations, with a focus on behavior and localization. Our custom-

made simulator abstracts away the control and perception system, to facilitate testing the behaviors and localization in a simpler environment. Additionally, the team is currently migrating its simulation framework to MuJoCo, which is already being adopted in the development of the new robot.

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?

We are currently using Ubuntu 20.04 and ROS1 Noetic. However, we are in the process of transitioning to Ubuntu 24.04.

Is there anything else you would like to share that did not fit any previous question?

No.