Survey response 17

Software

Team Name
WF Wolves

Is your software fully or partially OpenSource. If so, where can it be found:
No

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)?
Yes, Robotmodels exist in Fusion are and are exported to URDF via fusion2urdf (https://github.com/syuntoku14/fusion2urdf)

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?
Not anymore (used in old walker - analytic); New Walker uses CPGs

Are you simulating your robot? If so what are you using simulation for?
Yes, current simulation approaches exist in pybullet.

What approach are you using to generate the robot walking motion?
Current Walking Gait is generated by Central Pattern Generators (CPGs)

What approach are you using to generate motions for standing up?
Joint-Space Keyframe Animations

What approach are you using to generate kicking motions?
Dynamic Kick-Engine developed by Krebs, O.: Closed-Loop Kinematic Scheme for Kicking Motion of Humanoid Soccer Robots - 2015. Direction, Strength, Foot height is individually calculated based on current foot position to target position.

Do you use any other motions than the previously mentioned? If so, what approaches are you using to generate them?
None

Which datasets are you using in your research? If you are using your own datasets, are they public?
Current only Datasets are used in Vision. Training relies on private Imagedatasets from previous RoboCup Events on Imagetagger.

What approaches are you using in your robot's visual perception?
current Implementation utilizes YOLOv7 on above mentioned Datasets. Additionally trained for balls and goal posts.

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?
Image Transform is used for field lines in localization via OpenCV. Other transformations are solved via linear algrebra.

How is your robot localizing?
Early approaches using ROS internal AMCL exist: Lorenz, T.: Visuelle Lokalisierung eines humanoiden Fußballroboters anhand von Feldlinien. - 2016 Currently working on enhancing the ROS navigation stack by additional field highlights.

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)?
Motion planning is simplistic. Local and Global motion planners from ROS will be evaluated. Current focus is directed onto a stable walking algorithm.

How is the behavior of your robot's structured (e.g. Behavior Trees)? What additional approaches are you using?
Behavior is based on a state machine utilising FlexBe Behavior Engine (https://wiki.ros.org/flexbe)

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?
Yes, ball is focused during play.

Do you apply some form of filtering on the detected objects (e. g. Kalman filter for ball position)?
Local robot models contain ball position. These were filtered due to old and unreliable ball detection. Current vision renders filtering obsolete (until now).

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)?
MiteCom protocol is implemented but not utilized fully yet.

Please list contributions your team has made to RoboCup
See System Descritption Paper please. Kick-Engine; common League ROS-Framework together with Bit-Bots, etc.

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 (except Bachelor- & Masterthesis); Current Paper for Walking-Algorithm

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.
Robots are based on old Nimbro-Op Model. Common ROS-interfaces are shared with Bit-bots (humanoid_league_msgs); GameController integration was early developed together with bit-bots

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?
Intel Nuc - Ubuntu 20.04 with ROS Noetic; Jetson Tx2 - Ubuntu 18.04; ROS Melodic / Noetic (Docker)

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