RoboCup 2023 Submission Survey

Survey response 1

Software

Team Name
NUbots

Is your software fully or partially OpenSource. If so, where can it be found:
Fully open source. https://github.com/NUbots

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. Some parts are exported from the CAD model (for center of mass) and others measured inside the CAD software (length of leg, hip offsets and so on).

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?
NUbots use an approximate analytical solution for the inverse kinematics of the NUgus robot based on geometry.

Are you simulating your robot? If so what are you using simulation for?
Yes, Webots is our primary simulator. This is for developing, testing and bug fixing without wearing out the hardware (ie servos), for remote work and for ease.

What approach are you using to generate the robot walking motion?
NUbots currently use Bit-Bots' Quintic Walk based off Rhoban's Quintic Walk and Rhoban's IK Walk. This is an open loop walk engine that creates quintic splines representing the three-dimensional trajectory of the feet and torso both in rotation and translation. The engine interpolates over these splines to find the next target position for the feet, which is then converted into servo joint angles using inverse kinematics.

What approach are you using to generate motions for standing up?
NUbots' NUgus robots use key frame animations to stand up.

What approach are you using to generate kicking motions?
NUbots' NUgus robots use key frame animations to kick.

Do you use any other motions than the previously mentioned? If so, what approaches are you using to generate them?
Head motions move the servos such that the robot is looking in a partiular position, with an exponential filter to smooth the movement.

Which datasets are you using in your research? If you are using your own datasets, are they public?
NUbots train the visual mesh network with a dataset generated from NUpbr, https://github.com/NUbots/NUpbr, of which the code is public and can be used to generate vision data with segmentation masks.

What approaches are you using in your robot's visual perception?
The Visual Mesh underpins the vision system, and is used for sparse detection of balls, points on the field, field lines, goal posts and other robots. From the Visual Mesh a series of specialised detectors are employed to detect field edges, balls, and goal posts. All points that the Visual Mesh has identified as either field points or field line points are clustered into connected regions and each cluster is then either merged or discarded using some heuristics until a single cluster remains, allowing an upper convex hull to be fitted. The ball detector finds all clusters which are below the upper convex hull. A circular cone is then fitted to each cluster. Different heuristics, such as degree of circle fit and different distance metrics, are then used to discard to cones. The goal post detector find all clusters which intersect the upper convex hull. Clusters are formed from goal post edge points and a determination of the bottom mid-point is made for each post.

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. The visual mesh algorithm converts points from image space to cartesian space using knowledge of the location of the ground plane from robot pose estimation. The algorithm assumes all points are on the ground.

How is your robot localizing?
The NUgus robots localise using a particle filter, with the measurement update using goal post locations from the vision system.

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 robot currently walks directly to the ball, however this is likely to change by the 2023 competition.

How is the behavior of your robot's structured (e.g. Behavior Trees)? What additional approaches are you using?
The old system is a subsumption based framework. The system is currently updating to an algorithm that creates a series of solution trees and chooses the best solution tree, based on various factors including access to lower submodules, priority, and required robot states. The algorithm allows for soft transitions and composable behaviours, and is implemented on both high level and low level modules.

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?
The head is moved to track the ball or follow a set search pattern the looks left and right, with an exponential filter.

Do you apply some form of filtering on the detected objects (e. g. Kalman filter for ball position)?
Heuristics are used to filter ball and goal measurements, using shape, size and distance. A simple exponential filter is applied to the closest ball measurement. The team is currently implementing a Kalman filter for the ball.

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)?
The current system does not have communication implemented, but it is planned to be implemented soon.

Please list contributions your team has made to RoboCup
- Participation since 2002, including Virtual competitions. Won Four Legged League in 2006 and Standard Platform League in 2008. - Open source code, hardware and debugging tools on https://github.com/NUbots - RoboCup Protocol https://github.com/RoboCup-Humanoid-TC/RobocupProtocol - Blender plugin for automatic dataset generation https://github.com/NUbots/NUpbr - Comprehensive documentation in the form of a website https://nubook.nubots.net/ - Participation in forum discussions

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).
M. Amos, R. Middleton, A. Biddulph and A. Mendes, Implementation and analysis of dynamic stability for bipedal robotic motion, 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, Australia, 2020, pp. 1950-1957. https://ieeexplore.ieee.org/document/9308374

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.
Walk engine: quintic walk engine from team Bit-Bots, which in turn was from Rhoban. Base hardware design: Nimbro/iGus.

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?
Arch Linux 29/08/2021 NUClear https://github.com/fastcode/nuclear

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

If you have a description document of your software you would like to share, you may do so here.
[{ "title":"NUbots Software Description","comment":"Document describing the system.","size":"124.0029296875","name":"NUbots-Software-Description.pdf","filename":"fu_i3m7xe22fdzimww","ext":"pdf" }]

filecount - If you have a description document of your software you would like to share, you may do so here.
1