Software Survey 2026

Team name

RFC-Tsudanuma

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?

Fully opensource, https://github.com/RFC-Tsudanuma

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.

We currently use Booster perception module. But we plan to replace this for our custom model for this competition.

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

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).

No. We are a new team.

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

No.

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

We are currently trying Reinforcement learning and MPC based methods.

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

Reinforcement learning.

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 kinetic and dynamic model which is provided by the manufacturer.

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

We use YOLOv8-based object detection on RGB images and geometry-based 3D position estimation, together with a segmentation model for field line extraction. All perception results are synchronized in the robot’s coordinate frame.

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?

Cartesian space. A 2D detection from the image, such as the center of a bounding box, is back-projected into a 3D viewing ray using the camera intrinsics. The ray is transformed into the robot base frame using calibrated extrinsics, and its intersection with the ground plane is computed to obtain the target’s 3D position.

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

When the ball is not detected, the robot enters a search mode by rotating its head. Conversely, when the ball is visible, the head actively tracks it. Additionally, if the ball is temporarily lost, the robot predicts its position and orients its head toward that direction.

What approach are you using to localize your robot?

We localize primarily with EMCL (an expansion/reset variant of MCL). We also have a separate white□line matching–based self□localization module. Next, we plan to fuse the white□line matching and EMCL estimates through weighted integration.

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

We use UDP and Google Protobuf for team communication.

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

Our robot avoids obstacles. We use the A algorithm for path planning. We set waypoints around obstacles to generate a collision-free path.

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

HTN planner and some optimal method(e.g. Potential method)

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

Using webots, IsaacSim and our own 2D simulator. We use webots and IsaacSim for developing robot motion and strategy. 2D simulator is used for only developing decision making.

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 ROS2 Humble

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

No.