RoboCup 2023 Submission Survey

Survey response 1

Software

Team Name
Bold Hearts

Is your software fully or partially OpenSource. If so, where can it be found:
Our code is partially open source. We open source the code when it is entirely stable and tested in competitions. Software can be found: https://gitlab.com/boldhearts/

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 measured our physical robot.

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?
The IKWalk library is a C++ implementation of an open loop walk engine for small and mid-size humanoid robots. Motor target positions are generated online based on splines in Cartesian space and inverse kinematics. No ZMP is used but several parameters have to be manually tuned.

Are you simulating your robot? If so what are you using simulation for?
Yes, we have a simulation for Gazebo and recently added a model and libraries for the simulation in Webots.

What approach are you using to generate the robot walking motion?
We use the IKWalk library with a ROS2 wrapper.

What approach are you using to generate motions for standing up?
We use manual key frame animations.

What approach are you using to generate kicking motions?
We use manual key frame animations.

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

Which datasets are you using in your research? If you are using your own datasets, are they public?
We use a combination of data collected during RC competition (currently not released), and dataset created by the Hamburg Bit-Bots and built by the RC community.

What approaches are you using in your robot's visual perception?
We wrap the Darknet framework as a ROS2 node and use the xYOLO network developed by the Electric Sheep team. Additionally, we have improved their image pre-processing and network detection accuracy for a greater number of object detections.

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 currently use image space, but will move towards Cartesian space with the addition of localisation.

How is your robot localizing?
We are currently implementing an appearance based mapping techniques for localization (citations in the pdf attached). This technique uses visual features in the environment and odometry to detect whether the robot is revisiting any of these features, referred to as closures. Once a closure is accepted, it is added as a constraint to the map of the environment. We plan to incorporate a variation of RTAB-map.

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)?
It is not currently. However, we are investigating possible techniques, such as image-based dynamic window approaches to obstacle avoidance.

How is the behavior of your robot's structured (e.g. Behavior Trees)? What additional approaches are you using?
Robots' behaviours are currently controlled by a simple Behaviour Tree.

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?
Not currently.

Do you apply some form of filtering on the detected objects (e. g. Kalman filter for ball position)?
Not currently.

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)?
Since we are using ROS 2 to develop our software, we decided to make full use of the functionalities provided by this framework. In particular, ROS 2 supports the quality of service (QoS) policy that allows us to tune the communication between nodes. For the team communication, we use the ROS 2 efficient non-blocking "best-effort", "UDP-like" and "TCP-like" policy.

Please list contributions your team has made to RoboCup
Since 2019, we decided to open our ROS 2 source code to the community, and since last year we opened also framework developed by us based on the Darwin-OP platform. Since last year, we also started annual RC hackathon events for students in Engineering and Computer Science for robot design and software solutions. In order to attract new members, we created and entirely designed a robotic module at the University of Hertfordshire that uses RoboCup as use-case for robotic applications.

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).
The team promoted and also actively participated in editing and contributing to the unique journal project, titled "The human in the loop: perspectives and challenges for robots' behaviours in RoboCup 2050" which it has currently submitted for publication.

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.
We are currently using IKWalk Engine from Rhoban team, and our vision is a modified version of the one developed my Electric Sheep.

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?
We run Ubuntu 22.10 and ROS 2 humble.

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

If you have a description document of your software you would like to share, you may do so here.

filecount - If you have a description document of your software you would like to share, you may do so here.
0