Software Survey 2025

Team Name

Hamburg Bit-Bots

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

Our software is fully open-source and available at https://github.com/bit-bots/

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 and dynamic model in the URDF and Webots .proto format. The URDF is exported from the CAD software
we use, onshape.com, using https://github.com/Rhoban/onshape-to-robot. We used https://github.com/cyberbotics/urdf2webots to
convert the URDF to a Webots .proto. Manual adjustments were required.

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

We are using BioIK for inverse kinematics: https://github.com/PickNikRobotics/bio_ik/tree/ros2

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

We use Webots and Mujoco. We use simulation for several machine learning algorithms: Parameter optimization of motion
algorithms, footstep planning using Reinforcement Learning, odometry correction.

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

We use an approach based on Rhoban's IKwalk, but heavily modified. It is based on quintic splines generating Cartesian
trajectories. Parameters defining these splines are optimized in simulation and adapted by an expert in the real world. The

approach is described in: Marc Bestmann, and Jianwei Zhang "Bipedal Walking on Humanoid Robots through Parameter
Optimization" RoboCup Symposium 2022, Bangkok, Thailand

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

We use a combination of a joint space spline keyframe animation which is recorded by puppeteering the real robot and an IK based rise movement in Cartesian space from the squat position onward. Our protective falling reaction is active during the later part of the standup motion, so damages from falls during a standup (e.g. due to a collision with another robot) are minimized.

What approach are you using to generate kicking motions?

We use our walk engine to generate normal steps that include a spline based Cartesian kick trajectory.

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

We generate different simple patterns as well as Cartesian look at actions for the head.

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

We use the TORSO21 dataset (https://github.com/bit-bots/TORSO_21_dataset) we published ourselves.

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

We use a Deep Learning approach based on YOLO called YOEO described in: Vahl, Florian, et al. "YOEO–You Only Encode
Once: A CNN for Embedded Object Detection and Semantic Segmentation." 2021 IEEE International Conference on Robotics
and Biomimetics (ROBIO). IEEE, 2021.

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 plan in Cartesian coordinates, depending on application relative to the robot or

relative to the playing field. We do inverse
perspective mapping (https://github.com/ros-sports/soccer_ipm/) using the orientation measured by an Inertial Measurement Unit and forward kinematics.

How is your robot localizing?

We use a particle filter whose particles are updated by matching a point cloud of observed line points with a precomputed distance map. Additionally, we work on image space localization using neural networks.

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 plans a path through the environment around obstacles. A* is used for generating a global plan. A carrot planner is used
to execute this plan.
In ongoing research, we are looking to replace this approach with a reinforcement learned footstep planner policy.

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

We use our own behavior framework called Dynamic Stack Decider described in: Poppinga, Martin, and Marc Bestmann. "DSD-
Dynamic Stack Decider." International Journal of Social Robotics 14.1 (2022): 73-83. Furthermore, we employ a potential field
method incorporating detected opponent robots to decide if we should dribble or pass.

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

Our head moves based on the state of the behavior. Several head modes such as looking around to detect obstacles or tracking
the ball are implemented.

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

We apply some handcrafted filtering approaches to model the ball position, accounting

for positive and negative observations as well as team data. A simpler filter is used for modelling observed robots for strategy and obstacle avoidance.

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 are performing team communication using the standard Protobuf based protocol.

Please list contributions your team has made to RoboCup

Open-sourcing our code has allowed several teams to use our algorithms (e.g. NUbots using our walking)

Our dataset TORSO21 includes 10464 fully annotated real world images to allow for comparing vision approaches in RoboCup soccer.

Additionally, we contribute to ROS-Sports, which provides standard software solutions for common tasks in the RoboCup.

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

"Deep Learning Based Measurement Model for Monte Carlo Localization in the RoboCup Humanoid League", RoboCup Symposium (2024) (to appear)

"Learning Footstep Planning using Deep Reinforcement Learning for Dynamic Walking", Workshop on Humanoid Soccer Robots 2024, Humanoids 2024 IEEE-RAS International Conference on Humanoid Robots

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

Our walking is based on IKWalk from Rhoban.

We use the onshape-to-robot tool for converting our CAD model to a URDF

We use the rot_conv_lib from Nimbro for using Fused Angles in our code.

Our goalie behavior is based on the one from BHuman

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

We are using ROS2 Jazzy on Ubuntu 24.04.

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.