Software Survey 2026

Team name

Invic

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

Middle Size (height < 125 cm, weight < 25 kg)

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

No

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.

No, we use Booster Studio developed by Accelerated Evolution.

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

Yes, No

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

Not yet, but we will share once we have results.

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

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

Reinforcement Learning Framework like booster_train, booster_assets and booster_deploy

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

Reinforcement Learning Framework like booster_train, booster_assets and booster_deploy

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)?

export from CAD model

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

The model is built based on the YOLOv8 architecture

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?

First, use object detection algorithms to identify objects in image space. Then, transform the detection results into Cartesian space using the camera's intrinsic and extrinsic parameters

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

Yes.
Passive Tracking: When a target is detected, actively adjusts the camera to keep the target centered in the frame.
Active Search: When the target is lost, scans the environment according to preset patterns.
Intelligent Decision-Making: Determines whether to use tracking or search strategies based on whether the target's position is known.
Scene Adaptation: Employs different camera control strategies under varying match conditions.

What approach are you using to localize your robot?

A Multi-Strategy Positioning Method Based on Particle Filtering and Feature Matching

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

Yes, using the UDP network protocol for communication between robots

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

This robotic system features comprehensive navigation and obstacle avoidance capabilities:

Navigation: Utilizes a waypoint-driven navigation system supporting multiple navigation strategies
Obstacle Avoidance: Employs depth cameras for real-time obstacle detection with dynamic avoidance capabilities
Flexibility: Enables or disables obstacle avoidance functionality across diverse scenarios
Safety: Ensures collision prevention through multiple safety distance thresholds

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

Behaviour Trees

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

No, the simulation has not yet been fully developed.

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

This robotic system utilizes ROS 2 Humble as its middleware, running on Ubuntu 22.04. It integrates with various sensors—including cameras and depth sensors—and control nodes to enable the robot's perception, decision-making, and control functions.

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