A Failure-Tolerant Approach for Autonomous Mobile Manipulation in RoboCup@Work

Jan Carstensen[1](✉), Torben Carstensen[1], Simon Aden[1], Andrej Dick[1], Jens Hübner[1], Sven Krause[1], Alexander Michailik[1], Johann Wigger[1], Jan Friederichs[2], and Jens Kotlarski[3]

[1] LUHbots, Leibniz Universität Hannover, Hanover, Germany
carstensen@stud.uni-hannover.de, info@luhbots.de

[2] Hannover Centre for Mechatronics, Leibniz Universität Hannover, Hanover, Germany
friederichs@mzh.uni-hannover.de

[3] Institute of Mechatronic Systems, Leibniz Universität Hannover, Hanover, Germany
jens.kotlarski@imes.uni-hannover.de

Abstract. In this paper we summarize how the LUHbots team was able to win the 2015 RoboCup@Work league. We introduce various failure handling concepts, which lead to the robustness necessary to outperform all the other teams. The proposed concepts are based on failure prevention and failure handling.

1 Introduction

The RoboCup@Work league was established in 2012 [9] to foster the development and benchmarking of robots in the industrial environment. The main focus of the league is to improve small but versatile robots capable of doing many different tasks and, therefore, are interesting not only to huge firms which can afford many robots, but also to small companies. After introducing the league and tests performed in 2015, we present our approach, which this year has been focussed on robustness and failure handling.

2 LUHbots

The LUHbots team was founded in 2012 at the Institute of Mechatronic Systems Leibniz Universität Hannover, consists of bachelor and master students. Most of the founding team members have participated in the research inspired practical lecture RobotChallenge [11]. Nowadays the team is a part of the Hannover Centre for Mechatronics. The team consists of students from mechanical engineering, computer science and navigation and environmental robotics (see Fig. 1a). In 2012 the LUHbots team first competed in the RoboCup@Work challenge and was able to win the competition [10], in 2013 a second place was achieved [1]. In 2015 the LUHbots won both events, the German Open and the RoboCup in Hefei.

L. Almeida et al. (Eds.): RoboCup 2015, LNAI 9513, pp. 95–105, 2015.
DOI: 10.1007/978-3-319-29339-4_8

(a) The LUHbots Team in Hefei

(b) The RoboCup@Work League

Fig. 1. RoboCup@Work in 2015

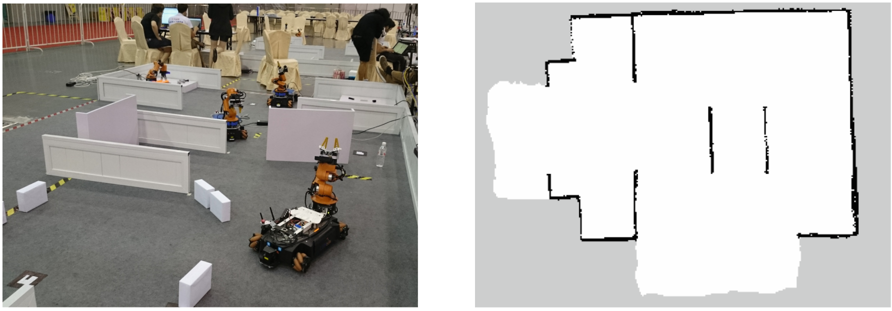
3 RoboCup@Work

In this section we introduce the tests of the 2015 RoboCup@Work world championship. The competition focussed on transportation tasks. In 2015 the rules allowed for picking complexity levels per test [12]. Six teams participating at the world cup in Hefei (see Fig. 1b).

3.1 Tests

In the following we are only going to discuss the complexity levels chosen by the LUHbots.

Basic Navigation Test: The purpose of the Basic Navigation Test (BNT) is testing navigation in a static environment. The arena is initially known and can be mapped during a set-up phase (see Fig. 2). The task consists of reaching and covering a series of markers in a specified orientation and covering a marker completely. In order to increase the complexity, obstacles are positioned in the arena. The position of the unknown obstacles is static.



(a) The arena during testing

(b) Map used for navigation

Fig. 2. The RoboCup@Work Arena in 2015

Basic Manipulation Test: The Basic Manipulation Test (BMT) focusses on manipulation tasks. The objective is to successfully grasp three objects and place them on a nearby service area. We increased the complexity level by choosing the hardest position, rotation, order and all decoy objects. Thus the complete set-up is defined by chance.

Basic Transportation Test: The Basic Transportation Test (BTT) combines manipulation tasks and navigation tasks. A task description is sent to the robot. The description includes information of starting and end positions for the objects to be transported. The task order and the specific transport tasks have to be determined autonomously by the robot. In order to increase complexity we had to pick objects in a randomly determined order, position and rotation. Furthermore the highest amount of decoy objects was placed on the service areas. We then had to place them according to specification. After placing all objects the robot has to leave the arena.

Precision Placement Test: The Precision Placement Test (PPT) consists of transporting objects and placing them inside small cavities, which are only a few millimetres larger than the object. An initially unknown position of the cavities increased the complexity.

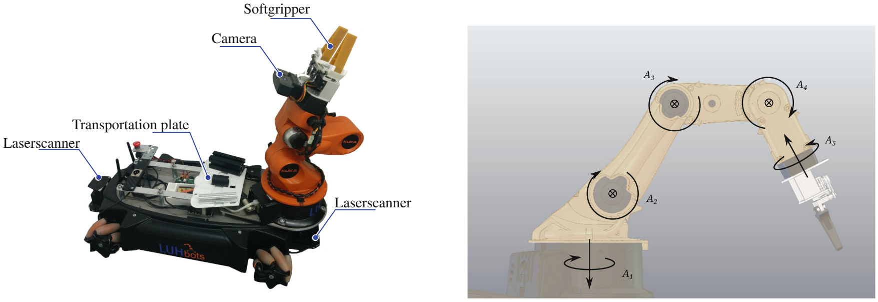
Final: Traditionally the final is a combination of all the above mentioned tests performed on the event. In 2015 the final task consisted of an extended BTT with ten objects. Some objects needed to be placed according to the PPT rules. To further increase the complexity level it was possible to add obstacles. We performed with high manipulation complexity but without additional navigation obstacles.

4 Hardware

Our robot is based on the mobile robot KUKA youBot (see Fig. 3) [2]. The robot consists of a platform with four meccanum wheels [8] and a five degrees of freedom (DoF) manipulator. Additional a gripper is attached at the end of the manipulator (see Fig. 3). The internal computer of the youBot has been replaced by an Intel Core i7 based system. In addition, the robot is equipped with an emergency stop system, allowing for keeping the platform and the manipulator in the actual pose when activated. The manipulator has been remounted to increase the manipulation area. The hardware itself does not offer failure tolerance, this is only achieved in combination with software.

4.1 Sensors

The youBot is equipped with two commercial laser range finders (Hokuyo URG-04LX-UG01) at the platform's front and back. A RGB-D camera (Creative Senz3D) mounted on the wrist of the manipulator (see Fig. 3a).



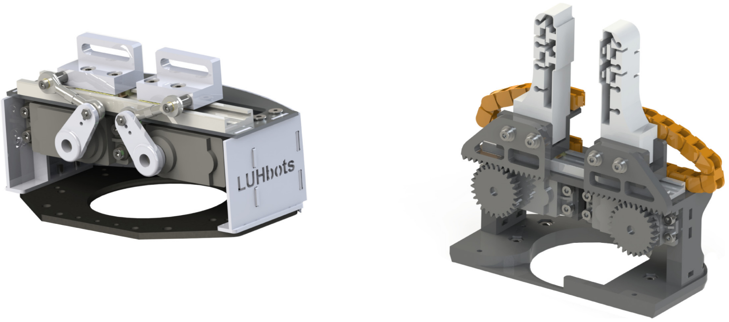
(a) LUHbots youBot

(b) The youBot manipulator

Fig. 3. Hardware overview

4.2 Gripper

One of the major hardware advances performed by the team is the development of a custom gripper. The original gripper has a low speed and stroke. As a result, it is not possible, to grasp all objects defined by the RoboCup@Work rule book, without manually changing the gripper-fingers. Besides the limited stroke, the low speed limit does not allow for an appropriate grasping of moving objects. Even though, in the 2015 competition the Conveyor Belt Test, has not been performed, the hardware design is optimized to meet future requirements. An advancement was to include force feedback into the gripper. Thanks to the integrated feedback within our custom made gripper, we are able to verify performed grasps. If a failure occurs during grasping, we are able to recover. In the current version the gripper uses soft-grippers to allow for a better handling of all objects (see Fig. 4a). A different approach using hall effect sensing has been tested (see Fig. 4b).



(a) Force sensing gripper, without soft-fingers

(b) Advanced sensing gripper

Fig. 4. The LUHbots grippers

5 Approach

We take advantage of an open source software framework called Robot Operating System (ROS) [14]. We used the Indigo release in 2015. Since 2012 we have a series of custom approaches. Each new development is tested extensively in predefined test scenarios before being included in our competition code base. This way we can guarantee a good robustness. Robustness, fail-safes and recovery behaviours are the cornerstone of our development.

5.1 Overview

Since our software architecture is based on ROS, different nodes are used (see Fig. 5). The yellow nodes are drivers they give access to the sensors. The youBot driver in red, can be accessed via the youBot OODL node. The camera data is first processed by the vision node and then filtered and clustered by the observer node, which is triggered by the state machine. The laser scanners are publishing to the navigation stack and the navigation watchdog. The watchdog filters the navigation commands. The task planner and the referee box connection communicate with the state machine. The laser scanner nodes are used unmodified. The ROS navigation stack is used but the global and local planners have been replaced. The youBot OODL driver is heavily modified. All other Nodes are developed entirely by the team.

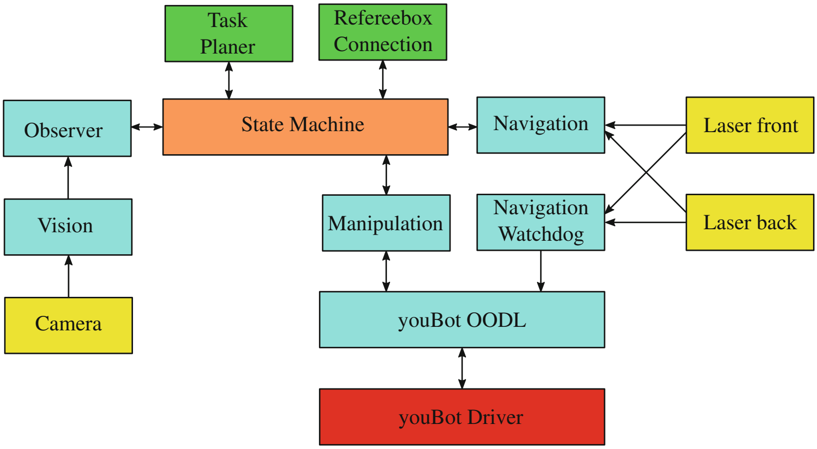
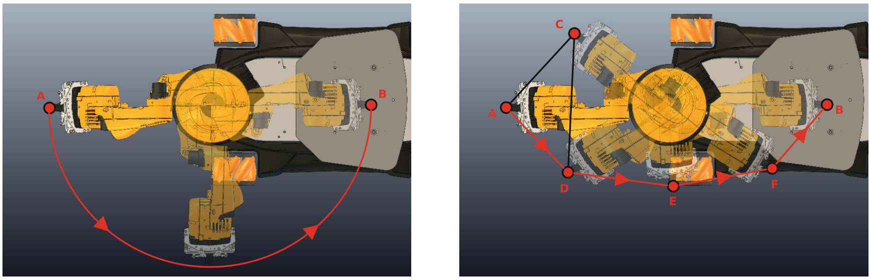


Fig. 5. Overview of the software architecture (Color figure online)

5.2 Manipulation

During the last year we developed a new software system that can be seen as a software development kit (SDK) for manipulation tasks with the youBot. The aim was to facilitate the development of applications for the youBot by



(a) Interpolation based planning

(b) Graph based path planning

Fig. 6. Graph based approach for the path planning, thanks to the proposed approach (b) a shorter motion is executed

providing advanced functionality for the manipulator and the mobile platform combined with user friendly interfaces. Some of the features for the manipulator are: inverse kinematics, path planning, interpolated movement in joint- and task-space, gravity compensation and force fitting. Features for the mobile platform include incremental movement, collision avoidance and movement relative to the environment based on laser scans. The provided interfaces contain a documented API and a graphical interface for the manipulator. In the RoboCup we use this software e.g. to grab objects using inverse kinematics, to optimize trajectories and to create fast and smooth movements with the manipulator. Besides the usability the main improvements are the graph based planning approach (see Fig. 6) and the higher control frequency of the base and the manipulator. Planning on a graph which is based on known and, therefore, valid positions leads to a higher robustness. Using an A*-approach the best path is generated [6]. The higher frequency leads to better executed motion plans and an overall smooth and more accurate motion.

5.3 Navigation

The navigation is based on the ROS navigation stack. The main improvement has been done in the local and global planners. The global planner has been extended to calculate the orientation for each pose of the global plan (see Fig. 7). The plan provided by the global planner is executed by a local planner with high reliance on the global plan. Since the RoboCup@Work arena is mapped before the tests, only a few obstacles are unknown at the beginning of the run. After a short time, the complete arena is mapped including additional obstacles and, therefore, the global plan is very close to the optimal path. Besides improving parts of the navigation stack we implemented a watchdog which operates based on the laser scanner data and is therefore much faster than a costmap-based local planner. The watchdog reduces velocities if an obstacle is too close, or permits the execution of a movement command if a collision would be eminent.

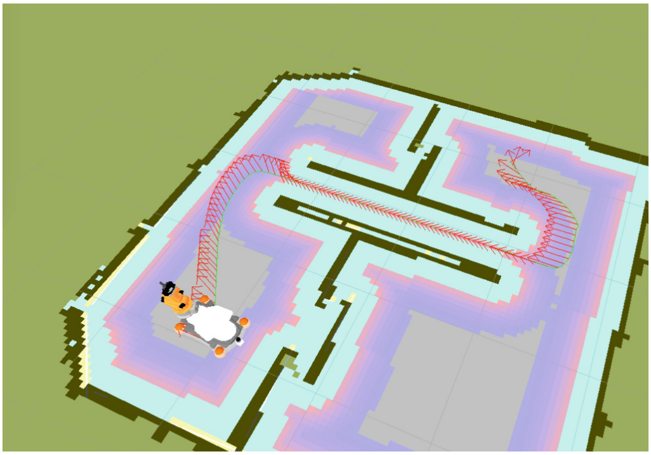


Fig. 7. Global plan with orientations, from start to end pose considering obstacles

5.4 Vision

We use the Creative Senz3D for object recognition, witch has two basic advantages in comparison to similar devices. Firstly it works at close range. Secondly, it is relatively small. Since the camera is not intended to be used in high precision tasks, the obtained 3D points are too noisy to be used directly for object recognition. Instead, we use the 2D images of the infra-red and RGB camera to segment the image, to extract features and to classify the objects. From the infra-red image first the objects are separated using the canny algorithm [4]. Then, the objects are then classified using Hu-moments and a random forest classifier [7,15]. Finally, the 3D points are used to determine the object's position and orientation. In order to get a robust vision system that can handle miss detections and which can memorize detected objects, all detections are clustered using a modified version

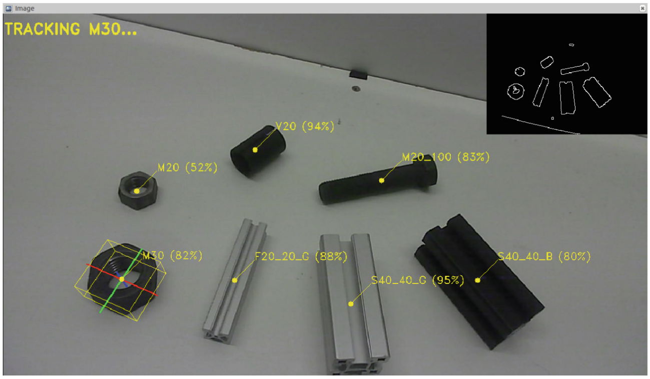


Fig. 8. Detected objects, classified and scored

of DBSCAN [5]. Each cluster is weighed, filtered and the positions are averaged. Then, the clusters are classified as objects or as failures (Fig. 8).

5.5 Task Planning

Our task planning is based on a graph based search. In each step all known service areas are used as possible navigation tasks. All objects on the back of the robot (there are up to three allowed) are used as possible placing tasks and the objects on the service area are used as grasping tasks. A greedy-based planning [13] is used up to a max depth and repeated until a complete plan is produced. The greedy algorithm is based on a cost function taking the time to perform the task, the probability to fail and the expected output. For the navigation tasks the distances are precomputed based on the known map. The manipulation time costs are averaged based the last respective manipulation action. When the state machine is not able to successfully recover a failure, the task is rescheduled and replanned increasing the probability to fail.

5.6 State Machine

The state machine is based on SMACH [3]. Which is a python library for building hierarchical state machines in ROS. Due to the capabilities of SMACH our state machine is modular and consists of the main components task planning, task execution, navigation and manipulation (see Fig. 9). The state machine acts as an action client, which sets the goals in navigation and manipulation to accomplish the tasks and receives feedback in case of issues. The state machine is designed for recovery. Each state is analysed. Foreseeable failures are considered. Depending on the failure a direct recovery is applied, the current task will be retried or postponed and tried again later, respectively.

The subroutine for manipulation is basically a linear sequence of actions with several cascaded loops that are repeated if an action fails. The idea is to detect failures as soon as possible and try to recover immediately. If the recovery fails after multiple attempts, the higher level recovery loop is repeated. A manipulation task is defined by a list of objects and an action to be performed with them. For a picking task e.g. the robot first approaches the service area and moves the arm to perform a scan of the objects. The scan is repeated until all objects are found with sufficient certainty or until a maximum number of scan movements is reached. Then, the found objects are grasped and placed on the cargo beginning from left to right. For grasping, the robot moves sideways to the object, then a second close scan is performed to verify the object and to further improve pose estimation. If the certainty of the scan is too low, the scan is repeated from a slightly changed perspective. If the scan fails multiple times, the object is postponed. Otherwise, the object is grasped and the force feedback is evaluated to verify that the grasp was successful. Depending on the result the object is either placed on the cargo or the close scan loop is repeated. If there are any postponed objects left in the end, the outer loop is repeated two times

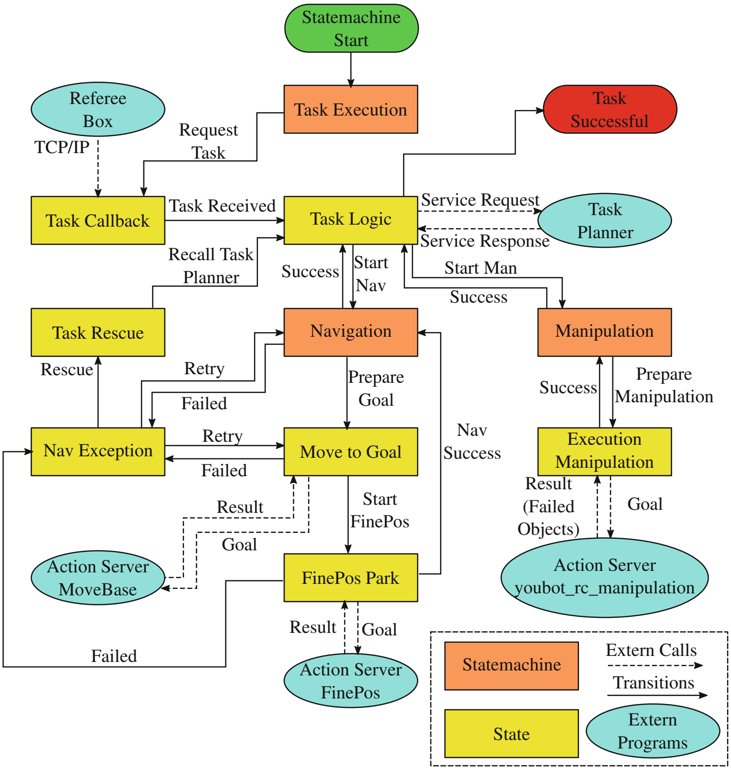


Fig. 9. State machine

with a preceding movement first to the left and then to the right. All postponed objects are reported back to the superior state machine.

6 Results

The results can be seen in Table 1.

Table 1. Results of the RoboCup@Work competition in 2015

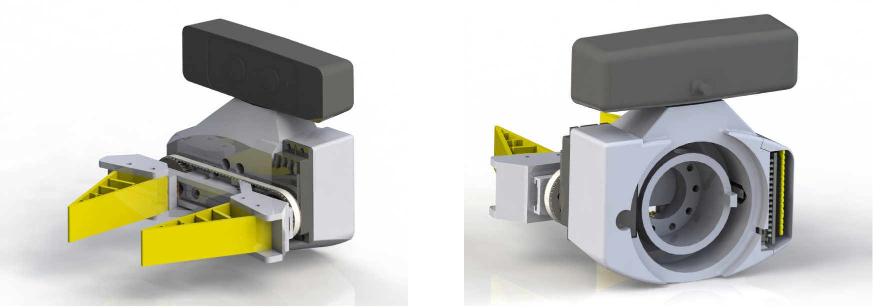
Place	Team	BNT	BMT	PPT	BTT	BTT2	Semi Final	Final	Total
1	LUHbots	280	480	630	1265	1920	2050	1840	8465
2	Robo-Erectus	0	480	1405	0	1785	1700	825	6195
3	b-it-bots	575	0	170	420	1020	1105	1615	4905
4	KeJia Worker	260	530	240	270	225	450	600	2575
5	WF Wolves	180	0	0	210	375	600	525	1890
6	robOTTO	195	0	0	0	0	560	375	1130

7 Conclusion

In our opinion robustness through failure tolerant approaches are the key the succeed in RoboCup. Based on different approaches we were able to improve our overall stability. Every failure or miss behaviour ever occurred is either fixed or a recovery behaviour is created to minimise the consequences of the failure. Furthermore, the overall vision approach lead to a robust object recognition. Even though our segmentation and classification had significant problems with the service area's surfaces, our combination of different scan poses, the filtering and clustering done by the observer resulted in an appropriate solution. Besides being able to recover, we had a very fast and optimised manipulation which was able to perform grasps faster then all the other teams and, therefore, giving us an edge. Even though we had a stable navigation the speed was rather slow.

8 Future Work

Even though we already have a good stability we are going to further increase our testing scenarios and the recovery behaviours. The navigation is a topic to work on, even though we are still focussing on a stable collision free navigation, we would like to improve the speed. Since the scenarios are increasing in complexity, we work on improving our task-planner and plan to test different approaches. Changing the vision system is planed. Another major point we will work on will be focussed on improving the gripper we are using. Even though we had a good performance we are looking forward to further increase the speed and robustness. A new gripper is in development, using only one servo motor and including a controller to further increase reliability (see Fig. 10).



(a) Front view

(b) Back view

Fig. 10. Model of the next generation gripper

Acknowledgements. We would like to thank a couple of institutes and persons supporting our work. The team is supported by the Institute of Mechatronic Systems, the Institute of Systems Engineering Real Time Systems Group, the student affairs office of the faculty of mechanical engineering, the society for the promotion of geodesy and geoinformatics and the Hannover Centre for Mechatronics. The team is being supervised by Jan Friederichs, Johannes Gaa and Daniel Kaczor.

References

1. Alers, S., Claes, D., Hennes, D., Tuyls, K., Fossel, J., Weiss, G.: How to win RoboCup@Work? In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013. LNCS, vol. 8371, pp. 147–158. Springer, Heidelberg (2014)
2. Bischoff, R., Huggenberger, U., Prassler, E.: Kuka youbot - a mobile manipulator for research and education. In: ICRA (2011)
3. Bohren, J., Cousins, S.: The SMACH high-level executive [ROS news]. IEEE Robot. Autom. Mag. **4**(17), 18–20 (2010)
4. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996). pp. 226–231. AAAI Press (1996)
6. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
7. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory **8**(2), 179–187 (1962)
8. Ilon, B.: Directionally stable self propelled vehicle, US Patent 3,746,112, 17 July 1973
9. Nowak, W., Dwiputra, R., Schneider, S., Berghofer, J., Hegger, F., Bischoff, R., Hochgeschwender, N., Kraetzschmar, G.K.: RoboCup@Work: competing for the factory of the future. In: Bianchi, R.A.C., Akin, H.L., Ramamoorthy, S.,Sugiura, K. (eds.) RoboCup 2014. LNCS, vol. 8992, pp. 171–182. Springer, Heidelberg (2015). http://dx.doi.org/10.1007/978-3-319-18615-3_14
10. Kaczor, D., et al.: RoboCup@Work league winners 2012. In: Chen, X., Stone, P., Sucar, L.E., van der Zant, T. (eds.) RoboCup 2012. LNCS, vol. 7500, pp. 65–76. Springer, Heidelberg (2013)
11. Munske, B., Kotlarski, J., Ortmaier, T.: The robotchallenge - a research inspired practical lecture. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1072–1077, October 2012
12. Nowak, W., Kraetzschmar, G., Hochgeschwender, N., Bischoff, R., Kaczor, D., Hegger, F., Carstensen, J.: Robocup@work rule book (2015). http://www.robocupatwork.org/
13. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall Inc, Upper Saddle River (1982)
14. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
15. Statistics, L.B., Breiman, L.: Random forests. In: Machine Learning, pp. 5–32 (2001)