Survey response 25

Software

Team Name
TH-MOS (Kid Size)

Is your software fully or partially OpenSource. If so, where can it be found:
Our software is partially OpenSource. It can be found at https://github.com/MosHumanoid/bitbots_thmos_meta

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 designed our robot model in Solidworks. Therefore, we get the corresponding kinematic and dynamic model from solidworks model.

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?
We are using inverse kinematics and typically the closed-form solution is used.

Are you simulating your robot? If so what are you using simulation for?
We are using Webots to simulate our robot. We use it for conducting experiment of new algorithm and test motion stability in simulation.

What approach are you using to generate the robot walking motion?
We are using an open-loop walking pattern by trajectory adjustment to generate the robot walking motion.

What approach are you using to generate motions for standing up?
We are using animation design based on interpolation to generate the motions for standing up.

What approach are you using to generate kicking motions?
We are using animation design based on interpolation to generate the kicking motions.

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

Which datasets are you using in your research? If you are using your own datasets, are they public?
We are using our own datasets, and they haven't been made public yet.

What approaches are you using in your robot's visual perception?
Our vision algorithm for perception is based on OpenCV library of computer vision, using deep learning method and training through artificial neural network.

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 the camera matrix, including intrinsic and extrinsic matrix to perform transformation between the image space and cartesian space.

How is your robot localizing?
Our robot localizing is implemented majorly through the scan matching and probabilistic model. Detailed algorithm includes particle filter, Iterative Closest Point, etc.

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)?
Our robot employs the ros-navigation package in order to implement path planning and obstacle avoidance. A combination of A* algorithm and dynamic window approach is also employed.

How is the behavior of your robot's structured (e.g. Behavior Trees)? What additional approaches are you using?
Inspired by the hierarchical state machine (HSM) programmed in XABSL, we introduce HSM into our algorithms utilized to generate behavior.

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?
Our camera would make corresponding action according to the behavior model, including changing the yaw and pitch of the camera.

Do you apply some form of filtering on the detected objects (e. g. Kalman filter for ball position)?
Our vision model uses deep learning method and training through artificial neural network, while particle filter is employed in the localization model.

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)?
We haven't achieved team communication right now, but hopefully we will implement the multi-agent reinforcement learning for team communication, which we have already carried out some relative research in simulation.

Please list contributions your team has made to RoboCup
Ranked 3 in Robocup Asia-Pacific 2019 Ranked 4 in Robocup 2023

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).
To achieve robust robot locomotion more efficiently, bipedal walking algorithms based on deep reinforcement learning methods are highly desired. To this end, we proposed to design an adaptive reward function for imitation learning from the references and used Adaptive Mimic algorithm to train the agent to achieve a fast and stable walking pattern from an infeasible reference. This work was published on https://arxiv.org/abs/2112.03735.

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.
Our hardware designs are made on our own, while our approaches of walking refers the quintic walk algorithm of team Bitbots, which uses a large number of parameters to generate the center of mass and the origin of coordinates of feet, to obtain the current position and the target position of the next frame and uses the interpolation algorithm of quintic polynomial to calculate the trajectory.

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?
We are running our robot on Ubuntu 20.04 and ROS1

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