Software Survey 2025

Team Name

PUMAS

Is your software fully or partially OpenSource. If so, where can it be found:

Our software is fully OpenSource avaible at: https://github.com/mnegretev/Humanoids/tree/master

Do you have a kinematic or dynamic model of your robot(s)? If so, how did you create it (e.g. measure physical robot, export from CAD model)?

We have only a kinematic i n URDF provided by the manufacturer.

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?

We use inverse kinematics with pseudo inverse jacobian

Are you simulating your robot? If so what are you using simulation for?

We use gazebo simulator, the simulator is used to simulate the dynamics of the robot before

testing it on the real robot

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

We use an open loop control by the inverted pendulum approach.

What approach are you using to generate motions for standing up?

We use an approximation for predefined positions.

What approach are you using to generate kicking motions?

We use an open loop control by the inverted pendulum approach.

Do you use any other motions than the previously mentioned? If so, what approaches are you using to generate them?

No

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

We don’t use datasets.

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

We have implemented computer vision algorithms to enhance environmental perception and

interaction. Gaussian filters are applied to smooth images and reduce noise. Then, the Hough CircleTransform is utilized to identify circular shapes, aiding in the recognition of specific objects. By converting images to the HSV color space, we improve object detection under varying lighting conditions and significantly reduces computing needs. Additionally, a Proportional (P) control system adjusts actuator responses based on positional errors, ensuring stability and precision in the robot’s movements. The integration of these algorithms significantly enhances the robot’s ability to interpret and respond to visual stimuli in dynamic environments.

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 Cartesian coordinates making the transformation between the height of the robot,

the angle of the camera obtained by the centroid of the ball in coordinates of imaged with

trigonometric equations.

How is your robot localizing?

Landmark base localization with extended Kalman filter.

Is your robot planning a path for navigation? Is it avoiding obstacles? How is the plan executed by the robot (e.g. dynamic window approach)?

No

How is the behavior of your robot's structured (e.g. Behavior Trees)? What additional approaches are you using?

We don't use.

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

We center the head pan and tilt to the ball based on the real ball position.

Do you apply some form of filtering on the detected objects (e. g. Kalman filter for ball position)?

We filter the image by changing color spaces, Hough circles, Hough transform, Hu moments filtering and Gaussian filters for ball and goal detection.

Is your team performing team communication? Are you using the standard RoboCup Humanoid League protocol? If not, why (e.g. it is missing something you need)?

They don’t communicate they have predefined possitions.

Please list contributions your team has made to RoboCup

It is our first participation

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

NA

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.

NA

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

We are using ROS noetic on Ubuntu 20.04.

Is there anything else you would like to share that did not fit to the previous questions?

If you have additional materials you would like to show, please link to them here.