Teaching Practical Robotics During the COVID-19 Pandemic: A Case Study on Regular and Hardware-in-the-Loop Simulations

José Lima[1,2,4](✉), Felipe N. Martins[3], and Paulo Costa[4,5]

[1] CeDRI - Research Centre in Digitalization and Intelligent Robotics, Polytechnic Institute of Bragança, Bragança, Portugal
jllima@ipb.pt

[2] Laboratório para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Bragança, Portugal

[3] Sensors and Smart Systems Group, Institute of Engineering, Hanze University of Applied Sciences, Groningen, The Netherlands
fe.nascimento.martins@pl.hanze.nl

[4] INESC TEC - INESC Technology and Science, Porto, Porto, Portugal

[5] Faculty of Engineering of University of Porto, Porto, Portugal
paco@fe.up.pt

Abstract. Laboratory experiments are important pedagogical tools in engineering courses. Restrictions related to the COVID-19 pandemic made it very difficult or impossible for laboratory classes to take place, resulting on a fast transition to simulation as an approach to guarantee the effectiveness of teaching. Simulation environments are powerful tools that can be adopted for remote classes and self-study. With these tools, students can perform experiments and, in some cases, make use of the laboratory facilities from outside of the University. This paper proposes and describes two free tools developed during the COVID-19 pandemic lock-down that allowed students to work from home, namely a set of simulation experiments and a Hardware-in-the-loop simulator, accessible 24/7. Two approaches in Python and C languages are presented, both in the context of Robotics courses for Engineering students. Successful results and student feedback indicate the effectiveness of the proposed approaches in institutions in Portugal and in the Netherlands.

Keywords: Hardware-in-the-loop simulation · Robot simulation · Engineering education · Distance learning · COVID-19 pandemic

1 Introduction

Robotics is a multidisciplinary area of study that requires students to carry out practical experiments to fully understand and learn its concepts properly. When access to hardware is not possible, simulation can be used to promote practical education. Recently, the importance of simulation was greatly emphasized

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
D. Tardioli et al. (Eds.): ROBOT 2022, LNNS 589, pp. 538–549, 2023.
https://doi.org/10.1007/978-3-031-21065-5_44

during COVID-19 pandemic because social distancing rules made it difficult or impossible for practical classes to take place physically. During this period, many universities switched to teaching remotely, including practical/lab classes [7], and the use of simulators and on-line tools increased rapidly. The impact of such change on the quality of education will take a few years to be fully understood, but some recent studies indicate that undergraduate engineering students were happy with virtual laboratory settings [8], and perceived simulation software as supporting their learning to a moderate degree during the social distancing restrictions [1].

In this paper we describe two proposals that allow the practice of robotics concepts at home using simulation, and present results from a pool conducted with students regarding the use of simulation in one of our Robotics courses. One proposal is based on pure simulation and the other uses a hardware-in-the-loop (HIL) approach. Both proposals are especially useful when students have limited or no access to the laboratory facilities at the University. Comparing with ROS, Gazebo, RViz and similar tools, both proposed approaches are easier to install and use. The HIL just requires a microcontroller connected to the computer.

Considering the advantages of simulation, the first proposal is the *Robotics Simulation Labs*, a website with a set of tutorials for practicing concepts of mobile robot control with an open-source robot simulator. Its main goal was to replace the practical activities of the Robotics course at Hanze University of Applied Sciences, which were canceled due to Corona-related restrictions in 2020 and 2021. The Robotics course is an introductory-level course for engineering students, focusing on wheeled mobile robots and motion control. Its laboratory activities are intended to give some hands-on experience with a mobile robot while practicing concepts of differential drive kinematics, state-machine behavior selection, odometry-based localization, and simple motion controllers. The activities described in the *Robotics Simulation Labs* are presented as a series of tutorials intended to be followed in sequence, with some extra challenges for the students that want to develop further.

Despite of allowing students to learn at home without the need of dealing with hardware, pure simulators sometimes hide real hardware limitations because real platforms are equipped with embedded systems. To overcome such limitation and get closer to real conditions, the second proposal uses a hardware-in-the-loop (HIL) approach, in which a microcontroller is used to perform control, navigation, and decision tasks while interfacing with a simulated robot. The same microcontroller can also be used to control a real robot. According to [5], authors performed a review of physics simulators for robotic applications but the communication that allow the hardware-in-the-loop capability is neglected. The SimTwo simulation environment, proposed by [15], allows to perform communication with a serial connection and enables the possibility of send and receive information from an external microcontroller [9]. Moreover it allows to edit the model of the robot and change sensors. By this way, it is possible to implement the HIL. The proposed HIL environment allowed to replace the practical experiments of Robotics course at Faculty of Engineering of University of Porto and Polytechnic Institute of Bragança.

In this paper we describe both the *Robotics Simulation Labs* and the above mentioned hardware-in-the-loop system operating with SimTwo. Student feedback is also discussed in the context of the simulation labs. The main objectives of the paper are: (1) to describe two proposals that allow the practice of robotics concepts at home using simulation, and (2) to present the results of a pool carried out with students regarding the use of simulation in one of our courses of Robotics. Moreover, allow to improve the education quality in pandemic periods for the generations of students who are inserted in the technological culture, as well as in education 4.0.

2 Related Works

A great number of laboratory classes were quickly replaced by simulations [7], and even some robotics competitions adapted to include or expand the use of simulation in their portfolio, like RoboCup Junior Soccer [12] due to the COVID-19. Besides the sudden increase in use during the pandemic, simulation tools have been applied to promote practical education and access to educational robotics and competitions for many years [11,13]. In fact, computer simulations have been a valuable pedagogical tool in a wide variety of engineering disciplines for decades.

The hardware-in-the-loop methodology can be applied to obtain more realistic results from simulations. It was already applied in the last two decades in space robotics and On-Orbit Robotic Missions, such as proposed by [6]. HIL simulation was also used for other applications, like design and testing of robot manipulators [10] and control [3], focusing on its metamorphosis from a control validation tool into a system development paradigm.

Engineering education also benefited from the HIL simulation. As an example, a combination of physical, remote and virtual laboratories was used to teach automation engineering [16]. An approach in HIL simulation was introduced as an education tool in robotics, mechatronics and control, in the teaching of control and design aspects of on-site and remote robotics courses [17]. We also developed previous work in the HIL [4,9].

In the following sections we describe our proposals for applying simulation in robotics courses for engineering students. First, we describe the *Robotics Simulation Labs* that make use of an open-source robotics simulator, then we introduce new HIL approach using ESP32 microcontroller with PCRobot application to get data from the robot in real time. Other tools can be combined to control the robot, such as Matlab/Simulink.

3 Robotics Simulation Labs

Robotics Simulation Labs is a website with a set of tutorials for practicing concepts of mobile robot control. The activities are intended to give some robot programming experience while dealing with concepts of differential drive kinematics, state-machine behavior, odometry-based localization, and simple motion controllers. Those concepts are part of the learning objectives of the Robotics course offered to second year students of the Electrical and Electronics Engineering program at Hanze University of Applied Sciences, in the Netherlands.

The simulation labs were designed to be executed in Webots because it is a powerful, versatile, and open-source robotics simulator that can run on Linux, MacOS or Windows [14]. According to a recent review on simulators for robotics applications [5], Webots is a good option for simulating ground mobile robots with no soft-body contacts, which applies to our case. Robots simulated in Webots can be controlled by code written in many different languages, including Python. Besides being a popular language, our students follow a prior course on Python, so this language has been chosen for the simulation labs.

Figure 1 shows an impression of Webots while running one of the lab activities. The window in the center shows the simulated robot in its environment, which in this case is a rectangular space surrounded by walls, with flat white floor and a black line painted on it. The simulated robot is the circular object with a coordinate frame on top. A bottle used as obstacle can be seen in the center. On the left side several parameters of the simulation can be adjusted and monitored, like position, orientation and size of the objects, simulation time-step, program associated to the robot controller, etc. On the right side of the window, the Python code used to control the robot can be visualized and edited. At the bottom, a console can be used to print messages and data during the course of a simulation, which can assist on debugging. In Fig. 1, the console is printing the simulation time-stamp and the pose of the robot as calculated via dead reckoning every time-step (32 ms in this case).

There are many options of simulated robots already available in Webots, but the user can also build your own model. Simulated robots can have independently controlled actuators, several sensors, and physically interact with the environment. The robot used in most of the tasks is the e-puck, a small differential-drive robot originally developed for teaching purposes. In Webots, the e-puck model[1] includes simulation of the wheel motors, encoders, infra-red sensors used for proximity detection around the robot, ground sensors (capable of detecting a line) among other sensors not (yet) used in the simulation labs.

The lab activities were divided in five modules:[2] Lab 1 - Installation and introduction to Webots Robot Simulator; Lab 2 - Line-following behavior with State-Machine; Lab 3 - Odometry-based Localization; Lab 4 - Trajectory Tracking Controller; Lab 5 - Combine Behaviors to Execute a Complex Task.

Each lab has a dedicated page that includes its objectives, some background information, extra links (including Webots tutorials), a detailed description of the tasks to be executed by the student (including snippets of code), templates with bare-bones code (except for lab 5), and solution for the main assignment

[1] https://cyberbotics.com/doc/guide/epuck.

[2] The version at the time of writing this manuscript is available at https://github.com/felipenmartins/Robotics-Simulation-Labs/tree/ROBOT2022.

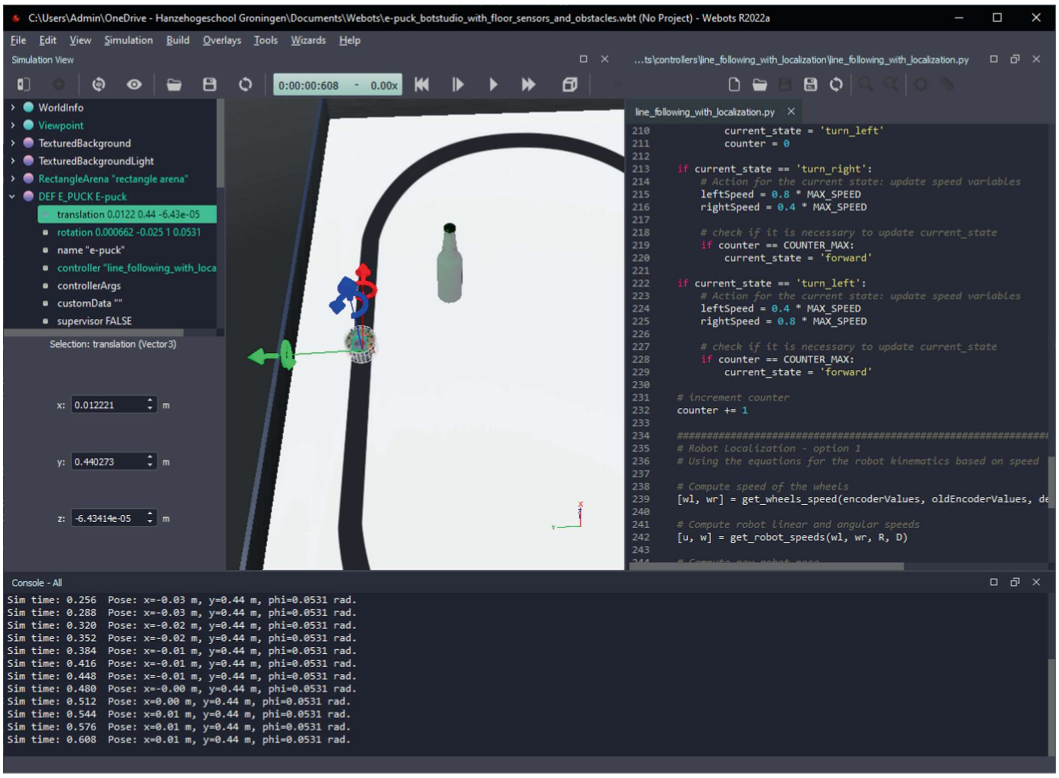


Fig. 1. Webots screenshot while running one of the lab activities. The simulated environment with the robot is seen in the center and (part of) the Python code that controls the robot is seen on the right side of the Window.

(except for lab 5). The reason for providing both a bare-bones template code and a solution code for labs 2–4 is to give students some general idea on how to start and organize their program (template code) and an explanation on how to completely solve some problems (solution code).

Labs 2, 3 and 4 also include the so-called challenges, to which no solution is provided: Lab 2 challenge - Obstacle avoidance with state-machine; Lab 3 challenge - Go-to-goal behavior with PID controller; Lab 4 challenge - Generate a trajectory for the robot to follow.

The main idea behind the challenges is that students use what they learned to build new code while implementing concepts discussed in the theory lessons and textbook. Figure 2 illustrates the challenge of lab 2: a state-machine that implements a simple obstacle-avoidance algorithm. Students need to implement the obstacle avoidance state-machine in Python, so they need to understand the state-machine and translate it into Python code. The state-machine given in Fig. 2 is only capable to avoid small round objects, like the bottle shown the figure, so students have to expand the state-machine to successfully avoid obstacles of different sizes and shapes. The goal is to activate learning because students build knowledge “by modifying and refining their current concepts and by adding new concepts to what they already know” [2]. The same idea is applied in the other challenges.

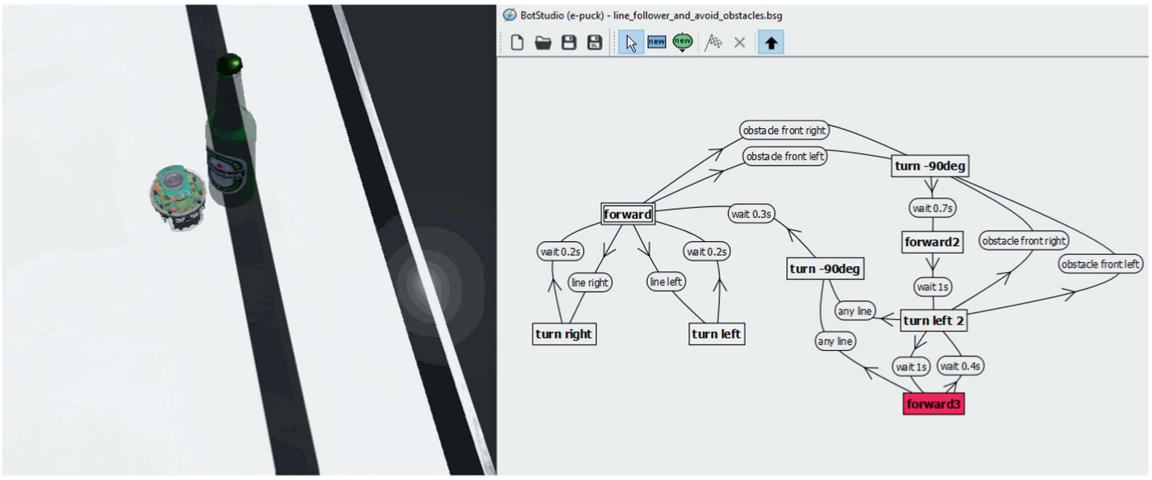


Fig. 2. Example of a state-machine that implements the line-following behavior with simple obstacle avoidance.

Finally, a *bonus* lab was included to motivate the interested students to go beyond the content of the regular curriculum. It invites them to build a robot soccer team and run a competition with other classmates using the RoboCupJunior SoccerSim platform [12], that also runs on Webots and uses Python.

Robotics Simulation Labs is hosted in GitHub and is licensed under the terms of the MIT license, which gives any person the rights to use, copy, modify, merge, publish, and distribute it, free of charge. Its most recent version can be accessed as a webpage[3] that is generated via the service github-pages.

4 Hardware-in-the-Loop Simulator

For the HIL simulator, the SimTwo simulation environment was used. It provides a simulation considering the dynamic constraints that the real scenario has, and has a realistic 3D model of the robot and the scenario. All the data (robot model and scene) can be freely modified, using XML language, according to the real model. Moreover, a scene of a robotic competition, Robot at Factory Lite (RaF Lite) from the Portuguese robotics Open, is used to exemplify the Hardware-in-the-loop approach.

The RaF Lite competition emulates, with real robots, the small-scale shop floor of an industry where it is possible for autonomous mobile robots (AMR) to transport materials from the supply warehouse to the machines, waiting for the material to be processed, and then transported back to the final product warehouse. As objective, the robot must be able to collect, transport, position the materials in the right position, locate and navigate within the given environment without collisions. The RaF Lite competition aims to stimulate students and researchers to develop solutions to the challenges it presents. 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. 3.

[3] https://felipenmartins.github.io/Robotics-Simulation-Labs/.

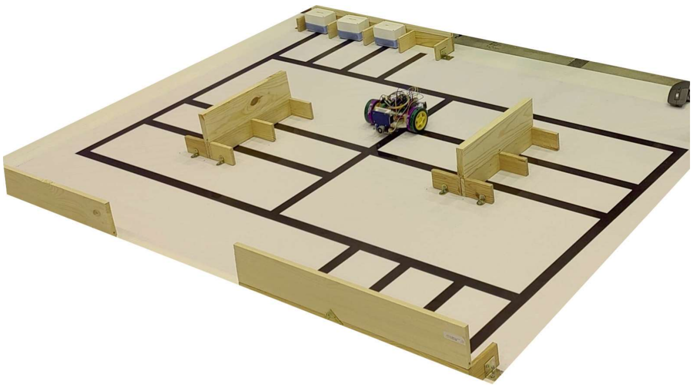


Fig. 3. Robot at Factory Lite competition arena with the differential-drive robot used in the tasks.

The organizers of the competition propose as a *start−kit* the tools, available online[4] with a mechanical design with the .stl files (for 3D printing), an electrical schematics (for real robot prototyping), a sample of code to control the robot and a simulation environment based on SimTwo.

Regarding the hardware, previous publication addressed a common Arduino Uno with an 8-bit microcontroller (ATMega328) [9]. In this paper, a 32-bit microcontroller ESP32 is used and also an application that allow students to access the data from the robot in real time, via WiFi. This facilitates debugging and logging of variables used in the robot program.

The main architecture of the real robot is presented in Fig. 4. Since the ESP32 has WiFi connection, an application was developed to access data from the robot to allow students a debug tool, the PCRobot presented at the left side of Fig. 4.

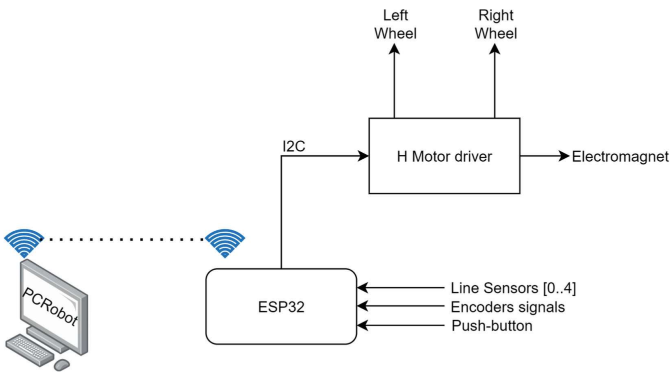


Fig. 4. Real robot architecture

The ESP32 microcontroller sends to the PCA9685 I2C the required PWM for each motor. Two drivers TB6612 as places in the same shield that allow to control two motors and also two more outputs. One of them is used to connect

[4] https://github.com/P33a/RobotAtFactoryLite.

an electromagnet that can be polarized to attract the metallic part of the box. Besides that, ESP32 also reads the five line analog sensors placed under the robot. They will allow to follow the line. A push button is also connected to detect the box proximity. Finally, four line signals from left and right encoders are connected to use odometry. Students can use encoders to assist on turns and measure the moving distance.

The on-board ESP32 microcontroller should be programmed by teams of students. In order to test their system, usually a team would require access to a competition arena and a robot. Instead, students can use the simulation environment to test their developments in the code. But, in order to keep the simulation conditions as close to real conditions as possible, SimTwo [15] is used combined with the HIL approach. SimTwo can be downloaded for free[5]. It provides a realistic simulation that can be employed in preliminary testing of team control software. Thus, a simulation model was developed as a reference for the teams to test their prototypes. Furthermore, the HIL methodology allow a real hardware control of the simulation environment. Previous work of authors, presented in [9], used an Arduino Uno. This embedded system is too limited and a powerfully microcontroller is proposed in this paper. Thus, an ESP32 will be used in this HIL. Figure 5 presents the HIL architecture. The same ESP32 microcontroller is used to receive the data from sensors and send commands to the actuators, virtually on Simulation, by USB connection with a loop period of 40 ms. Also the PCRobot developed application is enabled, in the same or another computer, to log data and real-time visualization by the programmer.

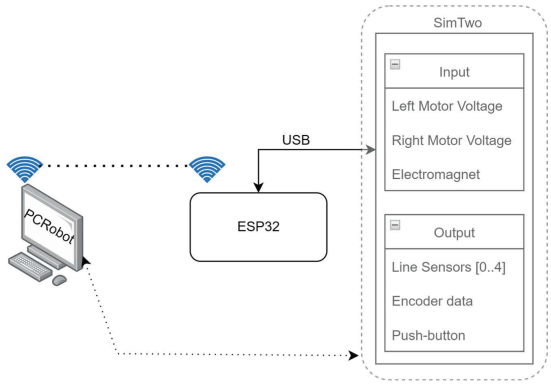
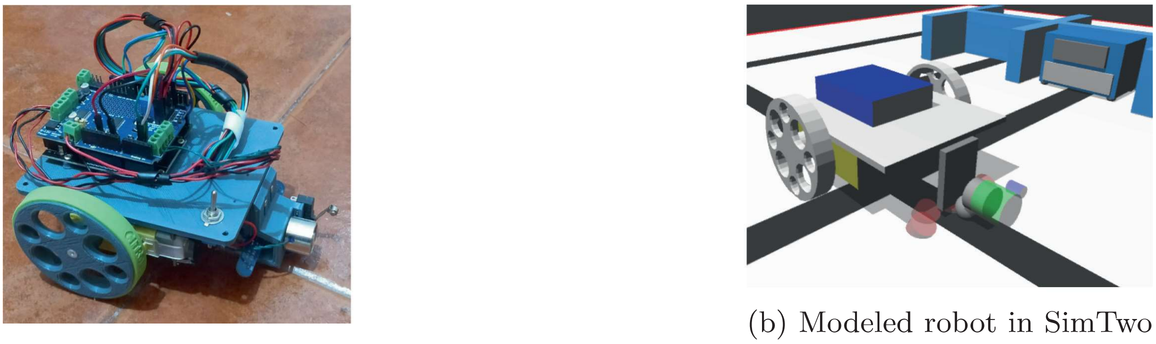


Fig. 5. Hardware-in-the-loop architecture and data transfer

The real robot is just a proposal from the competition organizers. Teams can develop their own robot, according to the rules, namely dimensions and weight.

[5] https://github.com/P33a/SimTwo.



(a) Real assembled robot

(b) Modeled robot in SimTwo environment

Fig. 6. Real and simulated model robots

5 Results

In this section, the results for both proposed approaches are presented, namely the *Robotics Simulation Labs* and the SimTwo HIL simulator environment.

5.1 Robotics Simulation Labs

The *Robotics Simulation Labs* were used for the first time in the period between April and July, 2020. In the following years (2021 and 2022) the content of the labs was made available via GitHub. The simulation labs completely replaced the lab activities originally planned for real robots since 2020. Students did not have the chance to work with real robots, but had the opportunity to program robots in simulation. To better understand the impact of the simulation labs, students were asked to give feedback, on a voluntary basis, anonymously. Between 20% and 30% of the active students of each year filled in an on-line form (42 students in total). They answered 10 multiple-choice questions, all of which are available on-line along with a summary of their answers[6].

Students also had the option to give written feedback (open topic). Only 17 actually wrote something, but some of their comments was not directly related to the simulation labs. Some of the relevant ones are reproduced below, exactly as written by the students: *"It was interesting to see and make a robot simulation."*; *"I really enjoyed working with WeBots. I enjoyed it so much that next to doing the tasks I also messed around with the program seeing what was possible with it."*; *"If it is possible to have the practicum in physical form with real robots that would be great."*; *"physical education lessons are waaaay beter"*; *"Real robots have some pro's and con's. What I liked about the simulation is that I could work all by myself from home. And it always worked once you understood how to use the sofware."*; *"I really like the way of including this kind of exercise and practical lessons. So long you are willing to ask for help this course is really doable!"*

[6] https://tinyurl.com/ykxh5syh.

A summary of relevant multiple-choice answers is given as follows. More than 80% of students found the simulation labs helped their learning process. The level of difficulty of the tasks was also assessed as good or excellent by around 80% of respondents. Much to our surprise, only 70% of the students reported that they prefer to practice with real robots. For 20% of them, it makes no difference if they are practicing the concepts with a real robot or on a simulator, while 10% actually prefer the simulator. These numbers show that the use of real robots is very important for teaching, but also that a third of the respondents don't mind (or prefer) learning robotics without dealing with real robots. We speculate that those students are more interested on programming aspects.

5.2 Hardware-in-the-Loop Simulator

Figure 6a) presents the real assembled robot with its hardware whereas b) presents the simultaed robot in the simulations environment.

The array of the simulator screenshots, presented in Fig. 7, shows the robot on simulation controlled by the external microcontroller, ESP32 with the provided demo code. It can be seen the robot moving, picking the first part from the incoming warehouse and take it to the outgoing warehouse.

Students feedback allow to conclude that this approach was determinant to the learning outcomes on robotics classes during the COVID-19 pandemic. While the limited accesses to the University, the presented tools allowed students to work at their homes. Moreover, the HIL methodology was used during the Portuguese Robotics Open (robotics Competition) where students could compete in a virtual arena.

6 Discussion and Future Work

In this paper, two simulation tools were addressed, namely the *Robotics Simulation Labs* and the SimTwo Hardware-in-the-loop environment. They allow students to work from home with robot experiments, programming and evaluating results. The presented tools allowed to cover the content and achieve the learning objectives of the robotics courses using simulation, which would have been very difficult or impossible due to social distancing rules during the COVID-19 pandemic. From the feedback given by students about the simulation labs, it is clear that some students enjoyed it, while others prefer to work with real robots. The main conclusion is that simulators do not to replace physical labs but they can work as an extra tool to give students more opportunities to practice, including at home. In simulation, students can test concepts quickly and play with different robots in safe and diverse environments. Simulation increases flexibility: students can do the experiments at any time, and repeat them as many times as they need or want. They can also vary the parameters of the experiment to better understand how they influence the overall behavior of the system. As one of the students pointed-out, the interested student can explore the simulator to go beyond the content and tasks of the course. It is important to highlight the

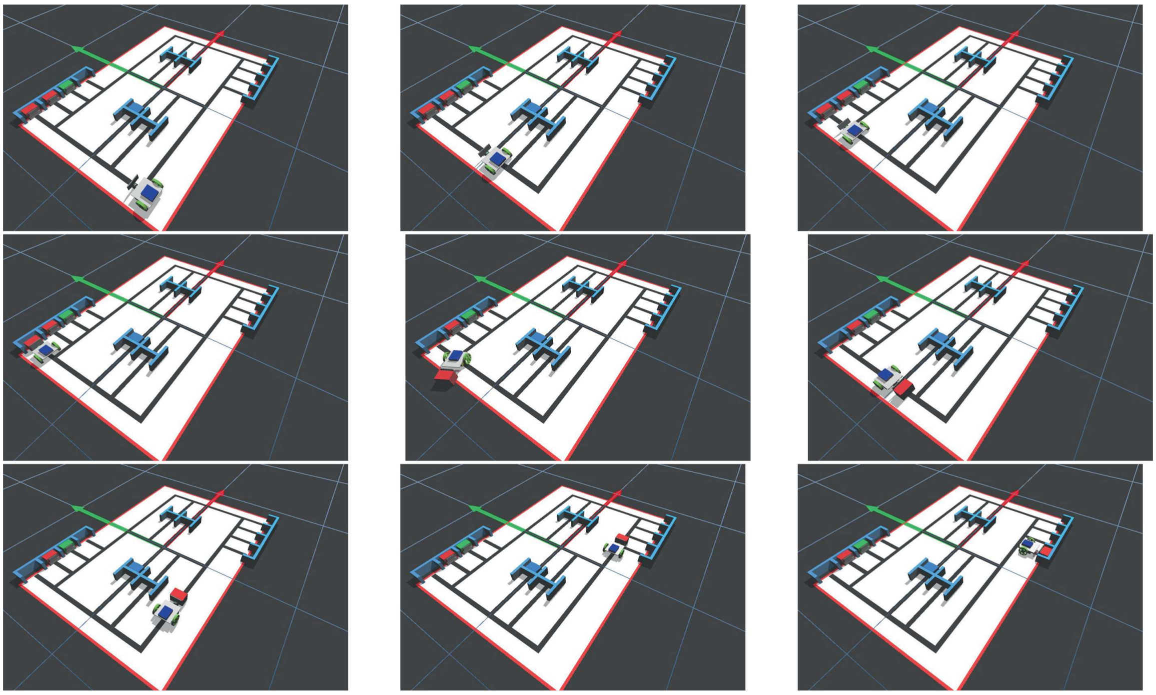


Fig. 7. SimTwo environment: mobile robot controlled by ESP32 microcontroller (HIL). Robot picks the left box, transport it to the outgoing warehouse and place the box there (sequence from left top to bottom right).

importance of the presented paper on the sustainable development goals such as Quality Education, for the area of Engineering Education (Robotic) focused on digital technologies. As future work, more simulation scenes can be added to the proposed tools, so that students have different environments to develop and program. Moreover, the use of other sensors can be integrated in the simulation labs, like accelerometers, gyroscopes and a camera. By doing so, other topics related to robotics can also be dealt with in the courses, such as sensor fusion (like Kalman Filter), sensors and actuators modeling, computer vision techniques and navigation.

Acknowledgment. J. Lima and F. Martins share first authorship. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020. The authors also thank the Research Centre Biobased Economy and the Institute of Engineering, Hanze University of Applied Sciences, The Netherlands.

References

1. Alsalhi, N.R., et al.: Undergraduate students' perceptions of the use of simulation software through online learning in colleges of engineering during the Covid-19 pandemic: a case study at al-Balqa applied university, Jordan. Periodicals Eng. Nat. Sci. (PEN) **10**(1), 678–691 (2022)
2. Kauffman, A.B.: Pedagogy in action - guided discovery problems. (2022). https://serc.carleton.edu/36080. Accessed 28 June 2022

3. Bacic, M.: On hardware-in-the-loop simulation. In: Proceedings of the 44th IEEE Conference on Decision and Control, pp. 3194–3198. IEEE (2005)
4. Braun, J., et al.: Robot@Factory Lite: an educational approach for the competition with simulated and real environment. In: Silva, M.F., Luís Lima, J., Reis, L.P., Sanfeliu, A., Tardioli, D. (eds.) ROBOT 2019. AISC, vol. 1092, pp. 478–489. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-35990-4_39
5. Collins, J., Chand, S., Vanderkop, A., Howard, D.: A review of physics simulators for robotic applications. IEEE Access **9**, 51416–51431 (2021)
6. De Stefano, M., Mishra, H., Giordano, A.M., Lampariello, R., Ott, C.: A relative dynamics formulation for hardware-in-the-loop simulation of on-orbit robotic missions. IEEE Robot. Autom. Lett. **6**(2), 3569–3576 (2021)
7. Gamage, K.A., Wijesuriya, D.I., Ekanayake, S.Y., Rennie, A.E., Lambert, C.G., Gunawardhana, N.: Online delivery of teaching and laboratory practices: Continuity of university programmes during Covid-19 pandemic. Educ. Sci. **10**(10), 291 (2020)
8. Kapilan, N., Vidhya, P., Gao, X.Z.: Virtual laboratory: a boon to the mechanical engineering education during Covid-19 pandemic. High. Educ. Future **8**(1), 31–46 (2021)
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. Martin, A., Emami, M.R.: An architecture for robotic hardware-in-the-loop simulation. In: 2006 International Conference on Mechatronics and Automation, pp. 2162–2167. IEEE (2006)
11. Martins, F.N., Gomes, I.S., Santos, C.R.F.: RoSoS - a free and open-source robot soccer simulator for educational robotics. In: Santos Osório, F., Sales Gonçalves, R. (eds.) LARS/SBR -2016. CCIS, vol. 619, pp. 87–102. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47247-8_6
12. Martins, F.N., Matejov, A., Šuppa, M.: Moving robotics competitions virtual: the case study of RoboCupJunior soccer simulation (SoccerSim). Frontiers in Robotics and AI **9** (2022). 10.3389/frobt.2022.915322
13. Martins, F.N., Gomes, I.S., Santos, C.R.F.: Junior Soccer Simulation: providing all primary and secondary students access to educational robotics. In: 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), pp. 61–66. IEEE (2015)
14. Michel, O.: Cyberbotics Ltd. Webots^{TM}: professional mobile robot simulation. Int. J. Adv. Robot. Syst. **1**(1), 5 (2004)
15. Paulo, C., José, G., José, L., Paulo, M.: Simtwo realistic simulator: a tool for the development and validation of robot software. Theory Appl. Math. Comput. Sci. **1**(1), 17–33 (2011)
16. Pereira, C.E., Paladini, S., Schaf, F.M.: Control and automation engineering education: combining physical, remote and virtual labs. In: International Multi-Conference on Systems, Signals & Devices, pp. 1–10. IEEE (2012)
17. Temelta, H., Gokasan, M., Bogosyan, S.: Hardware in the loop robot simulators for on-site and remote education in robotics. Int. J. Eng. Educ. **22**(4), 815 (2006)