Software Survey 2026

Team name

ICHIRO ITS

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?

Our software is partially open source and can be found on our GitHub at github.com/ichiro-its.

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.

At the moment, we are experimenting with software developed by Rhoban. Specifically, we are adapting the planning and control library Placo (https://github.com/Rhoban/placo) for implementing LIPM-based walking, and Frasa (https://github.com/rhoban/frasa) for reinforcement learning–based fall recovery.

Both components are still in an early evaluation and reset phase, and we have not yet determined whether they can be fully integrated into our robot for the competition.

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

We use our own datasets for YOLO training, which are currently not public.

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

None

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?

We generate the robot’s walking motion using a sine wave–based gait generation approach.

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

Other motions are generated using manually recorded joint-space keyframes with linear interpolation.

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 have both kinematic and dynamic models of the robot. The models were created from the robot’s CAD design and exported into a URDF file, including link geometries and mass properties.

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

We use a deep learning–based visual perception approach with YOLOv8 (small) for object detection, including the ball, goalposts, robots, and line intersections. In addition, we apply color-based detection using HSV or LAB color spaces.

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 use both approaches. For ball positioning and kicking, we use a combination of image space and the head pan-tilt joint space. For Cartesian space estimation, we apply Inverse Perspective Mapping (IPM) to estimate the distance between the robot and objects detected by YOLO.

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

Yes, we use active vision for ball tracking and scanning by controlling the head pan-tilt based on visual feedback.

What approach are you using to localize your robot?

We use odometry with a defined starting point. Currently, we are researching localization using Monte Carlo Localization (MCL) with vision and IMU data, and we hope to fully implement it for the competition.

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

Yes, we perform team communication using UDP and TCP protocols through our custom C++ communication library.

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

Currently, we do not implement a navigation algorithm or obstacle avoidance.

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

The robot behavior is structured using a rule-based finite state machine, where transitions between behaviors are defined by predefined conditions from perception and game state.

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

We use MuJoCo and PyBullet as simulation environments for walking development and reinforcement learning research.

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

Ubuntu 24.04 and ROS2 Jazzy

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