JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Effectiveness of Digital Twin Framework for Collaborative Robotic Manipulation

Quang-Huan Dong[1], Tuan-Khanh Nguyen[1*], Chi-Cuong Tran[2], The-Thinh Pham[4], Duy-Tan Do[3], Hoang-Vinh-Khang Nguyen[1], Quang-Chien Nguyen[3]

[1]*Vietnamese-German University, Vietnam*
[2]*National Taiwan University of Science and Technology, Taiwan*
[3]*Ho Chi Minh University of Technology and Education, Vietnam*
[4]*Can Tho University of Technology, Vietnam*

[*]Corresponding author. *Email: khanh.nt@vgu.edu.vn*

ARTICLE INFO

Received: 18/02/2025
Revised: 25/03/2025
Accepted: 29/04/2025
Published: 28/11/2025

KEYWORDS

Automation and control engineering;
Collaborative robots;
Digital twin;
Simulation and modeling;
Industrial case study.

ABSTRACT

This work examines the effectiveness of a digital twin (DT) framework using an industrial pick-and-place case study with a collaborative robotics arm. The problem addressed is the need for improved production process planning and visualization in robotics. The employed method involves creating a DT and evaluating its fidelity to a physical robotic system performing a pick-and-place task. Evaluations included comparing the digital and real robot trajectories, utilizing ISO 9283 performance testing, and analyzing metrics like RMSE, MAPE, and R2. The evaluation shows initial positive results indicating that the proposed DT framework fits the real data well, thus, demonstrates feasibility of this approach. The results can be used to improve the planning and visualization of the production process for collaborative robot arm with 3D printers or adapted for more complex industrial machine tools. These improvements can support the enterprises to spot potential problems before they occur, optimize performance and reduce costs.

Doi: https://doi.org/10.54644/jte.2025.1835

1. Introduction

Applying new technologies to the production process enables many benefits such as reducing risks and increasing competitiveness for the manufacturers. Among the advanced technologies, digital twin (DT) allows manufacturers to improve the production quality by reducing errors. Recently, the application of DT has increased very rapidly thanks to the significant progress of technologies such as robotics, cloud computing, internet of things and especially artificial intelligence. In DT technology, a system or real object is virtually represented. In the digital realm, it is comparable to a mirror image. DT enables us to forecast the performance of the physical device, understand how it functions, and even improve its design by simulating its behavior with this virtual counterpart.

The latest research from Japan's AIST [1] presents a summary of software platforms and tools proposed for human-robot interaction applications aimed at developing DT engineering systems. The paper surveys and presents VR/AR/MR technology solutions that humans can control and interact with physical robot systems using extended reality devices, as described in the most recent published articles. However, these studies are quite general and do not highlight specific applications. In the studies mentioned in [2], there are surveys on the development trends of intelligent manufacturing systems with the application of DT technology. The applications of this technology have been successfully implemented in various fields such as product manufacturing, management, and real-time diagnosis of the status of production equipment. The paper also points out that DT technology applied in robotics will be a necessary development direction in the near future with the era of Industry 4.0.

The combination of DTs and robotic arms offers numerous benefits, especially in manufacturing. To delve deeper into this approach, scientists at a Russian research center [3] have created a 3D model of a UR10e robotic arm in a digital environment using the Unity3D tool. This model will be linked and controlled with its physical counterpart using the ROS (Robot Operating System) platform. The paper

employs 3D scanning technology to display the system as point clouds, accurately determining the position of the actual robot in the workspace. By linking the robot within a synchronized digital environment, information from the real system is synchronized. The results show that tasks can be planned and trajectories can be executed accurately in both the simulated and real robot environments. The paper also integrates AR/VR technology, allowing users to control the robot system in a visualized workspace.

Recognizing the significance of 3D modeling and development in a virtual environment, the work in [4] developed a 3D robotic system model in a virtual environment where humans can interact through VR/MR technology. The 3D robotic system in the virtual environment utilizes Unity3D for design and modeling, and communicates with VR/MR devices and the actual robotic system through ROS. Additionally, an application of DT technology for industrial robots performing metal welding tasks [5] provided valuable insights for further research. This is a virtual robotic arm system equipped with a welding torch, developed using Unity3D, allowing users to interact through a virtual reality (VR) integrated controller. This virtual system is digitized and synchronized in real-time with the actual welding robot system, enabling the inspection, evaluation, analysis, and recording of the user's welding process. Furthermore, the paper applies machine learning methods to classify the skill levels of different users from the welding process data stored by the virtual system. From this, the welding path can be planned and refined in the virtual environment before being applied to the actual welding robot system. The results of the paper demonstrate the effectiveness of the proposed method by ensuring that each user with different skill levels can complete the same welding task consistently.

Methods for connecting the virtual environment to the real work environment by studying relevant articles on applications in manufacturing are explored in [6], [7]. Paper [6] proposed a method for connecting robots in a virtual environment to real robots based on an improvement to the ROS platform called the ROSIE adapter. The results of the paper demonstrated the ability to connect and respond quickly and easily to platforms such as AR/NSI. Additionally, [7] addressed the issue of real-time monitoring of an industrial robot system that does not have a built-in monitoring system. The paper proposed using Unity3D to create a DT of the industrial robot process for planning and monitoring the system. This digital system is connected to the industrial robot system through ROS communication to perform monitoring tasks. On the other hand, applications of DTs and robotic arms are also widely used in the medical field [8], [9]. A prime example is the design and control of robotic manipulation systems for non-contact remote diagnosis in otolaryngology [8], which has been presented in detail and is highly applicable in special conditions such as during the COVID-19 pandemic. Regarding user health, a very beneficial DT-related study on the combination of DTs and the Metaverse for consumer health using a typical research approach [9] is relevant. The paper described this typical research as follows: virtual health consulting supported by DTs in the Metaverse, surgery training supported by DTs in the Metaverse, and self-health assessment supported by DTs in the Metaverse. The authors used inverse dynamics to recreate the process of converting physical movement into movement in the virtual world. These typical studies have demonstrated the functionality and feasibility of combining DTs and the Metaverse for consumer health.

Overall, the integration of DTs with robotic arms offers significant advantages across various sectors, particularly in manufacturing and healthcare. However, research on the effectiveness of DT frameworks for collaborative robotic arms remains limited, especially within manufacturing applications. Existing DT applications in robotics often lack comprehensive validation that bridges simulation and real-world performance, particularly concerning position deviation and processing time for each stage of tasks in specific applications. This study addresses this gap by investigating the efficacy of a DT framework through a case study involving an industrial pick-and-place task using the UR10e collaborative robotic arm. The work examines the position discrepancy between the DT and the real robot by employing metrics such as RMSE, MAPE, and R2. The processing time for motion planning and real robot task execution in each stage of the pick-and-place application is reported. Position performance testing is conducted in accordance with ISO 9283 to validate synchronization between the DT and real robots in different scenarios. The findings provide insights into improving the planning and visualization of production processes, particularly in relation to 3D printing. Consequently, this work advances the

understanding of relevant processes and establishes a foundation for further exploration and development in this area.

The remainder of the paper is structured as follows: In Section 2, the study procedures to develop a DT framework and a case study utilizing DT technology for collaborative robot arms are reported. In Section 3, the analysis of the framework's performance through the pick-and-place case study and ISO 9283 performance testing is described. In Section 4, the paper summarizes the results and methods implemented and proposes future work.

2. Materials and Methods

2.1. Constructing a mathematical model of DTs for collaborative robots

According to Pieper's principle [10], if a 6-DOF serial robot has three consecutive coordinate frames meeting at a common origin, then there exists a solution to the nonlinear inverse kinematics problem. However, this issue does not arise with UR cobots, as their wrist design lacks a spherical configuration where the origins of the three consecutive wrist frames converge at a single point. Instead, as depicted in Figure 1, the offset introduced by link 5 disrupts this desirable kinematic property. Nevertheless, Pieper's condition does not preclude the existence of alternative analytical solutions to the inverse kinematics problem, even when the robot's structure deviates from this configuration. Therefore, an analytical solution to the inverse kinematics problem for UR cobots can still be pursued, despite the absence of a spherical wrist structure.

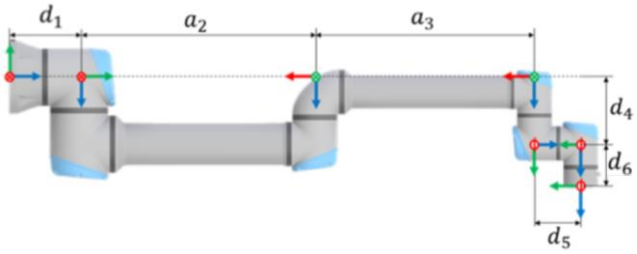

Figure 1. *Allocate robot coordinate frames using the DH convention.*

Numerous studies have analyzed the kinematics of UR cobots in references [11] - [13], with some solutions compared statistically in [14], and a recent classification method for inverse kinematic branches of UR robots has been proposed in [15]. Forward and inverse kinematics of robots can be studied using various methods. A well-known technique employs the original or modified Denavit-Hartenberg (DH) parameters [16]. The inverse kinematics solution allows for the calculation of configurations leading to a specific pose, which is crucial when controlling the position and orientation of the end-effector. Unlike the forward kinematics solution, the inverse kinematics solution is computed sequentially from joints 1, 5, 6, 3, 2, and 4. The inverse kinematics solution we adopt is based on [17], [18]. Avoiding singular configurations in the inverse kinematics solution is beneficial for robot design, control, and its DT [19].

2.2. Building a DT framework for collaborative robot arms

The digital model of the collaborative robot arm, including the manipulator, is built based on the support of CAD design technology and then converted into a URDF (Universal Robotic Description Format) file. This URDF format facilitates seamless integration into the Unity3D virtual environment and is employed in ROS to describe the robot's components and their respective properties. The URDF file encapsulates details about the robot's structure, geometry, DH parameters (kinematic model parameters), and inertia of the joint components, specifying connections between joints and links, joint types, link lengths, and their mass and inertia properties. The main components of UR10e, basics of URDF and an excerpt of the generated URDF file are shown in Figure 2. The UR10e components are depicted on the left, including a series of links and joints that describe the robot arm structure. For instance, the upper arm structure consists of the shoulder_link, shoulder_lift_joint, and upper_arm_link, denoted as (1), (2) and (3), respectively. The shoulder_link connects the rotating shoulder joint to the fixed base and enables horizontal rotation. The shoulder_lift_joint provides vertical rotation to the arm

JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

by rotating the upper_arm_link. The upper_arm_link extends from the shoulder to the elbow, acting as a rigid connection that links movement from the shoulder_lift_joint to the next joint in the kinematic chain.

For each link (cp. Figure 2 - link box), a link origin defines the position and orientation of the visual, collision, and inertial relative to the link's reference frame of the link, which is determined by its joint. For each joint (cp. Figure 2 - joint box), a joint origin specifies the transform from the parent link to the child link. The joint is positioned at the origin of the child link. Therefore, the origin represents the relative position and orientation of the child frame with respect to the parent frame. The joint axis is defined within the joint frame.

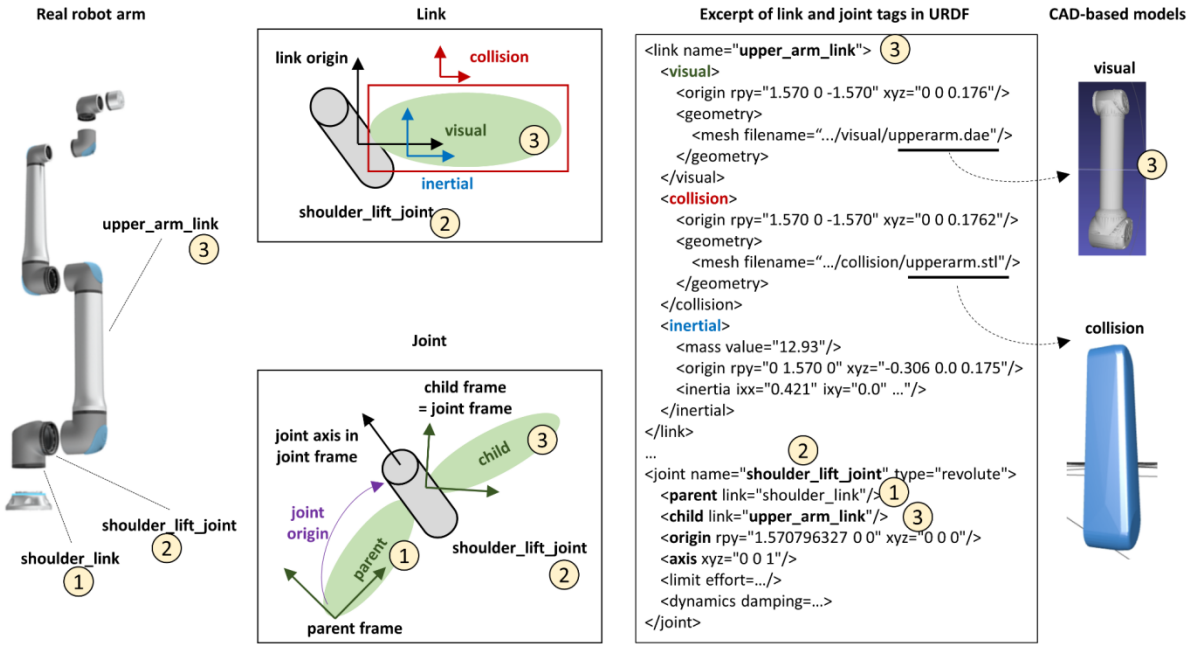

Figure 2. *An excerpt of the generated URDF file for UR10e robot arm.*

The link tag (cp. Figure 2 - excerpt of link and joint tags in URDF - upper_arm_link) describes a rigid body with visual features, collision properties, and inertia characteristics. Specifically, the visual element defines the shape of the link used for simulations or visualizations. The collision element describes the geometry employed for collision detection. The inertial element specifies the mass and inertia of the link, which are crucial for simulating dynamics such as gravity or torque. The joint tag (cp. Figure 2 - excerpt of link and joint tags in URDF - shoulder_lift_joint) describes relative motion between parent and child links. Different joint types can be defined such as revolute or prismatic. The axis element, for instance, specifies the rotation, as shoulder_lift_joint is revolute. Additionally, the limit element defines constraints on the movement of the joint. For example, the effort attribute can be used to specify maximum torque or force applied at the joint. Finally, the dynamic element specifies the dynamic properties used during simulation, with the damping attribute representing how much the joint resists movement when it is moving.

The CAD-based models visually represent the link's geometry are shown on the right of Figure 2. The geometry elements refer to two mesh-format files which represent 3D objects by defining their shape and surface details using vertices, edges, and faces. The collision is usually a simplified version of the robot's actual shape (e.g. boxes or cylinders) to make physics and planning computations more efficient. The CAD-based collaborative robot arm model is initially converted to the URDF file format containing only geometric properties (meshes). This initial URDF file allows for visual representation of the robot in Unity3D, verifying the geometric accuracy of the CAD model. Additional parameters, such as joint mass, geometric inertia, joint types, and link connections, are subsequently added and adjusted to reflect manufacturer specifications. Configuration files are also incorporated to define joint limits, initial positions of links and joints, and constraints based on the model. As a result, this digital

collaborative robot arm model accurately represents a physical system on a digital platform. Figure 3 shows a screenshot of the robot arm model rendered in Unity3D, demonstrating the visual fidelity of the DT.

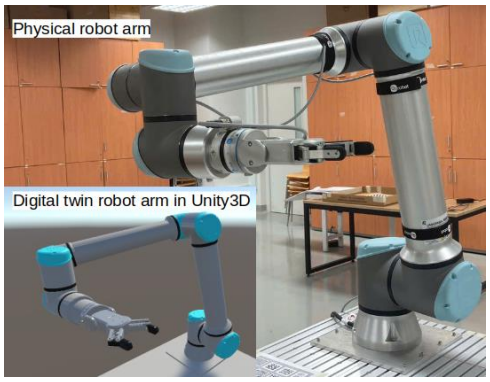

Figure 3. *Physical robot arm and its virtual replica in Unity3D environment.*

Unity3D communicates with ROS via the ROS-Unity3D bridge, enabling real-time data exchange and visualization. For robot arm motion control, MoveIt [20] is employed. Built on top of ROS, MoveIt is an open-source platform which provides key functionality for robotics manipulation. There are some planners available in MoveIt such as Pilz (pilz industrial motion planner) or OMPL (open motion planning library). The OMPL supports many sampling-based algorithms such as RRTConnect, EST, RRTstar, RRT. The pilz_industrial_motion_planner supports linear (LIN), circular (CIRC), or point-to-point (PTP) motion planning.

The UR10e collaborative robot offers a 1300 mm reach, defining a substantial work volume for different automated operations. With up to 12.5 kg payloads and 0.05 mm repeatability, the UR10e provides consistent and precise task execution. Therefore, the robot is a suitable candidate for applications that require both stable operation and flexible integration into diverse industrial settings such as 3D printing or CNC tending. An open-source ROS driver provided by the manufacturer defines an interface between the robots and ROS. Consequently, the UR10e robots are fully compatible with ROS. The supported interface allows MoveIt – built on top of ROS, as aforementioned – to read values from the robot' sensors and send commands to it for following a planned trajectory.

UR robots support some read-only broadcasters that read states from the robot and publish them as ROS topics. For instance, the joint_state_broadcaster publishes all joints' positions, velocities, and motor currents to a topic named joint_states. To synchronize with the physical robot, a subscriber is implemented in the Unity3D digital environment, subscribing to the joint_states topic and updating the joints of the virtual robot accordingly.

The results validate the successful development of a virtual robot arm model, which will be progressively refined for integration with a physical robot arm. As illustrated in Figure 3, the implementation and practical operation details will be discussed in the subsequent sections.

2.3. Developing a case study of an industrial pick-and-place robot utilizing DT technology

The results demonstrate the successful development of a robot arm model within a virtual environment. An outline of the collaborative robot arm DT framework and industrial pick-and-place robot case study is presented in Figure 4. A digital environment in Unity3D has been established to create a representative industrial case study, focusing on the following components:

- Pick-and-place operations for machine tool components, specifically a 3D-printed product.
- Integration of the collaborative robot arm's mathematical model (described in Section 2) with the DT framework (outlined in Section 3) to execute the industrial pick-and-place task.
- The foundational hardware system required to implement this case study, consisting of a computer, robot, gripper, sensors, support frame, machine tools, and machine parts, as illustrated in Figure 5. The real UR10e robot sends real-time data through the ROS to a computer, the data can be stored in a ROS database (ROS bag) for further analysis (e.g. studying position accuracy).

The DT consists of a Process Twin that simulates the entire grasping process (e.g. Move, Pick, Move, Place) and System Twin which comprises of Components Twin (i.e. DT robot) and DT 3D printer.

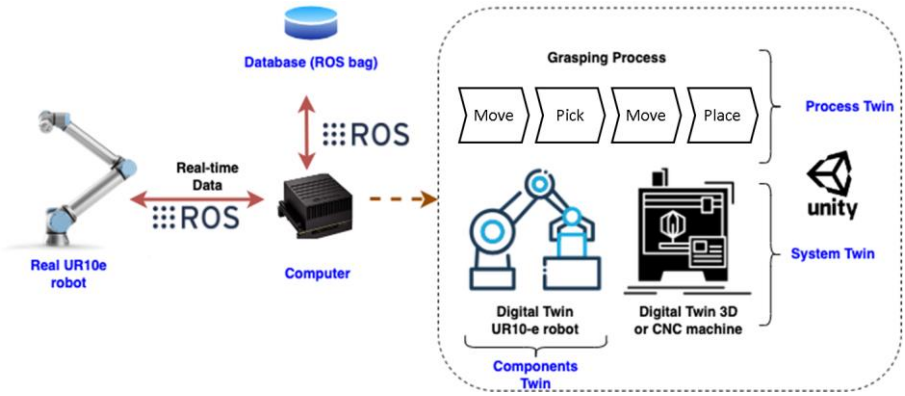

Figure 4. *Collaborative robot arm DT framework and industrial pick-and-place robot case study.*

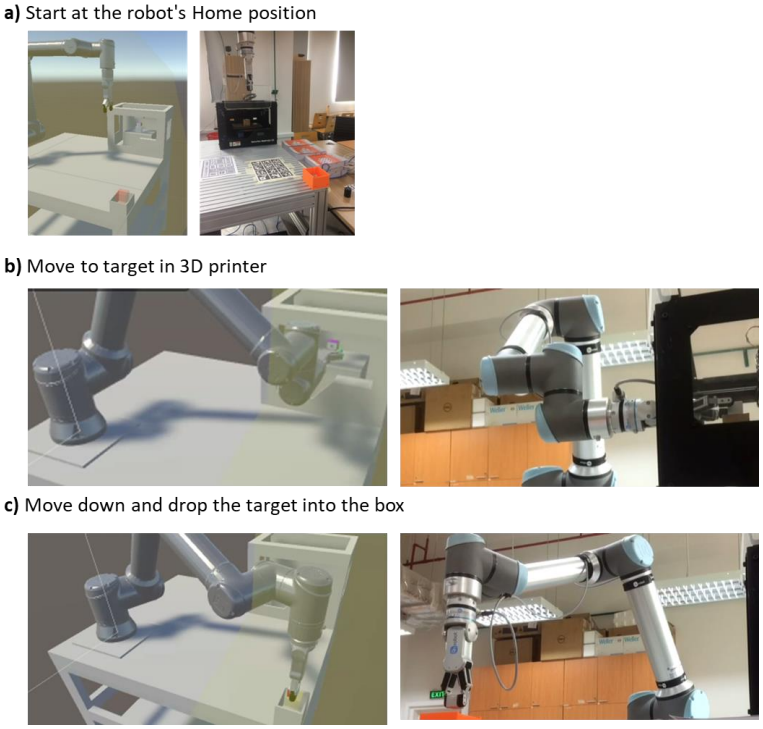

Figure 5. *DT in Unity3D (left) and real system (right) in the 3D printer pick-and-place case study a) Home position, b) Pick target, c) Drop target.*

3. Evaluation Results and Discussion

The effectiveness and applicability of the proposed framework are evaluated through the case study outlined in Section 2.3. Key evaluation parameters include performance, efficiency, and the scope of application. It should be noted that the UR10e has a built-in real-time controller running at 500 Hz which is considered high for most robotics applications, allowing it to execute motion smoothly and publish the joint_states topic in real time without relying on external commands. Additionally, since the Unity3D digital environment directly subscribes to the joint_states topic to update the virtual robot's joints and network is controlled and stable (e.g., a dedicated local wired connection), the communication time between the physical and digital models is considered fixed. Therefore, latency is not in the focus of this work.

The position discrepancy between the Tool Center Point (TCP) of the DT robot and the actual tool center of the real robot is summarized in Table 1. The TCP is a specified point defined relative to the

end of the robot arm, typically located at the tip of the attached tool, such as a gripper. The position of the TCP in the physical robot is obtained by using MoveIt's built-in pose retrieval function, get_current_pose, in the MoveGroup interface, which acquires real values through the robot' sensors. The purpose of the discrepancy calculation is to validate the synchronization capability between the physical robot and the DT robot. The joint values collected from the robot's sensors are calculated based on the physical robot's kinematic model. This information is considered a reliable reference, since the manufacturer specifies a precision of 0.05 mm, as mentioned earlier. The joint positions are then published through a communication interface, typically over Ethernet. In contrast, the TCP position of the DT robot is calculated using a kinematic model implemented on a computer (MoveIt). The synchronization can be considered successful when the TCP position discrepancy is within an acceptable threshold.

The position discrepancy results from both robot modeling inaccuracies and sensor noise during the synchronization between the DT and physical robot. The Root Mean Square Error (RMSE) quantifies the deviation between the DT and actual robot sensor readings, while the Mean Absolute Percentage Error (MAPE) measures model accuracy, lower MAPE values indicate a more accurate model. The R2 coefficient, used in the regression model, assesses the relationship between the dependent and independent variables. In the context of DT, a higher R2 value signifies a better fit to real-world data.

Table 1. *Compare TCP position between DT and UR10e using three indicators: standard deviation of residual (prediction error) RMSE, mean relative error MAPE, and coefficient of determination R2.*

	X position	Y position	Z position
RMSE	0.005	0.003	0.008
MAPE (%)	0.492	0.702	1.182
R2	0.997	0.999	0.992

Table 2 details the collaborative robot motion control algorithm for a typical industrial pick-and-place task. The default ROS ompl-RRTConnect algorithm explores random positions between the starting and ending points, which may result in suboptimal paths, sudden joint accelerations, or pathfinding failures. Specifically, the algorithm occasionally cannot establish a direct path between the Pre-Pick and Pre-Place positions (steps 05 and 07). To address this issue, an additional "return to Home position" step (step 06) is implemented, ensuring the seamless execution of the pick-and-place task. For fixed path scenarios, algorithms that eliminate random position exploration should be used to minimize the risk of collisions between the robot and the operator.

Table 2. *Key steps in the pick-and-place process in a case study.*

Step	Details
01	Starting at the robot's Home position, move the TCP to the front of the 3D printer
02	Pre-Pick - Move the TCP into the printer
03	Pre-Grasp - Lower the target and close the gripper
04	Move back to the Pre-Grasp position
05	Move back to the Pre-Pick position
06	Move back to the Home position
07	Pre-Place - Move to the top of the container
08	Place - Move down and drop the target into the container
09	Move back to the Pre-Place position
10	Move back to the Home position

JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Table 3 presents the processing times for five typical algorithms at each step of the case study, averaged over three executions. The pilz_industrial_motion_planner-LIN algorithm, based on linear interpolation, computes trajectory planning in approximately 0.1 seconds and is suitable for low-dimensional spaces. Conversely, algorithms from the OMPL motion planning suite, including ompl-RRTConnect, ompl-EST, ompl-RRTstar, and ompl-RRT, are more suited for complex, high-dimensional spaces. Among these, ompl-RRTstar generates the most optimal path but requires the longest computation time (~5 seconds).

Table 3. *The processing time of the algorithms (unit: seconds).*

Step	Algorithm				
	pilz_industrial_motion_planner-LIN	ompl-RRTConnect	ompl-EST	ompl-RRTstar	ompl-RRT
01	0.117	0.159	0.160	5.022	0.078
02	0.031	0.195	0.690	5.038	0.066
03	0.022	0.167	0.127	5.033	0.076
04	0.021	0.176	0.149	5.029	0.058
05	0.034	0.179	0.096	5.023	0.074
06	0.110	0.151	0.364	5.038	0.061
07	0.054	0.170	0.332	5.043	0.096
08	0.022	0.145	0.207	5.035	0.056
09	0.022	0.158	0.22	5.023	0.081
10	0.052	0.144	0.237	5.031	0.068

Table 4 shows the execution times for each stage of the task in the case study. With the robot speed set at 28% of the maximum allowed velocity, execution times for each step were approximately uniform (~18 seconds), as a result of the default ROS time interval setting being applied. These times can be adjusted by modifying the ROS time interval parameter. The consistency in execution times may also be attributed to the simple environment of the 3D printer pick-and-place task, where objects are fixed and there are minimal obstacles.

Table 4. *Real robot task execution time (speed: 28%, unit: seconds).*

Step	Algorithm				
	pilz_industrial_motion_planner-LIN	ompl-RRTConnect	ompl-EST	ompl-RRTstar	ompl-RRT
01	18.494	18.510	18.512	18.494	18.526
02	17.904	17.904	17.904	17.904	17.904
03	18.952	18.952	18.948	18.950	18.951
04	17.904	17.904	17.904	17.904	17.904
05	17.904	17.904	17.904	17.904	17.904
06	17.954	17.954	17.954	17.954	17.954
07	17.904	17.904	17.904	17.904	17.904
08	18.951	18.951	18.951	18.950	18.948
09	17.904	17.904	17.904	17.904	17.904
10	17.904	17.904	17.904	17.904	17.904

JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Figure 6 displays the robot's active or work area, determined by parameters such as the reach radius of the robot arm, as provided by the manufacturer. Initial performance evaluation demonstrates promising results with low RMSE, small MAPE, and high R2, indicating a strong fit to real-world data. However, errors due to sensor noise and modeling inaccuracies persist. Future improvements could involve implementing path-fixing algorithms to reduce collision risks and enhance overall operational efficiency.

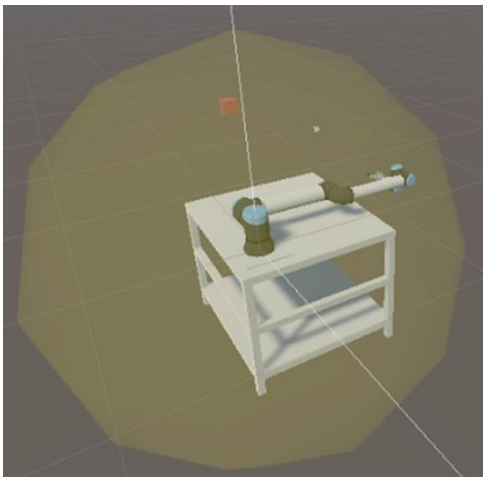

Figure 6. *The sphere in Unity3D shows the robot's active area.*

In addition to the pick-and-place scenario, a position performance testing is conducted in accordance with ISO 9283. The standard describes suitable measurement positions located in a plane placed inside a cube (cp. Figure 7-left) within the robot's working space. As the positions P1–P9 were specified in the Cartesian space, the MoveIt's built-in function for computing Cartesian paths is employed to plan the trajectory for the performance test. This function facilitates the computation of a series of waypoints that enable the end-effector to travel in segments of a straight line that correspond to the position-specified poses. The test also includes two different circles whose centers shall be P1. The large circle is inside the defined plane and as large as possible. Its diameter shall be 80% of the cube side length. The small circle diameter shall be 10% of the large circle diameter. Thus, the positions include P1–P9 and discretized points along the circles. A comparison of TCP position between DT and real robots is illustrated in Figure 7-right. Collected data indicates a quite good synchronization between the DT and real robots. Thus, the performance test results would seem to confirm the findings from the pick-and-place case study in which the trajectory is planned in the joint space, but future research is necessary to further validate the obtained outcomes for, e.g., other industrial case studies or settings.

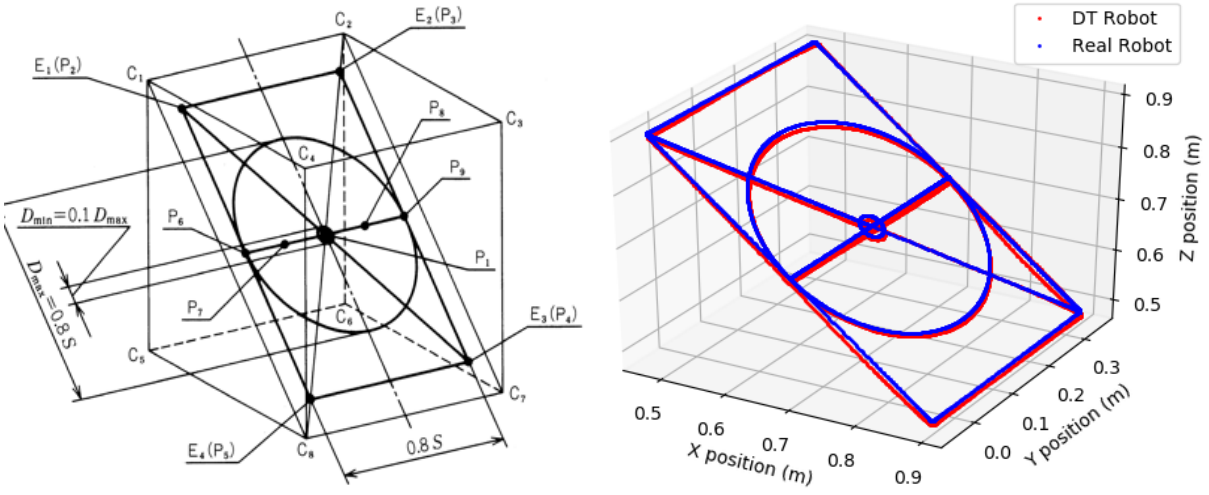

Figure 7. *Performance testing: ISO 9283 cube (left); TCP position compared between DT and real robot (right).*

In summary, the system was tested with operational conditions at a speed of 28% in two scenarios: a pick-and-place application and a recommended trajectory following ISO 9283 cube. Consequently, further system tests under different operational conditions could be developed in future work.

4. Conclusions

This paper explores the effectiveness and applicability of a digital framework in the context of an industrial pick-and-place robot case study. The small error between the DT and physical robot arm – as demonstrated by the evaluations using prediction error RMSE, mean relative error MAPE, and coefficient of determination R2 indicators – validates the effectiveness of the proposed framework for accurately mirroring real-world robotic behaviors. By employing real-world data integration and ISO 9283-compliant testing, a reliable virtual replica capable of predicting physical robot performance is established. In addition, the comparative analysis of some motion planning algorithms allows further optimization potential for the framework. Furthermore, these results confirm that DT technology offers a robust method for validating, optimizing, and enhancing the efficiency and reliability of industrial robotic applications before physical implementation.

Future research will focus on expanding the investigation of intelligent control algorithms, which includes enhancing existing algorithms and developing new ones to improve control efficiency. These algorithms will undergo simulation and verification using mathematical models and numerical simulations to evaluate their feasibility and stability prior to implementation on collaborative robot arms. Additional experiments with different equipment (e.g., CNC machines or injection molding machines) will be conducted to gather and analyze further data, enabling a more comprehensive assessment of the system's performance and reliability. This will ensure the results meet both scientific and industrial standards. The collected data will serve as a foundation for further algorithm refinements.

Acknowledgments

This research is funded by the Vietnam Ministry of Education and Training under grant number B2024-VGU-03.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] E. Coronado, S. Itadera, and I. G. Ramirez-Alpizar, “Integrating virtual, mixed, and augmented reality to human–robot interaction applications using game engines: A brief review of accessible software tools and frameworks,” *Appl. Sci.*, vol. 13, no. 3, p. 1292, 2023.
[2] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.
[3] M. Ostanin, S. Mikhel, A. Evlampiev, V. Skvortsova, and A. Klimchik, “Human-robot interaction for robotic manipulator programming in mixed reality,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, 2020, pp. 2805–2811.
[4] M. Q. Tram, J. M. Cloud, and W. J. Beksi, “Intuitive robot integration via virtual reality workspaces,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, London, U.K., 2023, pp. 11654–11660, doi: 10.1109/ICRA48891.2023.10160699.
[5] Q. Wang, W. Jiao, P. Wang, and Y. Zhang, “Digital twin for human-robot interactive welding and welder behavior analysis,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 334–343, Feb. 2021, doi: 10.1109/JAS.2020.1003518.
[6] G. Pisanelli, M. Tymczuk, J. A. Douthwaite, J. M. Aitken, and J. Law, “ROSIE: A ROS adapter for a modular digital twinning framework,” in *Proc. IEEE Int. Conf. Robot. Human Interact. Commun. (RO-MAN)*, Napoli, Italy, 2022, pp. 1297–1304.
[7] E. Sita, C. M. Horváth, T. Thomessen, P. Korondi, and A. G. Pipe, “ROS-Unity3D based system for monitoring of an industrial robotic process,” in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Taipei, Taiwan, 2017, pp. 1047–1052, doi: 10.1109/SII.2017.8279361.
[8] S. H. Lee *et al.*, “Robotic manipulation system design and control for non-contact remote diagnosis in otolaryngology: Digital twin approach,” *IEEE Access*, vol. 11, pp. 28735–28750, 2023.
[9] S. Sai, M. Prasad, A. Garg, and V. Chamola, “Synergizing digital twins and metaverse for consumer health: A case study approach,” *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2137–2144, Feb. 2024.
[10] D. L. Pieper, “The kinematics of manipulators under computer control,” Stanford Artificial Intelligence Project, Memo, Stanford Univ., Stanford, CA, USA, 1968.
[11] Q. Liu, D. Yang, W. Hao, and Y. Wei, “Research on kinematic modeling and analysis methods of UR robot,” in *Proc. Int. Technol. Eng. Conf. (ITOEC)*, 2018, doi: 10.1109/ITOEC.2018.8740681.
[12] R. Zhao *et al.*, “Inverse kinematic solution of 6R robot manipulators based on screw theory and the Paden–Kahan subproblem,” *Int. J. Adv. Robot. Syst.*, vol. 15, Dec. 2018, Art. no. 1729881418818297, doi: 10.1177/1729881418818297.

[13] B. Sheng, W. Meng, C. Deng, and S. Xie, "Model based kinematic & dynamic simulation of 6-DOF upper-limb rehabilitation robot," in *Proc. Asia-Pacific Conf. Intell. Robot Syst. (ACIRS)*, 2016, pp. 21–25, doi: 10.1109/ACIRS.2016.7556181.
[14] J. Villalobos, I. Y. Sanchez, and F. Martell, "Statistical comparison of Denavit-Hartenberg based inverse kinematic solutions of the UR5 robotic manipulator," in *Proc. Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, 2021, pp. 1–6, doi: 10.1109/ICECCME52200.2021.9591104.
[15] L. T. Schreiber and C. Gosselin, "Determination of the inverse kinematics branches of solution based on joint coordinates for UR-like serial robot architecture," *J. Mech. Robot.*, vol. 14, pp. 1–13, Oct. 2021, doi: 10.1115/1.4052805.
[16] H. Wang *et al.*, "Research on the relationship between classic Denavit-Hartenberg and modified Denavit-Hartenberg," in *Proc. 7th Int. Symp. Comput. Intell. Des. (ISCID)*, vol. 2, pp. 26–29, Apr. 2015, doi: 10.1109/ISCID.2014.56.
[17] K. P. Hawkins, "Analytic inverse kinematics for the Universal Robots UR-5/UR-10 arms," 2013. [Online]. Available: [repository or publisher not specified].
[18] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Upper Saddle River, NJ, USA: Pearson/Prentice Hall, 2005.
[19] F. Pires, V. Melo, J. Almeida, and P. Leitão, "Digital twin experiments focusing virtualisation, connectivity and real-time monitoring," 2020, doi: 10.1109/ICPS48405.2020.9274739.
[20] *Planners | MoveIt*, Moveit.ai, 2024. [Online]. Available: https://moveit.ai/documentation/planners/

Quang-Huan Dong received a PhD degree from the TUM School of Engineering and Design, Technical University of Munich, Germany in 2023. He is currently a postdoctoral researcher at the Faculty of Engineering, Vietnamese-German University, Vietnam. His research focuses on factory automation and software engineering.
Email: huan.dq@vgu.edu.vn. ORCID: https://orcid.org/0000-0002-3085-3464

Tuan-Khanh Nguyen received the B.S. and M.S. degrees in electronic telecommunication from Can Tho University, Can Tho, Vietnam, in 2010 and Ho Chi Minh University of Technology, Ho Chi Minh City, Vietnam, in 2014, respectively. He got the PhD degree in Electronic and Computer Engineering at National Taiwan University of Science and Technology in 2023. He is currently pursuing a postdoctoral degree at the Vietnamese-German University. His research interests include semiconductors, radio-frequency biomedical sensors, noncontact vital-sign radar sensors, and microwave circuits and modules.
Email: khanh.nt@vgu.edu.vn. ORCID: https://orcid.org/0000-0002-5162-4417

Chi-Cuong Tran received the B.S. and M.S. degrees in Automation and Control Engineering from Can Tho University, Vietnam, in 2015 and 2017, respectively. He received his Ph.D. degree in Mechanical Engineering from National Taiwan University of Science and Technology (NTUST), in 2023. He is currently working as a postdoctoral researcher at National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include intelligent robotics, intelligent automation, computer vision, and machine learning.
Email: tccuong09@gmail.com. ORCID: https://orcid.org/0009-0004-7698-0970

The-Thinh Pham received his Bachelor degree in Mechatronic Engineering from Can Tho University of Technology, Vietnam in 2019. He received his Master degree in Mechanical Engineering from National Taiwan University of Science and Technology, Taiwan in 2023. He is currently pursuing a Ph.D. degree with Mechanical Engineering, National Taiwan University of Science and Technology. His current research interests include computer vision, deep learning and robotics.
Email: ptthinh@ctuet.edu.vn. ORCID: https://orcid.org/0009-0009-4426-0457

Duy-Tan Do received his B.S. degree from Ho Chi Minh City University of Technology (HCMUT), Vietnam, and M.S. degree from Kumoh National Institute of Technology, Korea, in 2010 and 2013, respectively. He received his Ph.D. degree from Autonomous University of Barcelona, Spain, in 2019. He is currently an Assistant Professor/Lecturer at the Department of Computer and Communication Engineering, Ho Chi Minh City University of Technology and Education (HCMUTE) in Vietnam. His main research interests include real-time optimization for resource allocation in wireless networks and coding applications for wireless communications.
Email: tandd@hcmute.edu.vn. ORCID: https://orcid.org/0000-0003-4570-0441

Hoang-Vinh-Khang Nguyen received his B.S. degree in Electrical Engineering and Information Technology (EEIT, now Electrical and Computer Engineering) from the Vietnamese-German University (VGU) in 2015 and his M.S. degree in Mechatronics and Sensor Systems Technology (MST) from VGU in 2018. As a recipient of the DAAD scholarship, he spent one year at Karlsruhe University of Applied Sciences, Germany, where he conducted research for his master's thesis. His research interests include biomedical signal processing for rehabilitation applications and exoskeleton robot control.
Email: khang.nhv@vgu.edu.vn. ORCID: https://orcid.org/0000-0003-0256-2237

Quang-Chien Nguyen received the B.S. degree in Automation and Control Engineering from Ho Chi Minh City University of Technology and Education (HCMUTE) in 2023. He is currently pursuing a Master's degree in Automation and Control Engineering at Ho Chi Minh City University of Technology and Education. His research interests include robotics, mobile robot, intelligent control and motion control.
Email: 2431101@student.hcmute.edu.vn. ORCID: https://orcid.org/0009-0008-1332-7746