Advances in Technology Innovation, vol. 8, no. 4, 2023, pp. 267-277

Virtual Modeling of an Industrial Robotic Arm for Energy Consumption Estimation

Jin-Siang Shaw[1,*], Yi-Hua Huang[2]

[1]Department of Mechanical Engineering, National Taipei University of Technology, Taiwan, ROC

[2]Institute of Mechatronic Engineering, National Taipei University of Technology, Taiwan, ROC

Received 13 April 2023; received in revised form 03 July 2023; accepted 04 July 2023

DOI: https://doi.org/10.46604/aiti.2023.11957

Abstract

This study aims to improve the traditional control methods of industrial robotic arms for path planning in line with efforts to conserve energy and reduce carbon emissions. The digital twin of a six-axis industrial robotic arm with an energy consumption model is innovatively designed. By directly dragging the end effector of a digital twin model, the robotic arm can be controlled for path planning, allowing path tuning to be easily made. In addition, the dynamic equation of the industrial robotic arm is derived, and the energy consumption of the corresponding path can be estimated. Four cases are designed to test the validity of the digital twin. Experimental results show that the physical robotic arm follows its digital twin model with the corresponding energy consumption computed. The estimated energy consumptions agree quite well with each designed case scenario.

Keywords: digital twin, robotic arm, Unity, Euler-Lagrange equation, energy consumption

1. Introduction

In the advanced 21st century, technologies, such as the internet of things (IoT), cloud computing, big data analysis, and intelligent robot systems, will cause changes in the global manufacturing industry and lead to an era of intelligent manufacturing in Industry 4.0. With the advent of Industry 4.0, there has been remarkable growth in the synergistic integration of existing and emerging technologies, for example, the digital twin (DT) in robotics [1], the predictive maintenance based on DT [2], and the DT-driven machining [3]. Moreover, Gartner predicted that more than 20 billion US of equipment (mostly from the manufacturing industry) would be applied to IoT technology [4].

In Industry 4.0, engineers need to establish IoT and Cyber-Physical Systems for coordinating smart factories, with digital twins playing a crucial role [5]. Through computer-integrated manufacturing, digital information technology, data analysis, and data-driven services, appropriate and practical production integration can improve the efficiency, productivity, and flexibility of the production process [6-8]. A virtual model in a DT can use this information to monitor, optimize, and predict real objects by accessing their physical properties, behaviors, and specifications [9]. Therefore, the DT concept is considered a promising and innovative research field. However, owing to the complexity of constructing a digital model corresponding to physical space in a virtual space, its application in the field of industrial manufacturing is rarely observed.

Although Digital twinning has been widely recognized as a futuristic and innovative research field, its applications in industrial manufacturing are uncommon due to the complexity of developing digital models in a virtual space that corresponds to and replicates a physical space. Currently, most of the research papers and applications related to robotic arms and digital twinning are focused on establishing teaching and demonstration scenarios [10-12].

* Corresponding author. E-mail address: jshaw@ntut.edu.tw

English language proofreader: Chih-Wei Chang

The Windows operating system is widely used worldwide, as shown in Fig. 1 [13]. The most significant difference between ROS and Windows is the complexity of ROS operations. While most software today is designed based on the Windows operating system architecture. Its scalability is far more convenient than that of ROS, and the resources are more abundant. Since most people have been in contact with computers set up under the Windows operating system architecture since childhood, they are familiar with the operating interface and use computers with much greater proficiency than those under ROS architecture. As a result, there is no need to invest additional time learning the two operating systems, Ubuntu and ROS, and only one operating system can be used to complete the robot research.

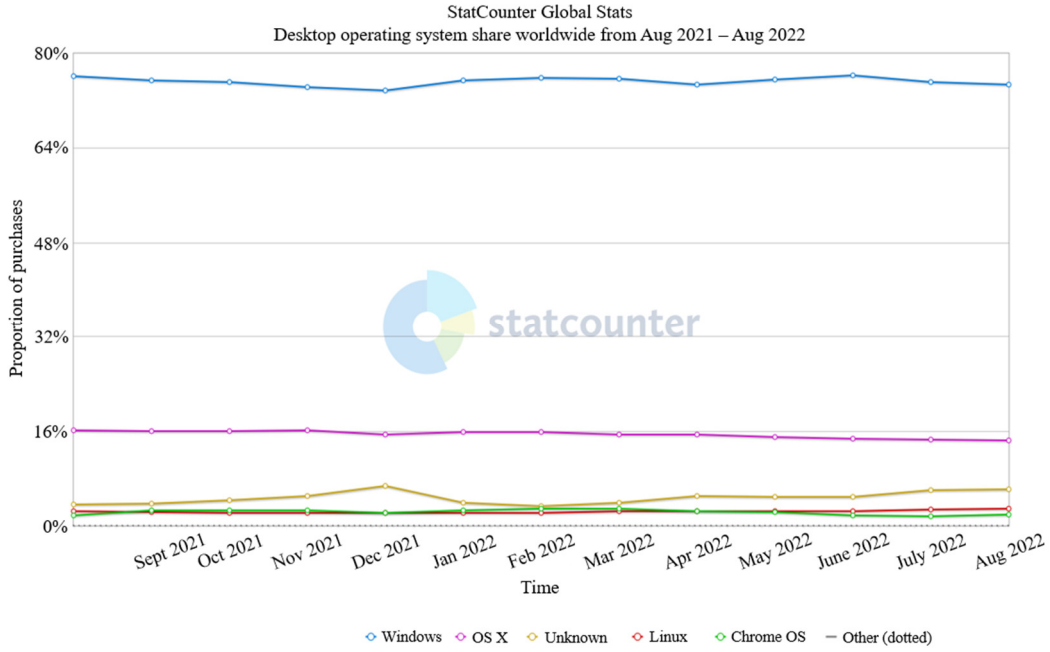


Fig. 1 Global desktop operating system market share [13]

According to the World Robotics 2022 Industrial Robots report released by the International Federation of Robotics, there are currently 3.5 million industrial robots used in global factories, and approximately 510,000 units were delivered globally in 2021. The applications include handling, welding, assembly, cleaning, distribution, etc. [14]. However, many industries have not invested in robot-related technologies, and the main problem lies in the limitations of traditional robot control methods. Considerable knowledge and technological intervention are required to make it operate normally, which seriously constrains the development of the robotics industry. Furthermore, revenue from industrial robots continues to grow globally, and there is a huge energy cost to drive these automated systems with multiple robots operating [15].

Typical energy consumption modeling methods for robots rely on kinematic and dynamic characteristics to calculate energy consumption. Heredia et al. [16] established mathematical models of robotic arms for energy consumption calculation. These models were parameterized and trained through collected experimental data to evaluate their accuracy. Another approach involved using a trained multilayer perceptron (MLP) or ResNet model to predict the relevant energy consumption of the robotic arm [17-18]. However, the aforementioned data-driven modeling required a large amount of training data collected from various sensors. In this study, the corresponding energy consumption can be computed solely by the derived energy consumption model with the provided angular speeds of each joint from the DT, and no need to collect training data.

This study aims to create a DT and energy consumption model in the Windows framework, in which the physical robotic arm moves by dragging the end effector of the virtual robotic arm. In other words, the physical robotic arm can be collaborative in this manner, and the energy consumption of the corresponding path can be computed. With the energy consumption evaluated, the planning path with the lowest energy consumption value can be selected for the physical robotic arm to operate, resulting in significant energy savings.

2. Methodology

Although the simulation of a robotic arm in ROS using Moveit, Gazebo, and Rviz is quite common, no cases have been found that utilize the ROS framework in constructing a digital twin model capable of estimating energy consumption for robotic arms. In fact, to the best of the authors' knowledge, no research has been reported on using the DT model that can predict the corresponding energy consumption for robotic arms under either the Windows or ROS system.

Since Windows is the most popular operating system in the world, the development of a DT of an industrial robotic arm under the Windows architecture is planned. The system architecture is illustrated in Fig. 2. Model preprocessing of the industrial robotic arm is primarily performed because the initial industrial robotic arm model is not similar to that under the ROS architecture. Instead, similar to the unified robot description format (URDF) file, the URDF file includes all relevant information regarding the industrial robotic arm, such as the pivot point, rotation range, and hue of each joint. Therefore, it is necessary to preprocess the original model using computer graphics software.

After processing, the model is exported to Unity, a software used as a DT. To allow data transmission between the industrial robotic arm and DT software, it is necessary to establish network communication between the two, enabling the real-time transfer of important data required to simulate the state of the physical robot arm in the DT, and vice versa. This study used the "Socket" developed in the TCP/IP environment to transfer data between the two. This approach differs from previous control methods of transmitting the world coordinate position of an industrial robot arm and the current rotation angle of each joint. This decision allows users to understand the movement patterns of an industrial robot arm better when it is being controlled for the first time.

In addition, a human-machine monitoring interface is established for allowing users quickly understand the system status and control the industrial robotic arm in real-time. To meet the requirements of remote connection control, this interface integrates functions, such as robot arm communication and the real-time angles of each joint. Finally, the dynamic equation of the six-axis industrial robotic arm is calculated using the Euler-Lagrange equation. Hence, the required torque of each motor can be estimated for any specific path.

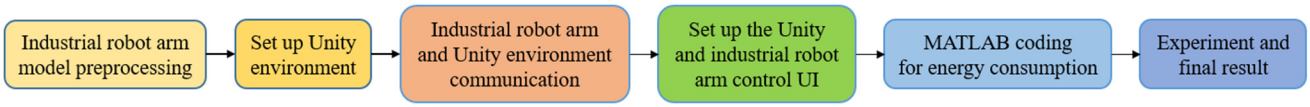


Fig. 2 System architecture diagram

2.1. *Building digital twins – Unity*

In this study, the Unity software was used to build the DT, which is a cross-platform 2D/3D game engine developed by Unity Technologies, a game software development company. Generally, Unity supports the following aspects. First, Unity can be used to develop stand-alone games for Windows, MacOS, Linux, and other operating systems. Second, it's also available for games on mobile devices such as iOS and Android. In addition to game development, Unity is widely used in the production of real-time 3D animations, architectural visualizations, and other types of projects. Lastly, it supports the PhysX physics engine and particle system and provides network multiplayer connection functionality without requiring the user to learn a complex programming language. Given this advantage, it meets the various requirements of game production. Thus, the launch of Unity has lowered the threshold for game development and made game creation for individuals and small teams realistic.

2.2. *Industrial robotic arm model preprocessing*

Before exporting the model to the Unity game engine, the industrial robotic arm model must undergo preprocessing. Meanwhile, the relationship between the joints and connecting links must be defined. The Staubli TX60L industrial robotic

arm used in this study has a CAD model in the standard triangulated language (STL) format, which is the output from the STP format provided by the manufacturer Staubli through SolidWorks. This format can be used in Blender computer graphics software. The parent-child relationship must be defined in Blender computer graphics software. Therefore, by defining the parent-child relationship of each link of the industrial robotic arm, the concept of this relationship can be controlled effectively, as shown in Fig. 3. For example, when turning "Axis-1," all the links that have a parent-child relationship will follow because "Axis-1" is defined as the parent of "Axis-2" to "Axis-6." It does not affect the base, so the base will not move. However, if the base is moved, the entire model will move along with it because the base is the parent from "Axis-1" to "Axis-6," and it indicates the relative relationship of the six axes (Axis-1, Axis-2, Axis-3, Axis-4, Axis-5, and Axis-6) as shown in Fig. 4 [19].

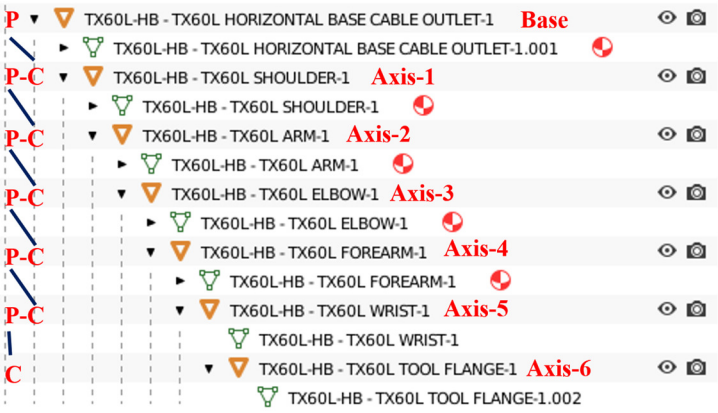


Fig. 3 Parent-child relationship

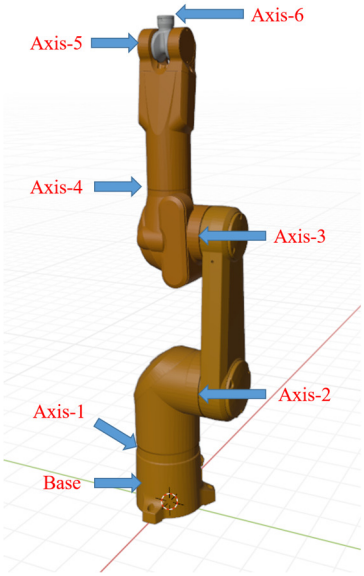


Fig. 4 3D model and the relationship of each axis

In addition to the above settings, the pivotal point of each link must be specified. The pivot point, also known as the joint center point, defines the motion of the links [20]. The parent-child relationship specifies the motion between the links and the pivot point specifies the motion of the joints, as shown in Fig. 5.

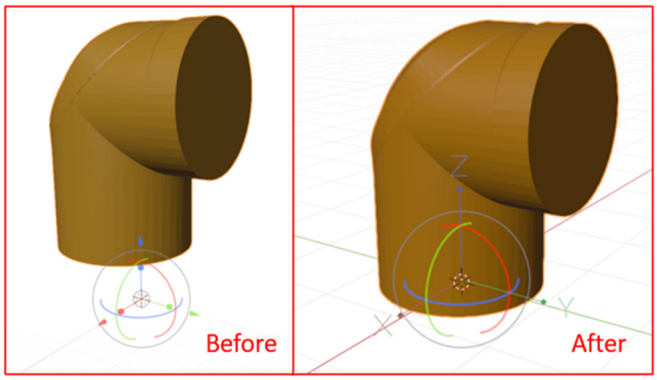


Fig. 5 Axis-1 pivot point setup

2.3. *Kinematics of the industrial robotic arm*

There are several methods to achieve the movement of a robotic arm by dragging the end effector in a virtual environment. One of the most commonly used methods is the cyclic coordinate descent (CCD) algorithm [21-22]. A popular approach for implementation is to use Bio IK [23]. Bio IK is the solution of inverse kinematics problems by utilizing a CCD algorithm for calculations. This study used the CCD algorithm for solving inverse kinematics problems and the traditional Denavit-Hartenberg (DH) parameter method for solving forward kinematics. By using the fast and less computationally intensive CCD algorithm and traditional DH parameter method, this study aimed to solve forward and inverse kinematics problems.

The CCD algorithm is currently one of the simplest and most popular methods for solving inverse kinematics, and it has been widely used in the computer game industry. The CCD algorithm calculates from one joint to another joint angle such that the endpoint is as close as possible to the target. After each iteration update, the algorithm measures the distance between the endpoint and the target to ensure that the endpoint is far enough from the target or already required to stop computing. Additionally, to avoid infinite recursive loops caused by unreachable and conflicting target points, the algorithm must be limited by setting a maximum number of iterations.

Essentially, the base of an inverse kinematic-driven mechanism is immovable, and the axis and angle are calculated using the formula, respectively, as shown in Fig. 6.

$$\theta = \cos^{-1}\left(\frac{P_e - P_c}{\|P_e - P_c\|} \bullet \frac{P_t - P_c}{\|P_t - P_c\|}\right) \tag{1}$$

$$\vec{r} = \frac{P_e - P_c}{\|P_e - P_c\|} \times \frac{P_t - P_c}{\|P_t - P_c\|} \tag{2}$$

This method is also applicable to 2D and 3D coordinates by aligning a joint (P_c) with the endpoint (P_e) and target, and the endpoint is finally approached by reciprocating the above steps towards the target.

To solve the problem of forward kinematics [24], this study used the DH parameter method. The DH parameter method designs a coordinate system for each connecting rod according to set rules, and then this coordinate system can be used to describe the transformation relationship from one connecting rod coordinate system to the adjacent connecting rod coordinate system. The transformation of the adjacent coordinate system is decomposed into several steps with only one parameter respectively. The combination of the corresponding transformations completes the transformation of the adjacent coordinate systems.

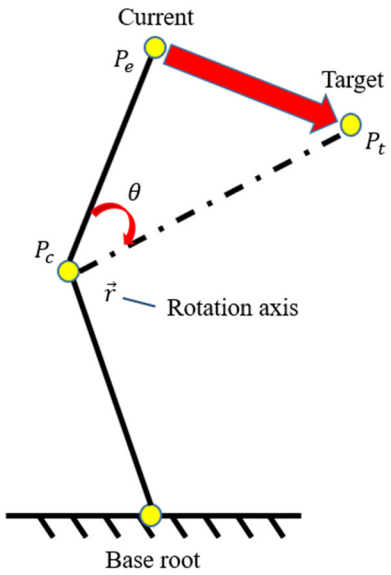


Fig. 6 CCD algorithm flow

2.4. Digital twin user interface for the industrial robotic arm

Two scenarios were considered in this study. The first scenario used forward kinematics to control the virtual robotic arm and synchronously moved the physical robotic arm. When the physical robotic arm was controlled by the robotic arm user interface, the virtual robotic arm moved synchronously. As shown in Fig. 7, after pressing the Start Received button (button 1), the arm begins to receive the six joint information read from the cs8c controller. This information was used to move the virtual robotic arm into a physical robotic arm. Each joint of the virtual robotic arm can be rotated by controlling the plus/minus

buttons (button 2) of each joint. The rotation angle information was transmitted to the physical robotic arm which rotated accordingly. After achieving the goal of mutual control between the two parties, press the Quit Game button (button 3) to end the currently executing scene.

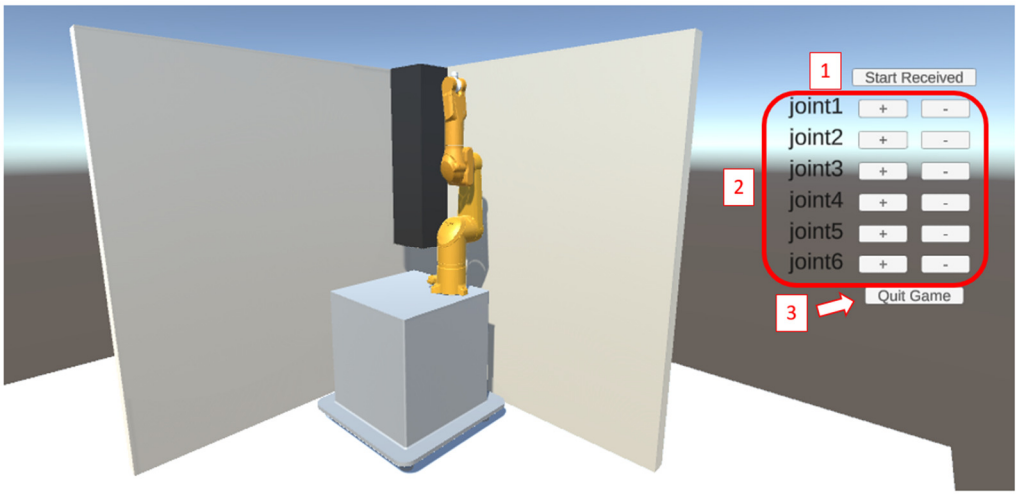


Fig. 7 Unity digital twin scene 1

The second scene primarily involves moving the virtual robotic arm through inverse kinematics, as shown in Fig. 8. In this scene, a red sphere was designed as the target object (target). The end effector moved with the target object. The movement of the virtual robotic arm was controlled by dragging the target object. When it moved to a position, the dragging was stopped, and the angle of each joint was calculated using the CCD script designed in Unity. This angle information was sent to the physical robot arm such that it could move like a virtual robotic arm. The goal was to control the physical robot arm by dragging the DT end effector. The point correction of a traditional industrial robotic arm was significantly reduced by dragging the DT end effector to the designated target point for the corresponding work settings, which makes it more flexible and can be used in complex environments. Such a DT with a user interface can also be used to establish a fully realistic workplace and perform path planning and other operations on the robotic arm through a remote control, thereby enabling task completion without entering the factory.

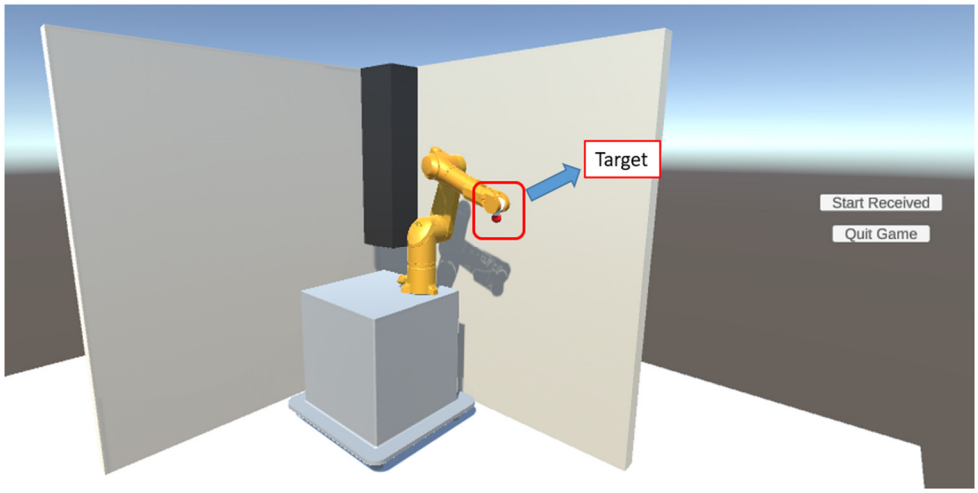


Fig. 8 Unity digital twin scene 2

2.5. *MATLAB energy consumption calculation*

With the gradual advancement of science and technology and the development of digital avatar-related technologies, factories are moving towards smart manufacturing, and smart factories are becoming a major trend. Through these technologies, factory personnel can monitor the internal conditions of a factory in real-time using computers and mobile phones. With the

rise of international oil prices and the improvement of green energy-related technologies, Taiwan officially announced the "Taiwan 2050 Net Zero Emissions Path and Strategy General Description" in March 2022 [25]. Energy conversion is a critical issue for achieving the goal of net-zero emissions. In the current situation, where green energy and other related technologies are not yet mature, effective management of equipment energy consumption in factories is crucial. This study aimed to address this issue using the DT of the robot arm in Unity and inputting the simulated robotic arm path into MATLAB for energy consumption estimation. Therefore, a dynamic model of a robot arm, which is a complicated and nonlinear system, is required.

This study used a dynamic model of the robotic arm to calculate the torque of each joint motor through the moving path of the robotic arm DT environment established above and computed the total energy required for this moving path. The purpose was to allow users to use Unity to first perform path planning with the DT of the robotic arm and then obtain the total energy consumption of the path through MATLAB calculations. This method enabled users to carry out various path planning through the digital avatar of the robotic arm and the realistic environment of the actual scene. By analyzing different paths, users can select and execute the path that consumes the least amount of energy. After obtaining the path that consumed the lowest energy, the physical robotic arm could operate according to the path. In robotics, the Euler-Lagrange equation can be used to calculate torque [26].

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i}-\frac{\partial L}{\partial q_i}=\tau_i,\ i=1,\ldots,m \tag{3}$$

The Lagrangian, *L*, is defined as the kinetic energy minus the potential energy.

$$L=K-P \tag{4}$$

The kinetic and potential energy equations of the n-link robotic arm are expressed in:

$$K=\frac{1}{2}\dot{q}^T\sum_{i=1}^{n}\left[m_i J_{vi}(q)^T J_{vi}(q)+J_{wi}(q)^T R_i(q) I_i R_i(q)^T J_{wi}(q)\right]\dot{q} \tag{5}$$

$$P=\sum_{i=1}^{n}P_i=\sum_{i=1}^{n}g^T r_{ci} m_i \tag{6}$$

where $\dot{q}$ and q denote vectors of joint angular velocity and position, respectively, n denotes the total number of joints, τ_i denotes the torque of the ith motor, m_i denotes the mass of the ith link, $J_{v_i}(q)$ and $J_{w_i}(q)$ represent the Jacobians of the ith link speed and angular velocity, respectively, $R_i(q)$ denotes the orientation matrix of the ith link, I_i denotes the inertia tensor of the ith link, g denotes the gravity vector, and finally r_{ci} denotes the mass center vector of the ith link.

After obtaining the equations for kinetic and potential energies, Eqs. (5) and (6) are substituted into Eq. (4), and Eq. (4) is substituted into Eq. (3), which can be used to obtain the dynamic equation of the robotic arm in a vector-matrix form.

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau \tag{7}$$

where $\ddot{q}$ represents the vector of joint angular acceleration. M represents the n × n inertia matrix, C represents the n × n matrix of the Coriolis force and centrifugal force, and G represents the gravity of the n × 1 vector.

Parameters in M, C, and G in Eq. (7) will be identified by the CAD model provided by the manufacturer Staubli, such as the geometry and property of each link, the link's moment of inertia, etc. The joint variable q can be obtained by applying the CCD algorithm for the inverse problem. Therefore, the torque of each link executing the path $\ddot{q}$, $\dot{q}$, and q can be computed in Eq. (7) by summing the three terms on the left-hand side. The total power required for the robotic arm is then estimated by

$$P=\frac{\sum_{i=1}^{n}\tau_i\dot{\theta}_i}{\eta_i} \tag{8}$$

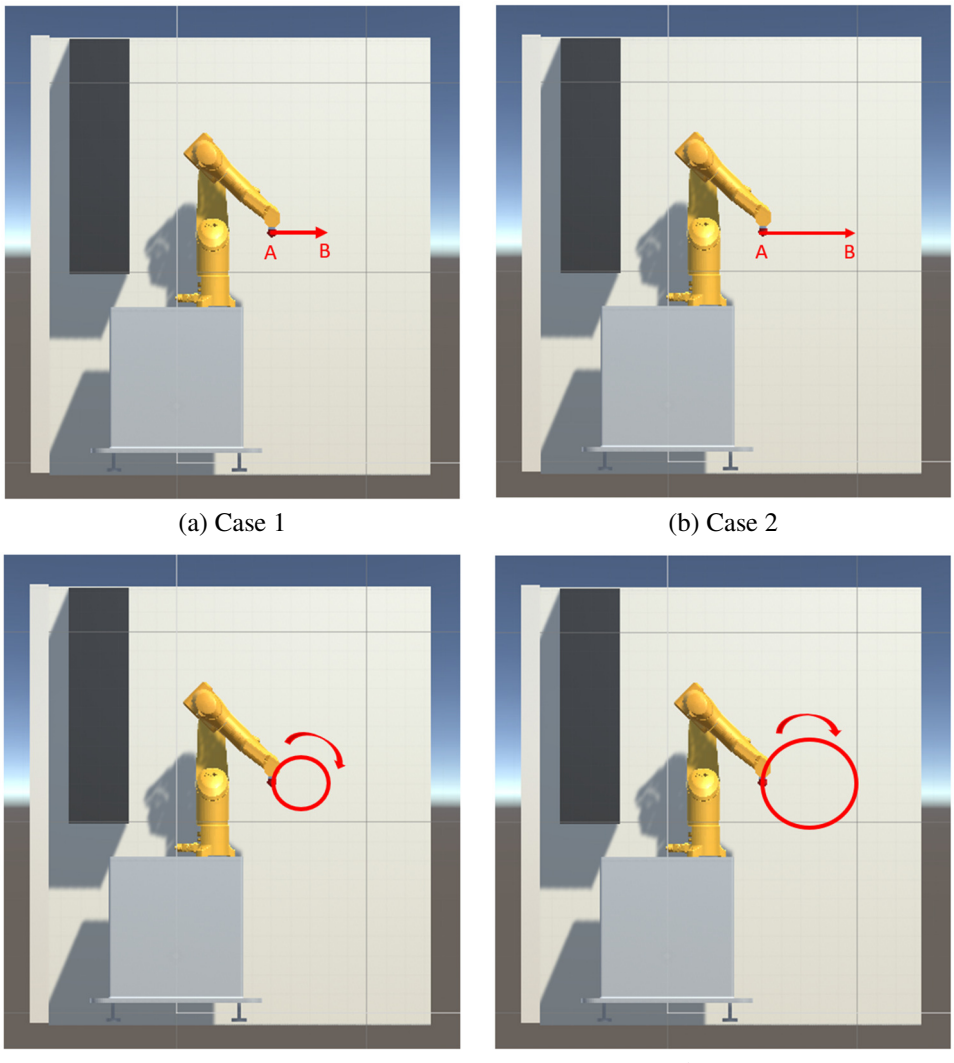
where η_i is the transfer efficiency of the ith motor. The resulting energy consumption of the robotic arm for this path is determined by integrating the power over time,

$$W = \int_{t_0}^{t_f} P dt \tag{9}$$

where t_f is the final time; t_0 is the initial time. It is noted that the dynamic equation in Eq. (7), an often-seen model for a robotic arm in the literature, does not consider friction between the robots' joints. This is because friction in robot joints including Coulomb friction and viscous friction is difficult to exactly identify the power it consumes constitutes only a small part of the total power. Here η_i is used to represent the transfer efficiency of the ith motor to reflect this friction power loss. By performing these calculations in Eqs. (7)-(9), it can determine the energy consumption required for different paths and hence select the path with the lowest energy consumption to perform the work.

3. Exmental and Result

This study moved the end effector in four different scenarios, as shown in Fig. 9: Case 1 involved moving in a straight line, Case 2 involved moving twice the distance of Case 1, Case 3 involved moving in a small circle, and Case 4 involved moving in a large circle. These scenarios allowed us to compare the energy consumption of Cases 1 and 2, where the moving path of Case 2 was twice that of Case 1, resulting in higher energy consumption. Similarly, the energy consumption in Case 3 was expected to be lower than that in Case 4.



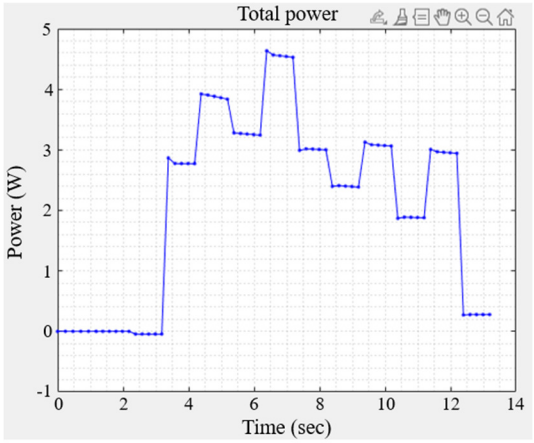
(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

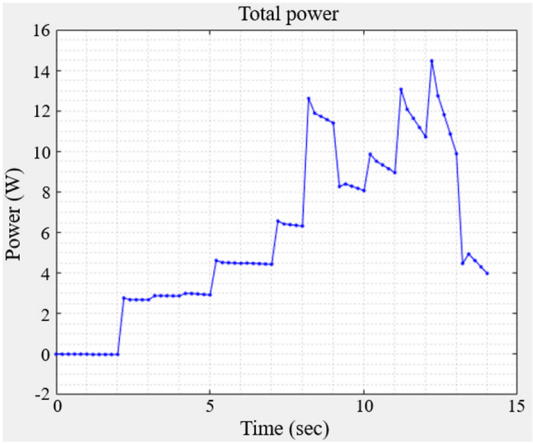
Fig. 9 Different cases of the simulation for energy consumption

In Case 1, the distance between A and B was 0.5 m. In Case 2, it was 1.0 m. In the case of circular motion, the large circle had a radius of 0.5 m, and the small circle had a radius of 0.25 m. The movement speed was set to 4 cm/sec for Case 1, where Case 2 had twice the speed of Case 1. The movement speed was set to 11 cm/sec for Case 3, where Case 4 had twice the speed of Case 3. The reason for choosing low speed in experiments was to facilitate the integration of the virtual and physical environments, which made it easier to control the robotic arm and avoid collisions and other factors.

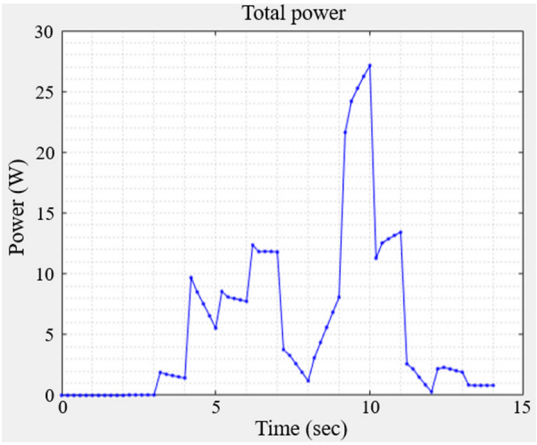
As shown in Fig. 10, the total power diagram was calculated using Eqs. (7) and (8), while Eq. (9) was used to compute the resulting energy consumption. Table 1 listed the final energy consumption for each case, which agreed with the predictions. Specifically, when the robotic arm's travel distance and travel speed were doubled, the corresponding energy consumption was also more than doubled (nonlinear), as seen in Case 2 over Case 1 and Case 4 over Case 3. Moreover, the ratio of energy consumption of Case 2 over Case 1 (2.9) was slightly larger than that of Case 4 over Case 3 (2.4). This can be attributed to the fact that a straight-line trajectory (Cases 1 and 2) requires more joints to execute compared to a circular trajectory (Cases 3 and 4) for an articulated robotic arm. Additionally, in Eq. (9), it was evident that the energy consumption was highly nonlinear concerning the operational conditions. To verify and validate the accuracy of the energy consumption model, appropriate sensors such as torque and angular speed sensors can be installed on the robotic arm to compute the true energy consumption values. However, it remains a task for future work.



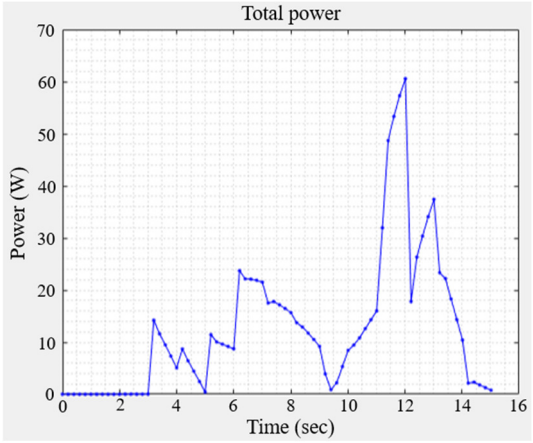
(a) Case 1



(b) Case 2



(c) Case 3



(d) Case 4

Fig. 10 Power response for each case

Table 1 Energy consumption of each case

Case	Energy consumption (J)
1	38
2	91
3	95
4	195

4. Conclusions and Future Work

The research utilizes the Unity game engine and the robotic principle of the Euler-Lagrange equation to establish a digital twin and energy consumption model for an industrial robot arm. The final energy performance is described through four case studies. It was observed that the motion length also influenced the calculated energy expenditure accordingly. By directly manipulating the end effector of the DT model, the industrial arm can be easily controlled remotely for task scheduling without the need to physically enter the factory, introducing a new form of control. Additionally, the digital twins and energy consumption models have demonstrated a reasonable level of accuracy.

In future studies, DT models of robotic arms capable of handling various loads in real-world applications will be considered. To enhance the accuracy of the energy consumption model, installing torque, comparing the results with the model's estimated energy consumption, and the angular velocity sensors on the physical industrial robotic arm to measure its energy consumption are required.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] A. Mazumder, M. F. Sahed, Z. Tasneem, P. Das, F. R. Badal, M. F. Ali, et al., "Towards Next Generation Digital Twin in Robotics: Trends, Scopes, Challenges, and Future," Heliyon, vol. 9, no. 2, article no. e13359, February 2023.

[2] D. Zhong, Z. Xia, Y. Zhu, and J. Duan, "Overview of Predictive Maintenance Based on Digital Twin Technology," Heliyon, vol. 9, no. 4, article no. e14534, April 2023.

[3] S. Liu, J. Bao, and P. Zheng, "A Review of Digital Twin-Driven Machining: From Digitization to Intellectualization," Journal of Manufacturing Systems, vol. 67, pp. 361-378, April 2023.

[4] Y. Cui, S. Kara, and K. C. Chan, "Manufacturing Big Data Ecosystem: A Systematic Literature Review," Robotics and Computer-Integrated Manufacturing, vol. 62, article no. 101861, April 2020.

[5] M. A. M. S. Lemstra and M. A. de Mesquita, "Industry 4.0: A Tertiary Literature Review," Technological Forecasting and Social Change, vol. 186, no. part B, article no. 122204, January 2023.

[6] J. Cheng, H. Zhang, F. Tao, and C. F. Juang, "DT-II: Digital Twin Enhanced Industrial Internet Reference Framework towards Smart Manufacturing," Robotics and Computer-Integrated Manufacturing, vol. 62, article no. 101881, April 2020.

[7] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of Digital Twin about Concepts, Technologies, and Industrial Applications," Journal of Manufacturing Systems, vol. 58, no. part B, pp. 346-361, January 2021.

[8] Y. Lu, C. Liu, K. I. K. Wang, H. Huang, and X. Xu, "Digital Twin-Driven Smart Manufacturing: Connotation, Reference Model, Applications and Research Issues," Robotics and Computer-Integrated Manufacturing, vol. 61, article no. 101837, February 2020.

[9] A. V. Barenji, X. Liu, H. Guo, and Z. Li "A Digital Twin-Driven Approach towards Smart Manufacturing: Reduced Energy Consumption for a Robotic Cell," International Journal of Computer Integrated Manufacturing, vol. 34, no. 7-8, pp. 844-859, 2021.

[10] C. Constantinescu, S. Giosan, R. Matei, and D. Wohlfeld, "A Holistic Methodology for Development of Real-Time Digital Twins," Procedia CIRP, vol. 88, pp. 163-166, 2020.

[11] A. Gallala, A. A. Kumar, B. Hichri, and P. Plapper, "Digital Twin for Human–Robot Interactions by Means of Industry 4.0 Enabling Technologies," Sensors, vol. 22, no. 13, article no. 4950, July 2022.

[12] Y. Palazhchenko, V. Shendryk, and S. Shendryk, "Digital Twins Data Visualization Methods. Problems of Human Interaction: A Review," In International Conference "New Technologies, Development and Applications", pp. 478-485, May 2023.

[13] Statcounter GlobalStats, "Desktop Operating System Market Share Worldwide," https://gs.statcounter.com/os-market-share/desktop/worldwide, September 30, 2022.

[14] J. Oyekan, M. Farnsworth, W. Hutabarat, D. Miller, and A. Tiwari, "Applying a 6 DoF Robotic Arm and Digital Twin to Automate Fan-Blade Reconditioning for Aerospace Maintenance, Repair, and Overhaul," Sensors, vol. 20, no. 16, article no. 4637, August 2020.

[15] M. Gadaleta, G. Berselli, M. Pellicciari, and F. Grassia, "Extensive Experimental Investigation for the Optimization of the Energy Consumption of a High Payload Industrial Robot with Open Research Dataset," Robotics and Computer-Integrated Manufacturing, vol. 68, article no. 102046, April 2021.

[16] J. Heredia, C. Schlette, and M. B. Kjærgaard, "Data-Driven Energy Estimation of Individual Instructions in User-Defined Robot Programs for Collaborative Robots," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6836-6843, October 2021.

[17] J. Yan and M. Zhang, "A Transfer-Learning Based Energy Consumption Modeling Method for Industrial Robots," Journal of Cleaner Production, vol. 325, article no. 129299, November 2021.

[18] M. Yao, Q. Zhao, Z. Shao, and Y. Zhao, "Research on Power Modeling of the Industrial Robot Based on ResNet," 7th International Conference on Automation, Control and Robotics Engineering, pp. 87-92, July 2022.

[19] G. Garg, "Digital Twin for Industrial Robotics," M.S. thesis, Department of Robotics and Computer Engineering, University of Tartu, Tartu, 2021.

[20] G. Garg, V. Kuts, and G. Anbarjafari, "Digital Twin for Fanuc Robots: Industrial Robot Programming and Simulation Using Virtual Reality," Sustainability, vol. 13, no. 18, article no. 10336, September 2021.

[21] P. Yotchon and Y. Jewajinda, "Combining a Differential Evolution Algorithm with Cyclic Coordinate Descent for Inverse Kinematics of Manipulator Robot," 3rd International Conference on Electronics Representation and Algorithm pp. 35-40, July 2021.

[22] C. Jacob, F. Espinosa, A. Luxenburger, D. Merkel, J. Mohr, T. Schwartz, et al., "Digital Twins for Distributed Collaborative Work in Shared Production," IEEE International Conference on Artificial Intelligence and Virtual Reality, pp. 210-212, December 2022.

[23] G. Gheorghe, "Mathematical Modeling of the Walking Hexapod Robot Using Denavit-Hartenberg Parameters," Annals of 'Constantin Brancusi' University of Targu-Jiu. Engineering Series, no. 1, pp. 64-69, 2021.

[24] A. Vatankhah Barenji, X. Liu, H. Guo, and Z. Li, "A Digital Twin-Driven Approach towards Smart Manufacturing: Reduced Energy Consumption for a Robotic Cell," International Journal of Computer Integrated Manufacturing, vol. 34, no. 7-8, pp. 844-859, 2021.

[25] National Development Council, "Taiwan's 2050 Net-Zero Emission," https://www.ndc.gov.tw/Content_List.aspx?n=FD76ECBAE77D9811, March 30, 2022

[26] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, 1st ed., New Jersey: Wiley, 2005.