Software Survey 2026

Team name

HULKs

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 fully open-source: https://github.com/HULKs/hulk

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

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

We used the B-Human Soccer ball dataset for our ball detection.

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

Planning Steps for Humanoid Robots Using Model Predictive Control

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

Multiple theses finished this year, which can be found on the hulks.de website:
- Exploring Lightweight Data-Driven Methods for Image Segmentation in the RoboCup SPL
- Sim-to-Real Transfer for Locomotion Tasks on Legged Robots: A Survey

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

On the old NAO platform we used a custom walking engine based on the original rUNSWift paper. On the new K1 robot we plan to implement a RL-based approach

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

Right now we use a conventional keyframe based approach.

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 a model provided by BoosterRobotics

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

We used to have a 3-stage ball detector and conventional image processing. Right now we are transitioning to a pipeline based on Yolov12

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 are planning in robot-centric cartesian space. We transform from image to cartesian space using the inverse camera transform.

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

Yes, in certain scenarios locking onto the ball or when we are unsure about localization we search for landmarks

What approach are you using to localize your robot?

We use a EKF and Iterative-Closest Line algorithm on field lines.

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

We are communicating using WLAN

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

We use A* for navigation. We plan paths avoiding obstacles (but assuming they do not move)

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

Large state machine

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

We use MuJoCo for phyical simulations of the robot (checking motion algorithms) and a custom 2d simulator for behavior simulation.

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

We used a custom OS built with Yocto on the Nao. Right now on the K1 we use Ubuntu 22.04 and avoid the ROS2 ecosystem

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

We use a custom Rust-based framework aiming to replace ROS2