# Software Survey 2026

**Team name**

SABANA HERONS

**Which division(s) are you applying for? If your used software differs between divisions, please fill out the survey once per division.**

Small Size (height < 110 cm, weight < 15 kg)

**Is your software fully or partially OpenSource? If so, where can it be found?**

https://github.com/juanulloa2050/SabanaHerons2026

**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.**

Our software framework is based on the B-Human 2023 code release. However, we have significantly modified several core components to suit our specific research goals. Specifically, we have reconstructed the entire whistle detection module and developed a new gesture recognition system. We are also currently replacing the strategic positioning behavior with a custom solution.

List of Reused Components:

- Component: Base Framework & Infrastructure (Communication, Debugging, Logging)
  - Original Developer: Team B-Human
- Component: Motion Engine (Walking Engine & Kick Engine)
  - Original Developer: Team B-Human
- Component: Visual Perception (Ball, Field Lines, and Robot Detection)
  - Original Developer: Team B-Human
- Component: Self-Localization Module
  - Original Developer: Team B-Human

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

Our software leverages two custom datasets that we collected and curated in-house: one for whistle recognition and another for visual gesture recognition. We are currently organizing and

refining both datasets (e.g., cleaning, labeling validation, and documentation) prior to making them public.

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

Our research trajectory is profoundly shaped by the humanistic ethos of Universidad de La Sabana, where technology is viewed not merely as an industrial tool, but as a vehicle for social inclusion and the preservation of human dignity. This institutional philosophy has led our team to focus heavily on Human-Robot Interaction (HRI) and socially assistive robotics, specifically addressing the complex challenge of automating Colombian Sign Language for the hearing impaired.

However, we posit that this focus is inextricably linked to the core technical challenges of the RoboCup Standard Platform League. The engineering constraints required to emulate the fluidity of human communication—precise Inverse Kinematics for upper-body gestures, robust balance control during complex center-of-mass shifts, and real-time computer vision for user recognition—are isomorphic to the skills required for a high-performance soccer athlete. By rigorously solving these problems in the domain of social robotics, we have developed a robust technological foundation that we are now transposing onto the soccer field. Our published work represents the validation of these core competencies, demonstrating that our algorithms are not only theoretically sound but proven in real-world, interactive scenarios. Thus, the following publications showcase the technical maturity we bring to the competition:

Castro-Murcia, H., Garzón-Castro, C., & Castellanos-Rivillas, J. (2025). Comparison of Neural Networks with a Numerical Method for Obtaining the Inverse Kinematic of a Bipedal Robot. IEEE Conference on Applications of Computational Intelligence (ColCACI 2025), Armenia, Colombia. https://doi.org/10.1109/ColCACI67437.2025.11230902

Mora-Zarate, J. E., Garzón-Castro, C. L., & Castellanos-Rivillas, J. A. (2025). LSC-54: A landmark-based dataset for Colombian Sign Language. Data in Brief, 63, 112145. https://doi.org/10.1016/j.dib.2025.112145

Mora-Zarate, J. E., Garzón-Castro, C. L., & Castellanos-Rivillas, J. A. (2024). Learning signs with NAO: Humanoid robot as a tool for helping to learn Colombian Sign Language. Frontiers in

Robotics and AI, 11, 1475069. http://doi.org/10.3389/frobt.2024.1475069

Mora-Zarate, J., Garzón-Castro, C. L., & Castellanos-Rivillas, J. (2024). Construction and Evaluation of a Dynamic Sign Dataset for the Colombian Sign Language. 2024 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Bogotá D.C., Colombia, 1-5. https://doi.org/10.1109/LA-CCI62337.2024.10814896

**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 utilize the B-Human walking engine, relying on a model-based gait to generate the robot's walking motion. The walk is produced by a parameterized kinematic model that defines the alternating support/swing phases, the swing-foot trajectories, and the body/COM shift required to keep the ZMP/ground reaction forces inside the support polygon. The robot performs alternating single-support steps where the body shifts laterally to unload the swing leg and ensure foot clearance, while the step timing and phase progression are governed by an internal walk-cycle controller.
To make the motion robust to real contact dynamics, we use the foot pressure sensors (FSRs) for touch-down detection and phase synchronization: measured load distribution is used to confirm/support the transition between swing and support and to correct cycle timing when contact happens earlier/later than predicted. For stabilization, we use IMU feedback, specifically a low-pass filtered pitch-rate gyroscope signal, which is fed back primarily into ankle pitch modulation (and associated posture compensation) to reject disturbances and dampen oscillations during single support. In addition, the gait incorporates dedicated stabilization/compensation actions (e.g., posture and foot-placement adjustments within the model limits) to improve balance recovery under perturbations and uneven contact conditions.

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

We use the b human 2023 non-walking motions framework that is based on the MotionPhase

concept, where the behavior layer issues a MotionRequest and the central engine selects the appropriate generator to execute the specific motion phase. Regarding kicking, we employ two complementary approaches: stand kicks (KickEngine), which utilize parameterized Bezier-curve trajectories maintained by basic balance controllers, and in-walk kicks (WalkKickEngine), which translate relative ball positions into step sizes to execute the kick fluently within the walking cycle by interpolating between multiple sub-targets. Finally, for recovery motions and special postures, we rely on the KeyframeMotionEngine, a module that manages joint-space interpolation between pre-defined keyframes, enhanced by state-machine execution, joint stiffness compensation, and PID-based balancing to improve stability under external disturbances.

### 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 maintain the kinematic model of B human 2023 which is defined by the nominal link geometry and joint chains, which is utilized for forward and inverse kinematics as well as for computing critical internal transformations, such as the TorsoMatrix and CameraMatrix used in perception and projection.
In simulation, we additionally employ a dynamic rigid-body physics model from B-Human's SimRobot, where the Open Dynamics Engine (ODE) manages physical interactions and collision detection.

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

We utilize the perception pipeline based on the B-Human 2023 framework, which synergizes classical computer vision techniques with deep learning architectures to construct a robust environmental model. Pre-processing operates on raw YCbCr imagery, converting it into the ECImage format to facilitate YHS decomposition for efficient color and saturation analysis. Deep learning modules, implemented via CNNs and the CNS framework, are applied for ball recognition and contour extraction. These CNN models serve to validate candidates generated by classical methods—such as scan-line analysis—enabling the precise detection of field lines, intersections, and the center circle. Furthermore, the field boundary is delineated by a specialized deep neural network, while robot detection relies on the YOLOv2 algorithm to accurately identify teammates and opponents.

**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 conduct object planning and localization strictly in Cartesian space, utilizing the projection geometry provided by the B-Human 2023 framework. The transformation from image space to Cartesian coordinates is performed via inverse projection: a vector is constructed from the camera's optical center to the target pixel and rotated into the world coordinate system using the camera matrix to find its intersection with the ground plane. For non-ground objects, the system supports projection onto a specific Z-height plane. Conversely, we transform 3D relative positions back to image space by applying the inverse camera transformation followed by perspective projection using the camera's intrinsic parameters. The camera matrix is dynamically updated by combining the torso's pose relative to the ground (derived from inertial sensors and leg kinematics), the kinematic chain from the hip to the camera, and extrinsic calibration corrections. To ensure accuracy, these transformations are treated as valid only when the robot is determined to be stable and upright, preventing erroneous projections during falls or unstable motion.

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

Yes, we employ an active vision strategy integrated into the B-Human 2023 framework that operates on both sensor and motor levels. At the sensor level, the system dynamically optimizes image acquisition using an exposure weighting scheme that prioritizes relevant field areas—such as the expected ball position—while down-weighting distant regions or the robot's own body. Mechanically, the robot actively manages its gaze direction to maximize information gain; for instance, during autonomous camera calibration, it scans for goal-area field lines and repositions its head or body if reliable features are not detected. This active gaze control extends to task-driven scenarios like the visual referee challenge, where the robot uses self-localization to direct its attention toward the expected referee position, utilizing neural networks to detect keypoints and recognize gestures, and adjusting its perspective if the current view hinders detection.

**What approach are you using to localize your robot?**

We employ the self-localization approach provided by the B-Human 2023 framework, which

utilizes probabilistic state estimation. The system tracks robot pose hypotheses (2D position and orientation) using Unscented Kalman Filters (UKFs), while a complementary lightweight particle filter manages multiple hypotheses to facilitate recovery from localization errors. This estimation is continuously corrected using visual measurements derived from perceived field features, such as field lines, penalty marks, and the center circle.

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

Yes, our team actively performs team communication using the UDP broadcast protocol. Leveraging the B-Human 2023 framework, our communication infrastructure is managed within the Cognition thread by the TeamMessageHandler module. This system generates and processes messages using a compact custom format, allowing robots to share state data and strategic decisions efficiently in real time.

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

We utilize the B-Human 2023 navigation framework, which combines input from the vision module with high-level behavior decisions to generate a desired walking vector that is continuously modified in real time to avoid obstacles. The system features three distinct avoidance behaviors: robot-robot avoidance, where the robot actively sidesteps both teammates and opponents; ball-aware avoidance, ensuring the robot navigates around the ball to prevent accidental contact or obstruction; and priority-based interactions. In this hierarchy, navigation adapts to the role of nearby teammates: if a teammate holds higher priority (such as the active ball controller), the robot yields to clear the path, whereas high-priority robots are permitted to maintain their trajectory with the expectation that others will adjust.

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

The robots' behavior utilizes the B-Human 2023 framework, structured as a hierarchical, four-layer decision architecture evaluated continuously in real time. At the highest level, the Strategy layer acts as a state machine that selects the global team formation based on game-state conditions, such as ball position, the number of active players, and temporal stability. Once a strategy is active, the Tactic layer defines the spatial configuration, assigning target field positions to each player and managing automatic reassignment via Voronoi regions if players are missing. Based on these assignments, the Role layer specifies the immediate

objective for each robot, combining a static positional role (goalkeeper, defender, midfielder, forward) with a dynamic active role (e.g., chasing the ball, supporting, or defending). Finally, the Skill layer implements how these actions are executed at the motor level, translating role decisions into concrete motion commands like walking, passing, or shooting. All layers are evaluated every control cycle (approximately 33 ms), ensuring coordinated, adaptive, and reactive team behavior while maintaining a clean separation between strategic decisions, tactical layout, individual logic, and low-level execution.

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

We utilize B-Human's SimRobot primarily as a pre-deployment validation tool, allowing us to verify the behavioral logic and stability of our NAOs in a physics-based environment before executing code on physical hardware. Furthermore, we are currently developing a specialized simulation environment tailored for our Reinforcement Learning research. This custom simulator is designed to train and optimize our dynamic positioning agents, accelerating the learning process for high-level strategic decisions.

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

The robots run on Ubuntu 22.04 LTS and use LoLA (Low Level Abstraction) as their middleware.

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