Team Description Paper robOTTO RoboCup@Work 2023

Christoph Steup, Hauke Petersen, Leander Bartsch, Inga Brockhage, Frodo Hinze, Adrian Köring, Franziska Labitzke, Dario Müller, Wiebke Outzen, Jurek Rostalsky, Sanaz Mostaghim, and Arndt Lüder

Otto-von-Guericke University, 39106 Magdeburg, Germany
{steup@ovgu.de,
WWW: http://www.robotto.ovgu.de/

Abstract. Team **robOTTO** is the RoboCup@Work League team of the Otto-von-Guericke-University Magdeburg, which has been participating in the RoboCup Logistics League since its inception in 2010. Our team combines expertise from computer science, electrical engineering, and mechanical engineering to solve the unique challenges of @Work while promoting the exchange of knowledge between theoretical science and practical robotics applications. In addition, the team offers students the opportunity to learn about robotics, teamwork, and complex project management in a real-world scenario.

Keywords: RoboCup@Work, robOTTO, RoboCup2024

1 Introduction

robOTTO was founded in 2010 as a RoboCup Logistics [4] team by nine students from different disciplines, allowing the exchange of knowledge and views among the members. After winning second place in Singapore in 2010, the team continued to participate in subsequent RoboCup competitions, with further successes in 2012 (4th place) and 2013 (2nd place).

Due to major rule and equipment changes in the Logistics League in 2015, we decided to switch to the @Work League [3]. The Faculty of Computer Science already had some experience with the KUKA youBots [1], providing a pool of experienced students and easy integration into courses and research projects. The transition from RoboCup Logistics to the @Work League helped broaden the team's expertise and laid the foundation for the 2016 Crossover Challenge[8] in Leipzig, resulting in a first place finish in that challenge. The team competed at the 2015 World Cup in China, finishing 6th, followed by the 2016 World Cup in Leipzig, where we finished 4th. 2017 was very successful for us, resulting in a third place at the German Open and a second place at the World Cup in Japan. In Montreal, Canada, the team won first place in the Arbitrary Surface Challenge. At the last World Cup in Sydney, Australia, the team again won the Vice World Champion title.

In 2020, there was no real World Cup, and the team won the best presentation award in the Virtual RoboCup Asia Pacific Open (VRCAP). In the 2021 RoboCup@Home competition, which was entirely virtual, the team placed 3rd. The team's efforts in organizing the competition and streaming were recognized by the appointment of Franziska Labitzke and Hauke Petersen to the league's Organizing Committee, Leander Bartsch to the Technical Committee, and Christoph Steup to the Executive Committee. In addition, Christoph Steup is the current maintainer of the league's new official referee box.

The World Cup 2022 ended with another second place for the team and the re-election of all committee members. The WorldCup in 2023 in Bordeaux ended with a third place in the competition and the successful participation in the master communication challenge. Since Leander Bartsch was finishing his Master's degree, Wiebke Outzen was elected as his replacement for the Technical Committee. The other committee positions were again re-elected.

2 Team Structure

Task	Name	Field / Role
	Sanaz Mostaghim	Responsible Professor
	Arndt Lüder	Liaison to Mechanical Engineering
Organisation	Christoph Steup	Team Leader (EC-Member)
	Hauke Petersen	Team Leader (OC-Member)
	Franziska Labitzke	Public Relations Officer (OC-Member)
	Leander Bartsch	Electrical Engineering
Hardware	Inga Brockhage	Electrical Engineering
	Dario Müller	Computer Science
	Frodo Hinze	Electrical Engineering
Recognition	Adrian Köring	Computer Vision
	Wiebke Outzen	Computer Science (TC-Member)
Robot Coordination	Jurek Rostalsky	Mathematics

Table 1. Overview of **robOTTO** team members by task and field of studies or role

Currently, the team consists of 11 active members, while the professors are not involved in the development but only provide guidance and organizational support. The team is composed of undergraduates who will leave the team after graduation, as well as postgraduates. This approach allows the team to combine fresh ideas and approaches with the experience of veterans. Depending on the new members and their backgrounds, the team can benefit from a wide range of expertise. The current team members bring a diverse set of expertise to the team, from cybernetics and computer science to electrical engineering, as shown in Table 1. The new students bring new ideas and challenges to the team. Currently, the team is trying to attract new members to compensate for the future loss of members. However, recruiting new members is difficult because students already have busy schedules and understanding the software, hardware, and procedures is tedious and time consuming. Two years of Corona have also taken their toll.

3 Robot Description

Over the years, the team identified many shortcomings of the standard **KUKA youBot**. As a result, the original hardware platform was heavily modified to be competitive in the league. To minimize effort and maximize results, most of the additions are Components of the Shelf (COTS), used in the robotics community. The robot currently in use replaced almost everything from the original YouBot except the lower body, wheels, arm, and hardware to control the wheels and arm.

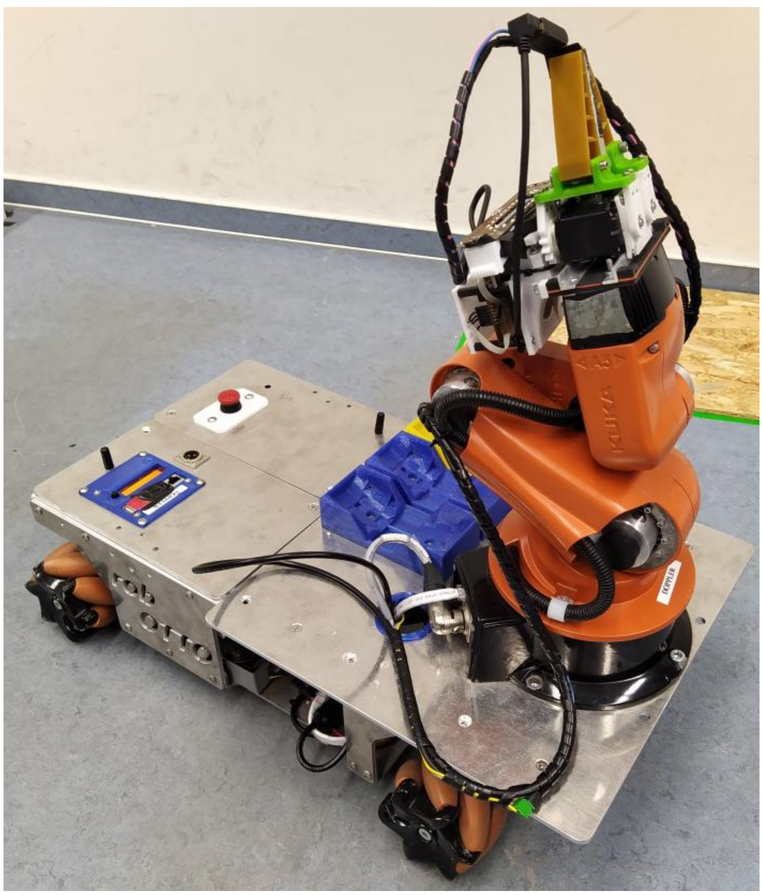

Fig. 1. Modified KUKA youBot

3.1 Changes to the standard Platform

Camera We use an Intel RealSense L515 RGB-D camera that provides registered point clouds as well as an RGB-D image. Currently, we focus on the color data for recognition and use the depth image as a highly flexible distance sensor. The camera position was changed from the end of the YouBot arm to the 3rd joint of the arm to allow more distance to the table surface. This allows us to detect more objects with less movement in front of the table, increasing speed. During navigation, the RealSense camera is used to detect barrier tape.

Gripper The objects in the @Work league have different shapes and sizes. After initial testing, we found that the standard metal gripper on the KUKA youBot could not reliably handle many of these objects. Our current gripper is based on a custom 3D-printed mount using a single servo from Dynamixel and Finray fingers from Festo controlled by a Teensy 3.2 microcontroller board. Current development focuses on correctly identifying and handling error cases such as the loss of an object. In addition, we aim to improve gripping by defining and updating individual gripping rules for each object.

Computing and Connections We upgraded our Intel NUC to a newer 8th generation version with 4 real cores and 45W TDP to enable more complex algorithms for navigation, path planning and task optimization. We now use two Teensy 3.2 ARM-Cortex M4 powered embedded boards to integrate the gripper and our custom power supply. The hardware and software architecture is modularized through defined interfaces to provide a stepping stone for students inexperienced in programming to experiment without needing the entire development stack used on the main robot. Additionally, modularity allows for upgrades and cross-overs between robots to test, debug, and repair hardware and software components.

Laser Scanner Mounting Brackets The team uses Hokuyo URG-04LX laser scanners. These provide appropriate distance measurements in a 240° radius with a maximum distance of 5.5 m. Reliable localization is possible when the laser scanners are exactly parallel to the ground. To achieve this, the team designed reliable, adjustable mounting brackets that the team 3D printed to prevent tilt errors even at the edge of the scanner's measurement range and to shield the expensive sensors from accidental collisions.

Replacement of the Upper Case The newly designed and manufactured top cover completely replaces the old top cover of the YouBot. It provides easy access to the electronics underneath while expanding the space inside the robot. This allows an additional Intel NUC, power supply for the grippers and our USB3 hub to be placed inside the robot. In addition, the inventory, emergency switch, and arm have pre-installed brackets on the new chassis. Multiple mounting holes allow the inventory to be offset and additional parts to be easily added to the surface. The new case has been very successful in the 2019, 2021 and 2022 competitions.

Lithium Polymer-based Battery System The YouBot's original lead-based batteries had two major drawbacks. First, they are incredibly heavy, which increases the cost of transportation when flying. Second, the energy they contain is rather low compared to the energy needed to power all the components. Currently, our bot lasts a maximum of 30 minutes, which is quite limited. For these reasons, we decided to switch to Lithium-Polymer based batteries. But instead of using single cells, we used Bosch Tool batteries, which are widely available and not suspicious at the airport. We used two off-the-shelf buck-boost DC-DC converters for 12V and 24V, as well as a custom printed circuit board for voltage and current sensing, and an emergency switch to disconnect the different rails. The current setup uses two batteries in parallel, which allows us to hot-swap the batteries without having to power down and up the bot. The system was extensively tested during the 2021 and 2022 World Cups and showed up to 1 hour of battery life, which is generally sufficient for competition runs.

4 Software Architecture

In this section we will describe the main software components and how they interact. We use the Robot Operating System 1 (ROS1) [5] in the Noetic version on Ubuntu 20.04 LTS. The main advantages are the communication abstraction and the large number of easy to use debugging tools. It also provides existing algorithms for typical robotics problems such as path planning and control.

4.1 Overview

Fig. 2 shows the interaction of our software components. They are described in detail in the following sections.

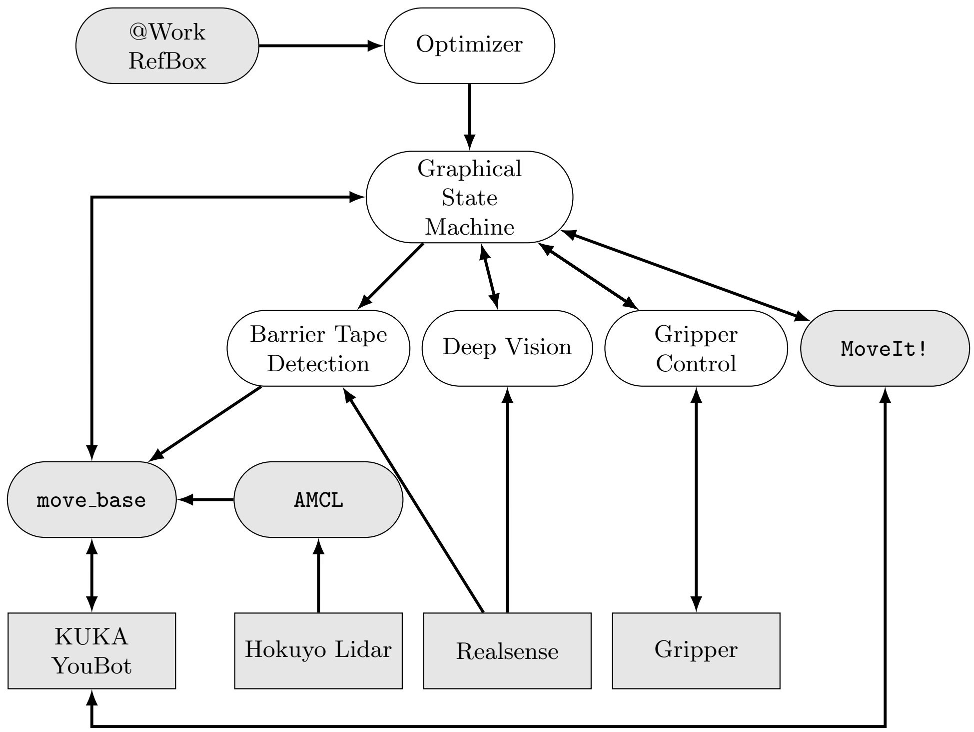

Fig. 2. Overview of the main software components of the robOTTO @Work framework.

State-Machine and Optimizer The core elements of the robot software architecture are the optimizer and the state machine, which are responsible for coordinating the other modules of the robot, such as vision, navigation, and arm movement. The transportation tasks to be performed by the robot are generated by the league's @work referee box and passed to the robot.

On the robot side, the `commander_receiver` is used to process and store the task in the world model. It also tells the optimizer to start the run. The optimizer splits the tasks into subtasks of picks, moves, and places, and searches for a sequence of transport tasks that achieves the maximum score within the time limit. It then passes these subtasks to the state machine.

The state machine contains a logic for each *SubTask*. The logic is determined by a graph of parametrized actions that are necessary to perform the *SubTask*. Common actions are the *MoveAction*, *ArmAction* and *VisionAction*. They control the substate machines of the corresponding robot modules. For example, a simplified *PickLogic* consists of the following actions

1. **ArmAction** moves the arm into the pose for the barrier tape detection.
2. **MoveAction** moves the robot to a specified workstation.
3. **ArmAction** moves the arm to the object recognition pose.
4. **DetectAction** Detects the specified object at the workstation.

5. ArmAction picks up the detected object and places it in the inventory.

Because a reliable state machine is critical to the success of a robot's performance, the state machine is built with many debugging options and a quick way to modify it to quickly handle unforeseen situations and errors on the cup.

World Model The World Cup 2017 in Leipzig showed that the complexity of the tasks and the environment is difficult to handle with our current software architecture. Incorporating additional information about the state of the arena or the robot required a lot of code and interface changes. To mitigate this engineering problem, we decided to manage the information about the world in a central component. The resulting World Model component allows us to store, track, and replay changes to the robot's and arena state. In addition, we added support to add, modify, and visualize the data in the world model of a specific task. The main benefit of this approach is the stability of the interfaces between our functional *SubTask* components, as well as the central point for team members to add and request information about the world. Finally, the visualization tool gives us better insight into the robot's current behavior and decisions, making debugging easier.

Deep-Learning-based Vision - Deep Vision Currently, we only use 2D images for object recognition. However, the RealSense also provides 3D data. Our object detection system is based on the TensorFlow Object Detection API. We have built a custom bootstrapping and augmentation system to generate additional training data. The main work in this system was the manual generation of training data. To facilitate the process, we developed a custom tool to check the detection and classification results and manually generate corrected training samples. The only addition to the original detection API is an additional object orientation detection code. The detection code to also output a 2D rotation of the object on the workstation.

Path Planning for Industrial Robots The navigation of industrial robots must be designed with many competing influences in mind, as fast movement and collision avoidance are both critical to successful participation in RoboCup@Work. Other factors are more subtle, such as predictability of behavior, ease of maintenance and adaptability. The last two are especially important in the context of RoboCup as a student competition, where team members and responsibilities change regularly and members must be able to learn the code, often on short notice. The current navigation stack used by robOTTO uses the TEB (Time-Elastic Band Local Planner) [6], which is open-source and on of the default planners for mobile platforms in ROS1. We slightly modified the parameters and tested it on the @Work Innovations Open in 2023, where it showed excellent performance and stability. The old specialized planner was deprecated due to lack of manpower to maintain it and no obvious advantages over the tuned TEB.

Integration of the MoveIt! Trajectory and Kinematics Stack The previously used kinematics stack for the team's manipulator, SAMIA, was developed by former member Stefan as a byproduct of his master's thesis and then adapted for the @Work competition. But with Stefan gone, we now face problems maintaining and extending the codebase. This led to the decision to abandon our stack in favor of MoveIt! [2]. MoveIt! is an open-source motion planning framework originally developed by Willow Garage that unifies motion planning kinematics, collision checking and dynamic three-dimensional environment rendering. Originally developed for use with ROS, it offers a high degree of integration with existing packages and tools such as RViz. The integration of the stack into our code, as well as the creation of the arm state machine Figure 2 and the interface to our main state machine. This was originally done by Hauke, but is now being reworked by Wiebke and Adrian. It has since proven to be a viable alternative to our previous solution and has been successfully used since the 2022 World Cup in Bangkok. One of the issues of the stack on our platform right now is the speed of the arm movement. We want to generate full end-to-end motion plans instead of our current point-to-point plans to increase the speed by limiting the accelerations during the motion and keeping the speed high.

Barrier Tape Detection Barrier Tape Detection is asked to spatially locate barrier tape strips given a camera image and position. To achieve this, we first detect the barrier tape in the image. This is done by semantic segmentation of the camera image, which is also used for object detection and recognition. In this case, however, a convolutional neural network is used, which combines the tasks of filtering, segmentation and recognition. We used Tensorflow to implement and train the network. The output of the network is an image of white pixels where the barrier tape has been detected. The training samples are generated by superimposing images of the barrier tapes used in the competition on different background images and lighting them with Blender. To get the world coordinates of the tape, we cast rays from the camera through the white pixels in the mask into the scene and find intersections with the ground plane. All of these intersection points eventually form the resulting point cloud, which is fed into the navigation stack as an additional sensor. The only manually configured part of the barrier detection is the calibration of the mapping of the 2D camera coordinates to the 2D map coordinates of the navigation. Currently, the results of the Barrier Tape Detection need better integration to the modified Nav-Stack. Consequently, the system is seldomly used because false positives strongly impact navigation performance and often cost more points than it gains.

Graph-based Visual State-Machine We have completed the integration of FlexBE [7] as our new state machine description mechanism. All of our old state machine code, as well as the previously separate RTT state machine, is now encoded in the new system. As a byproduct, we simplified the manipulator state machine and added advanced planning capabilities. Finally, our navigation state machine has also been transferred to the new state machine and has been extensively tested in the wild at the 2022 World Cup in Bangkok. The new state machine fully met our goal of simplifying development and speeding up changes, especially at the Cup. However, we still lack unit testing support for the state machine, which makes developing complex behaviors a bit cumbersome. We are currently implementing statistics tracking to automatically assess the performance of the robot and integrate this information into our optimizers.

5 Future Software and Hardware Components

The following components are currently being worked on and may be used in the 2024 World Cup when development is complete and they have been sufficiently tested.

Fig. 3. Our igus ReBeL®

5.1 Specialized Near-Field Localization

We are currently working on a specialized near-field localization mechanism to enable accurate localization of the robot near workstations, to eliminate our specialized navigation state machine that moves manually when very close to workstations. The new approach will enhance AMCL with semantic localization of tables and fuse the resulting pose information using an Unscented Kalman Filter.

5.2 Manipulator planned to be replaced in the future

The igus ReBeL
textsuperscript® pick and place cobot is a lightweight 6 DoF plastic robotic arm that can be used to replace the standard arm of our KUKA youBot
textsuperscript®, see figure 3. The new arm is clearly superior to the old one in terms of freedom of movement and reach. As a transitional measure, we will connect the existing gripper to the new arm using a module plate. Due to the conversion, a new location must be found for both the cameras and the inventory. The current system design uses a ROS2-based sub-system on a Raspberry Pi handling the low-level control of the manipulator and gripper over the integrated CAN-Bus. The existing ROS1_Bridge is used to combine the rest of the system with the new ROS2-Subsystem.

6 Conclusion

With the navigation settled and now performing extremely well and our new manipulator coming in, we are cautiously optimistic that we will be able to build on the work done last year. The last participation left us with a code base that solved most of the @work league tasks. This allows us to focus this year on improving the robustness and speed of the working solutions, while adapting them to the new rules.

Acknowledgements

We would like to thank Sanaz Mostaghim, Chair of Computational Intelligence, for her support throughout the year and for providing us with access to tools and the necessary facilities for storage and testing. The team and all former members would also like to thank Arndt Lüder for his dedication and support since our founding in 2010.

References

1. Rainer Bischoff, Ulrich Huggenberger, and Erwin Prassler. Kuka youbot-a mobile manipulator for research and education. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.
2. Sachin Chitta. Moveit!: an introduction. In *Robot Operating System (ROS)*, pages 3–27. Springer, 2016.
3. Gerhard K Kraetzschmar, Nico Hochgeschwender, Walter Nowak, Frederik Hegger, Sven Schneider, Rhama Dwiputra, Jakob Berghofer, and Rainer Bischoff. Robocup@ work: competing for the factory of the future. In *Robot Soccer World Cup*, pages 171–182. Springer, 2014.
4. Tim Niemueller, Daniel Ewert, Sebastian Reuter, Alexander Ferrein, Sabina Jeschke, and Gerhard Lakemeyer. Robocup logistics league sponsored by festo: a competitive factory automation testbed. In *Automation, Communication and Cybernetics in Science and Engineering 2015/2016*, pages 605–618. Springer, 2016.

5. Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3.2, pages 1–6. Kobe, 2009.
6. Christoph Rösmann, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. Efficient trajectory optimization using a sparse model. In *2013 European Conference on Mobile Robots*, pages 138–143, 2013.
7. Philipp Schillinger. An approach for runtime-modifiable behavior control of humanoid rescue robots. Master's thesis, Technical University Darmstadt, 2015.
8. Sebastian Zug, Tim Niemueller, Nico Hochgeschwender, Kai Seidensticker, Martin Seidel, Tim Friedrich, Tobias Neumann, Ulrich Karras, Gerhard Kraetzschmar, and Alexander Ferrein. An integration challenge to bridge the gap among industry-inspired robocup leagues. In *RoboCup Symposium*, pages 1–12, 2016.