Software Survey 2026

Team name

WF Wolves

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?

The software is under open source license, access will be granted if requested. Our Codebase is not hosted publicly.

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.

We are using implementations of the Hamburg Bit-Bots and Rhoban:
- gamecontroller package (developed by the Hamburg Bit-Bots)
- RhIO (developed by Rhoban)

Our dxl-board is based on bitbots_quaddxl from the Hamburg Bit-Bots.

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

We are using custom datasets as well as some public datasets from the Hamburg Bit-Bots image tagger (https://imagetagger.bit-bots.de).

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

None

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?

Last competition we used an open loop pattern generator, combined with an active controller for stabilization. We are currently working on a reinforcement learning based approach.

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

Keyframe-Animations

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 both a kinematic and a dynamic model. Our models are generated from fusion CAD-models, using the syuntoku14/fusion2urdf package.

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

Our robots visual-perception is done using a self-trained YOLOv7 network.

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?

Our planning is happening in cartesian space. The transform between image space and cartesion space is done by a transformer based on the pinhole-camera model.

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

Robots are tracking the ball with their head.

What approach are you using to localize your robot?

Most of our planning is done without a global localization. An approach based on a particle filter is currently being developed and tested.

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

Our robots are using a custom build protocol for team communication. Currently we only share position of points of interest on the field, as well as the current strategy.

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

We are using a basic movement planner. No obstacle avoidance is done at the moment.

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

The behavior of our robot is structured as a state-machine.

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

pybullet (animation and collision testing), mujoco (reinforcement learning)

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

Ubuntu 20.04 with ROS 1 noetic

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