Survey response 13

Software

Team Name

UTRA

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

Fully OpenSource https://github.com/utra-robosoccer/soccerbot

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. All main dimensions were exported from CAD model

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

Yes, Analytic solution based on the CAD model

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

We use simulation to enhance our workflow to minimise the time development takes. Trying to get the simulation as close to the real life version. So a lot of sensor modelling, localisation, computer vision and walking engine testing to at least give us a starting point for tuning and testing in real life.

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

We use Dynamically tuned splines curves. Trajectories are determined using inverse kinematics. First, a fixed trajectory for the robot’s body was generated. Using this trajectory for the body, two footstep trajectories were determined alongside the robot body to move the robot from point A to point B. The locations of the footsteps were then calculated. Inverse kinematics was calculated to move the robot whilst maintaining balance. A multi-state PID controller using the yaw and pitch angles of the robot as feedback was used to stabilise the movement

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

Static key-frame animations

What approach are you using to generate kicking motions?

Static key-frame animations

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

Tried using a walking engine developed through RL but have not been able to successfully transfer into the physical domain

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

Bitbot's torso and ball dataset

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

Neural network (yolov5) + geometry for ball, opencv for field lines.
Information is obtained from the images acquired by the camera through a series of filtering stages in the computer vision system. The filters are as follows:

– Field Detection - A color filter supported by a color space estimator (which obtains the color of the grass) to get the area of the field. The results can be seen in figure 2.
– Field Line Detection - From the field area message we detect straight lines and find the individual intersections of all those field lines using Hough Lines. The results can be seen in figure 2.
– Pole Detection - The pole is detected using the Hough Line Transform. A high threshold value is set to ignore the vertical net lines and hence to only detect the vertical pole lines.
– Ball Detection - A FCNN trained with Pytorch locates the ball and sends its coordinates. The results can be seen in figure 1.
– Robot/ Obstacle Detection - A FCNN trained with Pytorch locates obstacles and sends its coordinates.

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?

Cartesian space. For transformation from image to Cartesian space use a simple 2D to 3D camera transformation and assume all objects are all the ground.

How is your robot localizing?

Using a UKF that combines measurements from robot movement odometry and visual odometry calculated using field-lines and an ICP algorithm

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)?

The path is currently a turn, move forward turn, for short paths and for long paths use a bezier curve. It is not avoiding obstacles. The entire path is executed by the robot in one go until it is cancelled by the strategy

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

The behaviour uses py_trees https://github.com/splintered-reality/py_trees

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

The vision is mostly looking for the ball and following the ball when it sees it.

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

No

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)?

Yes, and Somewhat, it is derived from the standard protocol but a lot of information is removed because it is not necessary

Please list contributions your team has made to RoboCup

N/A

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

N/A

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

N/A

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

Ubuntu 20.04, ROS1 Noetic

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