EPFL@Home 2024
RoboCup@Home OPL Team Description Paper

Max Polzin, Kai Junge, James Ziadeh, Jessica Chang, Hugo Miranda Queiros, Dimitri Jacquemont, Ludovic Pujol, Octavio Profeta, and Josie Hughes

CREATE Lab, EPFL, Lausanne, Switzerland

Abstract. The EPFL@Home team from Lausanne, Switzerland, is a first time participant in RoboCup@Home. The robot is being developed from the ground up for both the hardware and the software. Relevant electrical and computation components were selected to build an omni-directional mobile manipulator. The software stack aims to integrate components for autonomous navigation, manipulation, visual perception, and human robot interaction. This report outlines the development of the team and the robot, the technical components of the current robot, and the future directions.

Keywords: Service robot · Autonomous navigation · Manipulation and planning · Visual perception · Soft gripper · Human robot interaction

1 Introduction

The EPFL@Home team was initiated in 2022 and is a first time participant at RoboCup@Home. The team is from EPFL in Lausanne, Switzerland, and supported by the MAKE initiative[4]: an internal organization at EPFL dedicated to support student-led interdisciplinary projects in the intersection of engineering and research. The team comprises of several Master students studying robotics or mechanical engineering and two PhD students as project and technical supervisors.

The project is organized by the CREATE Lab[9] with a research focus on design and fabrication of robots across a variety of environments, including development of robots which can interact and operate in human environments; a common shared goal with RoboCup@Home. In contrast to the RoboCup@Home project where the focus is predominately in the software integration, the lab's research is driven through novel hardware designs and integration often involving soft materials. For instance past works include a single 3D printed flexure-based robot hand[2] and a continuum robot arm for safe human-robot interaction [8].

A long term objective is to introduce the research hardware into the robot used to participate at RoboCup@Home, to introduce new hardware designs to the community alongside testing the feasibility of the hardware in a close-to-real-life scenario out of the lab. Likewise, the project also holds a pedagogical objective to train and expose students often working in an individual and academically driven environment to learn hard and soft skills necessary to operate in a team with a complex technical task.

The objective of participating in the 2024 RoboCup@Home is twofold. First, we would like to compete and pass Stage I. Second, we would like to gain valuable experience at the event to further improve our robot for subsequent years.

As the team has no prior experience in participation at RoboCup@Home, since the initiation of the project the team has focused on the quickest way to reach the robotic integration level necessary for the tasks. This includes finding the state of the art implementation of software packages for the different tasks while minimizing the amount of "custom" code for maximum efficiency. In practice this results in using established tools such as Nav2, MoveIt2, Behavior Tree, Detectron2, Yolo, Segment Anything Model (SAM), Docker. By doing so, we aim to stay up-to-date with the state of the art, and increase the re-usability of each module (due to them being standard implementations). The development process in the project has fed back into the research in the CREATE Lab. For example, recently the experience from using the Nav2 stack and Behavior trees was used to use this technology for autonomous navigation and morphing of a soft rover in an outdoor environment.

1.1 Prior experience

Despite our first attempt at participating in RoboCup@Home, a subset of our team has participated in other related robotic competitions.

RoboSoft Competition 2023 The most similar competition to RoboCup@Home that we have participated in is the RoboSoft Competition 2023[1] in Singapore, part of the IEEE Soft Robotics Conference (Participants include Hugo, James, Jessica, and Kai). In this competition, the task was autonomous manipulation of food objects for meal preparation using a soft robotic gripper. This competition allowed the development and testing of the manipulation, perception, planning, and gripper modules of the RoboCup@Home robot. The final result was 3rd place.

ICRA PUB-R Competition 2023 Similar to the RoboSoft 2023 competition, we participated in the ICRA PUB-R competition, also with a task of food preparation (Participants include Kai and Max). In this competition, we used a 12-dof five fingered robotic hand to prepare a full English breakfast[3]. The final result was 2nd place.

Past involvement in RoboCup Among the participants, Kai and Josie have past involvement in RoboCup. Kai has previously prepared for the RoboCup Rescue League (selected for participation in 2020, but cancelled due to covid-19), while Josie has participated in RoboCup Rescue Simulation. Both have also competed in the Junior league and organized as Junior Rescue Committee.

2 Robot Overview

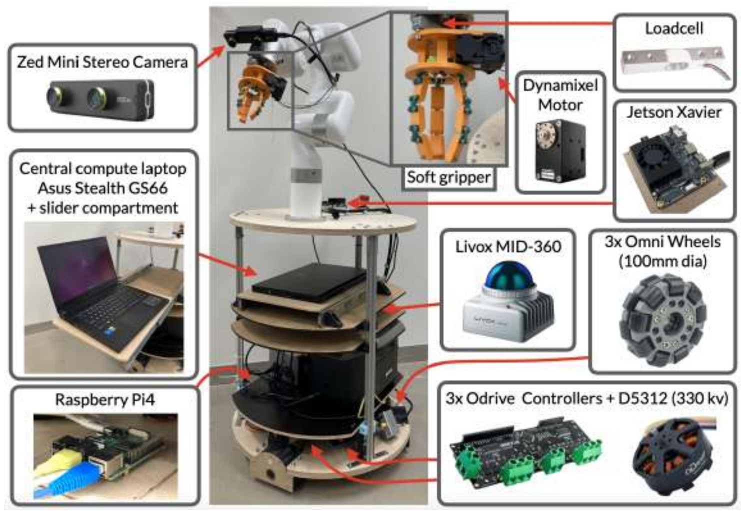


Fig. 1: Image of the robot annotated with major components used for the build.

The robot comprises of a custom built omni-directional base and a robotic arm/gripper (Fig. 1). The cylindrical robot base is divided into different layers. The bottom layer forms the drive module comprised of three omni-directional wheels actuated by brushless DC motors with a custom 15:1 reducer, two ODrive motor controllers, and a dedicated Raspberry Pi to execute low level motor commands. The second layer is comprised of power electronic controllers (main battery, robot arm control box) and a router. The third layer is dedicated the LiDAR (Livox MID-360), used for the Simultaneous Localization and Mapping (SLAM). The fourth layer is formed by the main computer. The fifth and final layer is where a Jetson Xavier is placed, which is used for all vision perception algorithms.

2.1 Gripper

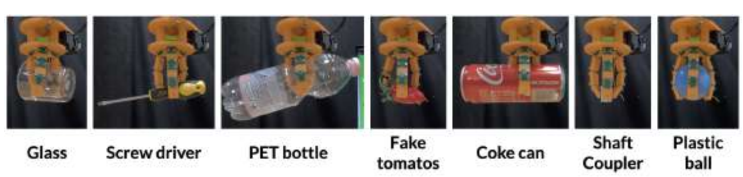


Fig. 2: Sample of different objects the soft gripper can hold with identical actuation commands.

A notable feature of the robot is the soft three fingered gripper. The gripper is a common design in the soft-robotic literature (such as [5]), and is a first step to integrate research hardware onto the Robocup@Home robot.

With a single cable actuation, the gripper is able to mechanically adapt its posture to grasp a multitude of objects reliably. Fig. 2 shows the grasping capabilities of the gripper on different objects of various sizes and weights with the same actuation motion. The gripper is also equipped with a loadcell, which can be used to measure any load on the gripper, and hence evaluate if a grasp was successful (shown in [6]).

2.2 Software integration overview

The robot's software integration is built from the ground up, and is continuously being improved.

Our low-level omni-directional base platform is integrated into the ros2control stack. We use the hardware interface implementation provided by Factor Robotics[7] and implement a custom kinematic model which translates velocity commands to the wheels and delivers accurate odometry measurements.

The robot's sensory and actuator stack is distributed across two embedded computing platforms. The robot's actuation stack includes the robot's base driver to execute basic locomotion commands and runs on a Raspberry Pi 4.

An Nvidia Jetson Xavier NX is tasked with our perception algorithms. It is equipped with the required hardware interfaces to connect to LiDAR and capture stereo camera images. The Nvidia Jetson Xavier NX runs algorithms to interface with those sensors and processes the data on its GPU.

The embedded computers are responsible to connect to the robot's hardware devices and are equipped with the required IO ports.

A central computer runs the robot's high-level algorithms, i.e. localization and mapping, state estimation and high-level behavior planning. Our robot's software stack tightly integrates high-level planning systems combining MoveIt2 for manipulator control and Nav2 for autonomous navigation. Coordination across frameworks to achieve task execution is implemented using behaviortreecpp.

Our development process leverages Docker for consistent software environment management across the robot's diverse computational platforms. Continuous integration and deployment are facilitated by GitHub Actions, automating the building of Docker images for each subsystem.

3 Navigation

The core of the navigation capability of our robot relies on the Nav2 stack. The navigation module takes as input a binary cost map, the current robot pose on the map, and a goal pose. At a high level, given the three inputs, waypoints for the robot trajectory are computed, and then executed based on an A* search algorithm. The execution of the robot motion is performed by

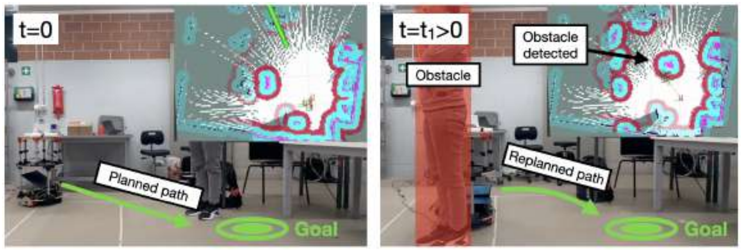


Fig. 3: Explanation of the dynamic navigation and replanning.

sending velocity commands to the robot base module, which is then converted to velocity commands executed by two ODrive BLDC motor controllers.

To work around dynamic obstacles, the cost map is constantly updated using a SLAM algorithm. Using the Livox Mid 360 LiDAR, we obtain a 3D point cloud of the environment. Since the standard configuration of the Nav2 stack requires a 2D cost map, the 3D point cloud is converted to a 2D cost map using the pointcloud to laserscan library.

The based on the 2D cost map from the Lidar measurements, the Nav2 stack updates the local and global cost map (see Fig. 3). The local cost map is the cost map physically nearby to the robot, which is updated at a high frequency to react well to dynamic obstacles.

3.1 Future work

The main future work of the navigation module is to test on different environments and situations to fine tune its parameters.

4 Manipulation

The arm is controlled thanks to MoveIt 2 ROS 2 library in order to modify at will the planned trajectory while avoiding any known obstacle. Several ROS services, topics and packages were created to allow a seamless communication between the arm, the camera and the gripper nodes. The tf2 ROS 2 library was used to associate coordinate frames with all known objects and joints of the robot and publish and compute the transforms between these frames.

The main control flow of the manipulation tasks is performed thanks to several Behavior Trees created with the BT library. To demonstrate what the robot is able to do, a simple demo where the xArm6 picks and places two pieces of broccoli in a plate and a water bottle in a box shown in [6]. In this demonstration, the manipulation module queries the perception module to determine the position of a given object in 3D space. The behavior tree allows for re-grasps and complex back-and-forth between different points in space. The behavior tree used for this motion can be found in Fig. 4.

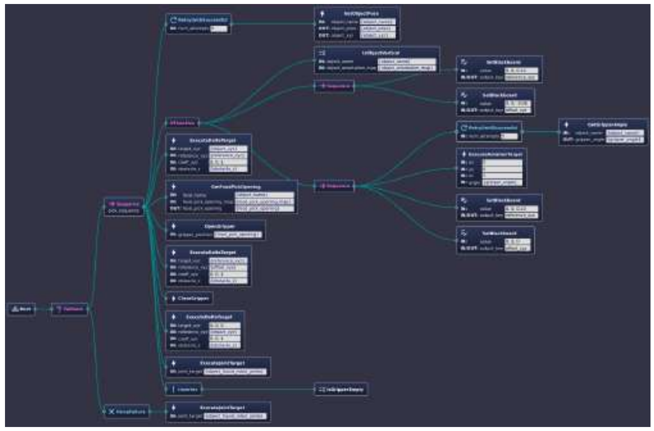


Fig. 4: Behavior tree used to perform the manipulation demonstration task.

4.1 Future work

The next steps are to connect seamlessly the Manipulation part with the Navigation module, to ensure a smooth and adaptive collaboration between the robot base and the arm.

5 Perception

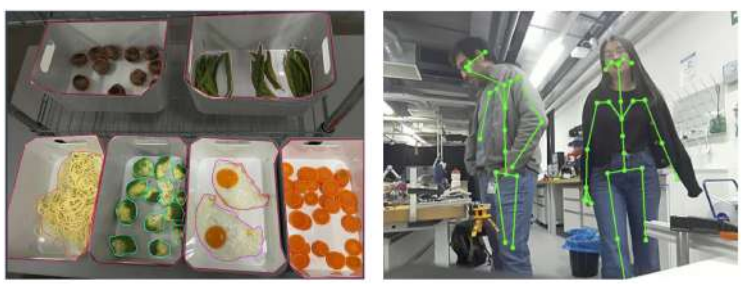


Fig. 5: Right: Detection and segmentation of food objects. Left: Human and skeleton detection.

Perception work revolves around the utilization of the camera. We employ Stereolab's SDK to connect with a Stereolabs ZED mini camera, capable of providing precise depth images. Additionally, the SDK automatically captures 3D human body poses represented by 38 body points (see Fig. 5).

Moreover, Detectron2 functions as our algorithm for object recognition and segmentation. We establish a real-time object detection system capable of recognizing objects in images and can also find the specified object given by the command. For labeling training data, we construct a pipeline for efficiently training a new model with fresh data. In addition, recognizing the importance of inference time in real-time applications, we implement another object detection system using YOLO to enhance efficiency. Furthermore, to enable the detection of unknown objects, we utilize SAM (Segment Anything) as a tool to localize and segment unidentified objects (see Fig. 5).

For real-time face recognition, we use deepFace as our primary tool to detect faces alongside estimation of other information such as age, gender, race, and emotions.

5.1 Future work

The prospective tasks for perception include the model training of object datasets used in RoboCup@Home, predicting the optimal grasping points on objects, and further improving the overall performance and efficiency of the systems mentioned. Gestures of humans from the skeletons must be automatically analysed. We must also develop a robust querying system for the main behavior tree to query the locations and/or most recent detection of a given object/face/gesture.

6 Human Robot Interaction

Human-robot interaction (HRI) plays a large role in the success of service. We began the development from implementing speech recognition, as it is the central component for generating high level commands. For this, we uses two main python libraries: vosk for an accurate speech-to-text conversion, and spaCy for the natural langage processing part (NLP). Compared to other speech-to-text models, Vosk offers a fast offline usage, and is easy to use with PyAudio, an other python library for sound recording. SpaCy is used to sparse the text and extract key elements such as the action, the involved object, the furniture, and the location. For example, a command like "Fetch a book from the living room bookshelf" is accurately processed to extract the action (fetch), object (book), location (living room), and furniture (bookshelf). For now the list of objects, furniture, etc. is predefined but this will have to change (see 6.2 Future work). Furthermore, by running the NLP module locally, this prevents from communication and network issues which may arise from querying external compute resources.

6.1 Future work

The main work on this would be linking the recognized commands to previously seen objects, faces, and relating them to a world position.

On the hardware side, there are a few updates to be made. Firstly, a dedicated microphone and speaker must be setup (currently, a headset is being used). Likewise, we will integrate a screen behind the robot arm that could provide the user with valuable feedback during interactions with the robot. To mitigate errors in speech parsing and for debugging purposes, we will also include a few buttons for manual control and operation

7 Behavior Planning

Our robot's behavior is controlled and defined using behavior trees. BTs provide an intuitive way to plan and execute complex behavior tasks and integrate different software stack such as manipulation or locomotion into a common framework, as shown in Fig. 4. This function is still largely under development, but initial integration is tested with the manipulation and perception modules.

7.1 Future work

The main focus of the next months' development will be centered around the BT, where all the modules must be called and executed in a given pattern. We first aim to integrate simple navigation and manipulation together, and gradually increase to add different components into the tree.

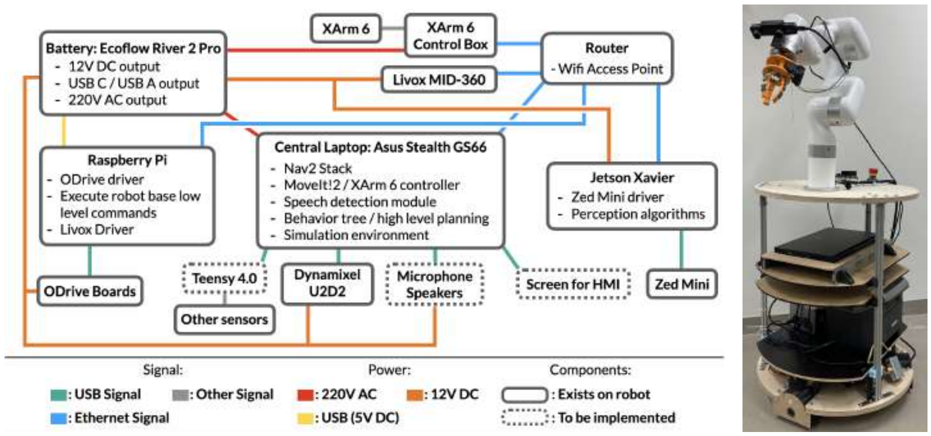
8 Conclusion

In this report we outline the current progression of developing the robot for RoboCup@Home for the team EPFL@Home. We explain the implementation and future work for the different modules necessary to solve the tasks presented for RoboCup@Home. Overall, most of the preliminary working implementations of the various modules are done, and the next steps remain to connect the different modules together. This is a challenge on its own, but we are eager to solve to demonstrate the capabilities of the robot we developed.

References

1. 2023, R.: Robosoft 2023 competition (2023), https://robosoft2023.org/2023-competitions/
2. Bosio, C., Junge, K., Hughes, J.: Scalable fabrication and actuation of a human inspired hand through 3d printed flexures and combinatorial actuation. Frontiers in Robotics and AI **9**, 878111 (2022)
3. createlabepfl: Can a robot cook breakfast? [icra 2023 pub.r competition] (Jul 2023), https://youtu.be/jln4d-4Vns8
4. Delisle, J.: MAKE projects interdisciplinary projects supported by the epfl (2023), https://www.epfl.ch/education/educational-initiatives/discovery-learning-program-2/interdisciplinary-projects/
5. Dollar, A.M., Howe, R.D.: The highly adaptive sdm hand: Design and performance evaluation. The international journal of robotics research **29**(5), 585–597 (2010)
6. EPFLRoboCup: Tdp 2024 video 1 (Nov 2023), https://youtu.be/C2BmDyehjlE
7. Factor-Robotics: odrive_ros2_control. https://github.com/Factor-Robotics/odrive_ros2_control (2021)
8. Guan, Q., Stella, F., Della Santina, C., Leng, J., Hughes, J.: Trimmed helicoids: an architectured soft structure yielding soft robots with high precision, large workspace, and compliant interactions. npj Robotics **1**(1), 4 (Oct 2023). https://doi.org/10.1038/s44182-023-00004-7, https://doi.org/10.1038/s44182-023-00004-7
9. Lab, C.: Create lab (2023), https://www.epfl.ch/labs/create/

Annex



(a) System integration diagram (b) Our robot

Fig. A1: Hardware description of the robot

Robot Hardware The hardware description is summarized by Fig. A1 through a component integration diagram and a photo of the robot (annotated figure of the hardware can be found in Fig. 1 in the main document). The robot has a custom built base with three omni-directional wheels. In this cylindrical base, all computation components (laptop and two single board computers), power electronics, and a LiDAR is placed. Above the base is a 6-dof robot arm with a custom gripper.

Table A1: 3rd Party Software and Purchased Components

3rd Party Software	Purchased Hardware
Camera SDK: Stereolab's SDK	Camera: Stereolabs ZED mini
Human skeleton detection: Stereolab's SDK	Lidar: Livox MID-360
Object detection: Detectron2, Yolov8, SAM	Base motor driver: ODrive 3.6
Face recognition: deepFace	Base motor: D5312S 330KV
Navigation: Nav2	Main laptop: ASUS Stealth GS66
Wheel control: odrive_ros2_control	Perception processing: Jetson Xavier NX
Lidar driver: Livox-SDK2	Motor command processing: Raspberry Pi 4
Manipulation: xarm_ros2, MoveIt 2, tf2	Robot arm: Ufactory XArm 6
Behavior Tree: BT	Battery: EcoFlow RIVER 2 Pro
Speech recognition: vosk,spaCy,PyAudio	Gripper motor: Dynamixel XM430-W210-R

Custom software The main repository is epfl_robocup and it aims at containing all the dependencies and git submodules of the different software parts of our robot. Here under, is a short description of the content of each repository:

- epfl_robocup : The main repository of our team, containing all our code dependencies and git submodules. Currently it contains all the Navigation and Localization stack, along the NLP models for the Human Robot interaction.
- arm_manipulation: The repository containing all the ROS 2 packages of the Manipulation part. This repository is meant to be used as a git submodule in epfl_robocup.
- face_recognition: The repository containing the code and models to perform face recognition. This repository is meant to be used as a git submodule in epfl_robocup.
- object_detection: The repository containing the code and models to perform object recognition. This repository is meant to be used as a git submodule in epfl_robocup.

Custom Hardware

- Robot structure: The structure of the robot base is custom build from plywood, MDF, 3D printed PLA, and aluminium extrusions.
- Robot wheel transmission: The transmission system from the BLDC motor to the omniwheel is custom built with a 15:1 cycloidal gearbox.
- Gripper: The soft gripper is custom built from 3D printed PLA and TPU.