Software Survey 2025

Team Name

CAU Mountain&Sea

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

Our software is not open-source at this time, as it is still under development. However, we may consider releasing it in the future once it reaches a more mature stage.

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, we have a kinematic model for this robot by exporting data from the CAD model. However, we have not derived a dynamic model, as we primarily rely on simulations to train the walking gait.

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

Yes, we utilize inverse kinematics to determine the joint angles of the actuators.

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

Yes, we utilize simulation for various aspects of our robot's development. First, we use a deep reinforcement learning framework to generate walking gaits. Additionally, our vision algorithms are tested in simulation to ensure their effectiveness. Finally, we also evaluate and refine our football-playing strategies within the simulation environment.

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

We employ a deep reinforcement learning (DRL) framework to generate the robot's walking motion. Specifically, we use a reward-driven training approach, where the robot learns optimal gait patterns through trial and error within a simulated environment. The training process involves defining reward functions that encourage stability, efficiency, and smooth movement while penalizing undesirable behaviors such as excessive energy consumption or instability. Additionally, we leverage domain randomization techniques to enhance the robustness of the learned policy, ensuring better generalization to real-world conditions. Once trained, the learned gait is transferred from simulation to the physical robot, with further fine-tuning if necessary.

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

We utilize a pre-defined motion sequence to enable the robot to stand up efficiently. This sequence is carefully designed based on the robot's kinematics and balance constraints to ensure stability and smooth execution. The standing-up motion is optimized by considering factors such as torque limits, center of mass shifting, and ground reaction forces to prevent tipping or excessive energy consumption. Additionally, we validate and refine the motion through simulations before implementing it on the physical robot. Future improvements may involve integrating reinforcement learning or optimization-based approaches to enhance adaptability across different terrains and environmental conditions.

What approach are you using to generate kicking motions?

The kicking motion is generally executed when the robot is in a stable, stationary position. To accommodate different scenarios, we have designed three pre-defined kicking motions—small force, medium force, and large force—allowing the robot to adjust the kick based on the required distance and target precision. These motions are carefully designed considering the robot's kinematic constraints, balance control, and force distribution to ensure stability and effective ball contact.

The predefined kicking actions are optimized through simulation, where we analyze joint torques, ground reaction forces, and ball trajectory outcomes. Additionally, we incorporate feedback from vision and localization systems to adjust the robot's positioning before executing the kick. Future enhancements may involve adaptive control strategies, where reinforcement learning or optimization-based approaches could refine the kicking motion dynamically based on real-time game conditions.

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

In addition to walking, standing up, and kicking, our robot utilizes various other motions depending on the task and environment. Generally, dynamic motions such as walking and turning are generated using a deep reinforcement learning (DRL) framework, which allows the robot to learn optimal movement patterns through trial and error in a simulated environment. This approach ensures adaptability and robustness, enabling the robot to handle different terrains and disturbances effectively.

For simpler actions like standing up and kicking, we use pre-defined motion sequences designed based on kinematic analysis and stability considerations. These motions are fine-tuned through simulation to ensure smooth execution and energy efficiency.

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

We primarily use our own proprietary dataset, which has been specifically collected and curated to support our research objectives. This dataset includes various types of sensor data, such as joint positions, torque measurements, vision inputs, and environmental interactions, which are essential for training and evaluating our models. The data is gathered from both real-world experiments and simulated environments to ensure robustness and generalizability.

At this moment, our dataset is not publicly available, as it is still under development and refinement. However, we may consider releasing portions of it in the future to support the research community, particularly after ensuring data quality, proper documentation, and compliance with relevant data-sharing policies.

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

For our robot's visual perception, we primarily utilize YOLOv5 for real-time object detection, enabling the robot to accurately identify key elements such as the ball and goal. YOLOv5's efficiency in handling object detection tasks allows us to achieve high-speed and reliable performance, which is crucial for real-time decision-making during gameplay.

In addition to deep learning-based detection, we incorporate traditional computer vision techniques to enhance robustness. For example, we employ particle filters to improve object tracking, allowing the robot to maintain a more stable and accurate estimate of the ball and goal positions even under occlusions or rapid movements. Furthermore, we apply color segmentation and contour detection methods to assist in refining object localization, especially in scenarios where deep learning models may face challenges due to varying lighting conditions or background noise.

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?

No, we do not use them in this robot.

How is your robot localizing?

Our robot primarily relies on its vision system for localization, using object detection and feature recognition to estimate its position on the field. Key elements such as the goalposts, field boundary lines, and the ball serve as reference points to help determine the robot’s location. We employ algorithms such as YOLOv5 for object detection and particle filters for tracking, enhancing the accuracy and stability of our localization 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)?

Our robot follows a straightforward path-planning approach, primarily navigating along straight-line paths toward its target. However, when obstacles are detected, we employ the Dynamic Window Approach (DWA) to enable real-time obstacle avoidance. DWA allows the robot to evaluate a set of possible velocity commands within a constrained window, considering factors such as collision avoidance, trajectory smoothness, and goal alignment. This ensures safe and efficient navigation, even in dynamic environments.

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

The behavior of our robot is structured using a finite state machine (FSM), which provides a clear and modular framework for decision-making and task execution. The FSM consists of well-defined states representing different behaviors, such as walking, standing up, kicking, and obstacle avoidance. Transitions between states are triggered by specific conditions, such as sensor inputs, game events, or environmental changes. This structured approach ensures predictable and reliable behavior while maintaining real-time responsiveness.

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

Yes, our robot incorporates an active vision system, where the camera mounted on the head can be dynamically adjusted using an actuator. This allows the robot to track key objects in the environment, including the ball, opponents, and the goal, improving situational awareness and decision-making.

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

Yes, the particle filter is applied for the object detection.

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)?

No, the team communication is not used.

Please list contributions your team has made to RoboCup

As a newly established team in RoboCup, we are actively working on developing and refining our robotic systems to contribute to the competition and the broader robotics research community. Our current focus is on advancing key areas such as locomotion, vision-based perception, localization, and strategy optimization.

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

[1].Chenyu Cai, Biao Hu*. BPNS: A Fast and Accurate Power Estimation Model for Embedded Devices. Journal of Circuits, Systems, and Computers, vo. 33, no. 15 pp. 2450275, 2024.

[2].Biao Hu, Yinbin Shi, Gang Chen, Zhengcai Cao, MengChu Zhou, Workload-Aware Scheduling of Real-Time Jobs in Cloud Computing to Minimize Energy Consumption, IEEE Internet of Things Journal, vo. 11, no. 1, pp. 638-652, 2024.

[3].Biao Hu, Xincheng Yang, Mingguo Zhao, Energy-Minimized Scheduling of Intermittent Real-Time Tasks in a CPU-GPU Cloud Computing Platform, IEEE Transactions on Parallel and Distributed Systems, vo. 34, no. 8, pp. 2391-2402, 2023.

[4]. Biao Hu, Xincheng Yang, Mingguo Zhao, Online Energy-Efficient Scheduling of DAG Tasks on Heterogeneous Embedded Platforms, Journal of Systems Architecture, vo. 140, pp. 102894, 2023.

[5]. Biao Hu, Zhilei Yan, Mingguo Zhao, Workload-Aware Scheduling of Multiple-Criticality Real-Time Applications in Vehicular Edge Computing System, IEEE Transactions on Industrial Informatics, vo. 19, no. 10, pp. 10091-10101, 2023.

[6].Biao Hu, Yinbin Shi, Zhengcai Cao, Mengchu Zhou, A Hybrid Scheduling Framework for Mixed Real-Time Tasks in an Automotive System With Vehicular Network, IEEE Transactions on Cloud Computing, vo. 11, no. 3, pp. 2231-2244, 2023.

[7]. Biao Hu, Yinbin Shi, Zhengcai Cao, Adaptive Energy-Minimized Scheduling of Real-Time Applications in Vehicular Edge Computing, IEEE Transactions on Industrial Informatics, vo. 19, no. 5, pp. 6895-6906, 2023.

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

None.

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

Ubuntu 22.04

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

No

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

No