Software Survey 2025

Team Name

NUbots

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

It is fully OpenSource. It can be found at: 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)?

We have a kinematic model which was made through exporting a URDF file from our CAD model.

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

We are using Inverse Kinematics. We use a combination of an approximate analytical and optimisation-based solution.

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

We simulate our robot with Webots to test various subsystems, namely behaviour, strategy, and localisation. We are also exploring MuJoCo for reinforcement learning.

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

We have an open-loop engine that generates polynomial splines. These splines represent three-dimensional trajectories of the feet and the torso between steps. The trajectories are generated in the planted foot frame as a function of the desired walk command. 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?

We have scripted keyframe animations whenever the robot gets up after falling. There are two scripts, one for getting up from the prone, and the other for getting up from supine. Each frame in the scripted animation has positions for each servo as well as an frame-duration.

What approach are you using to generate kicking motions?

We use a kick generated from a polynomial spline for the kicking foot and the torso.

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

The robot turns its head in order to look in a chosen direction using a vector from the camera. We have also a falling relax which unlocks sensitive servos, such as in the neck and arms, so that the robot can absorb the fall, and no parts are broken.

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

We have our own dataset of synthetic segmented images.
https://huggingface.co/datasets/NUbots/SoccerSegmentation
We also use Bit Bots' dataset to train Yolov8.
https://github.com/bit-bots/TORSO_21_dataset

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

We use at least one camera in the robot's head. Given a feed of images, we then use a combination of the Visual Mesh and a Yolov8 network. The Visual Mesh is a convolutional neural network that detects objects on the field such as field-lines, balls, other robots, etc. Its input is a mesh which sample pixels from a camera-image, such that the detection is depth-independent and has an efficient number of points. Yolov8 is a technique based on bounding boxes. It is used for redundancy to detect objects such as balls and goal-posts, but mainly field-line intersections such as L, T, and X-intersections. These detected intersections help the localisation of the robot.
https://github.com/Fastcode/VisualMesh
https://arxiv.org/abs/1807.08405

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 three-dimensional Cartesian space. The Visual Mesh already transforms from image space. Whereas, we use a spherical to Cartesian function on the output after Yolov8.
https://github.com/NUbots/NUbots/blob/36586a572afa5967802677c9a48ecff3941e8f82/shared/utility/vision/projection.hpp#L206-L252

How is your robot localizing?

We achieve localisation through a combination of both dead-reckoning odometry and an optimisation routine solved by NLopt's Constrained Optimisation BY Linear Approximations (COBYLA) algorithm using field-line points, intersections, and goal-posts. The odometry estimates the robot's pose over time through a Mahony Filter for roll and pitch, and through an anchor point method for translation and yaw.

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 does plan a path and can avoid obstacles. This plan is executed through localisation and through the Director framework which drives the overall behaviour. For obstacles specifically, the robot dynamically plans a path beside the obstacle until the robot is clear.

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

It is structured through the Director framework. The Director connects 'tasks', such as walking to ball, and 'providers' which either directly execute the the task or create smaller subtasks. These are organised in a tree.
https://arxiv.org/abs/2309.09248

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

If the robot loses sight of the ball, then it will begin moving its head around to widen its field of view. Once it sees the ball, it will then centre its head on the ball.

Do you apply some form of filtering on the detected objects (e. g.

Kalman filter for ball position)?

We use a Kalman filter for the ball. We also have Kalman filters for other robots on the field with association.

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 do perform robot-to-robot communication. The standard RoboCup Humanoid League protocol is used in UDP broadcast.

Please list contributions your team has made to RoboCup

The NUbots team participated in the 2024 Humanoid Kid-Size League and finished as semi-finalists. The NUbots have participated in the Four Legged League (2002-2007), the Standard Platform League (2008-2011), the Kid-Size Humanoid League (2012-2017, 2022-2024), and the Teen-Size Humanoid league (2018-2019). NUbots were the Four Legged League world champions in 2006. The team won the first Standard Platform League in 2008 as team NUManoid in collaboration with the National University of Maynooth, Ireland.

The team's RoboCup robot code, hardware, and debugging tools are open source on GitHub. A member of our team is also on the technical committee who often contributes to rule-changes. Another member has contributed to the Workshop on Humanoid Soccer Robots in 2024.
https://whsr-2024.github.io/

The NUbots team have developed a Blender plugin to generate semi-synthetic images with fully-annotated ground truth segmentation maps. The images contain random ball positions, robot positions and kinematic poses, obstacles, and viewer orientations. This tool is public on GitHub for anyone in the League to use.

The NUbots team maintains a comprehensive documentation resource in the form of a public website, providing detailed information about the hardware and software systems, as well as guides on various aspects of our systems. This resource aims to be useful to other RoboCup teams, as well as the wider robotics and AI community.

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 published three papers to the 2024 RoboCup Symposium and contributed to the Workshop on Humanoid Soccer Robots 2024. The RoboCup publications awaiting proceedings release are:

- The Director: A Composable Behaviour System with Soft Transitions
- Efficient Sequence Model for Early Fall Detection of Humanoid Robots
- SoS: A Semi-Synthetic RoboCup Soccer Dataset for Visual Segmentation

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

Our walk engine is inspired by Bit-Bots' (University of Hamburg) Quintic Walk, which itself was based off Rhoban's (University of Bordeaux) Quintic Walk and IK Walk. Our platform is based on the Humanoid OP platform developed by iGus and Nimbro (University of Bonn). We also use a dataset developed by Bit-Bots.
https://github.com/bit-bots/TORSO_21_dataset

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

We run Arch distribution of Linux and use the NUClear framework for our software.

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

If you have additional materials you would like to show, please link to them here.

https://nubook.nubots.net/