Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-9 (2019) 153–158

Using robotics to train students for Industry 4.0

Elisa Tosello * **Nicola Castaman** *,** **Emanuele Menegatti** *

* *Intelligent Autonomous Systems Lab (IAS-Lab), Department of Information Engineering, University of Padova, Via Gradenigo 6/B, 35131 Padova, Italy*
** *IT+Robotics srl, Contrà Valmerlara 21, 36100 Vicenza, Italy (e-mails: {toselloe, castaman, emg}@dei.unipd.it)*

Abstract: This paper presents the master course on Autonomous Robotics that we offer at the School of Engineering of the University of Padova (Italy). Its novelty is the assignment of a lab project carefully designed to train students on autonomous and industrial robotics in the framework of Industry 4.0: the "Industry 4.0 Robotics Challenge". Students have to program both a manipulator and a mobile robot, together with a 3D vision system, in order to collaborate in the fulfillment of a pick-place-transport industrial task. We adopt a constructionist approach: project-based learning and team-based learning are applied to robotics and Industry 4.0. The project is organized as a challenge to motivate students to propose innovative ideas. A survey on students' satisfaction is reported at the end of the paper. We made the description of both the hardware and software setup, together with tutorials and wikis, publicly available to let other robotics instructors replicate our proposal and make it a point of reference for teaching robotics in the frame of Industry 4.0.

Keywords: Education, Robotics, ROS, Industry 4.0, Robotic manipulation, Robot navigation

1. INTRODUCTION

The fourth industrial revolution, called Industry 4.0 (1), is underway. It sees the coming of Cyber-physical systems (CPS): computer-based machines, coupled with physical and software components, able to interact with each other, perceive their surroundings, and reason on assigned tasks. These systems operate within the production chain but do not slavishly repeat the same pre-programmed routines. They adapt their operations depending on the kind and type of goods they are producing by communicating and collaborating with each other and with human operators.

According to this prespective, the new generation of engineers should be able to integrate multidisciplinary and cross-domain knowledge. They should cope with new paradigms and concepts (e.g., modeling, simulation, interoperability and self-organization) and emergent technologies (e.g., IoT and Artificial Intelligence). They have to deal with the system's software implementation instead of mainly focusing on its hardware counterpart (2). These considerations triggered the re-design of the *Autonomous Robotics* course: a second year course of the Master of Science (MSc) in *Computer Science* at the School of Engineering of the University of Padova (Italy). In particular, we re-designed the laboratory activity proposed by the course itself: it now asks students to program two different robots in order to fulfill an assembly task in collaboration with a human operator (3). This means that students have to get in perspective of one single complex task that has to be subdivided into simpler sub-tasks, each of them has to be assigned to one single robot depending on its capabilities. This complex task is then achieved thanks to an efficient robot-robot and human-robot collaboration. Not only the laboratory activity was changed from the past years, but the entire course evolved from a course centered on classical mobile robotics to a course centered on intelligent robots, as such kind of robots conduces to the Industry 4.0 revolution.

In previous years, students had to solve six homework, homogeneously assigned throughout the duration of the course (4). Every lab asked students to face one single robotics problem by means of one robot: among others, we asked to solve a navigation problem by means of a Lego Mindstorms NXT robot, a manipulation problem by means of a Universal Robots UR10, and a teleoperation problem by means of a humanoid Robovie-X robot. In this way, students were able to program complex tasks with single, complex robots. However, they lacked a global view on how intelligent robots could be exploited in Industry 4.0. Thus, we decided to propose a single complex assignment to show how robotics systems should cooperate to fulfill complex tasks.

The rest of the paper is organized as follows. Sect. 2 shows related works on this field. Sect. 3 gives an overview of the *Autonomous Robotics* course by describing the content of both the lectures and the laboratory assignment, together with required prerequisites and skills. Sect. 4 and 5 detail the laboratory assignment. First, the intended learning outcomes are depicted. Then, both the hardware and software setups are given. Sect. 6 analyzes best solutions proposed by students of the A.Y. 2017/2018. Sect. 7 contains results in terms of student satisfaction. Finally, Sect. 8 contains conclusions and future work.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.08.185

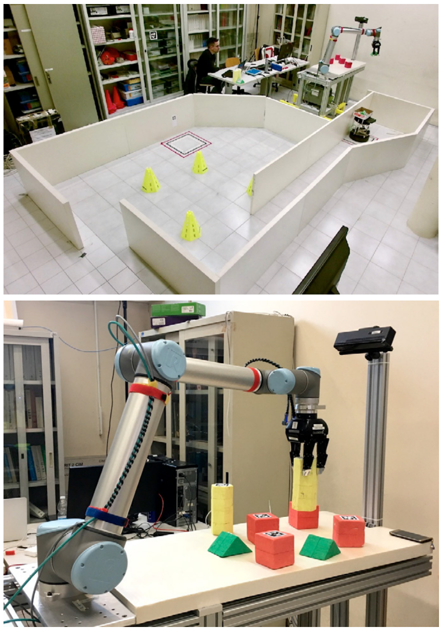

Fig. 1. The Industry 4.0 Robotics Challenge arena: on the top we show the global setup, on the bottom we zoom on the objects to be picked up and the picking station.

2. STATE OF THE ART

In the latter years, there has been an evolution on requirements needed to program a robot: from robots that had to accomplish single tasks in isolation to robots that have to communicate and collaborate with each other in Smart Factories of Industry 4.0. The same evolution is required on robotics courses offered by colleges and universities: teachers do not have to focus on classical robotics, they have to offer a multidisciplinary experience able to train the workers of the future. With this aim, several long-term and short-term academic learning programs emerged, having the purpose of offering a multidisciplinary robotics specialization (5; 6; 7; 8). Focusing on long-term programs, Duckietown (8) aims to train engineers capable of responding to the current needs of the autonomous automobiles market. The Robotic Decathlon (7), instead, offers a six-months introductory course composed of a set of lab projects, each one teaching a different aspect of robotics, from manipulation to navigation, and performed by teams of two students. Focusing on short-term programs, the Robot Operating System (ROS) (9) Summer School [1] is an interesting initiative. It lasts two weeks and provides both an introduction to the most important ROS packages and a deep overview of mobile robot routines, i.e., perception, localization, and navigation. At the end of the school, there is a competition in which participants, divided into teams, must develop a mobile robotics application. In (2), instead, Leitão recalls the Erasmus Intensive Program on Robotic Systems (10) and the Summer School on Industrial Agent in Automation (11). The first is a 60-hour course that provides a global understanding and practical experiences in industrial robotics, autonomous mobile robotics and robotics applied to medicine. The latter is a 28-hour course focused on multi-agent systems applied to industrial environments. To the best of our knowledge, we are not aware of a semester-long learning course focused on Industry 4.0. In particular, we did not find in literature a learning project that fuses all aspects of Industry 4.0 in a single project as the one we are proposing in this paper.

We are proposing a mixed approach that combines the benefits of competitions with the requirements of multidisciplinary teaching in the context of Industry 4.0. Indeed, competitions are widely recognized as strategies for motivating students and making the learning experience more extensive (12; 13). We can personally confirm this statement thanks to our experience in RoboCup (14) and in the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) (15). In order to solve the assignment, we ask students to program using ROS. Indeed, ROS is open source, fine-grained, and it consists of numerous reusable modules. By using it, students learn how to organize software in modules, reuse the structure and classes of data, and take advantage of class inheritance (16). ROS also provides tools and libraries to get, create, write and execute code on multiple computers with a powerful communication protocol. Therefore, students can easily implement a network of robots (and computers) that can exchange data and interact with each other.

3. THE COURSE

Autonomous Robotics is a second-year elective of the Master of Science (MSc) in *Computer Science* at the School of Engineering of the University of Padova (Italy). This master aims to train students in the fields of design, engineering, production, operation and maintenance of computer and information systems, computer laboratories, and company information systems, both in the context of industrial production and services. In order to clarify the final expertise of students enrolled in this master, *Programming of embedded systems*, *Network modeling*, *Internet of Things and Smart Cities*, *Distributed Systems*, *Computer Vision* and *Machine Learning* are other offered courses. MSc students of *ICT for Internet and Multimedia*, *Mathematical Engineering*, *Automation Engineering* and *Electronic Engineering* of the same School can also enroll for *Autonomous Robotics*. The course lasts 12 weeks and offers students methodological bases for programming autonomous robotic systems by combining class lectures and a laboratory assignment. While lectures give students a theoretical background on robotic fundamentals, the lab lets students apply these fundamentals for the resolution of a robotic task. Students are examined both on lectures (30% of the final grade) and on the programming assignment (the remaining 70%). As in competitions, the assignment is subdivided into sub-tasks and a scoring schema is made available to students, assigning a specific score to each sub-task.

3.1 Prerequisites and skills

No prerequisite is mandatory for course attendance. However, having basic C++ programming capabilities facilitates the implementation of the lab. Indeed, ROS accepts both C++ and Python commands, but we suggest the use of C++ because a course on C++ programming is offered within our same course of study.

[1] see `https://www.fh-achen.de/fachbereiche/maschinenbau-und-mechatronik/international/ros/`

3.2 Lectures

Three classes of two hours per week are taught. They give an introduction of both mobile and manipulator robots with a deep overview on the relationship between perception and action in robotic systems, existing algorithms for solving motion planning problems, and learning methods for the autonomous execution of assigned tasks. During class lessons, five tutorials of two hours each introduce students to ROS programming. Tutorials give an overview of the ROS architecture and on how to implement publishers, subscribers, actions and services. They teach how to use the ROS navigation stack, MoveIt!, and how to implement advanced perception routines by means of the AprilTags Visual Fiducial System (17) and the Point Cloud Library (PCL) (18). Tutorials give attendants the skills needed to face all project's sub-tasks.

3.3 Lab project

Course teachers and Teaching Assistants (TAs) designed the Industry 4.0 Robotics Challenge as follows: multiple different objects are placed on a table in front of a robot manipulator, and a vision sensor is mounted on this table. The manipulator has to detect these objects, pick them up, and place them on the top of a mobile robot previously docked next to the manipulator (see Fig. 1). The mobile robot has first to go through a narrow passage 5 cm larger than its base. Then, it has to navigate through an open arena, populated by obstacles (see Fig. 1), until reaching a second docking area where one student has to pick the objects and assemble a simple construction. Students have to cope with the following constraints:

(1) A timeout is imposed (30 minutes).
(2) Pieces are subdivided into three categories, each with a peculiar shape, color, and weight (red cubes, green triangles, and yellow prisms). Every category is associated with scores proportional to the grasping difficulty.
(3) The area on the top of the mobile robot is limited: students have to implement an optimal policy to transport all pieces in the minimum number of travels and to maximize their scores.
(4) The mobile robot has to reach fixed positions in the map (docking areas) in order to have the objects loaded by the manipulator or unloaded by the human operator.
(5) The arena is populated by both fixed and movable obstacles.
(6) The mobile robot always starts from a prefixed pose.

Both a real and a simulated setup are provided to students (see Sect. 5). Course teachers and TAs realized, sometimes mixing packages already available in ROS, all tools students need to interact with these environments (e.g., the packages suitable for the creation of the virtual environment, the robots bringup, the network connection and the manipulation and navigation configuration). After testing, these packages are provided to students so that they can focus only on the implementation of the perception, manipulation and navigation routines that perform the challenge. They have to estimate the pose of the objects. They have to move both the manipulator and the mobile robot. They have to implement a Finite State Machine able to coordinate robots while they are accomplishing their own sub-tasks. Every programming effort is proportionally scored in order to reward those students who adopt innovative solutions. As solutions should be innovative but working, we mediate the degree of difficulty with the time required to complete the project. Setting up the Challenge makes teachers and TAs aware of its sticking points. In this way, they will be able to empathize with the students and anticipate the time-consuming issues that the students will encounter.

4. LAB PROJECT: EDUCATIONAL DETAILS

We adopt a constructionist approach and we combine project-based, team-based, and peer-based learning. In detail, a complex, single project is assigned to students (*Project-based learning*). Students organize themselves in teams and every team has to propose its own solution (*Team-based learning*). We suggest groups of two/three people: larger groups induce confusion and unbalanced workload division within the group itself. Students can access the lab and use available robots every day during the course. One TA is available to them in case they need help. No more than one TA per time is required because students should act by themselves: during the class, tutorials are provided on how to perform different sub-tasks of the assignment. In the lab, every group decides how to face these sub-tasks, tries to implement their choices, and discusses what improvements are feasible (*constructionist learning*). Students have to present their own solutions within three months after the end of the course. At the exam, they must provide both a demonstration proving the correct functioning of their systems and an oral presentation motivating their implementation choices. For those students who want to present the project exactly at the end of the course, a challenge is organized. The winning team gains two more points on the final grade. In our opinion, facing a challenge motivates students to present innovative solutions and to take the exam by the end of the course (*Peer-based learning*).

Our Intended learning Outcomes (ILOs) follow. As for (8), we subdivide them into four categories: *operational tools*, *development methods*, *autonomy* (perception, control, coordination) and *documentation*. By the end of the course, students should be able to:

(1) Use *operational tools*, included Ubuntu and ROS.
(2) Acquire knowledge on *development methods*, such as:
  - (DEV-ROS) develop ROS software modules and integrate them in their systems;
  - (DEV-Tools) utilize standard tools for software development (e.g., source code repositories; branching and merging);
  - (DEV-Open) familiarize with the dynamics of open-source development, including the challenge of integrating independently-developed functionalities.
(3) Implement features guaranteeing robots *autonomy* by:
  - (AU-Perception) guaranteeing the autonomous detection and recognition of manipulation objects;
  - (AU-Control and Motion Planning) guaranteeing the autonomous manipulation and navigation. Collisions should be avoided, and the mobile robot should autonomously face the narrow passage;

- (AU-Coordination) guaranteeing the autonomous coordination and cooperation of available robots.

(4) Write an exhaustive *Documentation* by:
- (DOC-Presentation) preparing a presentation with design choices, system benefits, limitations and ideas on how to overcome these limitations;
- (DOC-Documentation) commenting their code and creating step-by-step instructions to enable future users to reproduce their systems.

5. LAB PROJECT: IMPLEMENTATION DETAILS

This section describes the hardware and software setups needed to reproduce our laboratory assignment.

5.1 Hardware setup

Adopted hardware (see Fig. 1) is low-cost, highlighting our aim of making an industrial setup easily reproducible. Details of each component follow, based on the routines it accomplishes.

(1) *Manipulation*: we have a UR10 manipulator robot with a Robotiq 3-Finger gripper attached on its end effector.
(2) *Navigation*: we have a Turtlebot 2 mobile robot with a box mounted on its top letting the transportation of collected objects. Turtlebot 2 has to navigate within an arena built of extruded polystyrene panels. In it, traffic cones are placed in order to simulate obstacles.
(3) *Perception*: Focusing on the perception needed for the manipulation routine, we mounted a Microsoft Kinect v2 on the table in front of the manipulator robot. This visual sensor streams both a 2D image in Full HD resolution and a depth image at 512x424 pixels. The information is streamed at 30 Hz. Students can choose which data to use to estimate the 6D pose of the objects on the table knowing that these objects have different colors and Apriltag markers [(19)] attached on their tops. Focusing on the navigation, we mounted a Hokujo URG-04LX-UG01 and a webcam on the Turtlebot. They can be used to map the environment and to detect obstacles and docking stations even when light conditions are disadvantageous.
(4) *Communication*: A desktop computer acts as ROS master and it is connected both to the robot (UR10 and gripper) and the Kinect. A laptop, instead, is used for Turtlebot 2 and its sensors. PCs can communicate with each other thanks to a network built ad hoc for the Challenge. It is composed of both a switcher and a router so to form both a cabled and a wireless network to which students can connect their laptops too.

5.2 Software setup

The project is based on ROS. We adopted Ubuntu 16.04 and ROS Kinetic: the last LST version at the time of project formulation. A simulated version of the real-world setup is provided to let students test their systems without the need of being in the lab. In this way, they can correct possible implementation errors before testing their proposal on real robots. Gazebo is used as simulator (20) as it is the official ROS simulator. The list of software packages needed to reproduce the project follow, subdivided according to the routines they accomplish.

(1) *Manipulation*: in order to control the manipulator arm, we provide the ROS-Industrial `universal_robot`, the `ur_modern_driver` (21), and the `robotiq` packages. We also make available the MoveIt! configuration of the provided setup.
(2) *Navigation*: the Turtlebot 2 ROS packages and its navigation stack are provided to move the mobile platform inside the arena while avoiding obstacles.
(3) *Perception*: to control the Kinect, we provide the IAI Kinect2 library. Students can use both the AprilTag library and PCL to accomplish the object detection.

Packages and the simulation setup are available through Bitbucket[2]. In this way, it is easy to solve bugs and distribute software updates. Moreover, it is easy for other institutions to replicate our assignment.

6. PROPOSED SOLUTIONS

In A.Y. 2017/2018, 16 students attended the course and passed the exam. 15 of them were MSc students of *Computer Science* and 1 was enrolled in *Electronic Engineering*. Below we report the two best solutions they proposed.

One team completed the lab by choosing the minimum difficulty coefficient in favor of a reliable and fast solution. For object detection, they exploited the AprilTag library, MoveIt! was used to plan the manipulation, and the ROS navigation stack was used for the navigation. They accurately turned navigation parameters to avoid collisions. To go through the narrow passage, they measured the passage width and combined this information with that extracted from the laser in order to implement a routine letting the robot travel a path equidistant from the passage walls. They completed the project within the end of the course, faced the challenge, and succeeded at their first run in about 20 minutes.

The second-best solution was the most innovative one. The team exploited objects' colors for object detection. MoveIt! was used for solving the manipulation problem and an ad-hoc routine was implemented for the navigation part. This routine exploited potential fields to face both obstacle avoidance and narrow passage. However, 3 additional months were required to complete the project and, during the trials, some penalties were assigned due to mobile robot inaccuracies.

These proposals confirm the lessons that we learned during the MBZIRC Challenge (15): a good trade-off has always to be found between a reliable, fast to implement, solution and an innovative, efficient, but maybe unstable, one.

7. STUDENTS SATISFACTION

At the end of the A.Y. 2017/2018 course, we asked students to compile an anonymous questionnaire and have feedback, on a 1-5 scale, on the perceived quality and usefulness of both teaching and lab experience. Feedback should highlight if goals of Sect. 4 were achieved. 9 students out of 16 completed the questionnaire. Obtained results demonstrate that students were overall happy with the experience: 58.3% of them assigned an average grade

[2] See `https://bitbucket.org/account/user/iaslab-unipd/projects/ROB2018`

Fig. 2. The challenge phases. From the left: obstacle avoidance, narrow passage, objects manipulation and delivering

Table 1. Most significant questions of the questionnaire assigned to students.

N	Question
Q1	Before starting the course, I had good programming skills.
Q2	Before starting the course, I was able to use ROS.
Q3	Before starting the course, I had knowledge of robotics.
Q4	ROS tutorials are enough to face the assigned lab.
Q5	The lab assignment lets me apply the theory taught in class.
Q6	The lab complexity proved discouraging and was difficult to manage.
Q7	The lab assignment asked for a lot of work, but this work is necessary if you want to become competitive in robotics.
Q8	Working with real robots is more interesting and challenging than working in simulation.
Q9	The overall lab experience was useful in terms of personal gratification and growth.
Q10	Working in teams was effective for achieving the assignment: we achieved results that I would not have reached alone.
Q11	By working in teams, I improved my ability to split tasks among people.
Q12	By working with ROS, I gained new programming skills.
Q13	After this lab experience, I'm able to use ROS.
Q14	Working with real robots made me face practical problems useful for my future job.
Q15	Programming an industrial robot make me competitive for my future job.
Q16	The course stimulated my robotics interest so that I decided to continue with a course of study/work based on these topics.
Q17	After this lab experience, I fell I have acquired the necessary skills to be included in a Smart Factory of Industry 4.0.

of 8.0 (out of 10.0) to the course, meaning that they were satisfied on how the course was held. 33.3% of students assigned a grade greater than 8.0 and only 8.6% of them assigned a grade lower than 6.0. An average grade of 8.46 has been assigned to the educational action, highlighting that teachers correctly stimulated and motivated the interest in the robotics discipline, beyond they clearly explained the course contents. Table 1 collects most significant questions, whose answers are reported in Fig. 3. In detail, students started the course with average programming skills (Q1), no knowledge of ROS (Q2), and a very low perception of what robotics was (Q3). They attested that, at the end of the course, their programming capabilities enhanced (Q12) and they recognized themselves as novice ROS programmers (Q13). The project is complex, but this complexity did not discourage students from facing it (Q6), they know that a lot of effort is needed to become competitive in robotics (Q7). Moreover, being able of autonomously carrying out a complex task gave a feeling of personal

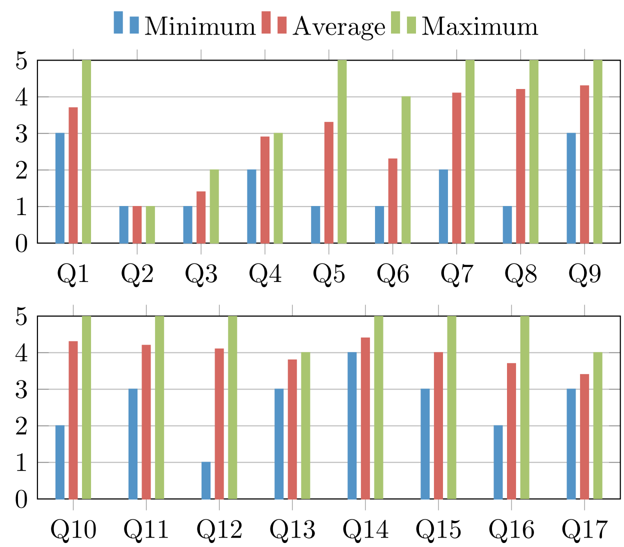

Fig. 3. Results of the survey: degree of agreement of the student (1-5) for each question (Q1-Q18).

grow and gratification (Q9). Focusing on the teachers' contribution, they were able to keep students' interest high by adopting a constructivist team-based approach (Q10, Q11) and employing real robots instead of a simulated setup (Q8). Indeed, working with real robots was perceived as useful for a future job (Q14, Q15).

The most critical feedback was related to the theoretical and tutorial support that teachers provided during the course. Results (Q4, Q5) highlight a margin for improvement on the integration of theory and tutorials in the hands-on activities. We believe this was a consequence of this being the pilot edition of the project: many structural "one-time" tasks had to be accomplished. E.g., setting up real and simulated environments, developing the necessary software, preparing wikis and necessary tutorials. In subsequent editions, less time and effort will be necessary for these tasks in favor of the integration between theory and practice.

As a result, students' robotics interest was stimulated in such a way that they decided to embark on a robotics course of study/job (Q16). Moreover, they felt they acquired good Industry 4.0 capabilities (Q17).

8. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the laboratory project proposed by the *Autonomous Robotics* course of the M.Sc. degree in *Computer Science* at the School of Engineering of the University of Padova (Italy). The project asks students to program two robots to cooperate for object picking and transportation in the context of Industry 4.0. The project is assigned to students in terms of challenge. A constructivist approach is adopted based on a project-based, team-based, peer-based learning. The effectiveness of our proposal is proved by students' satisfaction through an anonymous questionnaire. We made software and hardware setup, together with provided tutorials, publicly available. In this way, this project can be replicated by other institutions and can be adopted as a point of reference for teaching Industry 4.0. In the future, we plan to introduce both a more effective human-robot interaction and the use of Cloud-based services for AI and robotics. These are two other pivotal points of Industry 4.0.

REFERENCES

[1] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative industrie 4.0 – securing the future of german manufacturing industry," final report of the industrie 4.0 working group, acatech – National Academy of Science and Engineering, Apr 2013.

[2] P. Leitão, "Engineering education efforts to support industry 4.0," in *CLME2017/VCEM - 8º Congresso Luso-Moçambicano de Engenharia / V Congresso de Engenharia de Moçambique*, Sept 2017.

[3] E. Tosello, N. Castaman, S. Michieletto, and E. Menegatti, "Teaching robot programming for industry 4.0," in *Educational Robotics in the Context of the Maker Movement. International Conference of Educational Robotics (EDUROBOTICS), 2018. Proceedings of*, vol. 946, Springer, 2018.

[4] E. Tosello, S. Michieletto, and E. Pagello, "Training master students to program both virtual and real autonomous robots in a teaching laboratory," in *2016 IEEE Global Engineering Education Conference, EDUCON 2016, Abu Dhabi, United Arab Emirates, April 10-13, 2016*, pp. 621–630, 2016.

[5] M. Guo, L. Husman, N. Vullum, and A. Friesel, "Project in robotics at the copenhagen university college of engineering," in *IEEE International Conference on Robotics and Automation (ICRA), 2004. Proceedings. ICRA'04. 2004*, vol. 2, pp. 1375–1380, IEEE, 2004.

[6] N. Correll, R. Wing, and D. Coleman, "A one-year introductory robotics curriculum for computer science upperclassmen," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 54–60, 2013.

[7] D. J. Cappelleri and N. Vitoroulis, "The robotic decathlon: Project-based learning labs and curriculum design for an introductory robotics course," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 73–81, 2013.

[8] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, D. Hoehener, S. Liu, M. Novitzky, I. F. Okuyama, J. Pazis, G. Rosman, V. Varricchio, H. Wang, D. Yershov, H. Zhao, M. Benjamin, C. Carr, M. Zuber, S. Karaman, E. Frazzoli, D. D. Vecchio, D. Rus, J. How, J. Leonard, and A. Censi, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1497–1504, May 2017.

[9] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[10] P. Leitão, J. C. Fraile, V. Moreno, R. Harrison, H. Altun, A. W. Colombo, J. P. Turiel, and B. Curto, "Transnational lifelong education course in robotic systems," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 004181–004186, Nov 2015.

[11] P. Leitão, L. Ribeiro, J. Barata, and B. Vogel-Heuser, "Summer school on intelligent agents in automation: Hands-on educational experience on deploying industrial agents," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6602–6607, Oct 2016.

[12] P. Fiorini and D. Kragic, "Education by competition," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, p. 6, 2006.

[13] M.-T. Chew, S. Demidenko, C. Messom, and G. S. Gupta, "Robotics competitions in engineering eduction," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pp. 624–627, IEEE, 2009.

[14] D. Nardi, I. Noda, F. Ribeiro, P. Stone, O. von Stryk, and M. Veloso, "RoboCup Soccer Leagues," *AI Magazine*, vol. 35, no. 3, pp. 77–85, 2014.

[15] N. Castaman, E. Tosello, M. Antonello, N. Bagarello, S. Gandin, M. Carraro, M. Munaro, R. Bortoletto, S. Ghidoni, E. Menegatti, and E. Pagello, "RUR53: an unmanned ground vehicle for navigation, recognition and manipulation," *CoRR*, vol. abs/1711.08764, 2017.

[16] S. Michieletto, S. Ghidoni, E. Pagello, M. Moro, and E. Menegatti, "Why teach robotics using ros?," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 8, no. 1, pp. 60–68, 2014.

[17] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, May 2011.

[18] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), May 9-13 2011.

[19] E. Olson, "Apriltag: A robust and flexible visual fiducial system.," in *ICRA*, pp. 3400–3407, IEEE, 2011.

[20] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Sendai, Japan), pp. 2149–2154, Sep 2004.

[21] T. T. Andersen, "Optimizing the universal robots ros driver.," tech. rep., Technical University of Denmark, Department of Electrical Engineering, 2015.