Software Survey 2026

Team name

UT Austin Villa

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?

Not yet

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.

1. Booster Robotics SDK for low-level motor control
2. RoboCup Demo codebase Booster Robotics team developed as a skeleton code.
   - behavior tree for reasoning
   - fine-tuned YOLOv8 model for ball/marker detection

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).

1. Learning Agile Striker Skills for Humanoid Soccer Robots from Noisy Sensory Input

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?

RL policy trained on IssacLab

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

Teacher-Student RL policy trained on IsaacLab

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 the kinematic model in Unified Robot Description Format (URDF). Our robot is a commercial Booster T1 humanoid robot. The kinematic model is publicly available.

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

We use YOLO v8 for ball and marker detection and pose estimation.

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?

Specifically, we build upon Booster Robotics' perception pipeline, which utilizes the RealSense camera's intrinsic and extrinsic parameters to transform image-space pixel coordinates into Cartesian positions in the robot's base frame. The base frame is defined at the midpoint between the two feet, with the x-axis pointing forward, the y-axis pointing left, and the z-axis pointing upward.

In addition, we implemented robust synchronization and filtering system for head joints data, RGB image, and Depth image.

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

We adjust head to make ball always be centered in camera when it detected within 3sec. Otherwise, robot rotate turn in place or move to the center of the field to find the ball.

What approach are you using to localize your robot?

We used a partilce filtering combining the perception and odometry. It is built upon the RoboCup Demo from Booster Robotics and the localization approach open-sourced by BHuman SPL team.

The learned odometry estimation network is trained on our walking policy data.
Likelihood of particles are computed with Hungarian matching between detected markers and field markes (e.g., "L", "X", "T" shapes).

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

We are not performing communication at the moment. We plan to develop team communication with the standard RoboCup Humanoid League protocol, but our immediate short-term plan is to learn an end-to-end strike skill and implement a better perception system.

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

Yes. We are avoiding obstacles with state machine.

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

We use behavior trees to structure the high-level behavior and integrate policies trained with RL.

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

Our major simulation is IsaacSim, which we use to train the RL policy. We also use IsaacSim and MuJoCo for sim-to-sim transfer and Webots for sim-to-real transfer.

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

Ubunutu 22.04, ROS2 humble

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