

# Software Survey 2026

## Team name

Ruhrbot Devils

**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?**

Fully open source after the annual Code Release Deadline:

<https://github.com/NaoDevils/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.**

B-Human (SPL): Our Framework based on their 2015 Code Release

Berlin United (SPL): Unscented Kalman Filter for IMU filtering (integrated since 2018)

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

Public Datasets:

- Audio Datasets: <http://ruhrbot-devils.de/#datasets>

- ImageTagger annotations: <https://imageragger.bit-bots.de/users/team/28/>

Private Datasets (hosted on internal CVAT instance):

- Referee Pose Detection (NAO)

- Object Detection (NAO + K1)

- Semantic Segmentation (K1)

- Synthetic Depth Images (K1)

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

All published in RoboCup 2025 – Robot World Cup XXVIII:

- D. Brämer, D. Kleingarn, P. Reichenberg: "Multimodal Neural Network Approach for Real-Time

Fault Detection, Isolation and Recovery on a NAO Robot”

- T. Klute: “Performance Measurement and Software Quality Assurance in the RoboCup Standard Platform League: A Qualitative Interview Study”

- A. Moos: “Attention on the Pitch: Ultra-Light Vision Transformers Bring Global Context to Embedded Robot Vision”

- P. Brömmel, D. Brämer, O. Urbann, D. Kleingarn: “Robot Localization Using a Learned Keypoint Detector and Descriptor with a Floor Camera and a Feature Rich Industrial Floor”

- D. Brämer, D. Kleingarn, O. Urbann: “Graph-based Robot Localization Using a Graph Neural Network with a Floor Camera and a Feature Rich Industrial Floor”

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

Audio Processing Toolbox: <https://github.com/NaoDevils/AudioProcessing>

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

NAO: Flexible Linear Inverted Pendulum Model (FLIPM) with Linear Quadratic Regulator (LQR)

K1: End-to-end Reinforcement Learning (trained in MuJoCo Lab and Isaac Lab)

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

NAO: Keyframe-based motion sequences

K1: Reinforcement Learning combined with motion capture trajectory tracking

### **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)?**

NAO: Analytical dynamic model derived from Aldebaran assets

K1: Optimizer-based dynamic model derived from Booster assets robot specifications

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

NAO: Two-stage pipeline with hypothesis generation (heavily downscaled YOLOv3, scanlines, modeling) followed by patch-based classification (and position regression) using custom CNNs/Vision Transformer per object class

K1: Hypothesis generation via custom mobile-optimized CNNs and Vision Transformers for

object detection/segmentation; stereo-camera-based depth estimation; patch-based verification via custom classifiers

**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?**

Exclusively Cartesian space

NAO: Projection via camera matrix (pinhole model with known extrinsics)

K1: Combined camera matrix projection and stereo depth image triangulation

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

Points of Interest (POI) controlled head movement. POIs (ball, field lines, other robots) are prioritized and systematically tracked based on task relevance and uncertainty.

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

Multi-Hypothesis Kalman Filter: hybrid approach combining Kalman Filter state estimation with particle filter hypothesis management for robust multi-modal pose tracking

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

SPL-compliant UDP broadcast protocol; limited to 1200 messages per game at 128 bytes each

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

Custom path planning with dynamic obstacle avoidance; trajectories are optimized to maintain visibility of relevant POIs during motion

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

Hybrid architecture:

- Score-based behavior selection using utility functions (e.g., heat maps for opponent positions, pass evaluation)
- High-level state machines implemented in C++ Abstract Behavior Specification Language (CABSL) for (game) state transitions

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

SimRobot (ODE/MuJoCo physics backends): high-level behavior testing and debugging

K1 additionally:

- Isaac Lab / MuJoCo Lab: Reinforcement Learning policy training
- Webots / Unity: synthetic image data generation for vision training
- Blender: Retargeter for Imitation Learning Data Generation

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

- NAO: Custom Ubuntu 22.04 image (<https://github.com/NaoDeviils/Naolmage>), no middleware
- K1: Ubuntu 22.04 with ROS 2 Humble

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