



Robot@Factory Lite: An Educational Approach for the Competition with Simulated and Real Environment

João Braun^{1(✉)}, Lucas A. Fernandes¹, Thiago Moya¹, Vitor Oliveira¹,
Thadeu Brito², José Lima^{2,3}, and Paulo Costa^{3,4}

¹ Federal University of Technology, Paraná, Curitiba, Brazil
jabraunneto1@gmail.com, lucas_azevedo33@hotmail.com,
vfelipeoliveira@hotmail.com, thiagomoya@ymail.com

² Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Instituto Politécnico de Bragança, Bragança, Portugal
{brito,jllima}@ipb.pt

³ Centre for Robotics in Industry and Intelligent Systems - INESC TEC,
Porto, Portugal

⁴ Faculty of Engineering of University of Porto (FEUP), Porto, Portugal
paco@fe.up.pt

Abstract. Teaching based on challenges and competitions is one of the most exciting and promising methods for students. In this paper, a competition of the Portuguese Robotics Open is addressed and a solution is proposed. The Robot@Factory Lite is a new challenge and accepts participants from secondary schools (Rookie) and universities. The concepts of simulation, hardware-in-the-loop and timed finite state machine are presented and validated in the real robot prototype. The aim of this paper is to disseminate the developed solution in order to attract more students to STEM educational program.

Keywords: Robotics competition · Simulation · Factory logistics

1 Introduction

Robotics competitions are one of the methodologies that drives technology development, as robotics competitions encourage students and researchers to develop new ways to solve a task. Examples such as robotic soccer, autonomous driving and among others have contributed to the advancement of algorithms that are later used in both industry and service robotics. Besides, it is well known that robotics competitions captivate the students attention, improve their intrinsic motivation and skills and also improves teamwork and social collaboration. An example of this incentive for the development of robotics through competition can be seen in [1], which analyzes the educational performance of engineering students in the First Lego League (FLL).

Having as base the Robot@Factory competition that started in 2011 Portuguese Robotics Open, the Robot@Factory Lite (R@FL) is a simplified version where the parts should be moved between warehouses and processing machines through a magnet bringing this competition more accessible for younger students. The organization provides a prototype and some libraries to deal with the robot I/O (magnet, RFID, motors, etc.). This paper presents a solution for this R@FL competition, based on the proposed prototype, that uses both Hardware-in-the-loop (HIL) and real robot approaches. The results show that the adopted solution solves the challenge. It is desired to present a solution that encourages further students to participate in robotics competition, namely the R@FL. This competition is aligned with the STEM topics, i.e., the proposed challenges encourage students to approach the topics of science, technology, engineering and mathematics.

During the competition, the students develop skills such as communication skills (they have to socialize and present the developed work), problem-solving skills (problems appear during the competition), teamwork skills (the students need to work together to develop solutions and compete), self-motivation (student motivation increases as they are facing real problems), and conflict resolution (if a problem appears, the students should work to solve it), among the others.

The paper is organized as follows: after this introduction, the related work is presented in Sect. 2. Section 3 addresses the rules and simulation environment. The adopted solutions for all rounds are addressed in Sect. 4 whereas results and conclusions with future work are stressed in Sects. 5 and 6 respectively.

2 Related Work

The challenges presented in robotic competitions provide the opportunity for researchers, students and enthusiasts to come up with creative solutions. Over the years, the methods and solutions found are vast, and therefore, it is important to have a benchmark of the methodologies developed [2]. By means of metric comparisons of the performances of the participating teams, [3] demonstrates the difficulty in judging the approaches of the competitors during the competition. Maritime robots can have various shapes, sizes, and application solutions, therefore the measurement during the challenges of the euRathlon 2015 competition was based on references from previous editions. Service robot challenges are made in RoboCup@Home based on household activities. For over seven years, the challenges of domestic robots have been influenced by the complexity and performance of tasks [4].

Benefiting of the challenge proposed in the Robot@Factory competition as a test, [5, 6] demonstrates a platform for navigation, control, and localization of Automatic Guided Vehicle (AGV) robots. With the purpose of encouraging the development of human-robot cooperation applications, the RoCKIn@Work competition challenges its competitors to optimize small and medium factory processes [7]. Since the challenge is to simulate a real shop-floor situation, the developed system is able to avoid obstacles, determine the position of the mobile robot

and indicate the paths it must take to get the product to be processed. By comparing Student Autonomous Underwater Vehicles Challenge - Europe (SAUC-E), An Outdoor Robotics Challenge for Land, Sea and Air (EURATHLON) in the 2014 and 2015 editions, and The European Robotics League (ERL) EMERGENCY 2017, [8] demonstrates the instability in the scoring and judging system of the teams during the challenges. While some systems demonstrate to favor the implementation of the application, other systems favor the development of the application project. On the other hand, participants are free to undertake any approaches as long as they respect the rules of each competition.

3 The Competition and Simulation

The R@FL competition has the objective to stimulate students and researchers to develop solutions to the challenges it presents [9]. The idea behind the challenge is an AGV to organize the materials in warehouses with processing machines. The layout of the competition can be seen in Fig. 1.

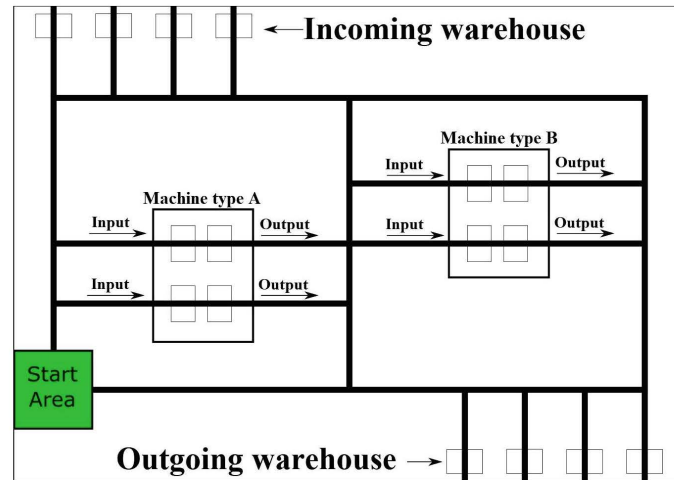


Fig. 1. Schematic of the competition environment [9].

As Fig. 1 illustrates, there are four incoming and outgoing warehouses alongside with two machines, A and B. In the incoming warehouses, there will be boxes that an AGV must deliver them to their correct locations simulating a real working warehouse. The outgoing warehouses represent the final destination to the processed products. In the same reasoning, machines A and B process the materials, and in this concept, they are pre-conditions for some of the materials before going to the outgoing warehouse, i.e., some materials need to be processed before being delivered to the outgoing warehouse. The start area can be chosen in the southwest or the northeast as the environment is symmetric in X and Y axes (machine A is always the machine near the start area). Thus, the competitors must implement an AGV capable of autonomously move, identify,

manipulate and deliver the type of materials to their correct locations. There are three types of materials in the competition, which are represented by parts. The parts projected can be seen in Fig. 2.

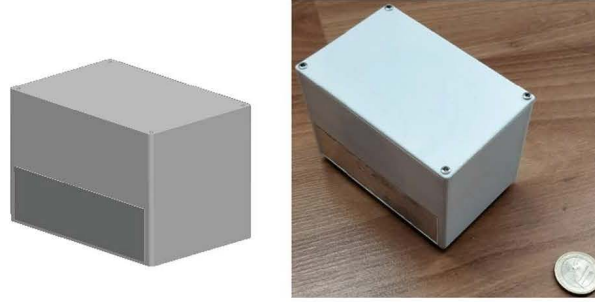


Fig. 2. Simulated and real parts with an RFID tag inside box em behind grey area [9].

The boxes dimensions can be seen in the competition rules [10]. As can be noted in Fig. 2, the boxes have a metal plate (grey area) and, this way the AGV can manipulate them by using electromagnets. Inside them, above the grey area, there are Radio Frequency Identification (RFID) tags. Thus the AGV, equipped with an RFID reader can identify the type of box that will be manipulated. In this sense, the types of boxes can be seen in the Table 1:

Table 1. Type of parts and its destinations.

Type of box	Destination
Raw	Machine A
Semi-processed	Machine B
Processed	Outgoing warehouse

As Table 1 displays, if the box identified is a raw one, this box will need to be processed by the two machines (A, B) before going to the warehouse. Therefore, if the box is a semi-processed material, it will be necessary just to process it through machine B before delivering to the outgoing warehouse. Finally, if the box is a processed one, the only task necessary is to deliver correctly to its final destination.

The competitors are free to implement the AGV they find suitable to the competition as long as the robot does not violate the dimensions rules [10]. However, the competition provides not only the full project (parts, bill of materials, project archives) for the recommended AGV but also a full manual that covers the implementation of the robot step by step. This was done to facilitate the integration of students to the competition. The recommended robot that was implemented can be seen in Fig. 3.

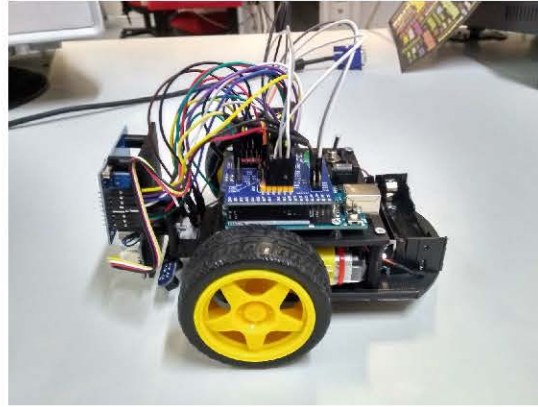


Fig. 3. The real AGV.

3.1 The Simulation

The competition also provides a simulation model. The SimTwo simulator provides a simulation considering the dynamic constraints that the real scenario has. Therefore, the simulator has a realistic 3D model of the robot and the competition scenario [11]. All those data can be freely modified, the graphic part in XML language and the simulator script in Pascal language. Therefore, the teams can adapt their robot model if they made a different one and, if they wish, modify the script provided by the competition staff. However, it is not necessary as all the tools needed are already coded. Thus, the competitors can validate their solutions easier and faster in the simulation before taking it to the real scenario. The simulation environment can be seen in Fig. 4.

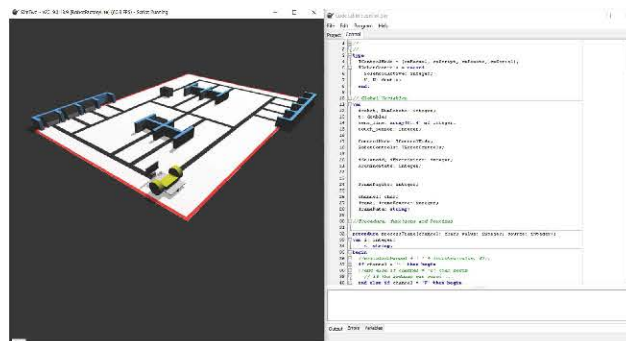


Fig. 4. Simulation environment. Left window is the graphic environment. Right window the code editor.

Although the simulator is realistic, it does not consider the microcontroller limitations such as available memory and processing speed. In this way, the competition staff provided a HIL tool coded in the simulator [9] as well. Therefore, the competitors can insert their microcontroller in the loop of the simulation,

i.e., they program their solution to their microcontroller and then, by serial communication (USB), the microcontroller is inserted to the simulator loop. In this sense, the simulator sends the sensor data (line sensor, electromagnet and the micro switch) to the microcontroller and the information is processed. Soon after, the microcontroller sends the motors speeds to the simulator which is then processed dynamically and graphically. The main loop in the simulator runs every 40 ms. In this way, the HIL loops roughly the same time. Figure 5 demonstrates this program loop.

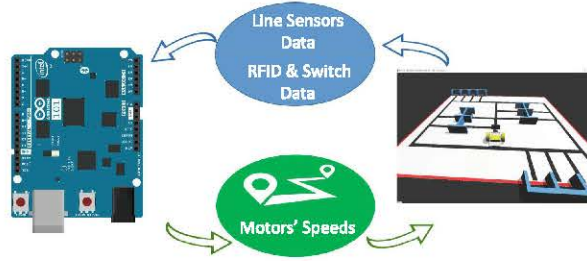


Fig. 5. HIL illustration within the code, provided by the organizers, it is possible to configure between the real movements or HIL mode. Adapted from [9].

4 Adopted Solutions

In this Section, the adopted solutions used in the three rounds of the R@FL competition are presented. As stated in the official rules [10], the boxes were identified through an RFID tag to differentiate the product type that they contain. However, in the first round this feature was not used because all boxes were processed materials and consequently had the same destination. The following subsections show the logic of the developed code and the states that the robot performs through illustrative figures.

4.1 First Case

To perform the first case, it was adopted the Timed Finite-State Machine (TFSM) approach. This technique and the code was provided by the competition organizer. The TFSM idea is simple, each path that the robot executes compose a different state and each state is achieved through a selected action. As an example, the pick-and-place of the first box using the technique explained is shown in Fig. 6 and described below.

The initial state of the pick-and-place movement consists of going straight until the robot reaches the box, then, when the robot touches the box, the microswitch is triggered and the second state is activated. Thus, the box is coupled to the robot through activating the electromagnet. After that, the third state is started and the robot drives backwards. When the conditions for changing this state are true, the fourth state is activated and an 180° turn is performed.

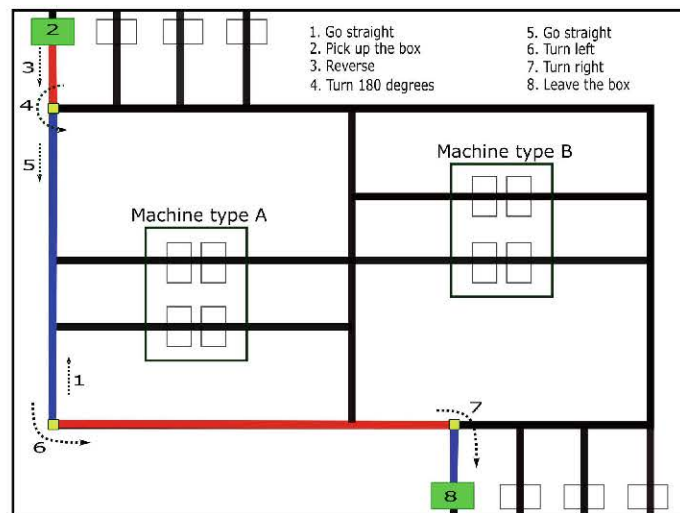


Fig. 6. Example of a path to deliver the first box.

Soon after, the fifth state is started, and the “go straight” command is given. When the requirements to activate the sixth and the seventh state are achieved the “turn to the left” and “turn to the right” commands are executed. Hereupon, the eighth and last state is accomplished leaving the box in the outgoing warehouse. The procedure for the other boxes is similar. Basically, a TFSM is based on the current state and the transitions between states when one or more conditions are satisfied.

4.2 Second Case

The second round includes the semi-processed materials and, consequently, new tasks to be performed. These boxes must be collected in the incoming warehouse and processed by either of the two machines, as can be seen in Fig. 8. From the different RFID tags, the robot must be able to identify the processed and semi-processed parts to correctly deliver the boxes. In this case the use of the TFSM is not recommended because the code would be large and inefficient resulting in a high memory consumption.

After studying the trajectories of the robot for all possibilities, it was noticed that several paths were repeated. For example, the path to go from the incoming warehouse to the outgoing warehouse is almost the same for all situations, as can be seen in Figs. 7 and 8. Thus, a generic path-travel function has been created. For each case the relevant information such as velocity and trajectory were transmitted to the function by parameters. In addition, the cases are determined according to the type of box, so the algorithm is able to identify and make the right decision in all situations.

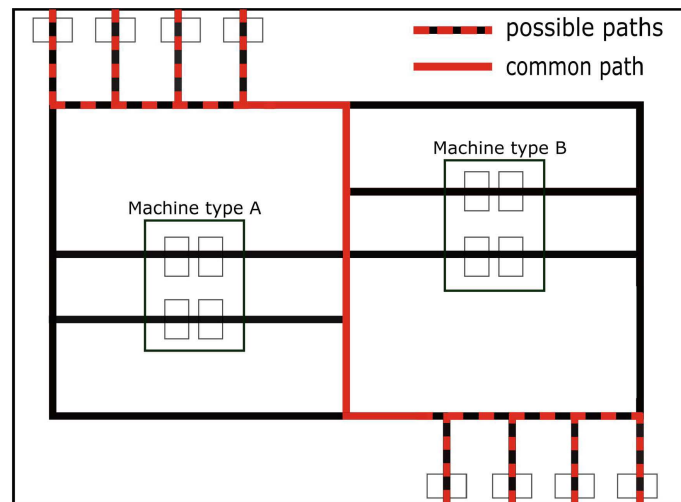


Fig. 7. Example of a path to deliver the processed parts.

The developed algorithm was ready to handle all position probabilities of all types of boxes for the first and second round. This means that if only the processed parts had been placed in the incoming warehouse, the robot could collect them and deliver them to the output.

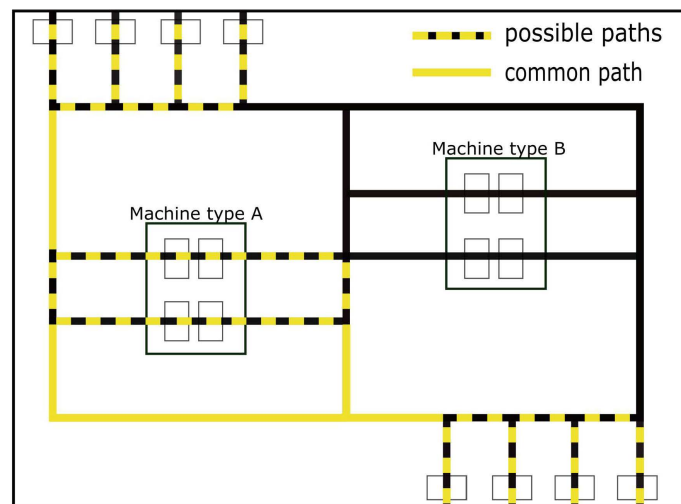


Fig. 8. Example of a path to the semi-processed parts.

4.3 Third Case

In the third round of the competition, the raw parts box was included. These boxes should be collected in the incoming warehouse, then be processed by Machine type A, after by Machine type B and finally delivered to the outgoing warehouse. From the logic used in the second round it was possible to develop

the algorithm for the third round by including new paths and reusing the previously established paths. The new paths were necessary to allow passage through machine type B, as shown in Fig. 9.

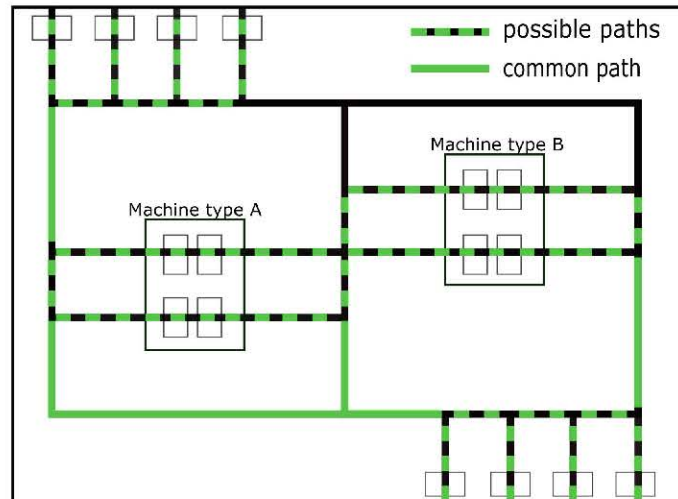


Fig. 9. Example of a path to the raw parts.

5 Results

The main objective of the experimental tests is to verify the performance of the robot and if the execution time of each round is in agreement with the time limit fixed by the rules. In this section will be presented the pseudo-codes of the implemented algorithms, the comparison between the applied methods, the memory consumption and the robot's performance in practice.

The first tests consisted of verifying the basic functions provided to competitors such as turn left or right, turn 180°, go straight, and drive backwards. Then the appropriate settings were set to adjust the movements of the robot. After that, the simulator was applied to become familiar with the competition scenario, as well as the possibility of performing the experiments without the physical robot. This resource was widely used in the initial tests. After the familiarization with the resources given by the competition, the development of the codes to solve the factory problems started. As previously mentioned, the technique used in the first case is different from the second and third cases. In the first one a TFSM technique was employed. To complete all steps, 61 states and 51% of the microcontroller's memory (SRAM) were used. This data is an estimated value extracted from Arduino IDE (Integrated Development Environment).

Algorithm 1. Main Function

```

1: function MAIN FUNCTION
2:   Go straight until touch the box
3:   Box ← Part_Type
4:   if Box == Processed Part then
5:     function ROUTE(1)
6:     end function
7:     function ROUTE(2)
8:     end function
9:   else if Box == Semi – Processed Part then
10:    function ROUTE(3)
11:    end function
12:    function ROUTE(4)
13:    end function
14:    function ROUTE(5)
15:    end function
16:    function ROUTE(2)
17:    end function
18:   end if
19: end function

```

The second technique was based on a single function that covered all the necessary paths, reducing the size and complexity of the code. The developed function applied was the switch case statement. In this sense, each path is described as a case, and, since the same path can be traveled more than once, the same case can be called several times. In the main function of the program, exemplified in Algorithm 1, the detection of the product type is performed through an RFID tag reading function and the execution of the pick-and-place process through the Route function, shown in Algorithm 2. The source code consumed 58% of the microcontroller SRAM but if the TFSM approach was used, that value would certainly be bigger.

Algorithm 2. Function ROUTE

```

1: function ROUTE(int NumCase)
2:   switch NumCase do
3:     case 1
4:       Pick up the box and leave in outgoing warehouse
5:     case 2
6:       Go back to incoming warehouse
7:     case 3
8:       Pick up the box and leave in the machine
9:     case 4
10:      Pick up the box in machine
11:    case 5
12:      Leave the box in outgoing warehouse
13:   end function

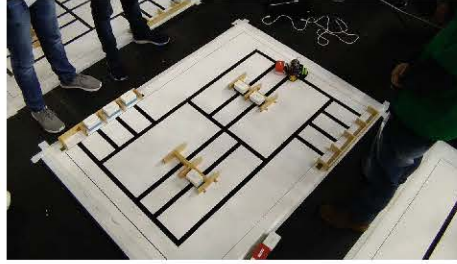
```

The third case was implemented based on the second round function, however, with some upgrades. For the robot to pick up the box and go through Machine A and Machine B, new paths were added. In the main function the novelty was the detection of a new RFID tag considering the raw parts. The source code consumed 67% of the microcontroller SRAM, a short increase of memory consumption in comparison with the second case.

Table 2. Performed times in the competition.

Round	Time	Boxes
1	2:00'57	4 blue boxes
2	3:48'80	2 blue boxes and 2 green boxes
3	2:58'00	1 red box, 1 green box and 1 blue box

The competition was divided into three days, which each day occurring a different round. In the first and second rounds the robot was able to pick-and-place all boxes required and, in the third round, the robot placed just 3 parts correctly. The performed times are presented in Table 2. Figure 10 shows the robot during the third round (real scenario).

**Fig. 10.** Real robot in the third round.

6 Conclusion and Future Work

This paper presented a solution for the R@FL competition of the Portuguese Robotics Open. The simulation environment and tools provided by the organization were used to develop strategies to complete the three rounds. The importance of HIL in mobile robot applications to optimize the time of project implementation was evidenced.

The proposed solution in the first round is based in a TSFM technique, which was a suitable option because this turn required few number of states. However, as the remaining rounds had a higher number of paths implying in a larger set of states, the TSFM technique was not possible. Therefore, the solution was to simplify the code using functions with the common paths to deliver the parts.

The approach by function while compared with the TFMS method presented some advantages, principally, the development time required and the computational consumption. The robot could not deliver the four boxes in the last round due to non-systematic errors (irregular floor and sliding of the robot). These errors jeopardized the runs as our robot's decisions were time dependent. As future work, it is proposed to include encoders on the shaft of the motors to improve the control by using odometry. This will allow better precision of movements and potentially discard the time dependency in the robot decision making.

Acknowledgements. This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project (POCI-01-0145-FEDER-006961), and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

1. Oppliger, D.: Using first Lego league to enhance engineering education and to increase the pool of future engineering students (work in progress). In: 32nd Annual Frontiers in Education, vol. 3, p. S4D. IEEE (2002)
2. Holz, D., Iocchi, L., Van Der Zant, T.: Benchmarking intelligent service robots through scientific competitions: the RoboCup@ Home approach. In: 2013 AAAI Spring Symposium Series (2013)
3. Petillot, Y., Ferreira, F., Ferri, G.: Performance measures to improve evaluation of teams in the euRathlon 2014 sea robotics competition. *IFAC-PapersOnLine* **48**(2), 224–230 (2015)
4. Iocchi, L., Holz, D., Ruiz-del-Solar, J., Sugiura, K., Van Der Zant, T.: RoboCup@ Home: analysis and results of evolving competitions for domestic and service robots. *Artif. Intell.* **229**, 258–281 (2015)
5. Costa, P., Moreira, N., Campos, D., Gonçalves, J., Lima, J., Costa, P.: Localização e navegação de um robô móvel omnidirecional: caso de estudo da competição robot@ factory. *VAEP-RITA* **3**(1) (2015)
6. Costa, P.J., Moreira, N., Campos, D., Gonçalves, J., Lima, J., Costa, P.L.: Localization and navigation of an omnidirectional mobile robot: the robot@ factory case study. *IEEE Revista Iberoamericana de Tec. del Aprendizaje* **11**(1), 1–9 (2016)
7. Bischoff, R., Friedrich, T., Kraetzschmar, G.K., Schneider, S., Hochgeschwender, N.: RoCKIn@ work: industrial robot challenge. In: RoCKIn: Benchmarking Through Robot Competitions, vol. 47 (2017)
8. Ferreira, F., Ferri, G., Petillot, Y., Liu, X., Franco, M.P., Matteucci, M., Grau, F.J.P., Winfield, A.F.: Scoring robotic competitions: balancing judging promptness and meaningful performance evaluation. In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 179–185. IEEE (2018)
9. Lima, J., Costa, P., Brito, T., Piardi, L.: Hardware-in-the-loop simulation approach for the Robot at Factory Lite competition proposal. In: 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 1–6. IEEE (2019)
10. Robot at Factory Lite Competition files. <https://github.com/P33a/RobotAtFactoryLite>
11. Piardi, L., Eckert, L., Lima, J., Costat, P., Valente, A., Nakano, A.: 3D simulator with hardware-in-the-loop capability for the micromouse competition. In: 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 1–6. IEEE (2019)