Romanian International Conference for Education and Research powered by Enformation

Open-Source Technologies for Industrial Robotics Education in Industry 4.0

Marica Marius-Constantin[1], Bostan Ionel[2]

1. Doctoral School of Electronics, Telecommunications and Information Technology, National University of Science and Technology POLITEHNICA Bucharest, marius.marica@upb.ro, ORCID
2. Department of Electronics, Communications and Computers, National University of Science and Technology POLITEHNICA Bucharest, Pitești University Centre, Pitesti, Romania, ionel.bostan@upb, ORCID

DOI: 10.63467/alls17.art8

Abstract

This paper addresses a gap in the current literature, namely the absence of a systematic, educationally oriented comparative framework for open-source robotics simulation platforms in the context of Industry 4.0 curricula. While platforms including Gazebo, Webots, and CoppeliaSim are widely used in academic settings, no consolidated study has evaluated them simultaneously against criteria specific to educational environments, such as ease of use, hardware accessibility, pedagogical resource availability, and alignment with ROS 2. Using a multi-criteria evaluation framework applied across four dimensions, specifically ROS 2 integration, ease of use, hardware accessibility, and availability of pedagogical resources, this paper systematically compares Gazebo, Webots, and CoppeliaSim as primary simulation platforms, with contextual references to emerging tools including NVIDIA Isaac Sim and MuJoCo (Multi-Joint dynamics with Contact). The evaluation is grounded in the specific requirements of engineering education programmes targeting Industry 4.0 competencies, where students must acquire practical robotics skills without direct access to industrial hardware. The results indicate that no single platform is optimal across all educational contexts. Webots offers the lowest barrier to entry and is most suitable for introductory courses, while Gazebo, being tightly coupled with ROS 2, is recommended for advanced undergraduate and graduate-level instruction. CoppeliaSim, through its educational license and modular scripting architecture, provides unique advantages for teaching control systems concepts. The paper concludes with a proposed learning pathway for progressive integration of open-source tools across an academic robotics curriculum, along with practical recommendations for laboratory setup and course design.

Keywords: engineering education, simulation-based learning, active learning, competency-based education, virtual laboratory, curriculum design, open-source tools, robotics pedagogy, Industry 4.0 education.

1. Introduction

The fourth industrial revolution, called Industry 4.0, has fundamentally redefined the skills profile required of engineering graduates in the field of automation and robotics. Industrial robotic systems are no longer isolated machines that perform repetitive tasks, but function as nodes integrated into cyber-physical systems, communicating in real time with digital twins, cloud analytics platforms, and distributed sensor networks. This transformation requires technical education programs to train graduates proficient not only in robot programming, but also in middleware communication architectures, simulation validation, motion planning, and sensor integration (Schwab, 2016).

A critical challenge for academic institutions is to provide students with meaningful practical experience in this field, without access to expensive industrial equipment. Open-source simulation platforms, especially Gazebo, Webots and CoppeliaSim, have emerged as a pedagogically and economically viable answer, allowing students to develop, test and validate robotic applications in realistic virtual environments at almost zero cost. Their increasing adoption in engineering curricula in Europe and North America reflects their recognized educational value (Audonnet et al., 2022; Faller & Feldmüller, 2015).

However, the diversification of these tools has created a new problem: the absence of a clear, educationally grounded framework for selecting the right platform according to the instructional context, level of study or learning objective. Most existing comparisons evaluate these tools on purely technical criteria, without addressing the decisive pedagogical dimension for curriculum designers (Macenski et al., 2022). This paper directly addresses this gap, analysing platforms as learning tools and proposing a structured learning path, grounded in education theory.

2. Theoretical Framework

The selection and sequencing of simulation tools in an engineering curriculum is not a purely technical decision, but a pedagogical one. This chapter presents the theoretical foundations that underpin the evaluation framework applied in the paper, starting from the learning theories relevant to the simulation and reaching the competency-based education models used in the accreditation of engineering programs.

2.1. Simulation-based learning in engineering

The use of simulation in engineering education is based on Kolb's (1984) Theory of Experiential Learning, which identifies four stages of effective learning: concrete experience, reflective observation, abstract conceptualization, and active experimentation. Process simulation environments engage all four stages in a single controlled activity: students interact with virtual robotic processes (concrete experience), observe the system's responses to their programming decisions (reflective observation), develop an understanding of control logic and system dynamics (abstract conceptualization), and iteratively refine their programs (active experimentation) (Kolb, 1984).

Simulation-based learning directly supports Bloom's Revised Taxonomy (Anderson & Krathwohl, 2001). While traditional lecture-based instruction predominantly addresses the lower cognitive levels of memory and comprehension, well-designed simulation exercises engage the higher levels: application (implementing control algorithms on a virtual robot), analysis (diagnosing unexpected system behavior), evaluation (comparing platform responses), and creation (designing autonomous robotic solutions for new scenarios) (Anderson & Krathwohl, 2001).

2.2. Competency-based education for Industry 4.0

Competency-based education frameworks, including EUR-ACE and ABET, define learning outcomes in terms of demonstrable skills. For Industry 4.0-oriented robotics education, the relevant set of competencies includes: competence in ROS 2 middleware, simulation system validation, implementation of motion planning, integration of sensor data and cyber-physical systems design. The assessment framework applied in this paper is explicitly calibrated on these competencies, ensuring that the platform selection criteria reflect real educational requirements (EUR-ACE, 2021). Essentially, these skills cannot be developed exclusively through theoretical instruction. They require experimentation in environments that closely approximate real industrial systems, a requirement that simulation platforms are uniquely positioned to meet at the scale and cost that academic institutions can sustain (Faller & Feldmüller, 2015).

3. Open-Source Platforms as Pedagogical Tools

Five tools were selected for analysis: Gazebo, Webots and CoppeliaSim as primary simulation platforms, ROS 2 as the reference middleware infrastructure and two emerging platforms, NVIDIA Isaac Sim and MuJoCo, as contextual reference. The selection is based on their current adoption in academic robotics programs and compatibility with the ROS 2 ecosystem, the dominant standard after the end of the ROS 1 lifecycle in May 2025. Each tool is examined below from a pedagogical perspective, highlighting the instructional value, the recommended student level, and the curriculum conditions in which it proves to be most effective.

3.1. ROS 2 as the reference infrastructure

ROS 2 is the dominant open-source middleware in modern robotics, constituting the foundation on which all other analysed platforms are built or integrated. Unlike its predecessors, ROS 2 has been designed for production deployment, incorporating real-time communication through the DDS (Data Distribution Service) standard, node lifecycle management and support for multi-robot systems.

From a pedagogical perspective, ROS 2 is not a simulator, but a communication framework that organizes robotic software into discrete nodes that exchange messages through publish-subscribe mechanisms, services and actions. This architecture directly shapes the philosophy of cyber-physical systems in Industry 4.0, making ROS 2 both a practical tool and a conceptual model for teaching systems integration. Long-term supportive distributions, such as the Humble Hawksbill (supported until May 2027)**,** provide necessary stability in the educational context (Macenski et al., 2022; Open Robotics, 2024).

3.2. Webots

Webots is the platform designed most explicitly for education. Its installation through a single package, native support for Windows, macOS, and Linux, eliminates the configuration overhead that frequently consumes lab time at the expense of learning time. Students can execute the first control program of a robot within minutes of the start of the session, a pedagogical advantage directly aligned with cognitive load theory, which recommends minimizing extrinsic cognitive load in introductory instruction (Sweller, 1988).

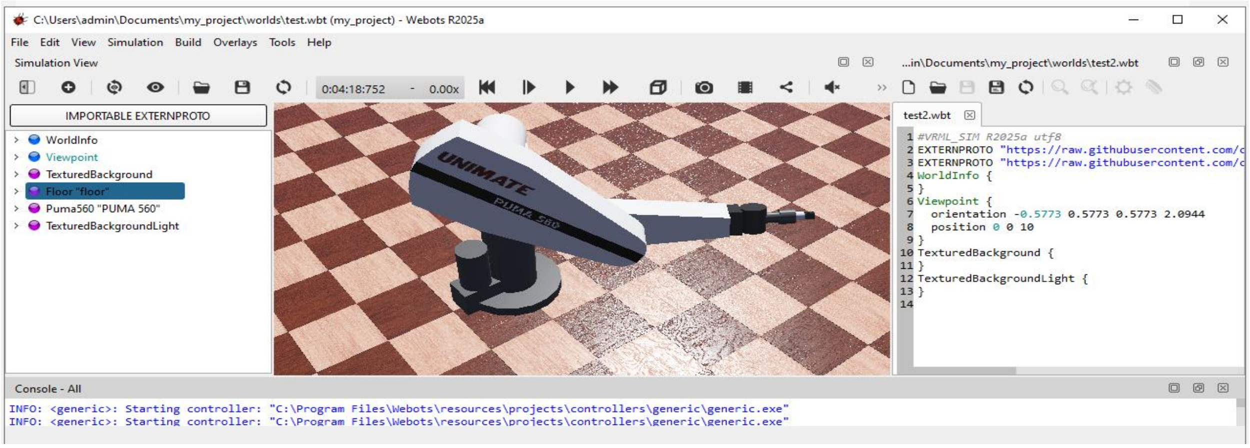

Figure 1. Webots simulation environment with industrial robot model and world editor.

Webots' Python API and extensive library of preconfigured robot and sensor models allow instructors to design exercises focused solely on the development of control algorithms, without requiring knowledge of Linux administration or ROS 2 workspace management. Its ROS 2 interface, through the webots_ros2 package, allows for progressive integration with the broader ecosystem when students are ready for that complexity (Michel, 2004; Cyberbotics, 2024).

3.3. Gazebo

Gazebo offers the tightest integration with ROS 2 of all the platforms evaluated. The native bridge allows ros_gz direct use of the entire ROS 2 toolchain, including RViz2, rqt, ros2bag, and ros2_control, without additional configuration. Pedagogically essential, the programming patterns, message types, and debug flows used in Gazebo simulation are identical to those used on ROS 2-compatible physical hardware (Koenig & Howard, 2004; Quigley et al., 2009). This simulation-reality continuity is a decisive pedagogical property for courses whose learning objective includes direct industrial employability (Koenig & Howard, 2004).

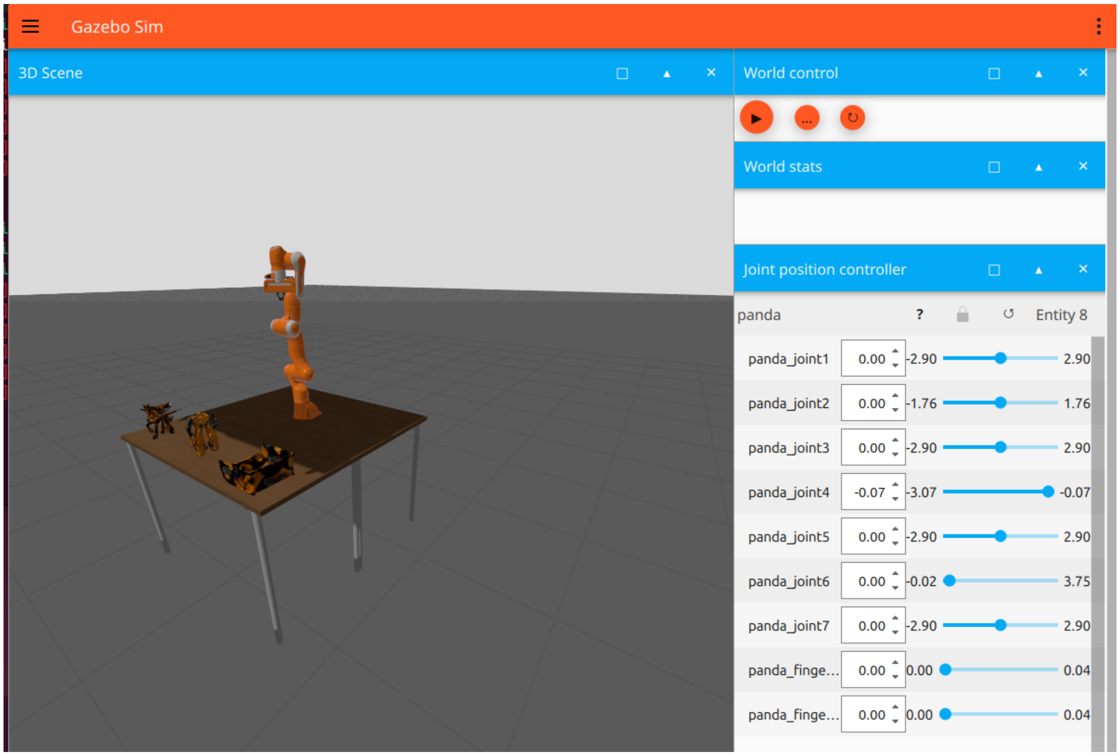

Figure 2. Gazebo simulation environment with industrial robot arm and joint position controller interface.

3.4. CoppeliaSim

CoppeliaSim follows a distributed control architecture in which the individual components of the scene are controlled by independent scripts integrated into Python or Lua. This architecture explicitly shapes the philosophy of decentralized and modular design of cyber-physical systems in Industry 4.0, making it a unique valuable pedagogical tool for courses focused on systems architecture and control design (Rohmer et al., 2013).

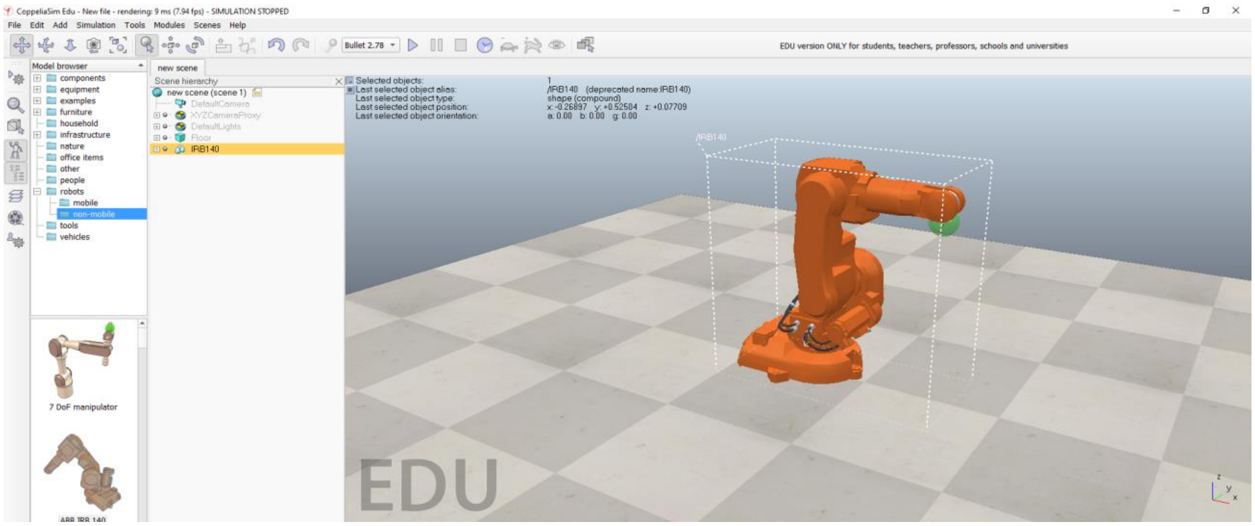

Figure 3. CoppeliaSim environment with distributed scripting architecture and physical engine selection.

Its support for five interchangeable physics engines, including MuJoCo, ODE, Bullet, Newton, and Vortex, enables comparative physics studies that are impossible on other platforms, a particularly valuable feature for advanced students investigating the relationship between simulation fidelity and real-world system behavior. The free educational license removes financial barriers while maintaining the platform's full functionality (Coppelia Robotics, 2024).

3.5. Emerging platform: NVIDIA Isaac Sim and MuJoCo

Two emerging platforms expand the horizon of open-source robotic simulation beyond the primary tools analyzed. NVIDIA Isaac Sim 5.0, released as open-source in 2025, offers GPU-accelerated photorealistic simulation, built on the NVIDIA Omniverse platform. Its main advantages are high visual fidelity, synthetic data generation for training artificial intelligence algorithms, and native support for digital twin workflows, aspects directly relevant to Industry 4.0 integration modules (NVIDIA, 2025). However, the hardware requirement for an NVIDIA dedicated GPU limits the wide educational deployment in standard laboratories for now.

MuJoCo (Model-based Optimization for Joints and Contacts), available for free from 2022 after acquisition by DeepMind/Google, offers a high-fidelity physical engine, optimized for contact-rich manipulation and reinforcement learning. Its MuJoCo Playground framework, recognized at the RSS 2025 conference, demonstrates simulation-reality transfer without additional adjustments on a wide range of robots. In an educational context, MuJoCo is recommended as a specialized module in the advanced stage of the curriculum, for courses that address robotic learning and model-based optimization, not as an introductory platform for simulation (Todorov et al., 2012).

4. Evaluation Methodology and Results

The comparison between the primary platforms, Gazebo, Webots and CoppeliaSim, is based on a multi-criteria framework whose dimensions and weights were established according to the priorities of a curriculum designer for engineering education, not according to pure technical performance (Noori et al., 2022). Unlike existing studies that evaluate robotic simulators predominantly on technical criteria such as physics accuracy or computational performance, the framework proposed in this paper places pedagogical accessibility and compatibility with the standard industrial ecosystem as priority dimensions.

The evaluation applies four criteria: ease of use (30%), availability of pedagogical resources (25%), ROS 2 integration (25%) and hardware accessibility (20%). The weights reflect a deliberate hierarchy: ease of use receives the highest weight because a platform inaccessible to a student without prior experience cannot generate educational value regardless of its technical qualities. The availability of pedagogical resources and the integration of ROS 2 receive equal weights, reflecting the balance between the teaching support needed by the teacher and the alignment with the de facto industry standard. Hardware accessibility receives the lowest share not because it would be unimportant, but because all three platforms meet the minimum requirement to run on standard hardware without a dedicated GPU, the differences between them on this dimension being smaller compared to the other criteria (Noori et al., 2022; Dimmig et al., 2025).

Each criterion is scored on a scale of 1 to 5, where 1 represents minimal capability and 5 represents benchmark performance in that category. The scores were determined through a structured analysis of official documentation, recent academic literature, and feedback from university programs that use these platforms. The weighted total reflects each platform's overall suitability for educational use, not absolute technical performance.

Criterion	Gazebo	Webots	CoppeliaSim	Weighting	Educational Rationale
Ease of use	3/5	5/5	3/5	30%	Reduce extrinsically cognitive arching
Pedagogical resources	4/5	5/5	3/5	25%	Supports autonomous and flipped learning
ROS 2 integration	5/5	3/5	3/5	25%	Alignment with I4.0 competency targets
Hardware accessibility	4/5	**5/5**	4/5	20%	Supports access on personal laptops
Weighted total	3,95/5	4,55/5	3,20/5	—	—

Table 1. Multi-criterion pedagogical evaluation matrix. Weighted total = Sum (score x weight).

Webots obtains the highest weighted score (4.55/5) due to the exceptional ease of use and availability of pedagogical resources, confirming itself as the most suitable platform for introductory instruction. The maximum usability-of-use score reflects the one-step installation, intuitive graphical interface, and browser-based execution option, which completely eliminates the dependence on local configuration. Gazebo obtains a weighted score of 3.95/5, positioning itself as optimal for intermediate and advanced courses aimed at direct industrial skills, due to the maximum score at ROS 2 integration. CoppeliaSim's lower overall score (3.20/5) reflects its steeper learning curve and less systematized pedagogical resources, without diminishing its value as a specialized complementary tool for advanced multi-physics simulation and control system instruction.

An important aspect of the proposed framework is its scalability: the same four dimensions can be applied for the evaluation of any new platform that emerges in the open-source ecosystem, including NVIDIA Isaac Sim and MuJoCo, allowing the comparison to be periodically updated as the ecosystem evolves (Audonnet et al., 2022).

5. The Proposed Learning Path

The results of the assessment, combined with the competence framework in Section 2, underpin a progressive learning path structured in four stages. The central design principle is the progressive disclosure of complexity: each stage introduces a new level of technical abstraction only after the competencies of the previous stage have been consolidated, reducing cognitive overload and maintaining student motivation throughout a multi-year program (Vygotsky, 1978). Unlike curricular approaches that introduce ROS 2 from first contact with robotics, the proposed path explicitly separates learning fundamental concepts from learning the technical ecosystem, allowing students to first build a mental model of robotic behaviour before being exposed to the complexity of software infrastructure.

The path integrates all the tools analysed in section 3, from primary simulation platforms to ROS 2 middleware and emerging platforms, ensuring that each tool is introduced at the curricular moment when it brings maximum pedagogical value. The cognitive levels in Bloom's Revised Taxonomy (Anderson & Krathwohl, 2001) are used as anchors for defining learning objectives per stage, ensuring vertical program coherence and facilitating the design of assessments aligned with the targeted competencies (Anderson & Krathwohl, 2001).

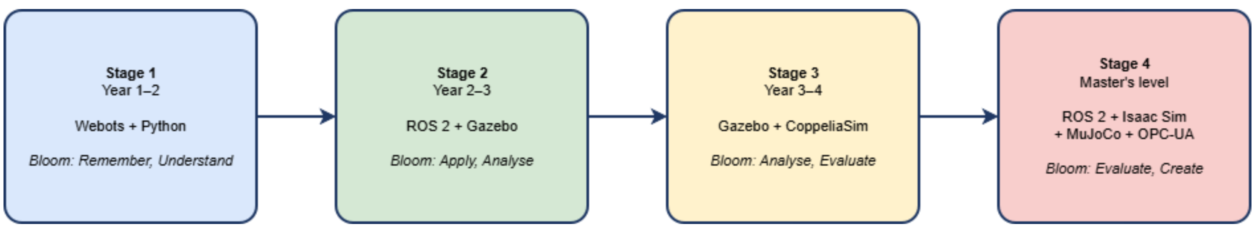

Figure 5. The proposed four-stage learning path for open-source robotics education.

Stage 1, corresponding to years 1-2, focuses on robotics fundamentals using Webots and Python. Students implement sensor-actuator loops, trajectory planning algorithms, and basic reactive behaviors without requiring ROS 2 or Linux knowledge. Webots is selected for this stage based on its maximum ease-of-use score in the proposed framework, which minimizes extrinsic cognitive load.

Stage 2, carried out in years 2-3, introduces middleware and integration through ROS 2 and Gazebo. Students configure node communication, integrate sensor data streams, and validate control logic in a native ROS 2 simulation environment. The transition to Gazebo is motivated by its maximum ROS 2 integration score, ensuring that programming patterns learned in simulation are directly transferable to physical hardware.

Stage 3, specific to years 3-4, addresses advanced robotic manipulation using Gazebo and CoppeliaSim. Students implement motion planning pipelines, compare physics simulation behavior, and design complete virtual robotic cells. CoppeliaSim is introduced at this stage for its distributed control architecture and multi-physics support, enabling comparative exercises unavailable in Gazebo.

Stage 4, addressed to master's programs, integrates Industry 4.0 concepts using a complete ROS 2 stack with OPC-UA, Isaac Sim, and MuJoCo. Students develop integrated automation solutions for cyber-physical systems and analyse platform trade-offs in complex industrial contexts. Isaac Sim and MuJoCo are reserved for this stage due to their high hardware requirements, which become manageable only after accumulating competencies from stages 1-3 (Sweller, 1988).

The vertical coherence of the proposed path lies in the fact that each platform selection is anchored in the evaluation framework results from Chapter 4, transforming the learning pathway into a reproducible model applicable beyond robotics education.

6. Implications for Educators

The effective adoption of open-source tools in robotics education depends not only on the selection of the right platform, but also on how it is integrated into the curriculum, the available technical infrastructure, and the assessment strategies used. The chapter summarizes the main practical recommendations addressed to teachers who plan or update robotics engineering programs oriented towards Industry 4.0.

6.1. Curriculum and assessment design

Performance-based assessment is the most appropriate approach for practical competence-oriented robotics disciplines. Within this approach, students develop and validate a robotic behaviour in a defined simulation scenario and then adapt the solution according to new requirements or modified conditions. This type of assessment aims not only to obtain a functional result, but also to the student's ability to analyse problems, diagnose errors and adapt the system in real time, skills directly required in professional engineering practice (Biggs & Tang, 2011). The robotbenchmark.net platform facilitates formative assessment through automatic scoring and immediate feedback, while reducing the administrative burden of teachers in the introductory stages (Cyberbotics, 2024).

6.2. Virtual laboratory infrastructure

Webots, Gazebo and CoppeliaSim run on standard university hardware, without a dedicated GPU, with a minimum of 8 GB of RAM and a modern multi-core processor. The use of Docker containers with preconfigured images is recommended for intermediate and advanced stages, ensuring a uniform software environment and eliminating configuration inconsistencies that can consume lab time. For institutions with limited infrastructure, platforms like "The Construct" provide access to ROS 2 environments directly from the browser, without local configuration (The Construct, 2024).

6.3. Limits and challenges

The most significant limitation of simulation-based education is the gap between the behaviour of virtual environments and that of real physical systems. Robots used in industrial applications are influenced by sensor noise, calibration errors and communication delays, which are difficult to fully reproduce in simulation. For this reason, predominantly simulation-based curricula should be complemented by demonstrations or exercises on real physical platforms, even short ones, to develop a realistic engineering perspective on the limitations of robotic systems (Noori et al., 2022). Another challenge is the rapid evolution of the open-source ecosystem, which requires a continuous effort to update teaching materials and software infrastructure. In addition, although the proposed route develops vendor-independent ROS 2 competencies, exposure to at least one major industrial platform, such as Siemens, ABB or KUKA, remains recommended for professional integration in sectors with dominant commercial ecosystems (Schwab, 2016).

7. Conclusions

This paper brings two main contributions in the field of engineering education oriented towards Industry 4.0. The first is a multi-criteria evaluation framework explicitly calibrated for the educational context, which evaluates open-source robotic simulation platforms not by technical performance, but by their pedagogical adequacy, aiming at ease of use, availability of teaching resources, integration with the ROS 2 ecosystem and hardware accessibility. The second contribution is a four-stage progressive learning path, grounded in Kolb's Experiential Learning Theory and Bloom's Revised Taxonomy, in which the selection of the platform per stage is directly justified by the results of the assessment framework, ensuring methodological coherence between benchmarking and curricular recommendation.

For an educator planning or updating a robotics program, the paper provides a concrete answer to a dilemma frequently encountered in teaching practice: choosing the right instrument and the optimal order of introduction of technical complexity. The proposed answer is not an arbitrary preference, but a decision anchored in a reproducible methodological framework, which places Webots in the fundamentals stage, ROS 2 and Gazebo in the integration phase, CoppeliaSim for advanced multi-physics control and simulation scenarios, and Isaac Sim together with MuJoCo at the master level. This model remains adaptable as the open-source ecosystem evolves, precisely because the logic of selection is pedagogical, not technological.

Future research may aim at the longitudinal assessment of the competencies developed by students in programs that adopt the proposed pathway, as well as the development of standardized open-source benchmarks for comparing educational performance between institutions. Also, the integration of generative artificial intelligence and simulation based on digital twins represent promising directions for the future evolution of simulation-based robotics education.

References

Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's educational objectives*. Longman.

Audonnet, F. P., Hamilton, A., & Aragon-Camarasa, G. (2022). *A systematic comparison of simulation software for robotic arm manipulation using ROS2*. arXiv. https://arxiv.org/abs/2204.06433.

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university* (4th ed.). Open University Press.

Coppelia Robotics. (2024). *CoppeliaSim user manual*. Accessed May 10, 2026, from https://coppeliarobotics.com.

Cyberbotics. (2024). *Webots reference manual*. Accessed May 10, 2026, from https://cyberbotics.com/doc/reference.

Dimmig, C. A., Silano, G., McGuire, K., Gabellieri, C., Hönig, W., Moore, J. L., & Kobilarov, M. (2025). *Survey of Simulators for Aerial Robots: An Overview and In-Depth Systematic Comparisons*. *IEEE Robotics & Automation Magazine*, *32*(2), pp. 153-166. https://doi.org/10.1109/MRA.2024.3433171.

EUR-ACE. (2021). *EUR-ACE framework standards and guidelines for accreditation of engineering programmes*. ENAEE. Accessed May 10, 2026, from https://www.enaee.eu/eur-ace-system.

Faller, C., & Feldmüller, D. (2015). *Industry 4.0 learning factory for regional SMEs*. *Procedia CIRP*, *32*, pp. 88-91. https://doi.org/10.1016/j.procir.2015.02.117.

Koenig, N., & Howard, A. (2004). *Design and use paradigms for Gazebo, an open-source multi-robot simulator*. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, pp. 2149-2154. https://doi.org/10.1109/IROS.2004.1389727.

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice-Hall.

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). *Robot Operating System 2: Design, architecture, and uses in the wild*. *Science Robotics*, *7*(66), eabm6074. https://doi.org/10.1126/scirobotics.abm6074.

Michel, O. (2004). *Webots: Professional mobile robot simulation*. *Journal of Advanced Robotic Systems*, *1*(1), pp. 39-42. https://doi.org/10.5772/5618.

Noori, F. M., Portugal, D., Rocha, R. P., & Couceiro, M. S. (2022). *A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion*. *Simulation Modelling Practice and Theory*, *116*, 102501. https://doi.org/10.1016/j.simpat.2022.102501.

NVIDIA. (2025). *Isaac Sim 5.0 documentation*. Accessed May 10, 2026, from https://docs.isaacsim.omniverse.nvidia.com.

Open Robotics. (2024). *ROS 2 Jazzy Jalisco documentation*. Accessed May 10, 2026, from https://docs.ros.org.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). *ROS: An open-source Robot Operating System*. In *Proceedings of the IEEE*

International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software.

Rohmer, E., Singh, S. P. N., & Freese, M. (2013). *V-REP: A versatile and scalable robot simulation framework*. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, pp. 1321-1326. https://doi.org/10.1109/IROS.2013.6696520.

Schwab, K. (2016). *The Fourth Industrial Revolution*. World Economic Forum.

Sweller, J. (1988). *Cognitive load during problem solving: Effects on learning*. *Cognitive Science*, *12*(2), pp. 257-285. https://doi.org/10.1207/s15516709cog1202_4.

The Construct. (2024). *The Construct: ROS development and training platform*. Accessed May 10, 2026, from https://www.theconstruct.ai.

Todorov, E., Erez, T., & Tassa, Y. (2012). *MuJoCo: A physics engine for model-based control*. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pp. 5026-5033. https://doi.org/10.1109/IROS.2012.6386109.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.