

Software Survey 2026

Team name

BigHeroX

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?

Our software is partially open source and can be found at <https://github.com/HNU-BigHeroX/>

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.

Yes, we use components from other teams' software, specifically the Bit-Bots framework (https://github.com/bit-bots/bitbots_main), including the decision layer, localization modules, and related utilities, which we have adapted and optimized for our humanoid platform.

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

Yes, we used our own datasets in the research; they are currently not publicly available, but we may consider releasing them in the future.

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

Since our last RoboCup application, our team members have contributed the following publications:

1. Y. Mao, Y. Wen, H. Chen, M. Liao, "Tungsten-oxide/liquid electrical-double-layer capacitive sensors and EIS characterization for liquid pressure monitoring," IEEE Transactions on Instrumentation and Measurement, 2023, doi:10.1109/TIM.2023.3309364. Application: Monitors hydraulic pressure in soccer robots to ensure stability and accuracy; explores

tungsten oxide for self-powered sensors.

2. Y. Mao, Y. Wen, H. Chen, M. Liao, F. Zhang, "Tungsten Oxide Flow Sensor and its Performance Regulation," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–8, 2022, Art no. 9506608, doi:10.1109/TIM.2022.3177202. Application: Flow sensor for robot hydraulics; performance tuning ensures robot stability and efficiency in competitions.

3. H. Chen, Q. Qu, Y. Lin, X. Chen, K. Li, "Authenticity verification on social data outsourcing," Computers & Security, vol. 100, 102077, 2021. Application: Ensures reliability of external data sources used by soccer robots for decision-making.

4. D. Tang, S. Zhou, H. Jiang, H. Chen, Y. Liu, "Gender-Adversarial Networks for Face Privacy Preserving," IEEE Internet of Things Journal, vol. 9, no. 18, pp. 17568–17576, 2022, doi:10.1109/JIOT.2022.3155878. Application: Protects privacy of spectators and players in soccer matches using GAN-based methods.

5. D. Tang, S. Zhou, M. Luo, H. Chen, H. Tang, "A new discriminative sparse parameter classifier with iterative removal for face recognition," Journal of Central South University, 2022, 29(4), 1226–1238. Application: Enhances robot facial recognition for game scene understanding and player/ball identification.

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

Beyond our participation in RoboCup 2026, we hope to share several contributions with the community: our partially open-source software framework, insights from adapting the DSD decision-layer architecture, experience in low-latency multi-robot communication and hierarchical navigation, high-performance hybrid visual perception techniques, and lessons learned from transitioning from the Middle Size League to the Humanoid League.

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

Reinforcement Learning

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

Mimic-based reinforcement learning, where the reference motions are obtained from human motion capture systems.

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. The model is generated by exporting from the robot's CAD model.

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

We employ a hybrid perception scheme that combines deep learning and traditional computer vision, fully utilizing the computational power of the NVIDIA Jetson Orin. Dynamic objects such as the ball, robots, and goalposts are detected using a quantized YOLOv8 model deployed with TensorRT to achieve real-time performance above 60 FPS. Static field features, especially white lines, are extracted with OpenCV through HSV-based color segmentation, Canny edge detection, and probabilistic Hough transforms, providing stable inputs for localization. GPU-accelerated image undistortion and enhancement are applied to handle illumination variations.

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 object planning in Cartesian space. Using Inverse Perspective Mapping (IPM), image pixel coordinates are transformed into 3D coordinates in the robot body frame. Specifically, the camera extrinsic parameters relative to the ground, including height, pitch, and roll, are obtained in real time via the robot kinematic model (TF2). Combined with the camera intrinsic matrix, a homography matrix is constructed. Assuming an ideal planar field, contact points detected in the image, such as the bottom center of the ball or field line points, are projected onto the 2D ground plane to estimate the target's distance and bearing relative to the robot.

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

We use an FSM-based active vision strategy that tracks the ball with PID-controlled head motion and triggers information-maximizing search patterns when the target is lost.

What approach are you using to localize your robot?

We employ ROS 2-based Monte Carlo Localization, fusing vision landmarks, odometry, and IMU data, with field-symmetry handling via GameController and magnetometer cues, inspired by and refined from the Bit-Bots localization framework.

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

Our team communication adopts a dual-layer architecture combining ROS 2 DDS for reliable, low-latency tactical data exchange and UDP for high-frequency, loss-tolerant coordination, with prioritized data flow and enhanced interference robustness.

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

We implement a hierarchical navigation system where the strategy layer plans optimal paths with A* and dynamic replanning, while the motion layer executes them via Pure Pursuit and `bitbots_quintic_walk`, combining global intelligence with local agility through real-time ROS 2 communication.

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

Our decision layer is built on the Bit-Bots DSD architecture, where a stack-based hybrid of behavior trees and pushdown automata drives role-aware, game-state-dependent actions through a ROS 2-integrated blackboard and modular decisions and actions.

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

Yes. We use MuJoCo during the deployment phase. The primary purpose of simulation is to evaluate the executability and performance of reinforcement learning results before real-robot deployment, serving as a deployment gate. In addition, simulation configurations are used to closely replicate real-world testing conditions, which helps diagnose and analyze issues encountered during hardware experiments.

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

Our robots run Ubuntu 22.04 with ROS 2 Humble, following the official configuration provided by Zhongke YunGu robots.

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

Our decision layer is built on the Bit-Bots DSD architecture, where a stack-based hybrid of behavior trees and pushdown automata drives role-aware, game-state-dependent actions through a ROS 2-integrated blackboard and modular decisions and actions.