

Software Survey 2026

Team name

SPQR

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?

the code will be fully opensource. At the moment the simulator is under construction and can be found here: <https://github.com/SPQRTeam/> under the name circus. The rest of the codebase is currently in a early development stage

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.

no

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

<https://spqr.diag.uniroma1.it/datasets>

there are more, we are updating the new website with them

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

https://scholar.google.it/citations?view_op=view_citation&hl=it&user=Y8LuLfoAAAAJ&cstart=20&pagesize=80&citation_for_view=Y8LuLfoAAAAJ:mVmsd5A6BfQC

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

We organize a tournament in Rome with 3 teams using the booster robots, and we provided some notes at the end of it, that maybe can be useful for the community:

<https://github.com/SPQRTeam/RoboCup-Humanoid-Community>

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

SPQR is developing a gait generation system based on Reinforcement Learning and Model Predictive Control (MPC). The goal is to leverage a hybrid learning/model-based approach to capture strengths from both sides and at the same time mitigate issues arising from both approaches, if used in isolation. We are adopting a hierarchical RL setting, where the MPC controller informs the low-level RL controller about feasible COM trajectories. This approach allows for real-time optimization of the robot's walking trajectory by predicting future states and adjusting control inputs accordingly. The robot is modeled as a linear inverted pendulum, thus we may identify the center of mass (CoM) and the zero-moment point (ZMP) as its main components. The latter, which captures the point where reaction forces at the contacts between the feet and the ground do not produce any moment in the horizontal direction, must always lie inside a stability region, called support polygon. By using MPC, we are able to enforce this constraint while tracking a desired trajectory. Practically speaking, the MPC controller generates a trajectory for the ZMP, based on the desired velocity and the planned footsteps, which then needs to be tracked by the CoM. To ensure the ZMP trajectory never diverges, we enforce the so-called stability constraint, which basically enforces a terminal constraint on the MPC horizon in order to stabilize the unstable component of the LIP dynamics. We formulate all MPC problems (footstep planner and controller) as QP problems. We also adopt heavy domain randomization through terrain variability, sensor noise etc., in order to deploy our walking policies on the Booster T1 and use it during our games.

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

We are exploring some imitation learning but we don't know what will be the final architecture for robocup 2026

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 export from the models available online for the booster robots

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

we are going to use Yolo on a custom dataset. At the moment we are not sure about the size of the model.

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 would like to plan in cartesian space and calibrating the robot in order to go back and forth between the transformations

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

not at the moment

What approach are you using to localize your robot?

particle filter

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

UDP

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

at the moment we do not have an obstacle avoidance system yet. We will have it for the competitions

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

we are currently using the approach based on behavior trees

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

we are building a simulatore (Circus), based on Mujoco. The simulator is open source and can be found in our github page <https://github.com/SPQRTeam/>

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 with ros2 humble

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