Software Survey 2026

Team name

Colmillos ITAM

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?

Not yet

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.

Yes, we are using the walking module from rUNSWift team

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

Yes, they are not public yet

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

Not in the last two years

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

As a new team we hope to have contributions in the future

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

we are using the walking module for the NAO robot from rUNSWift team.

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

We are using the motion engine from Naoqi that translates motions into joint angles by using trajectories.

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 are using the stantadard parameters and the CAD for the Nao robots.

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

We are working in the HSV space at QVGA resolution from any of the two cameras (as determined by the Decision module). We rely on scan-lines-based image processing algorithms for the color segmentation and we perform the object detection among the line segments found along these scan-lines.

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 both aproaches. For the tranformation to cartesians space we compute the homography using the angle of the camera and a previous camera calibration.

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

Yes, we track the lines of the field, the ball and the goals when it is possible and based on that we move the camera.

What approach are you using to localize your robot?

The localization algorithm is based on a classical particle filtering strategy, i.e. that maintains a probabilistic, sample-based representation of the robot position and orientation

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

Not yet

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

The motion planning strategy for the team is to implement a local motion planner that uses the

Nearness Diagram (ND) to avoid other robots in the field and navigate through them toward the ball or the goal.

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

We define a set of roles that the players can achieve during the game, each role is assigned a function to dynamically compute a set of candidate goals for state and environment variables relevant to achieve it. For example, one of the roles could be to intercept the ball, the function uses the estimation of the position of the ball and self-localization to determine a desired position for the robot. In addition, using Composable Controllers we can arbitrate among the available roles.

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

Yes, we use the Webots enviroment for the Robot NAO.

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

Currenly we are using Ubuntu 22.04 and a Docker to run Unbuntu 16.04 with Naoqi and ROS Kinetic.

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