OFFLINE PROGRAMMING AND SIMULATION PLATFORMS FOR INDUSTRIAL ROBOTS: A REVIEW WITH A RESEARCH-ORIENTED PERSPECTIVE ON ROBODK

Bogdan Marian Verdete[1,3][0009-0001-1856-9786], Sorin-Ionuț Badea[2][0009-0002-8485-5489], Cristian Radu Badea[2][0000-0002-0994-6834], Cristina Luciana Dudici[3,][0009-0003-8377-3120], Alexandra Cojocaru[3][0009-0000-1325-2854]*
[1]Robodictive SRL, Gheorghe Doja Street, Pitești, Romania
[2]National Institute of Research and Development in Mechatronics and Measurement Technique, 6-8 Pantelimon Road, Bucharest, Romania
[3]National University of Science and Technology POLITEHNICA Bucharest, 313 Splaiul Independentei, Bucharest, Romania
E-mail(s): cristina.ionita@upb.ro (✉)

Abstract - Offline programming and simulation have become core tools in industrial robotics because they allow robot programs, layouts, and motion strategies to be developed and validated before deployment in the physical cell. This review examines the current landscape of software platforms used for industrial robot offline programming, with attention to both vendor-specific solutions and multi-brand environments. The paper discusses major platforms such as ABB RobotStudio, Siemens Process Simulate, KUKA.Sim, Yaskawa MotoSim, Visual Components, DELMIA Robotics, and RoboDK, and compares them through criteria relevant to both engineering practice and academic work, including robot and controller support, simulation scope, path validation, collision analysis, calibration, extensibility, and suitability for reproducible experimentation. A special emphasis is placed on RoboDK as a research-oriented platform. The reason is that RoboDK combines multi-brand offline programming with an accessible API ecosystem, post-processing tools, simulation-based feasibility checks, estimated cycle-time analysis, and calibration-oriented workflows that support method development and repeatable evaluation. In this perspective, RoboDK is discussed not simply as an industrial software product, but as an experimental environment for studies on trajectory generation, path smoothness, obstacle avoidance, CAD/CAM-driven automation, and teaching-by-demonstration workflows. The review argues that vendor-native software remains important when controller-level fidelity and commissioning are the main priorities, while RoboDK offers strong advantages when flexibility, rapid prototyping, cross-platform comparison, and research productivity are the main objectives. The paper is intended to support researchers, educators, and engineers in selecting suitable software tools and in understanding how RoboDK can be positioned within current offline programming and simulation practice.

Keywords: Industrial robots, Offline programming, Robot simulation, RoboDK, Digital twin, Robot calibration, Trajectory planning, Path validation, Research workflows, Industrial automation.

1. Introduction

Industrial robots are widely used in manufacturing processes such as welding, machining, material handling, palletizing, inspection, and assembly. As robotic applications become more complex, the programming stage also becomes more demanding, especially when production cells include multiple devices, constrained workspaces, strict cycle-time requirements, and frequent product changes. In this context, classical on-site programming through manual teaching is often time-consuming, interrupts production, and limits early testing possibilities. Offline programming (OLP) emerged as a practical response to these limitations by allowing robot paths, workcell layouts, and task logic to be prepared and validated in a virtual environment before deployment on the shop floor [1].

Over time, OLP has evolved from a tool mainly used to reduce downtime into a broader digital

dx.doi.org/10.17683/ijomam/issue24.40

engineering workflow that includes 3D workcell modeling, collision checking, reachability analysis, cycle-time estimation, and, increasingly, digital-twin-oriented validation. In parallel, recent approaches have also explored virtual reality and digital twin methods as alternative ways to define robot motion and transfer human-guided operations into robotic execution, which further expands the role of simulation in industrial robotics research and development [1,2].

The current software landscape for industrial robot OLP is diverse. Vendor-specific platforms such as ABB RobotStudio, FANUC ROBOGUIDE, and KUKA.Sim are closely aligned with the ecosystems of their respective manufacturers and emphasize offline programming, simulation, and, in some cases, virtual commissioning capabilities [3],[4],[5]. At the same time, broader environments such as Siemens Process Simulate, Visual Components, and DELMIA Robotics extend the discussion toward multi-brand programming, cell-level validation, PLC-oriented virtual commissioning, and virtual-twin workflows [6],[7],[8]. This diversity is beneficial in practice, but it also makes software selection less straightforward, especially when the intended use is not only industrial deployment, but also research.

From a research perspective, the requirements are somewhat different from those of routine industrial commissioning. A research-oriented platform should support reproducible experiments, flexible scenario generation, data extraction, external scripting, and straightforward transfer from simulated tasks to executable robot programs. These requirements are especially important in studies on trajectory generation, path validation, calibration, obstacle avoidance, CAD/CAM-driven automation, and algorithm benchmarking, where the software platform becomes part of the experimental methodology rather than only a deployment tool.

Within this context, RoboDK deserves particular attention. According to its official documentation, RoboDK provides an API for Python and other programming languages, support for a broad robot library, post-processors for multiple robot brands and controllers, and calibration-oriented tools such as TwinTrack for improving the accuracy of offline-generated programs [9],[10],[11]. These characteristics make it suitable not only for industrial programming tasks, but also for rapid prototyping of research methods and repeatable experimental workflows.

This research-oriented use of RoboDK is already reflected in recent studies. For example, multiple palletizing scenarios following different trajectories has been tested by simulation and analysed in order to identify the least time consuming palletizing strategy in [12]. More recently, RoboDK was also used as the simulation environment for analysing the influence of geometric continuity levels on industrial robot trajectories, showing its suitability for structured comparative studies on motion quality and trajectory evaluation [13].

Other applications such as Dassault Catia and ABB RobotStudio were also used in research purpose such as in [14] where new forward kinematic models [15],[16] could be validated based on the virtual models of robots.

Based on this context, the present paper reviews major offline programming and simulation platforms for industrial robots, with a special focus on RoboDK as a research-oriented environment. The aim is twofold: first, to compare current platforms through criteria relevant to engineering practice and academic work; second, to clarify how RoboDK can be used not only as an OLP package, but also as a practical research instrument for developing, testing, and documenting new robotic methods.

2. Taxonomy of Offline Programming and Simulation Platforms

A practical taxonomy of current software solutions for industrial robot offline programming can be built around the dominant scope emphasized by each platform. In this review, three broad categories are considered: vendor-specific offline programming environments built around a single robot manufacturer, multi-brand offline programming platforms intended to support several robot brands in a common environment, and broader digital manufacturing platforms in which robot programming is embedded into virtual commissioning, process validation, and plant-level simulation workflows. RoboDK is treated here as a hybrid case because, although it clearly belongs to the multi-brand OLP category, its API-centered workflow and calibration-oriented functions also make it relevant for research-driven use.

The first category includes software environments closely linked to a specific robot vendor. ABB RobotStudio is presented by ABB as an offline programming and simulation tool for robotic applications, while FANUC ROBOGUIDE is described as software tailored specifically for FANUC robots and aimed at creating, programming, and simulating robotic workcells in 3D. In the same vendor-oriented group, KUKA.Sim is positioned as simulation software for efficient offline programming of KUKA robots, with real-time visualization, cycle-time analysis, reachability checking, and collision detection, whereas Yaskawa MotoSim EG-VRC [16] is dedicated to offline programming and simulation of the Yaskawa product family, including robots, gantries, tracks, and positioners. As a result, these platforms can be grouped together because they are strongly aligned with the hardware, terminology, and deployment logic of a single manufacturer ecosystem.

A second category includes multi-brand platforms whose main value lies in programming flexibility across different robot families. Visual Components fits well in this group because it presents its OLP software as compatible with major industrial robot brands and focused on fast, repeatable, and reusable robot programming without stopping production. RoboDK also belongs to this class at the software-definition level, since it is presented as a complete solution for robot simulation and offline programming that supports many robot types and integrates with most CAD/CAM software. In this category, the software emphasis is less on a single controller ecosystem and more on portability, process flexibility, and rapid deployment of robot programs across heterogeneous industrial settings.

A third category is formed by broader digital manufacturing environments in which robot programming is only one part of a larger validation chain. Siemens Process Simulate explicitly combines 3D robotic OLP with robot calibration, robot program download and upload, virtual commissioning, PLC-oriented validation, continuous manufacturing tools, human simulation, and VR-based process review. DELMIA Robotics is positioned in a similar high-level space, where robot programming is integrated with collaborative 3D workcell design, virtual validation, virtual commissioning, digital continuity between design and manufacturing, and virtual twin technology. Compared with the previous two categories, these platforms are therefore better described as factory- or PLM-oriented environments in which robotic simulation is embedded into a broader engineering and commissioning context,

RoboDK occupies an especially interesting position in this taxonomy because it connects the accessibility of a multi-brand OLP platform with features that are useful in academic and experimental work. Its official documentation presents RoboDK not only as software for simulation and offline programming, but also as a platform that supports API-based automation through Python, C#, C, C++, Visual Basic, and MATLAB. In addition, RoboDK provides calibration workflows through TwinTrack, with documentation stating that robot accuracy can be improved when programs are generated offline or through teach-by-demonstration workflows. This combination of multi-brand support, scripting accessibility, and calibration support helps explain why RoboDK is frequently suitable not only for industrial programming tasks, but also for research workflows that require repeatable scenario generation, program automation, and simulation-based validation.

This taxonomy is useful because it clarifies that "offline programming software" is not a single homogeneous class of tools. Some platforms are optimized for deployment inside one robot manufacturer ecosystem, some are optimized for flexible programming across multiple robot brands, and others are designed as broader digital manufacturing environments where robot programming is tightly coupled with virtual commissioning and production-system validation. Positioning RoboDK within this landscape is important for the present review because its strengths are not limited to generic OLP functionality but extend toward research-oriented experimentation and method development.

3. Comparative Criteria for Platform Evaluation

To compare software platforms that were developed for partially different purposes, a common evaluation framework is required. In this review, the comparison is based on criteria that affect both industrial deployment and academic work: controller compatibility, geometric and kinematic validation, process-level estimation, calibration and sim-to-real transfer, automation through APIs, and the possibility of extending the workflow toward virtual commissioning. This selection is consistent with the broader evolution of offline programming from simple path preparation toward digital-twin-based development and validation of robotic work cells [17].

A first criterion is the relationship between robot-brand coverage and controller specificity. In vendor-native systems, the main strength is usually the close alignment with the corresponding controller family. In multi-brand systems, the key issue is whether the platform can reliably convert generic simulated motions into controller-specific robot code. For this reason, post-processors should be treated as a central comparison point, because they determine how offline-generated programs are translated into executable instructions for the target controller [18].

A second criterion concerns geometric and kinematic validation. A useful OLP platform should allow the user to verify reachability, joint configuration, interference zones, and collision risk before transferring the program to the physical cell. Process Simulate explicitly includes collision-free robot motion path planning, reach testing, and robot-specific target and motion-path handling, while RoboDK documentation treats collision detection and path feasibility checking as standard validation steps within the offline workflow [19].

A third criterion is the quality of process-level estimation. In many industrial scenarios, it is not enough for a path to be feasible; the user also needs information about expected cycle time and, in some cases, energy use or synchronization within a larger workstation. Process Simulate describes cycle-time and energy optimization as part of its robotics workflow, while RoboDK states that simulated cycle time can be estimated but that accuracy depends strongly on controller behaviour, motion type, rounding, and speed and acceleration limits. Therefore, timing output should be interpreted as controller-dependent and model-dependent rather than as a universally exact value [20].

A fourth criterion is calibration and sim-to-real transfer. A simulated model can be geometrically correct and still be insufficient for precise deployment if the physical robot, tool, fixtures, or resources are not aligned with the digital cell. Recent work on industrial robot accuracy assessment has pointed out that experimental procedures for pose and path accuracy are often inconsistent, which makes systematic evaluation difficult. In parallel, recent digital-twin studies show that calibrated updates of resource locations are required when path planning must remain valid after cell reconfiguration. For these reasons, calibration support and accuracy evaluation should be treated as major criteria in any serious comparison of OLP platforms [21],[22].

A fifth criterion is extensibility through scripting, APIs, and data access. This point is especially important for research, where the software platform is often used to generate many scenarios, run batch evaluations, export results, and test custom algorithms. RoboDK documentation provides API examples for offline programming and states that many add-ins are implemented in Python, which shows that the platform can be integrated into automated experimental workflows. From a review perspective, this kind of openness separates software that is convenient for single-cell programming from software that can support repeatable and programmable studies [9].

Finally, virtual commissioning capability should be treated as a distinguishing criterion rather than a default requirement for every platform. For some studies, especially those centered on path generation, smoothing, or calibration, full PLC-level validation is not essential. For cell-level integration studies, however, the ability to simulate real PLC code and actual robot programs becomes highly relevant. This is why broader digital manufacturing platforms and lighter multi-brand research tools should not be judged by exactly the same priorities, even when both are described as offline programming environments [6].

Based on these observations, the comparison developed in the next sections is centered on six practical questions: how well a platform connects to real controllers, how reliably it validates motion in the digital cell, how informative its process-level estimates are, how well it supports calibration, how open it is to automation and experimentation, and how far it can be extended toward system-level commissioning. These criteria are broad enough to cover both industrial deployment needs and research-oriented use cases, while still allowing RoboDK to be evaluated in a fair and clearly defined context [9,17,22].

4. Comparative Review of Major Platforms

When the platforms introduced in the previous sections are examined through the evaluation criteria discussed above, it becomes clear that no single software environment is equally strong for all use cases. Vendor-specific tools tend to prioritize controller-oriented fidelity, deployment readiness, and process modules tailored to one robot family. Broader digital manufacturing platforms place more emphasis on virtual commissioning, lifecycle integration, and system-level validation. Multi-brand OLP tools, in contrast, are generally strongest when flexibility, portability, and rapid reconfiguration are required. This distinction is important because many comparisons between platforms become misleading when software designed for plant-level validation is judged by the same expectations as software designed for fast multi-brand programming or research prototyping.

4.1 ABB RobotStudio

As far as controller-centered OLP environments go, ABB RobotStudio is still among the best. With the express purpose of ensuring that the robot behavior seen in the virtual station corresponds to the behavior anticipated in the actual system, ABB introduces RobotStudio as an offline tool for programming and simulation based on its virtual controller technology. This puts RobotStudio in a good position when preparing ABB installations reliably is the primary goal, rather than doing experiments independent of brands. Additionally, the environment is not limited to modification, as shown by the availability of the RobotStudio SDK and Smart Component development support. Though its natural center of gravity is still ABB-specific rather than cross-platform, RobotStudio is nonetheless a solid choice for industrial projects centered on ABB and for sophisticated users requiring expansions inside the ABB software ecosystem [23].

4.2 Fanuc Roboguide

Similarly to how FANUC ROBOGUIDE follows vendor-native logic, it places a noticeable focus on application-oriented modules. Material handling, feasibility studies, conveyor tracking, machine modeling, and palletizing layout generation are some of the practical deployment tasks shown by the HandlingPRO and PalletPRO modules of ROBOGUIDE, which is described by FANUC as 3D offline programming and simulation software for FANUC robots. When it comes to broad categories, KUKA.Sim and Yaskawa MotoSim EG-VRC are on par. Welcome to KUKA.Sim is software that allows for optimization and efficient offline programming outside of production, whereas MotoSim EG-VRC is described by Yaskawa as a solution for complex systems with extensive control features and the ability to simulate robots, gantries, tracks, and positioners in various process types. When the user is heavily involved with a single manufacturer's ecosystem and requires software that adheres to the lingo, logic, and common application packages of that ecosystem, these platforms come together to make an ideal solution [24].

4.3 Siemens Process Simulate and Delmia Robotics

The profiles in DELMIA Robotics and Siemens Process Simulate are distinct. As part of a more comprehensive digital manufacturing workflow, this article presents robot offline programming with 3D process validation, virtual commissioning, and production-system analysis in general. Robotics virtual commissioning is defined independently by Siemens as verifying system behavior in simulation using the actual control software before shop-floor deployment; it is an explicit positioning of Process Simulate around robotics, offline programming, and virtual commissioning.

Similarly, DELMIA focuses on digital twin technology, virtual commissioning, validation of manufacturing designs, 3D robotic simulation, and related areas. Therefore, these settings are particularly effective when assessing the robot is not done in a vacuum, but rather as a component of a more comprehensive automated system that incorporates control logic, equipment interactions, and engineering continuity at the factory level. On the other hand, these systems tend to be better suited to heavy-duty industrial workflows with plenty of integration rather than light-weight experimental iteration [25].

4.4 Visual Components

Visual Components occupies an intermediate position between classical OLP and broader digital factory simulation. Its official materials present robot offline programming as a fast, multi-brand workflow compatible with major industrial robot brands and supported by 3D CAD-based programming. The platform also emphasizes the ability to create programs without interrupting production and to improve robot utilization in practical manufacturing scenarios. This suggests that Visual Components is especially attractive for users who need fast deployment across heterogeneous robot fleets and who value ease of use, rapid layout iteration, and accessible workflow design. Compared with vendor-native tools, however, its main value lies less in controller-family depth and more in speed, reuse, and multi-brand practicality [26].

4.5 RoboDK

Within this comparison, RoboDK stands out because it combines several characteristics that are rarely balanced together in the same environment. Official RoboDK sources describe it as a complete solution for robot simulation and offline programming, with support for more than 1,200 industrial robots from 80 manufacturers, integration with CAD/CAM software, post-processing for multiple controllers, API access through Python, C#, C, C++, Visual Basic, and MATLAB, and calibration tools intended to improve offline accuracy. RoboDK also highlights collision and singularity avoidance, external-axis integration, and offline generation of ready-to-run robot programs. From a comparative perspective, this means that RoboDK does not compete primarily by trying to replicate the deepest controller-specific behaviour of one vendor ecosystem or the broadest plant-level commissioning chain of a PLM platform. Its main strength is the combination of multi-brand reach, scripting accessibility, relatively low barrier to scenario generation, and calibration-aware sim-to-real improvement. For research, this combination is especially valuable because it supports repeatable experimentation, batch automation, and rapid testing of new planning or analysis methods without locking the methodology to a single robot brand [10],[27].

Overall, the comparison suggests a practical division of strengths. RobotStudio, ROBOGUIDE, KUKA Sim, and MotoSim EG-VRC are especially strong when deployment fidelity inside one robot family is the dominant priority. Process Simulate and DELMIA become more compelling when robotic programming must be integrated into virtual commissioning and wider production-system validation.

Visual Components is highly relevant for fast multi-brand industrial OLP. RoboDK, in turn, is particularly well positioned for research-oriented work because it offers enough industrial realism for meaningful simulation studies while remaining open, scriptable, and efficient enough for methodological experimentation. For the present review, this is the key reason why RoboDK deserves a dedicated discussion not only as a software product, but as a research instrument.

5. RoboDK as a Research-Oriented Platform

RoboDK can be regarded as more than a general offline programming package because its structure is well aligned with the needs of experimental robotics research. Official documentation presents RoboDK as a simulation and offline programming environment with multi-language API access, controller-oriented post-processing, collision checking, cycle-time estimation, calibration support, and CAD/CAM connectivity. In practical terms, this means that the same environment can be used to create a robotic scenario, generate executable code, automate repeated tests, and evaluate the resulting motion under comparable conditions. For research, this combination is valuable because it reduces the fragmentation that often appears when simulation, code generation, calibration, and experimental scripting are handled in separate tools.

A first research advantage of RoboDK is its scriptability. The API is available across several programming languages, and RoboDK add-ins can be created directly from Python scripts, which makes it possible to automate repetitive tasks and package custom research functions into reusable extensions. This is particularly important in article-oriented work, where the same station or algorithm must often be re-executed under multiple parameters sets in order to compare path planners, interpolation strategies, collision scenarios, or controller outputs. In this sense, RoboDK supports not only simulation, but also reproducible workflow design [28]. In addition to automating repetitive tasks, scripts can also be used to develop add-ons or applications that complement and extend the by-default capabilities of RoboDK. For example, scripts can be made that can read data files and automatically create target points for the robot, trajectories can be generated in different ways than they are generically generated by linear interpolation or interpolation in joint space, different aspects of trajectories can be analysed by extracting metrics such as: number of nodes, continuity, impulsivity, etc. Interfaces can be created to transmit signals or communicate with other applications, and much more. This type of repeated simulation-based evaluation is consistent with broader comparative methodology, where multiple computational variants are tested under the same conditions to assess performance differences in a controlled manner [29]

A second important strength is the bridge it offers between virtual and physical execution. RoboDK documentation explicitly treats post-processors as a key step in offline programming because they generate controller-specific robot programs, and the available documentation also states that these post-processors can be modified when the output format must be customized. In parallel, TwinTrack extends the platform toward teach-by-demonstration workflows by allowing users to teach points or curves manually, while RoboDK calculates and displays the robot position in real time and can create and simulate the corresponding program. This makes RoboDK especially useful for research on sim-to-real transfer, human-guided programming, and calibration-assisted path definition.

A third advantage is the platform's relevance for manufacturing-oriented experimentation. RoboDK CAM documentation states that the software supports strategies such as surface machining, drilling, roughing, and other toolpath operations, together with material-removal simulation. Separate documentation for robot machining indicates that generic 3-axis or 5-axis manufacturing operations and generic APT, NC, or G-code files can be converted into robot simulation and robot programs. This expands the platform well beyond simple pick-and-place examples and makes it suitable for research on machining, finishing, deburring, path conversion from CAD/CAM, and process-specific trajectory studies [30].

Another reason RoboDK fits well in academic research is that it can serve multiple roles in the same laboratory context. It can be used for teaching the principles of offline programming, for prototyping user interfaces or data-processing routines, and for generating the simulated results needed in research papers. This broader academic value is consistent with recent work. This educational relevance is also consistent with broader recent work showing that data-driven and software-supported approaches are increasingly important in improving learning outcomes and supporting academic decision-making [31]. Work presented in [32] on vision-based teleoperation of an articulated robotic arm shows how a robotic experimental framework can be rapidly developed around accessible sensing and control concepts, while work from [33] on software-supported learning highlights the importance of adaptable software environments in academic settings. The academic usefulness of such platforms is further supported by recent work on inclusive software environments that integrate gesture recognition and local AI models, highlighting the growing value of adaptable human-centered interfaces in learning and experimentation [34]. These publications are not RoboDK validation papers in the strict sense, but they strengthen the argument that research-oriented platforms are especially valuable when they can support both experimentation and teaching in the same institutional ecosystem.

Overall, RoboDK is best understood not as a replacement for every vendor-native or PLM-level platform, but as a flexible research instrument positioned between industrial realism and methodological openness. It offers enough fidelity to produce meaningful robotic simulations and executable programs, while remaining sufficiently accessible for scripting, extension, calibration-assisted improvement, and rapid iteration. For this reason, RoboDK is particularly well suited to studies involving trajectory generation, path validation, obstacle-aware planning, teach-by-demonstration, CAD/CAM-driven operations, and educational experimentation.

6. Limitations of RoboDK and Other OLP Platforms / Current Challenges and Future Directions

Despite the clear progress of offline programming and simulation software, one limitation remains common to almost all platforms: simulation fidelity is not equivalent to physical accuracy. A recent review on robot digital twin systems shows that manufacturing-oriented robot twins are becoming more connected, adaptive, and data-rich, but it also highlights persistent challenges related to model consistency, data integration, and practical deployment in dynamic environments. In robotic machining, these limitations become especially visible, because industrial robots are affected by coupled kinematic and dynamic errors, and virtual commissioning must account for these effects if the simulated model is expected to remain useful for accurate process preparation [35],[36].

A second challenge is that a digital model does not remain valid automatically when the physical cell changes. In [37] Bennulf et al. show that, in a Plug & Produce setting, the digital twin must be updated with calibrated resource locations when modules are moved, otherwise the path planner can no longer rely on the simulated scene. A related conclusion is reported by Kang et al., who argue that parameter uncertainty and model bias can significantly reduce the credibility of a digital twin if calibration is not treated in a systematic way. This means that, for both vendor-native and multi-brand OLP environments, the usefulness of simulation depends not only on path generation itself, but also on the quality and maintenance of the underlying model [38].

For RoboDK, these general issues appear in a very practical form. According to the official documentation, RoboDK can estimate cycle time during simulation, but the result depends strongly on the robot controller, movement type, rounding, and real speed and acceleration limitations; under some conditions, the cycle time calculation can be wrong by a factor of two or more. The calibration documentation also states that the Optical CMM workflow is limited to 6-axis robot arms, and that the accuracy obtained after calibration depends on the robot model and the setup. These statements do not weaken the value of RoboDK as a research platform, but they define important methodological limits for articles that rely only on simulated timing or unvalidated absolute accuracy claims [20],[39].

Vendor-native environments can reduce part of this gap by staying closer to the real controller behaviour. ABB explicitly presents the RobotStudio virtual controller as a way to achieve highly accurate simulation and virtual commissioning of robotic solutions. However, this advantage is linked by design to a manufacturer-specific ecosystem. For comparative research, educational work, or multi-brand studies, this creates a trade-off between controller-level fidelity and methodological portability across different robot families.

Looking ahead, the most important future direction is not simply "better simulation", but more adaptive and connected robotic software workflows. Recent review work on robot digital twins identifies artificial intelligence, Industrial Internet of Things technologies, cloud and edge connectivity, virtual reality, and adaptive control as major enablers of future manufacturing systems. In parallel, Asif et al. describe rapid and automated configuration of robot manufacturing cells through reconfigurable architectures and automated program generation, while in [40] Dhanda et al. show that human-robot collaboration, digital twins, and AI are becoming central in the transition toward Industry 5.0. In this context, future OLP platforms will likely be evaluated not only by their ability to simulate a robot path, but also by their support for reconfiguration, continuous model updating, and integration with data-driven decision layers [41]. In this broader context, AI-based perception and defect-detection methods are also becoming relevant for manufacturing-oriented robotic workflows, because they can complement simulation by providing data-driven feedback for process monitoring and quality assessment [42].

Related developments in human-centered control systems, including real-time BCI-based automation, further illustrate the broader movement toward more adaptive and accessible interaction paradigms in engineering applications [43,44].

7. Conclusions

This review has shown that offline programming and simulation platforms for industrial robots now cover a broad spectrum, from vendor-centered tools optimized for controller fidelity to multi-brand environments and larger digital manufacturing platforms oriented toward virtual commissioning and system-level validation. As recent literature on robot digital twins and reconfigurable manufacturing indicates, the field is moving toward more connected, adaptive, and data-rich workflows. For this reason, software selection should be guided by the real objective of the work: deployment inside one vendor ecosystem, broader manufacturing-system validation, or research-oriented experimentation.

Within this landscape, RoboDK is particularly relevant from a research perspective because it combines several capabilities that are rarely available together in such an accessible form. Official RoboDK sources present the platform as a multi-brand solution for robot simulation and offline programming, with CAD/CAM integration, controller-specific post-processors, API support, collision and singularity handling, and calibration tools. This combination makes RoboDK suitable for studies on trajectory generation, path validation, obstacle-aware planning, CAD/CAM-based robot programming, and educational experimentation in laboratories that need a common software environment across several robot brands.

At the same time, the review also shows that RoboDK should be used with clearly defined methodological limits. Simulation output alone should not be treated as direct proof of physical accuracy when cycle time, machining precision, or dynamic behavior are the main outcome measures. In these cases, calibration, additional measurements, and external validation remain necessary. This does not reduce the scientific value of RoboDK. On the contrary, it clarifies its strongest position: RoboDK is best understood not as a universal replacement for vendor-native commissioning software, but as a flexible and efficient research instrument that supports structured experimentation before final deployment on the real robot.

For academic work, this is precisely its main strength. RoboDK allows researchers to build, compare, automate, and document robotic methods in a common environment while remaining close enough to industrial workflows to keep the results relevant. In a research context increasingly shaped by adaptive digital twins, automated cell reconfiguration, and human-centered manufacturing systems, this balance between flexibility and industrial relevance is what makes RoboDK especially useful as a platform for both experimentation and publication.

Acknowledgement

This work has been supported by: (1) CERMISO Center—Project Contract no.159/2017, Program POC-A.1-A.1.1.1-F; (2) Research Program Nucleu within the National Research Development and Innovation Plan 2022–2027, carried out with the support of MCID, project no. PN 23 43 04 01; (3) Support Center for International RDI Projects in Mechatronics and CyberMix-Mechatronics, Contract no. 323/22.09.2020, project co-financed by the European Regional Development Fund through the Competitiveness Operational Program (POC) and the national budget; and (4) ERASMUS-EDU-2023-EUR-UNIV, Project 101124676—EELISA, funded by the European Union, https://eelisa.eu/ (accessed on 15 April 2026)

References

[1] Pan, Z.; Polden, J.; Larkin, N.; Van Duin, S.; Norrish, J. Recent progress on programming methods for industrial robots. Robotics and Computer-Integrated Manufacturing 2012, 28(2), 87–94. https://doi.org/10.1037/ppm0000185
[2] Burghardt, A.; Szybicki, D.; Gierlak, P.; Kurc, K.; Pietruś, P.; Cygan, R. Programming of Industrial Robots Using Virtual Reality and Digital Twins. Applied Sciences 2020, 10(2), 486. https://doi.org/10.3390/app10020486
[3] ABB. RobotStudio® Suite. Official product page accessed 14 April 2026. https://www.abb.com/global/en/areas/robotics/products/software/robotstudio-suite
[4] FANUC. Simulation Software ROBOGUIDE. Official product page accessed 15 April 2026. https://www.fanuc.eu/eu-en/accessory/software/simulation-software-roboguide
[5] KUKA. KUK Sim – Simulation Software. Official product page accessed 16 April 2026. https://www.kuka.com/en-us/products/robotics-systems/software/simulation-planning-optimization/kuka_sim
[6] Siemens. Process Simulate Software. Tecnomatix official product page, accessed 15 April 2026. https://www.siemens.com/en-us/products/tecnomatix/process-simulate-software/
[7] Visual Components. Robot Offline Programming. Official product page accessed 11 April 2026. https://www.visualcomponents.com/products/robot-offline-programming/
[8] Dassault Systèmes. Robotics Software: Simulation, Design & Programming. DELMIA official product page, accessed 12 April 2026. https://www.3ds.com/products/delmia/industrial-engineering/robotics

[9] RoboDK. RoboDK API. Official documentation accessed 17 April 2026. https://robodk.com/doc/en/RoboDK-API.html
[10] RoboDK. Supported Robots. Official documentation/product page accessed 18 April 2026. https://robodk.com/supported-robots
[11] RoboDK. TwinTrack / Robot Calibration. Official documentation accessed 15 April 2026. https://robodk.com/doc/en/TwinTrack.html
[12] Cristoiu, C.; Nicolescu, A.; Verdete, B, Robotic palletizing cell for optimum 5L water bottles palletizing. Proceedings in Manufacturing Systems 13 (4), 177
[13] Cristoiu, C.A.; Drăgoi, M.-V.; Nechita, R.-M.; et al. The Impact of Geometric Continuities (C1, C2, and C3) on the Trajectories of Industrial Robots. Technologies 2026, 14(3), 176 https://doi.org/10.3390/technologies14030176
[14] Cristoiu, C.; Nicolescu, A. New approach for forward kinematic modeling of industrial robots. Res. & Sci. Today 13, 136
[15] Cristoiu, C.; Nicolescu, A. New approach for forward kinematics modeling of industrial robots with closed kinematic chain. Res. & Sci. Today 13, 145
[16] Yaskawa. MotoSim EG-VRC. Official product page accessed 15 April 2026. https://www.yaskawa.eu.com/products/software/productdetail/product/motosim-eg-vrc_1686
[17] Tipary, B.; Erdős, G. Generic development methodology for flexible robotic pick-and-place workcells based on Digital Twin. Robotics and Computer-Integrated Manufacturing 2021, 71. https://doi.org/10.1016/j.rcim.2021.102140
[18] RoboDK. *Post Processors*. Official documentation accessed 15 April 2026. https://robodk.com/doc/en/Post-Processors.html
[19] RoboDK. Collision Detection. Official documentation accessed 13 April 2026. https://robodk.com/doc/en/Collision-Avoidance.html
[20] RoboDK. Cycle Time. Official documentation accessed 16 April 2026. https://robodk.com/doc/en/General-Cycle-Time.html
[21] Ferrarini, S.; Pedrocchi, N.; Villagrossi, E.; Beschi, M.; Ghidini, S.; Mazzoleni, M.; Tosatti, L.M. A method for the assessment and compensation of positioning errors in industrial robots. Robotics and Computer-Integrated Manufacturing 2024, 87. https://doi.org/10.1016/j.rcim.2023.102622
[22] Bennulf, M.; Falkman, P.; Hanson, L.; Vickhoff, J.; Halme, R.; Flodin, A. Automatic Calibration and Update of a Digital Twin for Plug & Produce. Sensors 2025, 25(22). https://doi.org/10.3390/s25226885
[23] ABB. RobotStudio SDK. ABB Developer Center, official documentation, accessed 15 April 2026. https://developercenter.robotstudio.com/api/robotstudio/index.html
[24] FANUC PalletPRO. Official product page, accessed 2 April 2026. https://www.fanucamerica.com/solutions/applications/palletizing-robots
[25] Siemens. Robotics virtual commissioning. Official technology page, accessed 18 April 2026 https://www.siemens.com/en-us/technology/robotics-virtual-commissioning/
[26] Visual Components. Robot programming. Official use-case page, accessed 14 April 2026 https://www.visualcomponents.com/use-cases/
[27] RoboDK. Offline Programming. Official product page, accessed 15 April 2026 https://robodk.com/offline-programming
[28] RoboDK. Add-ins. Official documentation, accessed 15 April 2026. https://robodk.com/doc/en/Add-ins.html
[29] Drăgoi, M.-V., Afram, R.M., Alzubaidi, A.A.A.M.H., Jiga, G.G., Ali, H.A., Puiu, R.-A., Petrea, G., Hank, A. Comparative Analysis of Interpolation and Extrapolation Methods in Regression Models. U.P.B. Sci. Bull., Series C, 2026, 88(1).
[30] RoboDK. RoboDK CAM. Official documentation accessed 16 April 2026. https://robodk.com/doc/en/CAM.html
[31] Buzducea (Drăgoi), C.-A., Drăgoi, M.-V., Cristoiu, C., Puiu, R.-A., Puiu, M., Petrea, G. & Navligu, B.-C. Machine Learning in Education: Predicting Student Performance and Guiding Institutional Decisions. Educ. Sci. 2026, 16(1), 76. https://doi.org/10.3390/educsci16010076
[32] Drăgoi, M.-V.; Frimu, A.-V.; Postelnicu, A.; Puiu, R.-A.; Petrea, G.; Hank, A. Interactive Teleoperation of an Articulated Robotic Arm Using Vision-Based Human Hand Tracking. Biomimetics 2026, 11(2), 151. https://doi.org/10.3390/biomimetics11020151
[33] Drăgoi, M.-V.; Puiu, R.-A.; Petrea, G.; Burtea, C.; Spiridon-Mocioacă, T.-M.; Puiu, M.; Suciu, C.P. Improving the Learning Process for Students. Personalizing Learning through Software Applications. U.P.B. Sci. Bull., Series C, 2025, 87(1).
[34] Drăgoi, M.-V., Nisipeanu, I., Puiu, R.-A., Tache, F.-G., Spiridon-Mocioacă, T.-M., Hank, A. & Cristoiu, C. An Inclusive Offline Learning Platform Integrating Gesture Recognition and Local AI Models. Biomimetics 2025, 10(10), 693. https://doi.org/10.3390/biomimetics10100693
[35] Qin, Q.; Liu, Z.; Zhong, R.; Wang, X.V.; Wang, L.; Wiktorsson, M.; Wang, W. Robot digital twin systems in manufacturing: Technologies, applications, trends and challenges. Robotics and Computer-Integrated Manufacturing 2026, 97. https://doi.org/10.1016/j.rcim.2025.103103
[36] Ni, H.; Hu, T.; Deng, J.; Chen, B.; Luo, S.; Ji, S. Digital twin-driven virtual commissioning for

robotic machining enhanced by machine learning. Robotics and Computer-Integrated Manufacturing 2025, 93. https://doi.org/10.1016/j.rcim.2024.102908

[37] Bennulf, M.; Ramasamy, S.; Zhang, X.; Danielsson, F.; Swathanandan, J. Automatic Calibration and Update of a Digital Twin for Plug & Produce. Sensors 2025, 25(22). https://doi.org/10.3390/s25226885

[38] Kang, B.; Park, C.; Kim, H.; Hong, S. A digital twin calibration for an automated material handling system in a semiconductor fab. Journal of Manufacturing Systems 2025, 80, 1013-1028. https://doi.org/10.1016/j.jmsy.2025.04.015

[39] RoboDK. Robot Calibration (Optical CMM). Official documentation accessed 15 April 2026. https://robodk.com/doc/en/Robot-Calibration-Creaform-Robot-calibration.html

[40] Dhanda, M.; Rogers, B.A.; Hall, S.; Dekoninck, E.; Dhokia, V. Reviewing human-robot collaboration in manufacturing: Opportunities and challenges in the context of Industry 5.0. Robotics and Computer-Integrated Manufacturing 2025. https://doi.org/10.1016/j.rcim.2024.102937

[41] Asif, S.; Bueno, M.; Ferreira, P.; Anandan, P.; Zhang, Z.; Yao, Y.; Ragunathan, G.; Tinkler, L.; Sotoodeh-Bahraini, M.; Lohse, N.; Webb, P.; Hutabarat, W.; Tiwari, A. Rapid and automated configuration of robot manufacturing cells. Robotics and Computer-Integrated Manufacturing 2025. https://doi.org/10.1016/j.rcim.2024.102862

[42] Tălîngă, A.-M., Hadăr, A., Drăgoi, M.-V., Nisipeanu, I., Ali, H.A., Suciu, C.P. YOLO-v8 in Capturing Imperfections Generated by Changing 3D Printer Parameters. U.P.B. Sci. Bull., Series C, 2024, 86(4).

[43] Drăgoi, M.-V., Nisipeanu, I., Frimu, A., Tălîngă, A.-M., Hadăr, A., Dobrescu, T.G., Suciu, C.P., Manea, A.R. Real-Time Home Automation System Using BCI Technology. Biomimetics 2024, 9(10), 594. https://doi.org/10.3390/biomimetics9100594

[44] Gamazeliuc, G.; Nechita, R.M.; Ulerich, O.; Bădiţă-Voicu, L.L.; Cristoiu, C. Fusion of AI and Robotics for Transforming Neurosurgery. International Journal of Mechatronics and Applied Mechanics 2024, 18, 175–183. https://doi.org/10.17683/ijomam/issue18.23