



Integrating SolidWorks, LabVIEW, and Arduino in Robotics Education

João Paulo Coelho^(✉), J. A. B. Coelho, and José Gonçalves

Research Centre in Digitalization and Intelligent Robotics (CeDRI), Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
{jpcoelho, joacoelho, goncalves}@ipb.pt

Abstract. This paper explores the integration of SolidWorks, LabVIEW, and Arduino as a comprehensive and cost-effective approach to teaching robotics to undergraduate students. In scenarios where real hardware is unavailable or prohibitively expensive, this methodology offers significant advantages. SolidWorks enables students to design and simulate robotic components in a virtual environment, fostering a deep understanding of mechanical design and engineering principles. LabVIEW provides an intuitive graphical interface for programming and control, allowing students to develop and test their algorithms. Finally, Arduino, as an open-source hardware platform, bridges the gap between virtual simulations and physical implementation, offering a hands-on experience with minimal financial investment. Together, these tools create a robust educational framework that enhances theoretical knowledge through practical application, encourages innovation, and prepares students for real-world engineering challenges. The paper concludes that this integrated approach not only mitigates the limitations of resource constraints but also enriches the learning experience by providing a versatile and accessible platform for robotics education.

Keywords: Education in robotics · mixed Simulation · 3D modelling · Control design

1 Introduction

From some time now, robotics has emerged as a cornerstone of modern electronic engineering curricula. For students enrolled in those types of courses, gaining a solid foundation in robotics is crucial, not only for their academic development, but also as a future professional skill. Understanding robotics equips students with a diverse set of competencies, including system integration, control theory, sensor interfacing, and programming, all of which are essential in today's technologically advanced world.

In this reference frame, a hands-on learning approach is fundamental. Engaging students through practical, experiential learning not only makes the educational process more enjoyable but also enhances comprehension and retention of

complex concepts. Hands-on experiences allow students to witness the immediate results of their designs and decisions, fostering a deeper understanding and sparking innovation.

However, one of the significant barriers to implementing effective industrial robotics education is the high cost of the machines. Industrial-grade robots are often prohibitively expensive, making them inaccessible to many educational institutions, especially in regions with limited financial resources. As a result, it becomes imperative to find alternative methods to provide students with a comprehensive learning experience. Nowadays, many different simulation platforms exist and offer the ability to create a virtual world with virtual robots. However, the time and effort students put into learning how to master those software packages does not translate into competencies that can, in the end, be used in the real world. This is because, in general, those tools, even if excellent from an educational point of view, are not of any practical use in the industrial world. Ideally, the robotic simulations should be carried out using software, whose time spent to learn how it works, would be converted into knowledge that, in the end, would promote useful professional skills.

In this sense, this article presents a methodology that involves the simulation and control of robots for educational purposes using industrial-grade computer tools over which the Polytechnic Institute of Bragança has purchased licenses: SolidWorks and LabVIEW. SolidWorks is an integrated environment that can be used for parametric 3D modelling and mechanical simulation. This tool allows the design and simulation of robotic components in a virtual environment, facilitating a thorough understanding of mechanical and structural aspects without the need for physical hardware. LabVIEW, with its intuitive graphical programming interface, enables the development and testing of control algorithms and system behaviors. By integrating these software tools with Arduino Uno, an affordable open-source hardware platform, students can transition from virtual simulations to real-world applications while applying knowledge acquired from different curricular units such as electronics and embedded systems.

This approach not only mitigates the financial constraints associated with purchasing industrial robots but also ensures that students gain practical skills in widely used industry-standard software. Through simulations, students can explore and solve mechatronic problems, thereby preparing them for real-world engineering challenges. The integration of SolidWorks, LabVIEW, and Arduino in the robotics curriculum provides a robust and accessible framework for comprehensive robotics education, equipping students with the necessary tools and knowledge to excel in their future careers.

This article focuses on these above context and describes the methodology in which a robotic manipulator, modeled in SolidWorks, is manipulated, via LabVIEW, using a joystick installed on an Arduino Uno. Students should be able to freely 3D model their robotic arms and proceed to its control and manipulation using signals provided to the microcontroller. This approach was tried within the “project” curricular unit of the “Automation, Robotics and Industrial Electronics” that takes place in the Polytechnic Institute of Bragança.

In order to place this work in the technical and scientific context, the following section presents a review of the works considered to be related to the present one. In the third section, details of the methodology used and an illustrative example are presented. The fourth, and final section, deals with the conclusions.

2 Virtual Environments for Robotics

Before starting this section, it is important to point out that the examples presented do not represent, in any way, an exhaustive compilation of all the simulation tools for robots that can be found commercially or in the literature. With the cases highlighted, the aim is to demonstrate that there is a great diversity of existing solutions and that the offer of this type of software is by no means recent, with examples dating back almost three decades. This is the case, for example, of Webots [1]. Webots, first released in 1996, is a robot simulator which provides a set of features for modelling, simulating, and programming robots, including support for various sensors and actuators. Moreover, Webots supports a wide range of programming languages, including C, C++, Python, Java, and Matlab. Even if at the beginning, Webots was a proprietary licensed software, in 2018 it shifted to open source.

Another example of a vintage robot simulation software is the well-known Gazebo. Similar to Webots, Gazebo [2] is an open-source robot simulation software that has been a cornerstone of robotics research and education since its release in 2004. It offers physics simulation and a library of robot models and environments. Gazebo is often used in conjunction with the Robot Operating System (ROS), providing a platform for developing and testing robotic algorithms. Its open-source nature allows for extensive customization and integration with various hardware and software tools, making it a popular choice in academia.

Still in the early 2000's, SimTwo [3] emerged as a robot simulation platform designed for educational and research purposes. It offers a physics engine and a variety of sensor and actuator models, enabling the simulation of different robotic systems [4].

Automation Studio, from Famic Technologies Inc., is a commercial software suite for simulating and designing industrial automation and fluid power systems [5]. While not exclusively focused on robotics, it offers robust tools for modelling and simulating automation systems, including PLCs, hydraulics, pneumatics, and electrical controls. This makes it particularly useful in courses that cover broader aspects of automation and control engineering bridging the gap between theory and practical application.

RoboDK is also a commercial product that provides robot simulation and offline programming software for industrial robots [6]. Available since 2015, this software supports over 500 robots from more than 50 different manufacturers, making it highly adaptable to various industrial scenarios. RoboDK enables users to create complex simulations and generate robot programs without the need for in-depth programming knowledge.

Finally, CoppeliaSim from Coppelia Robotics, despite having an educational version, it is also a commercial software aiming at robot simulation [7]. It was introduced in 2016 and supports a myriad of different programming languages such as Python, C/C++, Java, and Lua. CoppeliaSim allows users to create detailed simulations of robots and integrate them with real-time controllers.

The list of available robot simulation environments can go on and on. Microsoft Robotics Developer Studio, RobotStudio, RoboLogix, Anykode, LabVIEW Robotics and many, many more examples, can be tracked down by a simple web search. However, while that software offers powerful tools for education and research, they each come with their learning curves. Students must invest significant time and effort to become proficient in these platforms, learning not only the specific interfaces and functionalities of each tool but also the underlying principles of robot modelling and simulation. This learning process, while valuable for educational purposes, may not directly translate into readily applicable skills in the professional world. Many industries use different or more general-purpose software for design, modelling, and control systems. As a result, the time students spend mastering these simulation-specific tools may not provide them with skills that are immediately transferable to their future careers.

In contrast, learning and using software like SolidWorks and LabVIEW can offer more universally applicable competencies. While both software may not be specifically tailored for robotic simulation, their broad applicability and industry relevance make the learning effort more worthwhile and beneficial in the long term. By focusing on these widely-used platforms, educational programs can ensure that students are not only well-versed in theoretical and practical aspects of engineering but also equipped with competencies that enhance their employability and career prospects in various engineering disciplines.

The use of SolidWorks and LabVIEW as leverage tools to learn robotics is not new and there are already some “instructables” available in the WEB and papers published about this subject. For example, Magesh Jayakumar [8] maintain a blog where he provides the source files and some demonstration on how to interface SolidWorks with LabVIEW. However, no dynamics or closed-loop control systems are included in the models.

In a more academic framework, [9] describes the use of those software tools to simulate a Mitsubishi RV-2AJ industrial robot. However, they aimed to use this approach as a research tool and not within the education field. A search in the Web of Science using the query “TS=(‘SolidWorks’ AND ‘robot simulation’) OR TS=(‘LabVIEW’ AND ‘robot simulation’)” returned eleven papers from which only six are minimally related to the current work. Most of them are research-oriented and do not focus on education. For example, [10] presents a cloud-based system that can be used as a web-based robot simulation platform. SolidWorks was the CAD tool used to build the robot’s 3D geometric model.

In [11] describes a robot modelling and simulation approach aiming to operate within the LabVIEW robotics simulation environment and [12] describes a robot

built using the SolidWorks software while providing some details regarding its parameterization, assembly and motion simulation.

A multi-purpose 3D simulator, HNMSim, was reported in [13], where an interface, based on LabVIEW, was developed. Also, relying on LabVIEW, [14] describes the design and implementation of a remote virtual laboratory which can be used to study the sensor's behaviour in mobile robotics.

Finally, in [15], a hexapod beetle was 3D modelled using SolidWorks and its dynamics were simulated through ADAMS software.

In short, all the robot simulation software, even if very impressive, demands a significant learning curve, and the effort invested in mastering it will not yield transferable skills applicable in the professional realm. Furthermore, most academic papers, which resort to SolidWorks and LabVIEW, do not focus on educational aspects of robotics. The online materials that are accessible tend to be overly simplistic, lacking in-depth coverage of critical concepts such as dynamics and closed-loop control systems.

3 The Virtual Robot Challenge

To promote an integrated approach to teaching robotics, students are initially challenged to replicate the six degrees of freedom robotic manipulator illustrated in Fig. 1. However, instead of using a specific platform for robotic simulation, the 3D model should be designed using SolidWorks. It is worth noting that usually, students who attend electrical engineering courses do not have contact with CAD software for 3D modelling, and this is an opportunity to develop this competence since, nowadays, with 3D printing and additive manufacturing technologies, designing custom mechanical parts is a fundamental skill.

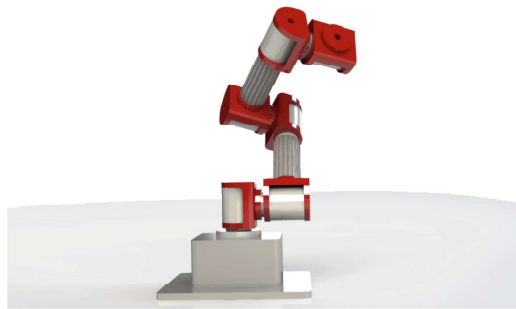


Fig. 1. Robotic manipulator that students are required to 3D model using SolidWorks.

The robotic manipulator comprises three parts that students must draw and assemble. To accomplish this task, they must refer to the part's dimensions provided in technical drawings such as the one presented in Fig. 2.

the next z-axis). For the current robot, the parameters, for each one of the six joints, are presented in Table 1.

Table 1. The Denavit-Hartenberg parameters associated to the 6-DoF robotic manipulator.

Joint/ i	1	2	3	4	5	6
a_i $\ z_{i-1}z_i\ _{x_{i-1}}$	0	0	0	0	0	0
α_i $\angle z_{i-1}z_i _{x_{i-1}x_{i-1}}$	0	$-\pi/2$	$\pi/2$	$\pi/2$	$-\pi/2$	$-\pi/2$
d_i $\ x_{i-1}x_i\ _{z_{i-1}}$	140	85	185	85	185	85
θ_i $\angle x_{i-1}x_i _{z_{i-1}x_{i-1}}$	θ_1^*	θ_2^*	θ_3^*	θ_4^*	θ_5^*	θ_6^*

The Denavit-Hartenberg parameters are then used to compute the transformation matrix \mathbf{T}_i for each joint i . The general format of this matrix is as follows:

$$\mathbf{A}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The overall transformation, \mathbf{T} , from the base frame to the end-effector frame, is given by:

$$\mathbf{T} = \prod_{i=1}^6 \mathbf{A}_i \quad (2)$$

After creating the robot's 3D model, students must associate virtual rotary motors to the robot joints. This is done by enabling the SolidWorks motion add-in and parameterize the virtual actuators as distance type motion rotary motors. Following that, each one of the virtual motion axis can be accessed through LabVIEW's SoftMotion toolbox. Moreover, interface with the Arduino Uno platform can be carried out using the LabVIEW's MakerHub toolbox. In this context, and after uploading the required firmware to an Arduino Uno board, the virtual instrument presented in Fig. 4 was able to change the position of the first joint of the motor according to the position of a joystick connected to one of the analog inputs of the microcontroller.

At this point, students are able to control each joint of the robot externally. However, at this point, the robot's movement does not include any dynamics. Dynamic behaviour affects the stability and control of a robot, especially for tasks that involve rapid movements or interactions with the environment. Proper dynamic modelling ensures that the robot can perform tasks without losing balance or control. To ensure proper dynamic behaviour, closed-loop controllers must be provided at different hierarchical levels. For example, any joint motor must be able to position the joint precisely and accurately to ensure the robot

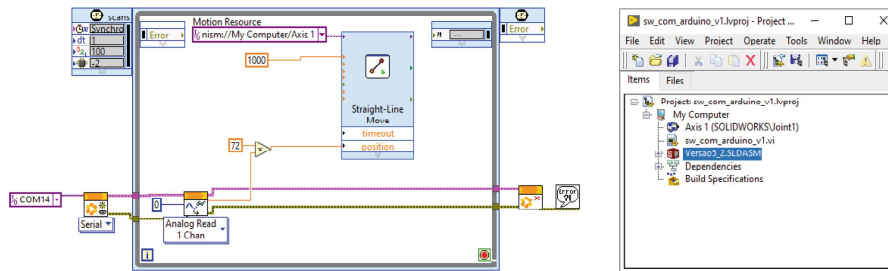


Fig. 4. Movement control of one of the robot joints from an external signal generated by a joystick connected to one of the analog Arduino Uno ports.

can perform its tasks correctly. This ability to follow an angle setpoint must be maintained despite any load disturbances, which requires a properly designed control system. In this work, students are expected to understand the mathematical model of a servomotor obtained from first principles and include its dynamics in the robot simulation. As an electromechanical device, the dynamical model of a servomotor must consider both electrical and mechanical dynamics. For the particular case of a DC, the following equation relates the input voltage $V(s)$ with the electrical current $I(s)$ in the Laplace domain:

$$V(s) = sLI(s) + RI(s) + K_e\Omega(s) \quad (3)$$

where L is the motor's inductance, R its winding resistance, K_e is the back EMF constant and $\Omega(s)$ is the angular velocity in Laplace domain.

On the other hand, the mechanical dynamics relate the torque, $T(s)$, generated by the motor to the rotational dynamics. The motor torque is proportional to the electrical current by a constant K_T and, following the Newton's second law for rotation:

$$T(s) = sJ\Omega(s) + B\Omega(s) \quad (4)$$

in which J denotes the moment of inertia, B the damping coefficient due to friction. Combining the previous two equations leads to the following servomotor transfer function:

$$\frac{\Omega(s)}{V(s)} = \frac{K_T}{JLs^2 + (JR + BL)s + (BR + K_TK_e)} \quad (5)$$

which describes the dynamical behaviour of a servomotor in the Laplace domain.

For particular values of K_T , K_e , J , B , R and L , this transfer function can be simulated directly in Laplace domain using LabVIEW's Control Design and Simulation Module. Assuming, for simplicity, that the model can be approximated to the first-order model,

$$\frac{\Omega(s)}{V(s)} = \frac{1}{s + 1} \quad (6)$$

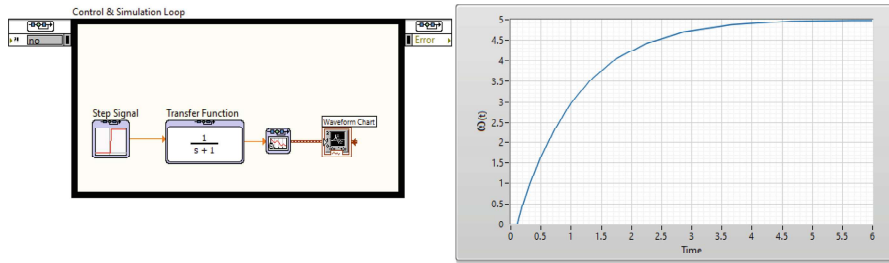


Fig. 5. Open-loop simulation of a joint's servomotor to a step type input.

then its open-loop step response can be simulated from the virtual instrument depicted in Fig. 5.

LabVIEW allows complete integration between blocks and functions of its various add-ons. Figure 6 shows an example of this where a signal generated by a joystick, connected to the Arduino Uno's zero analog input, is applied to the transfer function input. Both the original and processed signal are presented on a waveform chart.

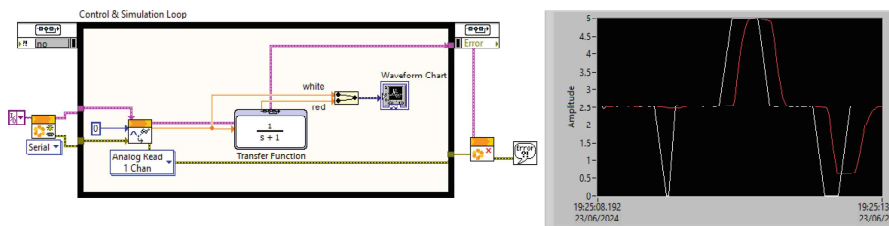


Fig. 6. Hardware-in-the-loop and dynamic behaviour coexisting in the LabVIEW environment.

In the end, after the students being able to perform each of the different parts in isolation, the final task consists of integrating all the knowledge acquired during each step, whether about robot's kinematics, dynamics or even hardware, to create a virtual environment capable of replicating the behaviour of a robotic manipulator.

4 Conclusions

This article described a robotic teaching methodology that relies on computer tools with wide applications in the industry. This teaching approach compels undergraduate students to explore various engineering disciplines, including control, kinematics, programming, and 3D modelling. It reinforces their learning

through hands-on integration, enabling them to apply previously acquired knowledge. Moreover, integrating SolidWorks, LabVIEW, and Arduino to create a virtual representation of a robotic manipulator offers significant advantages for student development compared to environments specifically designed for robot simulation. This is true because the skills and knowledge students acquire to develop the simulation task may directly apply to their future professional environments. Students can spend their time more effectively, gaining hands-on experience with software that they will likely encounter in their careers. This practical approach minimises the time spent learning specialized simulation environments that do not offer transferable skills.

As a final remark, it is worth pointing out that, at the time this paper is being written, this is an ongoing teaching strategy, and no feedback has yet to be gathered.

Acknowledgment. This work was supported by national funds through FCT/MCTES (PIDDAC): CeDRI, UIDB/ 05757/ 2020 (DOI: 10.54499/ UIDB/ 05757/ 2020) and UIDP/ 05757/ 2020 (DOI: 10.54499/ UIDP/ 05757/ 2020); and SusTEC, LA/ P/ 0007/ 2020 (DOI: 10.54499/ LA/ P/ 0007/ 2020).

References

1. Simulating your robots with Webots. <https://cyberbotics.com/#webots>. Accessed 12 June 2024
2. Simulate before you build. <https://gazebosim.org/home>. Accessed 21 May 2024
3. Simtwo - a realistic simulator for robotics. <https://github.com/P33a/SimTwo>. Accessed 18 Nov 2023
4. Costa, P., Gonçalves, J., Lima, J., Malheiros, P.: Simtwo realistic simulator: a tool for the development and validation of robot software. *Theory Appl. Math. Comput. Sci.* **1**(1), 17–33 (2011)
5. Famic technologies inc. <https://www.famictech.com/pt/>. Accessed 19 Nov 2023
6. Simulate robot applications. <https://robodk.com/>. Accessed 10 Dec 2023
7. Accelerate your R and D with next-gen automation and robotics prototyping. <https://www.coppeliarobotics.com/>. Accessed 10 Dec 2023
8. Robot kinematics using LabVIEW, Solidworks and Arduino. <https://labviewarduino.blogspot.com/2014/10/robot-kinematics-using-labview.html>. Accessed 10 Dec 2023
9. Serdarevic, M., Cabaravdic, M., Hadzikadunic, F., Cikmis, A., Tomasevic, D.: Simulation of robot RV-2AJ using softwares SolidWorks and NI LabVIEW (2018)
10. Lianzhong, L., Kun, Z., Yang, X.: A cloud-based framework for robot simulation using WebGL. *Sixth Int. Conf. Intell. Syst. Des. Eng. Appl. ISDEA* **2015**, 5–8 (2015)
11. Xiaowei, C., Chuanwei, S.: The study of robot simulation applications based on the LabVIEW robotics. In: *4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS)*, pp. 66–71 (2017)
12. Wang, T., Shenshun, H., Jun, X., Yan, D., Bi, J.: Simulation design and application of music playing robot based on SolidWorks. *Int. Conf. Measuring Technol. Mechatron. Autom.* **II**, 339–342 (2009)

13. Xu, J., Xie, L., Khanna, N., Chee, W.H.: Cooperative control in HNMSim - a 3D hybrid networked mas simulator. In: 12th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 626–631 (2012)
14. Chaos, D., Chacón, J., Lopez-Orozco, J.A., Dormido, S.: Virtual and remote robotic laboratory using EJS, MATLAB and LabVIEW. *Sensors* **13**(2), 2595–2612 (2013)
15. Yan, B.B., Li, Y., Ren, W.B., Yin, B.L.: Modeling and simulation of gait analysis of hexapod robot. In: Sung, W.P., Kao, J.C.M., Chen, R., (eds.) *Front. Mech. Eng. Materials Eng. II PTS 1 2*, vol. 457, pp. 639–642 (2014)