

Software Survey 2026

Team name

RedbackBots

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?

At present, the RedbackBots team are fully open source with yearly code releases on our team github page found here:

<https://github.com/rmit-computing-technologies/redbackbots-coderelease>

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.

(copied from TDP)

* Primarily the rUNSWift 2020 code release, for the software architecture, locomotion, vision, etc

* BHuman (and Nao Devils) 2022 code release, for general build configuration scripts, and general robot configuration management scripts

* BHuman 2024 code release for our vision system. Modifications have been made to the BHuman vision

system for compatibility with our code base

* Nao Devils 2024 code release for whistle detection

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

Not at the moment.

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

(copied from TDP)

Sanfilippo, F., Wiley, T., & Rousi, R. (2025). RoboCup Soccer Autonomy Uprising: How Crowds, Referees, and Humanoid Robots are Redefining the Future of Human-Robot Interaction. 2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 1131–1139. doi: 10.1109/hri61500.2025.10974154

Rodriguez, A. I. S., Ellis, T., Killeen, B., Owens, M., Griffiths, S., & Wiley, T. (2025). GameSight: Augmented Reality Visualisation for the RoboCup Soccer SPL. Lecture Notes in Computer Science (in press), pp. 1–12.

Zhen, K. Z. (2025). Learning Soccer Strategies Against Different Opponents in A Simulated Robot Soccer Environment. Master's Minor Thesis. Melbourne, Australia: School of Computing Technologies, RMIT University.

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

(copied from TDP)

Since joining the SPL, and first competing in soccer games in 2023, we have strived to actively contribute to the SPL, and anticipate continuing our contributions through the HSL. We actively believe that the league grows through all teams bringing new features to the league.

Since 2023, we have made the following contributions to the SPL:

- Developed and published GameSight, an open-source Augmented Reality (AR) visualiser of GameController information that overlays live player localisation and game state information over the top of the physical field, compatible with the Meta Quest platform. We have conducted research with GameSight into AR for HRI, with multiple published works (Sanfilippo, Zafar, Wiley, & Zambetta 2025; Rodriguez, Ellis, Killeen, Owens, Griffiths, & Wiley 2025), and a best-paper award at the HRI2025

conference.

- We have published works on the visual referee challenge (Lohani & Wiley 2023; Owens & Wiley 2024) at the RoboCup Symposium. We used this in the 2022 technical challenge, and attempted to use it in 2023 game play, though we had integration issues due to our first participation in the Challenge Shield.
- We support student Master's Minor Thesis and Honours projects (Zhen 2025; Owens 2023; Lohani 2022).
- We developed the 3D ball ramp for consistent testing as part of the 2025 SPL leader-board challenges.
- We contributed multiple software infrastructure components to the SPL. We provided a vision overhaul, integrating and simplifying the B-human vision systems outside their complex software architecture as a standalone module. We developed a simple Python 3 based behaviour decision-tree ideal for undergraduate students to build soccer behaviours. We ported our codebase to a buildable Nao V6 Ubuntu 20.04 image for the SPL, providing an independent codebase to be deployed as an independent image increases the diversity of SPL codebases.
- Since 2023, we have developed our own implementations for Ball detection, Whistle Detection, and visual referee signals, to varying degrees of success.

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

Manufacturer provided Booster K1 DRL walk-engine

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

Not fully implemented, but we are evaluating different options from: pre-computed animation from Booster's libraries, DRL trained kick behaviour, or implementing another team's

open-source kick animation/trained behaviour.

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. Provided with the robot's SDK and software stack.

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

- Vision processing pipeline, including scan line, colour, edge detection, RANSAC.
- Lightweight Neural Network for field element recognition (ball, field lines, robots), using CompiledNN library.
- Ball detection with YOLOV8 with Booster's provided model from their RoboCup Soccer Demo

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 use calculated and calibrated intrinsic and extrinsic image, camera, and world transforms
- These are dynamically calculated from the robot's pose which allows us to estimate a ground plane, which we can intersect with image points

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

- Yes, we move our robot's head actively to search for both features, and the ball, dynamically based on robot behavioural state, and team state.

What approach are you using to localize your robot?

- Mixed modal kalman filter (Sushkov, 2006)

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

- We use a custom constructed UDP packet with dynamic, event based delivery and contents

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

- For navigation to the ball we use custom behaviours to align ourselves with the goals, and typically approach directly with obstacle avoidance

- Vision based obstacle avoidance.

- Previously for the V6 we also used the Sonar to avoid obstacles, we will be looking to implement obstacle avoidance with the Booster K1

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

- We have a custom python based behaviour tree structure

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

- Our team is using Isaac Sim Booster simulation for allowing us to test behaviours, vision, locomotion, etc.

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?

As an executive member comment - this survey is not as agreed, and far too long, duplicating elements of the TDP.