

Software Survey 2026

Team name

PCMS-HRG

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

Large Size (height < 190 cm, weight < 80 kg)

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

It is not currently open-source as it is in the early stages of development. However, we plan to release our ROS 2-based stack and behavior tree implementations on our GitHub (<https://puiching-fablab.github.io/humanoid/>) following the competition.

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.

Yes, we are developing based on the open-source RoboCup demonstration project.

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

No, we have not yet utilized or developed specific datasets for model training.

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

No, PCMS-HRG is a newly established student team participating in RoboCup for the first time in 2026.

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

We hope to be the first team from Macau, China, in the AdultSize league, and provide a documented ROS 2 workflow for the Booster T1 platform to assist other new student teams entering the league.

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

We primarily use the proprietary learned gait provided by the Booster SDK.

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

We use pre-defined keyframe sequences managed by hierarchical behavior trees (py_trees). These sequences are triggered based on the robot's state (e.g., fall detection via IMU) or game logic (e.g., reaching the ball to kick).

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)?

Yes, we use a kinematic model in URDF format. Since we use the Booster T1 commercial platform, the model is provided by the manufacturer and integrated into our ROS 2 stack for coordinate transformations and sensor fusion.

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

We use YOLO-based multi-class detection (ball, goalposts, landmarks, and robots) optimized with TensorRT to run in real-time on the NVIDIA Jetson AGX Orin.

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 in Cartesian space. We transform detected objects from image-space coordinates to the robot's base frame using the camera's intrinsic parameters and the kinematic transform (head-to-feet). We use stereo depth data and the assumption that objects lie on the ground plane to fix the 3D coordinates.

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

Yes. The robot's head (pan/tilt) is controlled dynamically to track the ball or scan the environment for landmarks when localization confidence is low.

What approach are you using to localize your robot?

Our localization system utilizes a Team Coordinator node that maintains a world state based on detected field features (landmarks, goalposts) and IMU-based odometry.

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

Yes. We use a Team Coordinator node to broadcast high-level role commands (attack/defend) between our three robots. We are working toward full compliance with the standard RoboCup Humanoid League protocol.

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

The robot uses reactive navigation managed by behavior trees. It identifies the target (ball) and avoids detected obstacles (other robots) by adjusting velocity commands sent to the locomotion SDK.

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

We use hierarchical behavior trees (implemented via py_trees) to sequence skills like "search → approach → align → kick."

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

Not yet. We are currently focusing on hardware-first validation and baseline ROS 2 integration.

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

Ubuntu 22.04 and ROS 2 Humble.

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

No.