Software Survey 2026

Team name

HUST-HRT

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?

Our software is partially open-source, and the open-source repository is available at:https://github.com/OWO-OVO-QAQ/hust_hrt.git

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 use the software independently developed by our team.

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

In the course of our research, we have utilized public datasets for part of the work, while the rest of the datasets were collected by our team and remain unpublicized.

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 motions using the reinforcement learning method. We build the training environment based on the open-source Legged Gym framework, iteratively

optimize the walking strategy through multiple rounds of simulation training, and finally deploy the optimal trained strategy to the physical robot to achieve stable autonomous walking.

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

We generate the robot's movements such as kicking and standing up through teach pendant programming. We plan the core motion trajectories via manual teaching, and after system calibration and parameter optimization, the motion data is solidified into executable motion commands for the robot, ensuring the stability and accuracy of the movements.

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 established the kinematic and dynamic models for our robot. The basic model parameters are exported from the CAD 3D model of the robot, and the model parameters are subsequently calibrated and fine-tuned using the measured data of the physical robot to ensure the model accuracy and its matching with the actual motion of the robot.

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

We adopt the YOLOv5 recognition algorithm and leverage TensorRT for acceleration.

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 perform planning in the Cartesian space. We transform image space coordinates to Cartesian space via camera calibration and 3D pose estimation, combining the depth information (from depth camera) and the camera's external parameters (position and orientation relative to the robot base).

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

Yes, our robot can actively adjust the camera's pose based on the on-field elements identified via vision, such as the soccer ball and goalposts.

What approach are you using to localize your robot?

We utilize the triangulation positioning method to realize the robot's localization, which achieves accurate positioning by calculating the robot's spatial position with multi-point sensing and geometric triangulation principles.

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

Our team has implemented inter-robot team communication, and we adopt the ROS Topic communication mechanism to achieve real-time data interaction and command transmission among multiple robots.

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

We adopt the A algorithm* to implement the robot's navigation function, and we also achieve obstacle detection and avoidance based on this algorithm, ensuring the optimality of the navigation path and the safety of movement.

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?

We conduct simulation development and verification for our robot, and use the Isaac Gym and MuJoCo simulators to complete the simulation training of the robot's gaits and various motions. It is mainly applied to the preliminary debugging and effect verification of gait planning and motion control strategies, which greatly improves the efficiency of physical robot motion development and reduces debugging risks.

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

Ubuntu20.04; ROSnoetic

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