smARTLab@work Team Description Paper 2016

Bastian Broecker, Benjamin Burger, Daniel Claes, Joscha Fossel,
Benjamin Schnieders, Richard Williams, and Karl Tuyls

smARTLab, Department of Computer Science, University of Liverpool
{bastian.broecker, benjamin.burger, daniel.claes, j.fossel, benjamin.schnieders,
r.m.williams1, k.tuyls}@liverpool.ac.uk
smartlab.csc.liv.ac.uk/smartlabatwork

Abstract. With this paper, we apply for the RoboCup@Work competition with the smARTLab@work team that has won the 2014 RoboCup@Work competition.[1] smARTLab (swarms, multi-agent, and robot technologies and learning lab) is a robotics research lab that is part of the Agent ART group at the computer science department of the University of Liverpool. This team has won the competition for the second year in a row. The various techniques that have been combined to win the competition come from different computer science domains, entailing machine learning, (simultaneous) localization and mapping, navigation, object recognition and object manipulation. While the RoboCup@Work league is not a standard platform league, all participants use a (customized) Kuka youBot. The stock youBot is a ground based platform, capable of omnidirectional movement and equipped with a five degree of freedom arm featuring a parallel gripper. We present our adaptations to the robot, in which the replacement of the gripper was the most important upgrade comparing to the version of the robot that was used last year.

1 Introduction

Although fixed industrial robotics has achieved unrivalled productivity in mass production lines executing predefined repetitive tasks, such automation is still unreachable in customized flexible production or assembly situations. Such situations typically involve a high mixture of different types of products or objects that need to be transferred from one configuration to another. For example this could involve a task in a car garage in which robots need to collect specific parts from a bin or cupboard from one side of the workshop and bring these to another part or even put them back in a specific configuration, requiring planning, learning and reasoning capabilities as well as multi-robot and human-robot cooperation.

Recent years have seen new initiatives in this direction by the initiation of new robot competitions that focus on future industrial robot settings [1]. The most

[1] The team previously competed as swarmlab@Work and changed name after the move of professor Tuyls and his research group from Maastricht University to the University of Liverpool, UK.

famous competitions of its type are RoboCup@Work [2] and the RoCKIn@Work competition, which are specifically targeted to mobile industrial robotics in futuristic settings (the factory of the future). The competitions, in particular, focus on the use of robots in various work-related or industrial scenarios, such as navigating to specific locations and moving and manipulating objects. They aim to foster the research and development of innovative industrial robots for current and future industrial applications, where robots cooperate with humans on complex tasks ranging from manufacturing and automation to general logistics.

2 smARTLab@Work

smARTLab is the research laboratory of the Department of Computer Science at University of Liverpool that focuses on designing Autonomous Systems. The general research goal and mission of the lab is to create adaptive systems and robots that are able to autonomously operate in complex and diverse environments. The smARTLab@Work team has been established in the beginning of 2014, consisting of 3 PhD candidates and 2 senior faculty members. The team has won the @Work competitions of the 2013 RoboCup German Open, and the 2013 RoboCup world championship under the name swarmlab@Work. After migrating to the University of Liverpool smARTLab@Work also placed first in the 2014 German Open and the 2014 RoboCup world championship.

The team's mission is (a) to apply smARTLab research achievements in the area of industrial robotics and (b) identify new research challenges that connect to the core smARTLab research areas: autonomous systems, reinforcement learning and multi-robot coordination.

In the remainder of this section we introduce the smARTLab@Work robot platform that was used for the 2014 RoboCup@Work world championship. Especially, we describe the necessary hardware modifications that were made to adapt this standard platform to the winning configuration.

2.1 youBot Platform

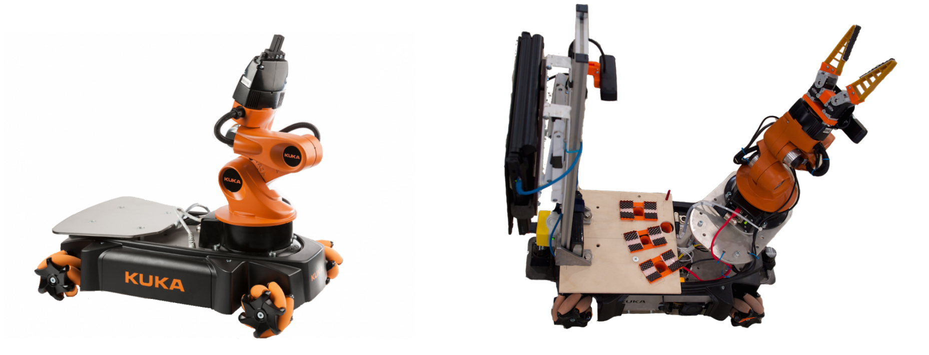
The youBot is an omni-directional platform that has four mecanum [3] wheels, a 5 DoF manipulator and a two finger gripper. The platform is manufactured by KUKA[2], and is commercially available at the youBot-store[3]. It has been designed to work in industrial like environments and to perform various industrial tasks. With this open-source robot, KUKA is targeting educational and research markets [4]. Figure 1a shows a model of the stock youBot.

2.2 Hardware Modifications

In order to meet the requirements we demand from the youBot platform to tackle the challenges, we made a number of modifications to the robot. In this

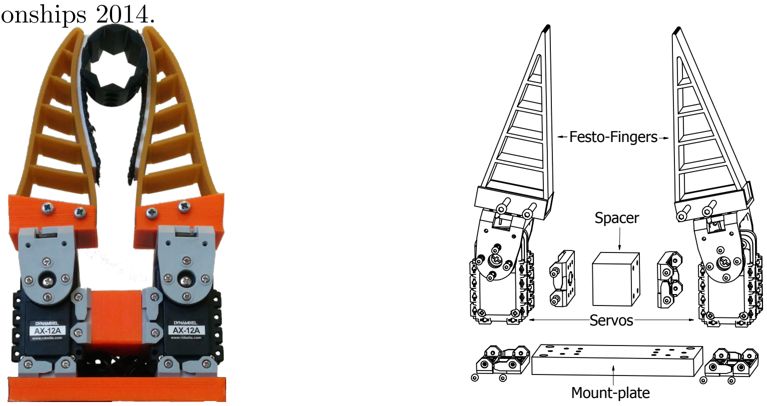
[2] http://kuka.com

[3] http://youbot-store.com/



(a) (b)

Fig. 1: (a) Stock youBot. (b) smARTLab@Work modified youBot for World Championships 2014.



(a) (b)

Fig. 2: Figure (a) illustrates the adaptation of the gripper to the round shape of the object. Figure (b) shows a drawing of the separate parts of the new designed gripper in more detail.

paragraph we describe which parts are modified and why these modifications are a necessity for our approach. Figure 1b shows the modified smARTLab@Work youBot setup that was used in the world championships in Brazil 2014.

The main adaptations with respect to last years version of the youBot are the replacement of the standard gripper with a custom made gripper and the design of a backplate to allow transportation of multiple objects. Since the new custom gripper is longer than the standard gripper, the arm was elevated by an additional 10cm to still be able pick objects in a top-down fashion.

The standard gripper is replaced by two FESTO FinGripper fingers[4] mounted on two Dynamixel AX-12A[5] servo motors. This increases the stroke to more than 20 cm and the speed of the gripper to up to 10 cm/s. Also the fingers passively

[4] http://www.festo.com/rep/en_corp/assets/pdf/Tripod_en.pdf

[5] http://support.robotis.com/en/product/dynamixel/ax_series/dxl_ax_actuator.htm

adapt to the shape of the objects as illustrated in figure 2a. Through the custom gripper is the youBot is now able to grasp every competition object, where the previous standard gripper was limited to the thinner objects due to his maximal opening width of 2cm.

The new backplate (see figure 1b) offers three high friction pockets to place the objects on the robot. These pockets are designed to hold the items in place during transportation.

In order to detect and recognize manipulation objects, an ASUS Xtion PRO LIVE RGBD camera is attached to the last arm joint. This camera is mounted, so that it faces away from the manipulator, as can be seen in Figure 4a. The main idea behind this mounting position originates from the minimal distance of the RGB-D camera, which is approximately ∼0.5m. Using this special pre-grip pose, we can scan the area and perceive objects from a top-down perspective. An additional camera is mounted on the back of the robot. This camera is used to get a second view of certain objects, i.e. an object is picked up and it is moved by the arm such that it can be seen from a different angle. This helps to resolve confusions that are inevitable when only looking from a top-down fashion with the ASUS RGBD camera.

In this section we described the new hardware modifications needed to optimally execute the required RoboCup@Work tasks, in the next section we will present the methods that we use to solve the specific tasks.

3 Approach

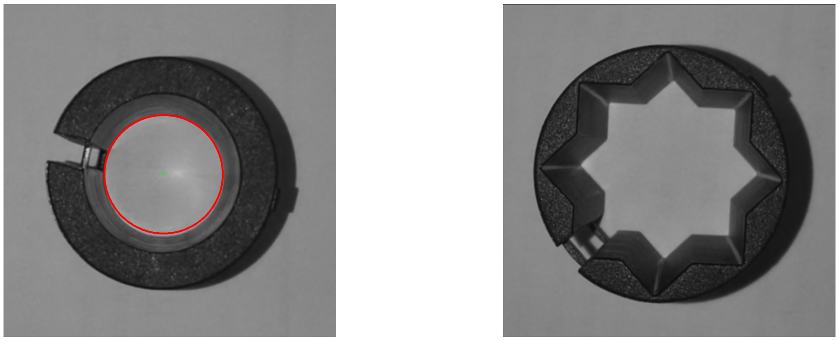
In this section, the different techniques used to tackle the above mentioned tests are explained. We developed different modules for many different capabilities, e.g. basic global navigation, fine-positioning, object recognition, inverse kinematics of the arm, etc. We also explain our various recovery methods. By combining these capabilities in state-machines we are able to solve the tasks specified above.

The main adaptation w.r.t. last year, is the addition of the possibility to transport objects on the backplate of the robot. This allows us to transport more than one object at a time and actually planning the task becomes an important factor in the execution of the test. Furthermore, the object recognition was improved by combining the depth information together with the RGB information for the features. Also, the additional camera on the back was used to distinguish the R20 and V20 object, since the object shapes are very similar.

Thus for the descriptions of the approaches for mapping, localisation, inverse-kinematics and fine-positioning, we refer to [5].

At the moment we only participated at the RoboCup@Work competition. So the approach is tailored to the tasks that need to be solved there.

Navigation A necessary capability of the robot is to navigate in the known environment without colliding with obstacles. The map is created with gmapping [6] and AMCL [7] is used for the localization in the map. The global path is computed by an A* algorithm and is then executed using a dynamic window



(a) (b)

Fig. 3: (a) R20 object with detected circle. (b) V20 object with no detected circle.

approach [8] trajectory rollout for the local path planning. This planner samples different velocities and ranks them according to the distance to the goal and the distance to the path, while velocities that collide with the environment are excluded. We tuned the trajectory rollout in such a way that it incorporates the omni-directional capabilities of the robot and also included scores for the correct destination heading, when being in close distance to the goal. The ROS implementation of the planner only takes the goal position into account and disregards the goal orientation until the robot has reached the goal position. The robot then performs a turn in place in order to achieve the correct orientation. Additionally, the robot tends to drive diagonally instead of straight, since the magnitude of the velocity can be higher and velocities with non-negative y values are sampled first. To tackle these problems, two additional scoring functions were added to the evaluation for each velocity sample:

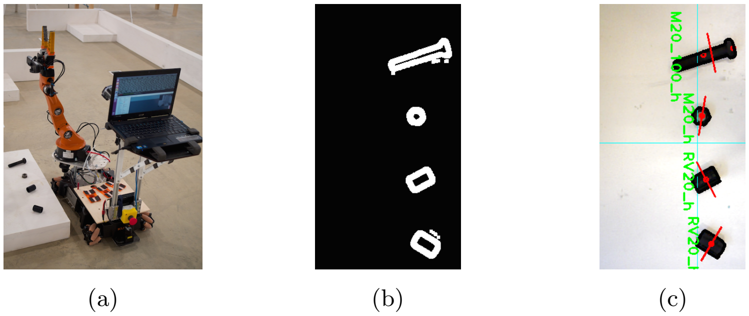
$$c_p = |y| * w_p \tag{1}$$

$$c_h = |\theta_r - \theta_g| * w_h \tag{2}$$

where c_p is the penalty for considering velocities that contain a y component. This ensures that the robot prefers driving straight instead of diagonally. c_h is the costs for the difference of the current heading in comparison to the goal heading. By adding this evaluation only when being reasonably close to the goal position, i.e. 1.5m, the robot starts to rotate towards the correct heading already before and does not need to turn in place as with the standard behavior of the planner. w_p and w_h are the corresponding weights for the two cost values.

Object/Hole Recognition Besides the navigation tasks, object detection and recognition is crucial to be able to interact with the environment, i.e. picking up objects and placing them in the correct target locations. We use the openCV-library[6] to detect the objects. We use two different detection pipelines, one is

[6] `http://opencv.org`



(a) (b) (c)

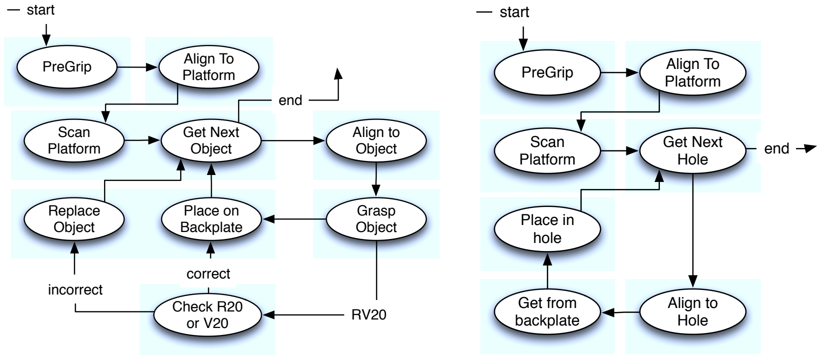
Fig. 4: (a) Pre-grip scan position. (b) Pre-processing of the image. (c) Detected objects, classification and grasp position of the closest object.

based on the depth image and the other is based on the RGB image. The depth image has the advantage that it is mostly independent of light conditions, while the RGB image is more accurate and allows to distinguish between black and silver objects.

An adaptive threshold filter is applied to the input image. Afterwards the image is converted into a black and white image and this is used to detect the contours of the objects as shown in Figure 4b. We use various features of the detected objects, e.g., length of principal axis, average intensity and area, and use a labeled data-set that we created on the location to train a J4.8 decision tree in weka [9] for the recognition of the objects. This decision tree is then used for the online classification of the objects. Figure 4c shows the detection in a service area.

In order to distinguish between the R20 and the V20 object, we use an additional camera on the back. The objects have the same shape from the outside, but inside there different, i.e. round or star-shaped as shown in Figure 3. This means that we have to pick the object up and hold in such that the additional camera can look inside the object. We use the Hough circle detection method [10] in order to determine if the inside of the object is round or star-shaped.

Recovery Methods Of course, in robotics many things can go wrong, so we include various recovery methods. The navigation can easily get stuck, since the robot is tuned to drive very close to obstacles. Therefore, we implemented a force field [11] recovery. More specifically, the robot moves away from every obstacle that is picked up by the laser scan for a couple of seconds and the navigation is restarted. The servos in the arm tend to overshoot sometimes, especially when they get hot. Since the new fingers are compliant, it is not too problematic to slightly hit the ground, but in some cases it can lead to an overload of the arm. Thus, we measure the voltages in the arm, and as soon as a certain voltage is exceeded, we recover the arm to an upwards position and back up a couple of states in the state machine to be able to try again.



(a) (b)

Fig. 5: (a) The simplified subtask-state-machine that is used for grasping objects at a service area. (b) The simplified subtask-state-machine that is used for dropping objects into the holes for PPT.

3.1 State machine

For the different tests in the RoboCup@Work competition, we combine the various capabilities explained in the previous section in different state machines to complete the tasks. For the BNT, we do not need any of the manipulation techniques, so we have a very small state machine that combines the navigation and the fine-positioning. More specifically, we use the navigation to get close to the position and then the fine-positioning moves the robot exactly to the asked position. If the navigation fails, the force-field recovery is activated. Afterwards it retries to move to the current location. When it fails more than a fixed number of times, the current goal is skipped.

In principle the BMT, BTT, PPT and the final tasks do not differ very much. For instance, the BMT is basically a BTT, in which only one source and one target location are used and the PPT is a BMT in which the objects have to be placed in a specific location instead of anywhere on the platform. Thus for all these other tests we created so-called subtask state-machines that are shown in Figure 5. For sake of simplicity, the recovery connections are left out. The subtasks are for example an behavior sequence for aligning to a platform a grasping all objects that need to be taken according to the task specification and placing them on the backplate of the robot as shown in Figure 5a. Before any object is taken, a full scan of the service platform is performed. This enables us to reason about the objects, if there are any classifications that are uncertain. For instance, if there is only one R20 or V20 found on the platform, and the specification says that we have to get one R20 from that service area, we know that the object should be a R20 and the extra check does not need to be performed. Another subtask is placing all objects that need to be precisely placed at this location, shown in Figure 5b. This task is very similar to the previously described behavior, since it is the reversed idea. The last subtask is the normal place of objects at target location. This is also similar to the PPT place, but since the objects can be placed anywhere on the platform, we only need to move once to the center of the platform and use the inverse kinematics of the arm to place the objects.

These subtask state machines are combined by a decision state that plans where to go and what to do next. This state gets all the necessary global and local information, i.e. which items are currently on the backplate, which items are still located in the arena, where is the robot located, etc. By looking at this information the best next location is chosen, i.e. to pick up more items, since there is space left on the backplate, or to go to a target area to (precisely) place items there. This state is called each time the subtask-state-machines are finished. If there is no item left on the backplate and also nothing more specified in the task description, the robot knows it is finished and moves out of the arena.

A video of the approach during a final run of the last RoboCup@Work world championships can be found here: `http://youtu.be/FzVgDqrWiOY`.

4 Future Work

The smARTLab@Work code and hardware designs will be released on Github[7]. For future competitions we plan on further improving the object recognition by adding state of the art techniques, such as deep learning algorithms in order to be even more robust against changing light conditions and also to be able to quickly adapt to new objects. Furthermore, we are currently working on a learning by demonstration approach to allow for learning new arm trajectories on the fly. This might enable grasping various 'more difficult to grasp objects', e.g. glasses. It will also help to cope with future RoboCup@Work tests, which are yet to be introduced, e.g. assembly tasks or picking up moving objects.

Additionally, this year we were almost able to compete on the highest difficulty level for the tasks, which includes the detection and grasping of upright standing objects. However, as the object recognition currently did not perform sufficiently reliable for this case, we opted to not tackle grasping upright standing objects in the competition.

Hardware-wise, we plan on mounting the arm on a rotating extension plate, so that the arm can be turned towards the optimal grasp position. Additionally, the backplate can be optimized, maybe even by including servos to hold different kinds of objects in place during transportation.

Lastly, we are looking into an open demonstration that involves the cooperation of many robots, e.g. some Turtlebots[8] for transportation and the youBot to grasp and place the objects on the Turtlebots.

[7] `http://github.com/smartlab-liv`

[8] `http://www.turtlebot.com`

References

1. TAPAS Project, "Robotics-enabled Logistics and Assistive Services for the Transformable Factory of the Future (TAPAS)," http://tapas-project.eu/, 2013.
2. G. K. Kraetzschmar, N. Hochgeschwender, W. Nowak, F. Hegger, S. Schneider, R. Dwiputra, J. Berghofer, and R. Bischof, "RoboCup@Work: Competing for the Factory of the Future," in *RoboCup Symposium*, 2014.
3. B. E. Hon, "Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base," 1975, uS Patent 3,876,255.
4. R. Bischoff, U. Huggenberger, and E. Prassler, "Kuka youbot - a mobile manipulator for research and education," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1–4.
5. S. Alers, D. Claes, J. Fossel, D. Hennes, K. Tuyls, and G. Weiss, "How to win robocup@work?" in *RoboCup 2013: Robot World Cup XVII*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, vol. 8371, pp. 147–158.
6. G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, 2007.
7. D. Fox, W. Burgard, F. D., and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99).*, 1999.
8. D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," in *IEEERobotics & Automation Magazine*, vol. 4, 1997.
9. M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, 2009.
10. L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016786559090042Z
11. N. Y. Chong, T. Kotoku, K. Ohba, and K. Tanie, "Virtual repulsive force field guided coordination for multi-telerobot collaboration," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA 2001)*, 2001.