Environment for UAV Education

Martin Sedláček, Eduard Mráz, Matej Rajchl and Jozef Rodina

April 14, 2023

Environment for UAV education [⋆]

Martin Sedláček[1], Eduard Mráz[1], Matej Rajchl[1], and Jozef Rodina[1]

Slovak University of Technology in Bratislava, Vazovova 5 Bratislava, Slovakia
{`martin.sedlacek, eduard.mraz, matej.rajchl`}`@stuba.sk`

Abstract. Simulation tools are often used in robotics education to transfer theoretical knowledge into practical experience. The correlation between practical and real-world experience depends on the quality of the simulation environment and its ability to serve as a replacement for the real world. In this study an environment for UAV education is proposed, where the transfer of theoretical knowledge into practical experience is achieved through the implementation of both simulation and real UAV deployment. Initially, this environment allows students to develop their applications in the simulation. Subsequently, in the later stages, they are able to easily deploy them into the educational UAV and test the solution in the drone laboratory. Compatibility between simulation and physical UAV is achieved by combining the appropriate components in both the simulation and physical UAV. The software technologies used include ROS, ArduPilot, and MavLink, while the hardware platform for the educational UAV was chosen as DroneCore.Suite by Airvolute s.r.o.. These technologies are not only suitable for the proposed environment but are also commonly used across the robotics field, which enhances relevant students competencies. Later in the article, an example assignment is presented and described how to use the environment for UAV education in a classroom setting.

Keywords: UAV · Simulation · Education

1 Introduction

Robotics as a field of study is a very broad topic, which incorporates quite a large range of subtopics e.g. computer science, mechatronics, control systems and a lot more. Given the complexity and multidisciplinary nature of robotics, it can be challenging to teach this subject effectively. It is often very difficult to apply students' theoretical knowledge into complex robotics problems which simulates real life practical examples. One of the obstacles is the mentioned complexity of a robotics system itself, which can lead to failures during the practical testing phase of student assignment. These failures are a natural part of the educational process. However, they can reduce the time effectiveness of students and increase

---

[⋆] This article was written thanks to the generous support under the Operational Program Integrated Infrastructure for the project: "Research and development of the applicability of autonomous flying vehicles in the fight against the pandemic caused by COVID-19", Project no. 313011ATR9, co-financed by the European Regional Development Fund. This publication was also supported by the project APVV-21-0352 - "A navigation stack for autonomous drones in an industrial environment" and by the company Airvolute s.r.o.

the cost of materials used in the class. To overcome these obstacles a simulation environment is commonly used [1]. Depending on the quality of the simulation environment the real practical experience gain can be reduced for the students. In the environment for UAV education, the impact of failures is intended to be reduced by utilizing the software in the loop (SITL) capabilities of the autopilot used in a real UAV. Additionally, a virtual environment was created to serve as a digital copy of an existing drone laboratory. The use of UAV as a platform is becoming increasingly common in higher education as well as in various industries. This trend is reflected in the growing number of applications of UAVs in various fields [2, 3]. This has also been observed in studies such as [4]. The proposed environment for UAV education is an implication of this trend and is expected to provide students with easier access to this growing industry.

One of the goals of the environment for UAV education is to create a virtual environment that simulates reality as realistically as possible. But the main goal is to create an environment that enables the development of easily deployable student projects in a real drone laboratory. The simulation part of the environment use the same technologies as those utilized in real hardware. By using the same technologies in both the simulation and the real hardware, it is possible to achieve that a solution developed in the simulation will also be able to run seamlessly in the real environment. The UAV used in this study is intended to fly indoors, which poses some new challenges for the UAV education environment. Those will be addressed in detail further in the article. Indoor UAV is a great platform to teach robotics across the whole spectrum of the students' years. The platform can be used by students in the early years of their studies to gain general knowledge about robotics, UAVs, and related topics through projects and education. The environment for UAV education can abstract students from critical low-level tasks such as localization and control. Students in the final years of their studies can use the UAV environment to develop much more complex projects. This environment will not only provide students with a simulation for rapid testing of their solution, but also enable them to test their solution in the real environment. Therefore, provide them with the motivation for the project, final thesis or a course and real world experiences.

As mentioned before, robotics is a very broad topic. The environment for UAV education also reflects this. It encapsulates many technologies that are used together to achieve working solution. In the selection of these technologies, the aim was to choose those commonly used in the field of robotics, with the intention of making the knowledge easily transferable to other subjects in the student curriculum [5]. Furthermore, technologies were chosen to be compatible with the capabilities of the UAV. In summary, the environment for UAV education includes several technologies such as ROS [6], ArduPilot [7], Gazebo [8], MAVLink[9] and the operating system Ubuntu [10]. Similar technologies are used by other researchers and technical teams to simulate UAV behaviour in various applications [12, 13]. The proposed environment differs from these works by providing a possibility to seamless transfer of the solution between simulation and real world as well as creating environment focused on teaching robotics.

Although the proposed solution is hoped to be viable for general use as described, certain limitations are expected to be encountered in achieving a seamless port between simulation and reality. These limitations are largely attributed to the unique precision of localization and other low-level systems for each UAV. As a result, extra safety or compatibility parameters may need to be incorporated into the simulation solution, which could make the porting process more complicated and require additional parameter adjustments.

2 Used technologies

2.1 Education UAV

To ensure high quality of education while maintaining accessibility of UAV systems in school environments, the requirements for the UAV are somewhat contradictory. The system needs to be robust, complex enough to demonstrate real world tasks and able to fly in an indoor environments, but on the other hand it should be affordable and also not too complicated. The requirements may be met by some of the commercial solutions, but their platforms are usually closed (proprietary) or protected by licenses, which makes it impossible to use them in the environment for UAV education. Therefore, custom solution had to be developed (Ref. 1).

Fig. 1. Prototype of education UAV.

2.2 Architecture of environmnet for UAV education

Electronics The UAV system from company Airvolute s.r.o. is used as the core of the custom UAV. This system (DroneCore.Suite [14]) combines UAV autopilot (ArduPilot) with the Nvidia Jetson Xavier NX system on a module [15].

DroneCore.Suite provides rich connectivity for various sensors, however for the custom UAV, the most relevant sensors are range finders and stereo camera pairs. This system is also capable of handling more complex and computationally expensive tasks such as reactive navigation or real time mapping elevating Nvidia Jetson Xavier NX coprocessors (GPU and other). DroneCore.Suite also includes DroneCore.Power, which is an electronic speed controller capable of working with ArduPilot.

Using DroneCore.Suite it was possible to build a relatively simple UAV capable of flying indoors, which is compatible with the environment for UAV education. The drone configuration consists of a DroneCore.Suite minimal viable configuration (see from the diagram in Fig. 2) and a camera stereo pair. Software compatibility is ensured by incorporating Nvidia Jetson Xavier NX that is powered by Nvidia JetPack SDK. Jetpack SDK is a collection of software components, which are compatible with Nvidia Jetson Xavier NX, the operating system of Jetpack SDK is Ubuntu. This operating system is used also in the environment for UAV education. This way we are able to port the application from the simulation environment to the real UAV comfortably and therefore reduce the overhead for the students.

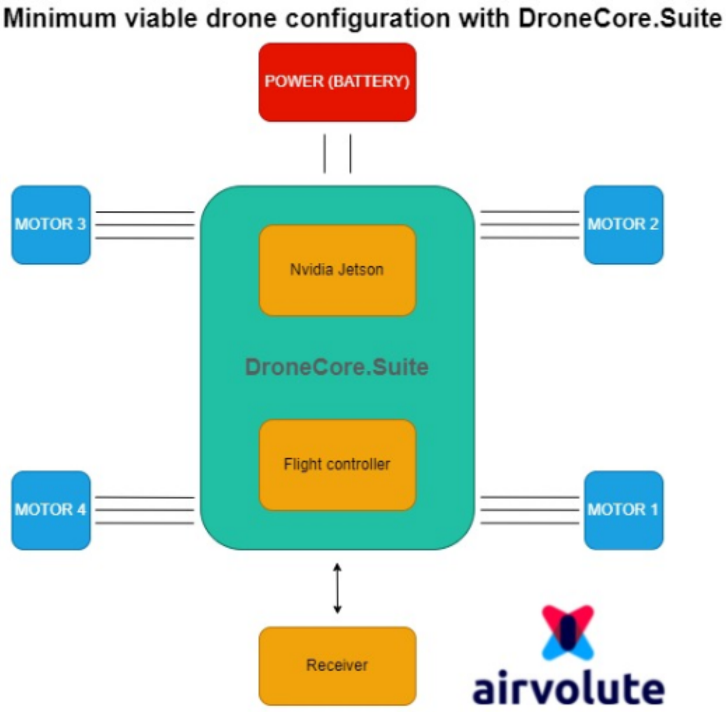

Fig. 2. Minimum viable drone configuration with DroneCore.Suite. [14]

Construction To ensure stiff construction which can be easily replaced a set of carbon plates has been used, which creates the main chassis of the UAV (see Fig. 1). It encapsulates all of the electronics in a safe robust covering whilst maintaining a very good weight ratio of the construction to the power of the motors. The bottom plate also seamlessly extends outwards to create four arms of the UAV. This enables to cut the whole body out of 3-5 mm carbon plates using a simple hobby grade CNC.

Main attributes of the whole construction is that it is very light-weight and very stiff. It can resist many of the student mishaps such as collision with other objects, ground crashes or high strain forces due to aerodynamic forces caused by for example regulator instability. Also the light-weight nature of the whole UAV is suitable for educational environment since the amount of kinetic energy which is produced by a drone is much lower compared to heavier drones. Thus chances of injury during collision is lowered.

Simulation In the simulation part of the proposed solution, an autopilot (ArduPilot) in Software-in-the-loop (SITL) configuration is used, which functions similarly to the actual autopilot. However, some sensors are either simulated or obtained from a simulation environment. As an alternative, autopilot PX4 could be used. Since our laboratories dispose of UAV's equipped with ArduPilot solution stack, ArduPilot is the preferred autopilot. The great advantage of ArduPilot is a very active development team and its opensource licensing. ArduPilot in SITL configuration is connected with a graphical simulator.

In the proposed environment for UAV education, Gazebo is currently utilized as the graphical simulator. One of the key advantage of Gazebo is, that it is relative lightweight in terms of computation demands, which allows it to run on lower-specification hardware without sacrificing performance. Additionally, Gazebo is well documented, with a large and active user community providing support and resources to users. These factors make Gazebo an excellent choice for use in educational settings, where students may not have access to high-performance hardware or the resources to troubleshoot complicated software. While Gazebo is an excellent choice for our task, there are other simulation software options available. One such option is AirSim [16], which is based on the Unreal Engine 4 and offers photo-realistic environments. While this may be advantageous for certain use cases, such as testing autonomous vehicle perception systems, there are significant drawbacks to using AirSim. One major disadvantage of AirSim is its computational complexity. The photo-realistic environment require significantly more computational power to simulate, which can limit performance of the simulation and in the worst case make the simulation invalid. Another issue with AirSim is that it is not fully compatible with ArduPilot SITL. While most of the features are supported, some have not yet been implemented. This is problematic as it can hinder the quality of the simulated flight, resulting in a deviation from real flight. These issues were considered critical for the given use case. Therefore, AirSim was not used, and instead, Gazebo was adopted.

Mavros ROS package [17] is used as an interface to ArduPilot SITL and graphical simulation environment. Mavros bridges and translates MAVLink protocol to ROS topics and provides other functionality related to ArduPilot.

Students applications are then written inside the ROS environment as a ROS nodes. This approach is considered favorable as it is one of the conventional methods of developing UAV control in the real world. The whole architecture of the simulation part of the environment of UAV education can be viewed on the Fig. 3. The ROS nodes created by students can access the ArduPilot SITL simulated UAV in the Gazebo graphical simulation via standard communication tools inside the ROS, such as topics and services. To communicate with ArduPilot SITL from ROS in a standard way, one can utilize Mavros (a ROS node) provided services and topics. These services and topic are converted and sent further via MavLink directly into ArduPilot SITL. So the ROS nodes created by students have access to autopilot and therefore are able to control the UAV. In addition to communicating with the autopilot, students are expected to work with other ROS topics that contain crucial data, such as the current position of the UAV in the simulation. By using this mechanism the simulation part of the environment for UAV education is capable to control the whole simulation using only a ROS node. By having all important data accessible through ROS topics, the complexity of the simulation environment is significantly reduced for students. This enables them to concentrate solely on the ROS node and handle all relevant tasks from there.

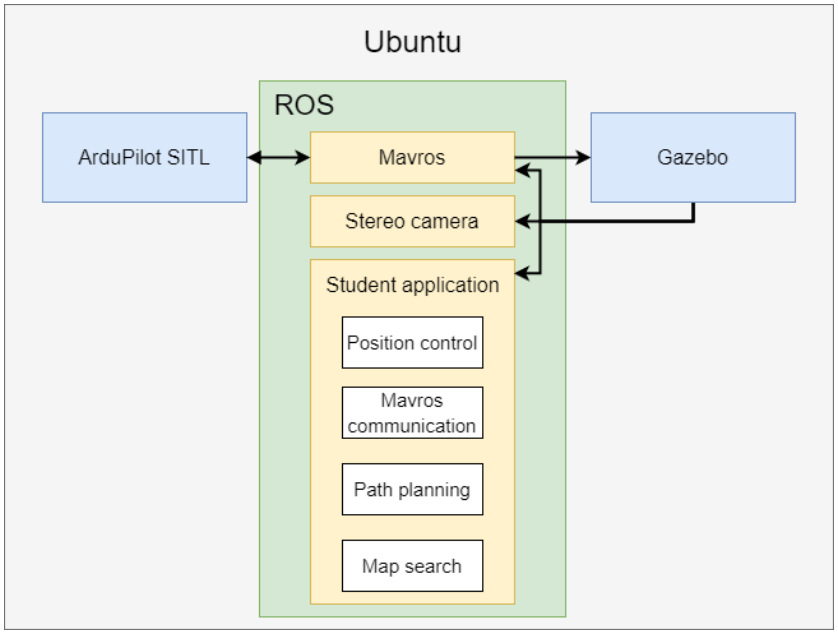

Fig. 3. Overview of the environment for UAV education.

Simulation and UAV system The software core of the environment for UAV education is the Ubuntu operating system both in the simulation and in the UAV.

This allows to use ROS and Mavros in the education UAV. Thus, it is possible to directly copy ROS node developed for the simulation to the board computer of the UAV, make minor modifications for subscribed topics names, compile it and launch it. This process is as seamless as possible, given the complexity of the UAV system. By utilizing ROS, Mavros, ArduPilot and Gazebo, the students are able to develop and test their application in a simulated environment, and then easily transfer it to the actual UAV. This approach greatly simplifies the development process and reduces the overhead for the students.

UAV system The DroneCore.Suite from Airvolute s.r.o. provides localization from stereo camera pair and other lower level tasks, which are launched automatically after the drone starts. This greatly simplify the development process for the students or the teachers, as they do not need to focus on implementing these tasks themselves. Instead, they can rely on the pre-existing capabilities provided by DroneCore.Suite and only use the data provided by ROS topics. But, if it is needed the system of DroneCore.Suite is open and it is possible to make changes..

Digital copy of the simulation environment and UAV Simulation scene or world, where students work on their assignment, is one to one scale digital copy of the existing drone laboratory at the university. Blender was used to model environment for Unreal Engine 4 (AirSim) and Gazebo. This approach gives us some flexibility to reuse model in some other projects due to Blender's rich export options. To use photo realistic textures in Unreal Engine 4 model, Substance3D software was used. Gazebo model was textured only with colors or simple textures, which are by default incorporated into Gazebo, motivation behind this was to keep this model as lightweight as possible. In this section, it is presented that a digital copy of the drone laboratory was created for both AirSim and Gazebo as part of the development process for the environment for UAV education. However, it should be noted that only Gazebo is adopted in the latest version.

In this virtual environment, a virtual UAV is placed. A simple static model can be placed almost directly into Gazebo or AirSim with the help of export from CAD software. It is essential to check that the virtual model's dimensions are as close to the real model as possible. This ensures feasibility of the proposed trajectories both in the simulation environment and in the drone laboratory. Virtual drone is also equipped with stereo camera pair, which needs to be placed in the right position with the right orientation.

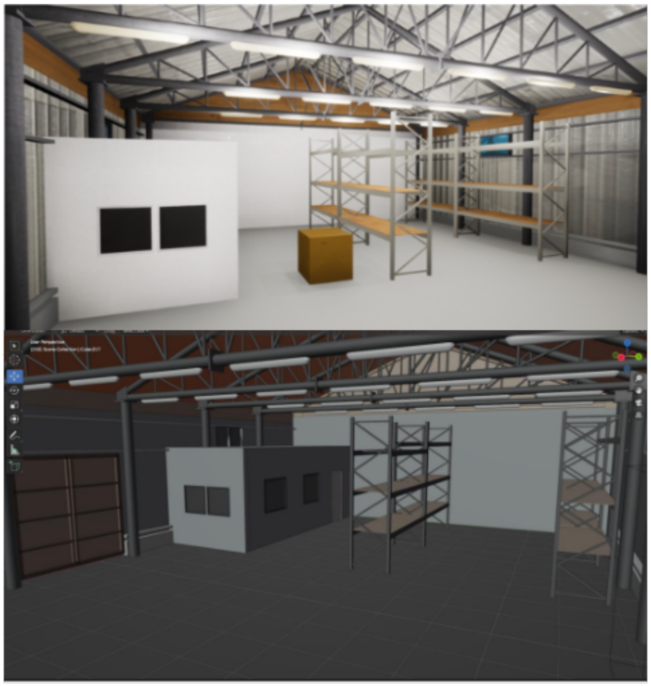

Fig. 4. Digital copy of drone laboratory. Up - Unreal Engine 4, down - Gazebo.

3 Environment in education

The environment for UAV education encapsulates all the technologies mentioned in section 2. It should be noted that not only virtual simulation is present, but also a drone laboratory and the education UAV are included. All these parts together ensure compatibility between simulation and reality.

The prerequisites for students to take full advantage of this environment include basic Linux skills (such as using bash, installing packages, and similar tasks), computer science skills (including knowledge of ROS, Python or C++ programming languages), and general robotics knowledge (such as understanding algorithms and basic control systems). Most of these skills are already present in a lot of robotics curriculum. This environment can also be used by students with less knowledge. For example, one can select such assignments, that will not require computer science or they can provided code templates. Our team believes, that even for starting students this environment can be a great motivation to further improve their skills and promote individual interest in robotics and UAVs.

3.1 Deployment of the environment for UAV education

To integrate the environment for UAV education in a course, it is important to make sure, that students are at least a bit familiar with each used technology. In our class, we attempted to achieve this by not providing pre-built Linux images or similar system distributions for desktops that already have the environment installed. Instead, students were given with an installation guide to set up the environment on their own. The guide is divided into several parts, each of these parts include control points. The control points do not only indicate the correct setup, but also demonstrate some basic functionality. This approach takes away some of the time in the class, which could be used to improve other talents. Despite this, the advantage of students getting familiar with the environment is more important to us. Therefore, it was decided to use this approach in our course. In general, we would recommend using the installation guide approach with students in higher years. It is expected that using the installation guide with new students would take too much time to be effective. In this case, each of the technologies used can be explained in the form of a presentation.

To deploy compatible system in the education UAV, the system image derived from Airvolute s.r.o. system version of Nvidia Jetpack for Nvidia Xavier NX was used. This system image, as mentioned in section 2.1, includes preinstalled software and features supporting hardware DroneCore.Suite, which the education UAV is running on. As already mentioned, students deploy their application into ROS workspace, so their contact with the whole UAV system is limited. Therefore, some faults can be avoided. In the case that some additional access to the system is needed, students are able to alter almost all the parts of the system. If some unrecoverable error occurs the DroneCore.Suite can be flashed again with the default system image in a relatively short time.

In addition to the application computer on DroneCore.Suite, where the students' application will run alongside ROS, Mavros, and other software components prepared by Airvolute s.r.o. The DroneCore.Suite also includes an autopilot hardware CubePilot running on ArduPilot software. It is possible to preset this autopilot with tuned values for regulators and parameters for other systems. These parameters needs to be tuned to work well. Due to the good mechanical properties of the education UAV, they can be identified on one drone and can be transferred to other UAVs of the same type. This is very beneficial, because a relatively large fleet of the educational UAVs can be serviced with ease, which is ideal for robots in education. In some more advanced and profiled courses students are able to tune autopilot by themselves as a part of an assignment. It can be achieved by changing parameters of the autopilot, which is supported by the education UAV.

Demands on drone laboratory To ensure good functionality and ease of use of the educational drones, it is necessary to enforce some properties of the drone laboratory. The dimensions of the laboratory should be enough to fly the drone freely. Good dimensions starts from at least 100m^2 with a square or rectangle footprint. Height of the ceiling is sometimes overlooked factor, it is recommended to be at least 3.5m. Space with these dimensions allows to plan a lot of types of automatic or autonomous UAV missions. At the same time it allows to fly UAV in manual mode, which can be really important during tuning. Flying drone, even in this size, can be dangerous. Therefore, the space should be equipped with safety nets or at least contain some protected space, where workspace for students can be set up. The lighting of the space is also important, good illumination should be present at all times, becasue of the use of visual odometry. In general, it is good to avoid some types of LEDs due to flickering in camera images. Visual odometry can be swapped for some other type of localization usable indoor, for example the education UAV is ready to be used with motion capture system, which can ease some demands on the lightning and improve stability of the localization.

3.2 Proposed assignment - Simple automatic mission

To better illustrate how the environment for UAV education can be used in courses, an exemplary practical assignment was created. In this assignment students have to apply a wide spectrum of theoretical knowledge about robotics to solve a simple automatic mission.

In this assignment, UAV have to fly on predefined trajectory (set of waypoints). Trajectory waypoints are given in a coordinate system of the simulation environment, each waypoint has predefined accuracy of reach and some waypoints have special tasks assigned to them. This task can be assigned to students using similar table to Table 1.

Table 1. Simple automatic mission

Waypoint	x[m]	y[m]	z[m]	Task	Precision
1	x0	y0	z0	takeoff	-
2	x1	y1	z1	-	soft
3	x2	y2	z2	land and takeoff	hard
4	x3	y3	z3	-	soft

To solve this assignment, multiple subtasks have to be solved. If the assignment is divided the right way, there will be some continuity between subtasks. Therefore, each subtask can be developed relatively standalone. This makes the assignment valid as a group project, which does not only promote teamwork, but also simulates real world technical problem solving. One possible way in which a simple automatic mission could be split is as follows: map-related tasks (including map generation and map handling), trajectory planning (involving pathfinding and trajectory planning), a ROS drone control node (providing an interface to the autopilot, managing mission tasks and implementing simple position control). After this division, it becomes apparent that the simple automatic mission involves a wide range of topics from robotics. It is possible, that some of the students have already solved similar tasks as a part of projects, final theses or other other courses. To avoid unnecessary implementation of the same algorithms, students are allowed to use already existing solutions in our course. The focus of the work of these students has shifted a little bit to integration with UAV environment. In our experience, this has led to increased motivation for the class and better overall solutions and class performance.

This assignment can be easily deployed from digital environment into the real one (if most of the suggestions for drone laboratory and digital copy has been satisfied). The process is not different as presented in section 2.2. To avoid potential issues during the deployment, waypoints from Table 1 should be defined with enough clearance from the walls and other obstacles.

4 Conclusion

This study presents a detailed simulation environment for UAV education. It is worth noting that the environment for UAV education includes not only a collection of software components and a simulator, but also a real-world UAV and a drone laboratory. The main advantage and benefit of this solution is that it bridges configuration in simulation environment into the real-world with relative ease. This helps students to transfer theoretical or simulator experience into practical experience. Therefore it helps them prepare better for future in robotics. In addition to practical experience, this solution supports students' motivation by enabling them to deploy their solutions and test them in conditions close to the real world.

In addition to the pointed out educational benefits, we believe that our environment can contribute to the trend of ever-increasing use of UAVs by preparing future engineers to face challenges related to UAV deployment and development.

Although it is possible to port the working solutions of the simulated UAVs to the real environment rapidly. This aspect of the work was tested only by a team of teachers and researches. Therefore, the complete integration of the simulation environment for UAV education into the courses remains as a part of future work. To be exact, the simulation part of this solution is already integrated into a university course. The part of the solution, where students deploy application into the real world is currently being tested by a team of students within their semestral project. After this test, the feedback will be integrated into the environment and the complete environment for UAV education will be incorporated into the course.

Thus far, the feedback from the students regarding the simulation part of this environment has been very positive. One of the most praised aspects is that students are able to integrate multiple skills from different areas and create a functional solution. It seems, that even the simulation part of the environment was motivational enough for the students.

Not any big or critical issue arisen during the integration of the environment into the course. Most of the issues stemmed from insufficient familiarity with ROS and computer science concepts.

In the future work, it is planned to continuously gather feedback from university students and apply relevant suggestions into the environment. It is also planned to add more exemplary assignments, which should target specific areas of robotics.

References

1. Fernandes D, Pinheiro F, Dias A, et al (2019) Teaching robotics with a simulator environment developed for the Autonomous Driving Competition. Robotics in Education 387–399. doi: 10.1007/978-3-030-26945-6_35
2. Ahmed F, Mohanta JC, Keshari A, Yadav PS (2022) Recent advances in Unmanned Aerial Vehicles: A Review. Arabian Journal for Science and Engineering 47:7963–7984. doi: 10.1007/s13369-022-06738-0
3. Maghazei O, Lewis MA, Netland TH (2022) Emerging technologies and the use case: A multi-year study of drone adoption. Journal of Operations Management 68:560–591. doi: 10.1002/joom.1196
4. Bolick MM, Mikhailova EA, Post CJ (2022) Teaching Innovation in STEM Education Using an Unmanned Aerial Vehicle (UAV). Education Sciences 12:224. https://doi.org/10.3390/educsci12030224
5. Zhang L, Merrifield R, Deguet A, Yang G-Z (2017) Powering the world's robots—10 years of Ros. Science Robotics. doi: 10.1126/scirobotics.aar1868
6. Quigley, M. et al.: ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software. (2009).
7. ArduPilot. https://ardupilot.org/. Accessed 22 Jan 2023
8. Koenig N, Howard A Design and use paradigms for gazebo, an open-source multirobot simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat No04CH37566). doi: 10.1109/iros.2004.1389727
9. Koubaa A, Allouch A, Alajlan M, et al (2019) Micro Air Vehicle Link (MAVlink) in a Nutshell: A survey. IEEE Access 7:87658–87680. doi: 10.1109/access.2019.2924410
10. Ubuntu. https://ubuntu.com/. Accessed 6 Jan 2023
11. Chen S, Zhou W, Yang A-S, et al (2022) An end-to-end UAV simulation platform for Visual Slam and Navigation. Aerospace 9:48. doi: 10.3390/aerospace9020048
12. Hentati AI, Krichen L, Fourati M, Fourati LC (2018) Simulation tools, environments and frameworks for UAV Systems Performance Analysis. 2018 14th International Wireless Communications amp; Mobile Computing Conference (IWCMC). doi: 10.1109/iwcmc.2018.8450505
13. Chen S, Zhou W, Yang A-S, et al (2022) An end-to-end UAV simulation platform for Visual Slam and Navigation. Aerospace 9:48. doi: 10.3390/aerospace9020048
14. DroneCore.suite - autopilot solution for autonomous drones. https://www.airvolute.com/product/dronecore/. Accessed 21 Dec 2022
15. Jetson Xavier NX Series. https://www.nvidia.com/en-gb/autonomous-machines/embedded-systems/jetson-xavier-nx/. Accessed 21 Dec 2022
16. Shah S, Dey D, Lovett C, Kapoor A (2017) AirSim: High-fidelity visual and physical simulation for Autonomous Vehicles. Field and Service Robotics 621–635. doi: 10.1007/978-3-319-67361-5_40
17. Mavros. http://wiki.ros.org/mavros. Accessed 23 Jan 2023