



HAL
open science

A Photorealistic Digital Twin for a Tank Truck Washing Robotic System

Luís Vicente, Pedro Lomelino, Fernando Carreira, Francisco M. Campos,
Mário Mendes, J. Calado

► **To cite this version:**

Luís Vicente, Pedro Lomelino, Fernando Carreira, Francisco M. Campos, Mário Mendes, et al.. A Photorealistic Digital Twin for a Tank Truck Washing Robotic System. 23th Working Conference on Virtual Enterprises (PRO-VE), Sep 2022, Lisbon, Portugal. pp.57-66, 10.1007/978-3-031-14844-6_5 . hal-04642043

HAL Id: hal-04642043

<https://inria.hal.science/hal-04642043v1>

Submitted on 9 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

A Photorealistic Digital Twin for a Tank Truck Washing Robotic System

Luís Vicente¹, Pedro Lomelino¹, Fernando Carreira^{1,3}, Francisco M. Campos¹,
Mário J. G. C. Mendes^{1,2}, J. M. F. Calado^{1,3}

¹ ISEL – Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa,
1959-007 Lisbon, Portugal

² CENTEC – Centre for Marine Technology and Ocean Engineering, Lisbon, Portugal

³ LAETA/IDMEC – Instituto de Engenharia Mecânica, Lisbon, Portugal
{a42292, a43640}@alunos.isel.pt, {fernando.carreira, francisco.campos}@isel.pt,
{mario.mendes, joao.calado}@isel.pt

Abstract. In the current industrial context, the adoption of a digital twin (DT) has proven to be an appropriate tool to optimize the entire lifecycle of a system. This vision includes the adoption of a DT, in the collaborative network of organizations from the virtual modeling of a system, through its control, operation, namely simulation, implementation, and monitoring/supervision. To take full advantage of DTs, it is essential that they are as photorealistic as possible, allowing all kinds of simulations to be carried out, for example testing vision algorithms. In this context, this paper presents the development of a DT that is integrated as a collaborative system to monitor all phases of the life cycle of a robot destined to wash tank truck. The Unreal Engine 4 software was used to develop a DT and the communication architecture was designed, using the standard OPC UA protocol.

Keywords: Digital twin, Robotic simulation, Factories of the future, Unreal Engine 4, Tank Truck Washing.

1 Introduction

The fourth industrial revolution (Industry 4.0) introduced several changes in manufacturing processes and service provision, driven by the need for their digitalization. Today's business competitiveness requires manufacturers and service providers to be able to respond and adapt to the increasingly diverse requirements of customers in ever shorter periods of time, demanding the adoption of collaborative networks [1]. One way to achieve this goal is to implement the smart factory concept, introduced by Industry 4.0 [2]. Cyber-physical systems (CPS) and digital twins (DT) are two important technologies for the digital transformation in organizations. Both concepts seek to connect the physical with the cybernetic world. However, while CPSs focus on the interaction between the two worlds, DTs provide a complete digital description of the physical world/process. A DT can be considered a digital representation of a physical system [3].

With the recent advances in computing technologies, the DT concept has evolved into a safe virtual space for simulation, testing and validation [4]. From this perspective, a solution for the creation of simulation models is to construct a DT that mirrors and observes the behavior of the real world, allowing to achieve a virtual commissioning (VC) scenario [5],[6]. VC is typically viewed as a hardware-in-the-loop (HiL) configuration that is used before the actual (physical) commissioning of an industrial device, for the purpose of testing and verifying the control logic. In HiL simulation, the real controller is connected to a DT that replicates the physical system [7].

In recent years, the DT concept has evolved quickly and been widely used. The authentic representation of physical systems, has advantages in several areas such as: manufacturing [5], health care [8], smart cities [9] and robotics [10], [11]. DT in robotics are embedded throughout their lifecycle, taking a key role in VC in order to test their design and improve control algorithms for robots at the development stage [10]. With DTs it is possible to test different scenarios to improve performance related to cost and time. In [11], the authors proposed a DT to monitor the collaborative work of various robots.

The implementation of a DT in an organization's communication network allows achieving a digital transformation towards collaborative manufacturing [6, 12]. In the development of a new system, the use of a DT facilitates the simulation of several scenarios previously related to being physically implemented, allowing the identification of bugs in efficiency systems, i.e. reducing costs related with tests of new processes in physical systems [6, 8, 9].

This paper addresses the DT implementation applied to a tank truck washing process automation, contributing with an effective solution for testing vision algorithms in a photorealistic virtual environment. Currently, tank washing is carried out through rotary jet heads which are manually inserted into the tanks. Motivated by the need to build collaborative, resilient and sustainable systems and meet the concept of factories of the future, this paper focuses on a robotic system to automatically perform the washing task. Specifically, this paper describes the development of a DT to support the entire lifecycle of the robotic system and promote its integration into the organization's collaborative network. One of the features of this system is the use of computer vision to detect the loading port. Given the need to validate computer vision algorithms, we propose a DT based on the Unreal Engine 4 (UE4), which allows rendering photorealistic images captured by virtual cameras.

To guarantee the smart factory concept, it is intended to develop a DT interoperable with the different systems present in organizations. The DTs must have the ability to simulate the changes that may occur in the production/service delivery processes. Several research works have been developed to implement a Service Oriented Architecture (SOA) in which all control, automation and supervision systems are designed as a combination of different services. These architectures allow the functions performed by the robot to be quickly improved and adapted to changes in the industry [10]. For that matter, the Informatics System of Systems (ISoS) architecture was adopted, where the developed DT constitutes a service. The implementation of this architecture allows the DT to communicate with the other services that are part of an organization, ensuring interoperability between all systems [13].

This paper is organized into the following sections: section 2 describes the various stages of the DT development for the robotic system for tank truck washing; section 3

presents the architecture developed for the communication between the DT and the system controller; section 4 presents the tests performed to verify the functioning of the proposed communication architecture, and finally, section 5 presents the work conclusions.

2 Development of a Digital Twin

This work addresses the development of a DT for an automated tank truck washing system. The use of a DT has advantages in the development of the robotic system and in its monitoring during operation. For example, a DT enables testing of control algorithms in parallel with the mechanical design, production and installation phases. Once implemented, the DT may be used to complement the existing supervision system, by providing a realistic visualization of the system in real time.

The choice of the supporting software is an important step in the development of a DT. In the present work, this choice is influenced by the requirement of detecting the loading port through an image processing system. Since image processing is essential for the robotic system operation, special emphasis was placed on the photorealism of the DT. For this reason, it was chosen the Unreal Engine 4 (UE4) software, from Epic Games®, for the virtual environment development, due to the sophisticated methods that it provides for rendering realistic images. The use of a game engine allows testing the system under varying lighting, by changing weather settings or artificial lighting, which is essential for validating a computer vision system. This software, which was initially developed for the creation of first-person shooter (FPS) games, has been successfully used in a wide variety of applications, including robotics [14],[15]. Additionally, UE4 is free to use, therefore it does not increase the robotic system cost.

According to Lei, et al. [16], the development of a DT can be divided into the construction of three models: 3D model, kinematic model and rules model. In the first stage, a three-dimensional model of the environment is built, including the robotic system and the tank truck. Next, a kinematic model of the system is defined, by establishing the mechanism constraints. The final stage defines the rules for interaction with the DT and the data exchanged between the controller and the DT. Next, the application of these stages in the DT development on the proposed robotic system is detailed.

The first stage in DT development is 3D modeling. To obtain a realistic DT, the 3D shapes, colors and textures used in the model must closely resemble those of the real world. Such models can be created with a computer aided design (CAD) software or by digitizing the physical system with 3D scanners. Since the proposed robotic system is still under development, the second approach was not viable and, thus, the 3D model was developed in the CAD software Solidworks®.

The DT model essentially includes a gantry robot, a jet head, and a tank truck, whose models were obtained from various sources. The gantry robot design was carried out through a specialized software from LUCAS [17], which also provided 3D models of that equipment (Fig. 1a). The jet head was selected from the catalogue of the Alfa Laval® manufacturer [18], which also provides corresponding 3D models (gantry robot

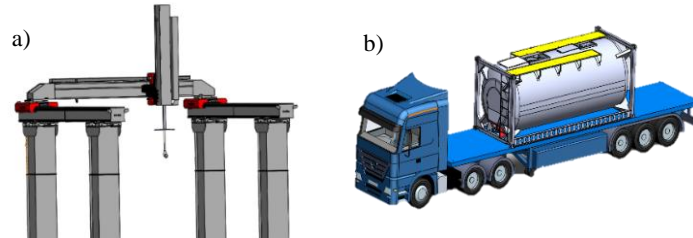


Fig. 1. 3D model: a) tank truck washing robotic system; b) tank truck.



Fig. 2. DT realistic environment developed in UE4.

end effector depicted in Fig. 1a). The tank truck model (Fig. 1b) was selected from the GRABCAD[®] repository [19].

Besides the aforementioned items, the virtual environment was enriched with a realistic model of the scenario where the washing operation takes place, by resorting to objects from the UE4 marketplace [20]. The complete virtual environment is shown in Fig. 2.

The 3D model described above represents the appearance of a static system. However, to develop a DT it is essential to set the kinematic properties that are relevant to the simulation. In the present work, the robot is described by a cartesian kinematic configuration, which yields linear relations. In this case, to impose the kinematic constraints on the objects of the system, it is necessary to: i) establish the rigid coupling among objects that are fixed together, ii) allow the translation movement of the x, y, and z axis and iii) establish the hierarchy between the axes. For this purpose, the objects in UE4 were organized in a hierarchical tree that defines the kinematic order, as usual in 3D modeling programs.

The purpose of the rules model is to define the interaction protocol with the DT and to identify the data exchanged between the virtual world and the controller. Within this interaction, the role of the controller is to send action commands in the form of x (longitudinal movement), y (transverse movement) and z (vertical movement) values that describe the movement of each robot axis.

The action commands should depend on visual feedback from the virtual world. For this purpose, a “Fusion Camera Actor” element was also added to the model (element A depicted in Fig. 3a). This element simulates a virtual camera and enables capturing images in the DT, as illustrated in

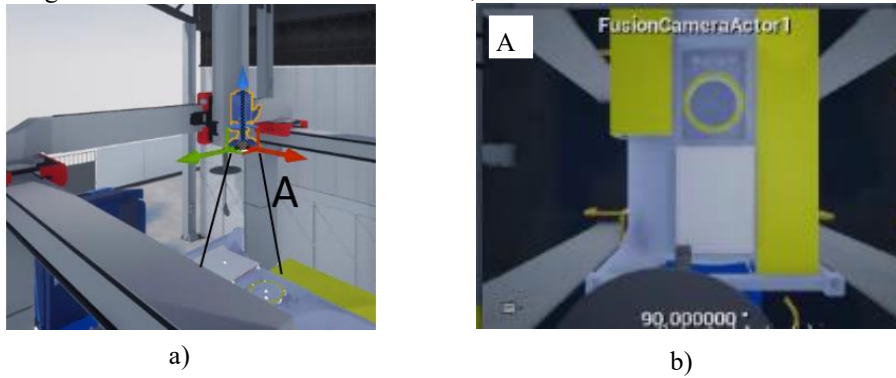


Fig. 3b. The data sent from the virtual model to the

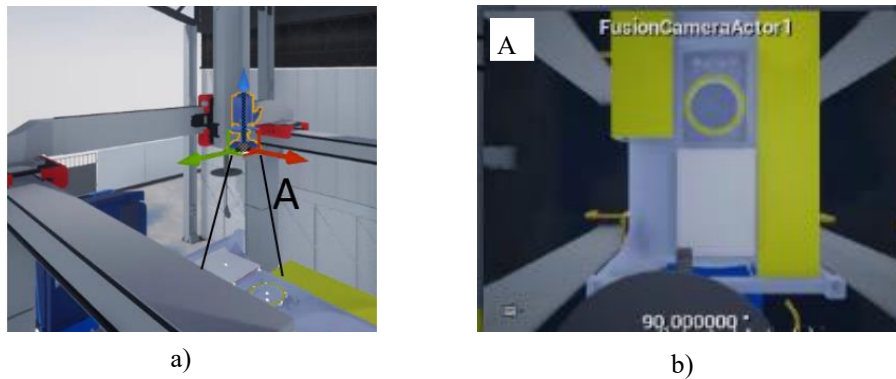


Fig. 3. Virtual camera in UE4: a) camera position and orientation; b) captured image.

controller consists of image data, which is later processed by image processing algorithms that compute the position targets for the gantry.

3 System Architecture

The proposed robotic system for tank truck washing is controlled by a programmable logical controller (PLC) connected to a computer, running a software platform for supervision, control and a data acquisition (SCADA).

In the present work, an HiL architecture is configured with the DT controlled by the same PLC that control the real system. This aims to validate the logical control

developed and the SCADA system functionalities. Furthermore, this architecture allows to use the DT in a hybrid commissioning, with the PLC connection to the real system, simultaneously.

As mentioned earlier, the robotic system DT was created with UE4. However, this software is not geared towards automation and the communication with a PLC is not provided as standard. For this reason, a custom designed communication architecture was required. Matlab[®] was used in the communication between the PLC and the DT due to need of its image processing capabilities, required by the current task, and to achieve future project goals that build on artificial intelligence (AI) algorithms. Taking this into account, Matlab[®] was selected to develop image processing algorithms. This allowed greater flexibility in choosing the most appropriate techniques from a wide range of classic and advanced tools available in the Matlab[®] computer vision toolboxes. From the communication architecture point of view, this choice implies that images recorded in the DT must be sent to Matlab[®] and that targets found by the detection algorithm must be sent back to the DT. To support this communication a custom library, Matlab_Unreal Interface, was developed, building on the existing UnrealCV library [21]. UnrealCV is a set of python functions developed to support computer vision research by providing an interface to UE4 synthetic worlds. Matlab_Unreal Interface extends this library with Matlab[®] functions that port the same functionalities to the Matlab[®] environment. In addition to exchanging image data, the library features include reading and setting the position of objects in the virtual environment.

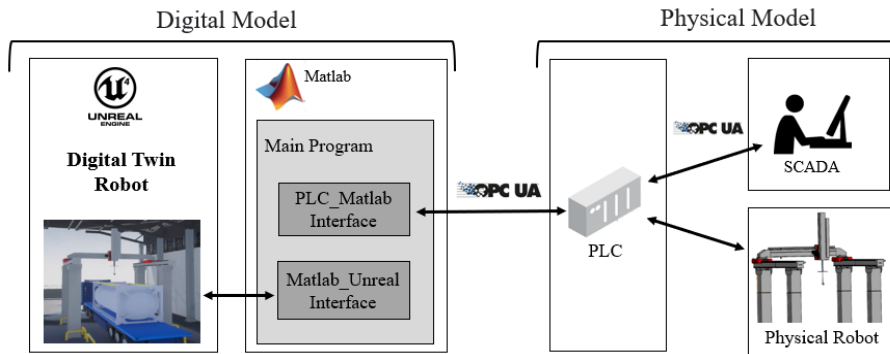


Fig. 4. Communication Architecture

Once a communication interface has been established between Matlab[®] and UE4, it can be used to mediate communication between the PLC and UE4. The OPC UA is the preferred PLC communication protocol, due to its high reliability, cross communication and uniformity of access [16]. This protocol is based on a client/server model, where the OPC UA server defines the address space and provides an access interface, and the OPC UA client implements the data acquisition that references that interface [11]. In the proposed communication architecture, shown in

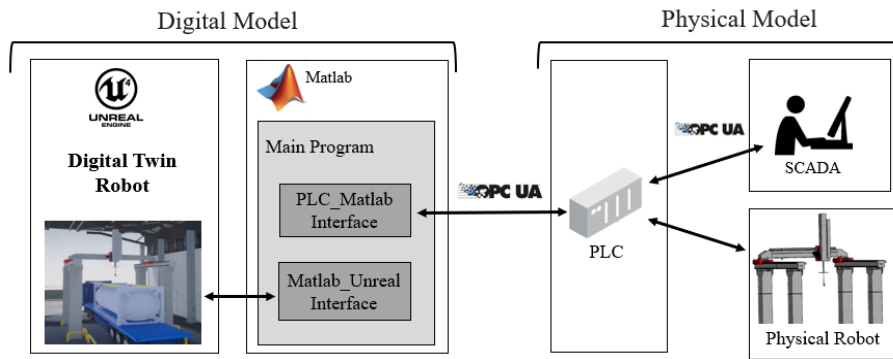


Fig. 4, the PLC performs the role of the OPC UA server while the other OPC UA nodes are clients. This is the case of Matlab®, which mediates the communication between the PLC and UE4. To achieve this functionality, another library was created, called PLC_Matlab interface. This library, based on the python library freeopcua [22], is a set of functions that enables Matlab® to perform the OPC UA client role. In the proposed architecture, Matlab® is responsible for both the image processing task and communication dispatch. All the operations underlying these tasks are integrated in a Matlab® program, *Main Program* (see

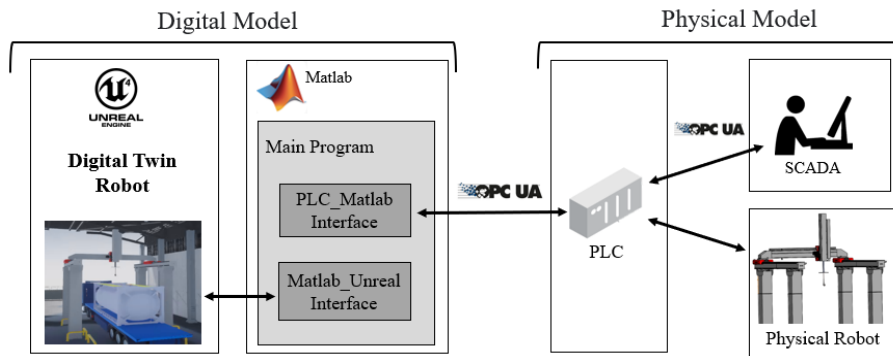


Fig. 4), which is executed in parallel with the other agents of the system.

4 Communication Architecture Verification

The developed system was tested according with the control real system specifications. It is considered a controller performed by the PLC CPX-E+CMMT, supplied by FESTO® and programed by the Codesys® software. Notice that the PLC has a server OPC UA, which specification is required in the developed architecture. The robotic tanks washing process was programmed in the PLC, according to the flowchart depicted in Fig. 5. The axes movements allow the validation of the developed DTs models and proposed architecture. The robot moves a virtual camera, coupled to the y

axis, to detect the loading port position through the image processing algorithms of captured images. The loading port position is estimated from the relative position to the virtual camera.

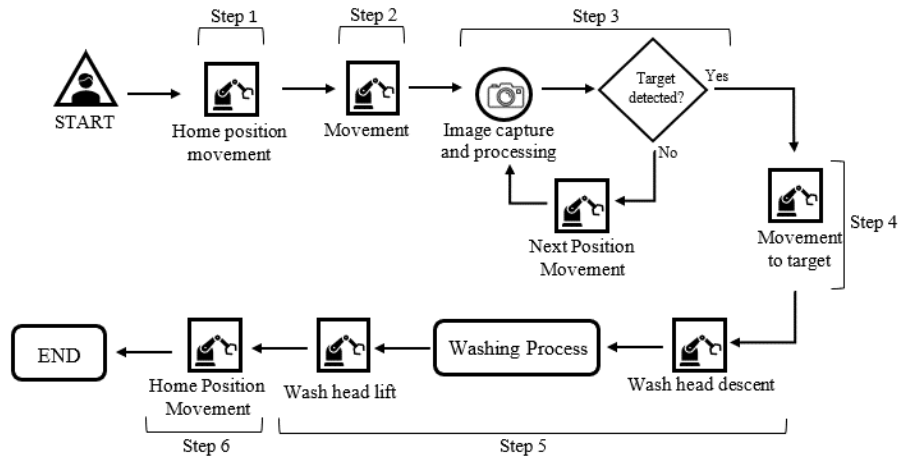


Fig. 5. Flowchart of the robotic system for road tank washing.

According with the flowchart depicted in Fig. 5, after the operator press the “START” bottom in the HMI, the washing cycle begins the following steps:

1. the robot moves to the home position, defining the cycle start position.
2. the y axis moves the jet head and the camera to the vehicle longitudinal axis.
3. the x axis moves the terminal element along the tank until the loading port is detected, through image processing algorithms.
4. both x and y axes are moved to ensure that the terminal element is over the loading port, resorting to image processing.
5. The jet head is introduced inside the tank, carried out the tank washing process for a predefined time and is removed again.
6. The axes are moved to the Home Position, finishing the washing cycle.

The process was executed, and the DT jet head positions were obtained by the Matlab[®] software.

Fig. 6 shows the DT jet head positions throughout the washing cycle, where it is possible to observe the steps executed by the implemented algorithm. Notice that the path does not show step 1, since the robot was already in the home position when the washing cycle was started. Observing the results, the proposed architecture was successfully validated.

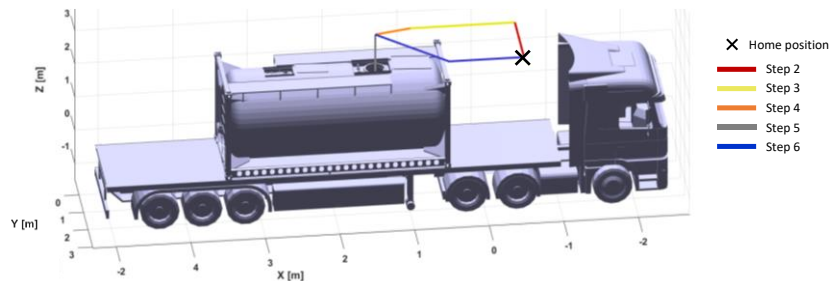


Fig. 6. Jet head trajectory described throughout the operating cycle.

5 Conclusion

A DT is one of the pillars of a digital transformation. This allows to create virtual replicas of physical systems, providing a simulation and supervision environment. The use of a DT makes it possible to configure and test a collaborative system before physical commissioning, to be integrated into the factory concept of the future. This work presented the DT development of a tank truck washing robotic system. The DT was implemented in UE4 in order to achieve a high level of realism. The use of this software allows the development of control algorithms based on image processing. For that purpose, the DT components were modeled in UE4, including a virtual camera to capture images from the environment. It was implemented an architecture to ensure the communication between devices and platforms: DT, PLC, SCADA software and Matlab®, resorting to OPC UA communication protocol. The results have shown that the architecture was successfully implemented, allowing the DT development in many real frameworks that requires a high-level photorealism and advanced computation. The DT will be tested for supervision of a laboratory gantry and is intended to be tested on a gantry in an industrial environment.

Acknowledgments. This work was supported by the project POCI-01-0247-FEDER-039764 under the Program PORTUGAL2020.

References

1. Camarinha-Matos, L.M., Rocha, A.D., Graça, P.: Brief Overview of Collaborative Approaches in Sustainable Manufacturing. *IFIP Advances in Information and Communication Technology*. 629 IFIPAI, 3–18 (2021). https://doi.org/10.1007/978-3-030-85969-5_1.
2. Karagiannis, D., Buchmann, R.A., Boucher, X., Cavalieri, S., Florea, A., Kiritsis, D., Lee, M.: OMILAB: A Smart Innovation Environment for Digital Engineers. *IFIP Advances in Information and Communication Technology*. 598, 273–282 (2020). https://doi.org/10.1007/978-3-030-62412-5_23.

3. Ward, R., Soulatiantork, P., Finneran, S., Hughes, R., Tiwari, A.: Real-time vision-based multiple object tracking of a production process: Industrial digital twin case study. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. 235, 1861–1872 (2021). <https://doi.org/10.1177/09544054211002464>.
4. Bilberg, A., Malik, A.A.: Digital twin driven human–robot collaborative assembly. *CIRP Annals*. 68, 499–502 (2019). <https://doi.org/10.1016/j.cirp.2019.04.011>.
5. Leng, J., Wang, D., Shen, W., Li, X., Liu, Q., Chen, X.: Digital twins-based smart manufacturing system design in Industry 4.0: A review. *Journal of Manufacturing Systems*. 60, 119–137 (2021). <https://doi.org/10.1016/j.jmsy.2021.05.011>.
6. Gavin, C.: Digital Twins Calibrated with Operational Data Drive Efficiency. *InTech Magazine*. (2021).
7. Martins, A., Costelha, H., Neves, C.: Supporting the design, commissioning and supervision of smart factory components through their digital twin. 2020 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2020. 114–119 (2020). <https://doi.org/10.1109/ICARSC49921.2020.9096072>.
8. Rasheed, A., San, O., Kvamsdal, T.: Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*. 8, 21980–22012 (2020). <https://doi.org/10.1109/ACCESS.2020.2970143>.
9. Qi, Q., Tao, F., Zuo, Y., Zhao, D.: Digital Twin Service towards Smart Manufacturing. *Procedia CIRP*. 72, 237–242 (2018). <https://doi.org/10.1016/j.procir.2018.03.103>.
10. Jhunjhunwala, P., Atmojo, U.D., Vyatkin, V.: Applying Skill-based Engineering using OPC-UA in Production System with a Digital Twin. *IEEE International Symposium on Industrial Electronics*. 2021-June, (2021). <https://doi.org/10.1109/ISIE45552.2021.9576342>.
11. Zong, X., Luan, Y., Wang, H., Li, S.: A multi-robot monitoring system based on digital twin. *Procedia Computer Science*. 183, 94–99 (2021). <https://doi.org/10.1016/j.procs.2021.02.035>.
12. Mrabti, N., Gargouri, M.A., Hamani, N., Kermad, L.: Towards a Sustainable Collaborative Distribution Network 4.0 with Blockchain Involvement. *IFIP Advances in Information and Communication Technology*. 629 IFIPAI, 41–52 (2021). https://doi.org/10.1007/978-3-030-85969-5_4.
13. Vicente, L., Lomelino, P., Carreira, F., Campos, F.M., Mendes, M.J.G.C., Osório, A.L., Calado, J.M.F.: Industrial Collaborative Robotics Platform. In: *IFIP Advances in Information and Communication Technology*. pp. 567–576. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85969-5_53.
14. Alvey, B., Anderson, D.T., Buck, A., Deardorff, M., Scott, G., Keller, J.M.: Simulated Photorealistic Deep Learning Framework and Workflows to Accelerate Computer Vision and Unmanned Aerial Vehicle Research. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 3882–3891. Institute of Electrical and Electronics Engineers (IEEE) (2021). <https://doi.org/10.1109/iccvw54120.2021.00435>.
15. Müller, M., Casser, V., Lahoud, J., Smith, N., Ghanem, B.: Sim4CV: A Photo-

- Realistic Simulator for Computer Vision Applications. *International Journal of Computer Vision*. 126, 902–919 (2018). <https://doi.org/10.1007/s11263-018-1073-7>.
16. Lei, Z., Zhou, H., Hu, W., Liu, G.-P., Guan, S., Feng, X.: Toward a Web-Based Digital Twin Thermal Power Plant. *IEEE Transactions on Industrial Informatics*. 18(3), 1716–1725 (2022). <https://doi.org/10.1109/TII.2021.3086149>.
 17. RC3 - 1 ROBOT - 3-AXES GANTRY ROBOT - LUCAS FRANCE, https://lucas-robotic-embedded.partcommunity.com/3d-cad-models/rc3-1-robot-3-axes-gantry-robot-lucas-france?info=lucas%2Frobot_cartesien_3_axes%2Frc3_1_robot.prj&cwid=3603, last accessed 2022/01/14.
 18. Alfa Laval: Alfa Laval TJ MultiJet 25 Rotary Jet Head - Product leaflet. (2021).
 19. grand - Recent models | 3D CAD Model Collection | GrabCAD Community Library, https://grabcad.com/library/tag/grand?page=1&time=all_time&sort=recent, last accessed 2022/01/14.
 20. Content Search - UE Marketplace, <https://www.unrealengine.com/marketplace/en-US/assets?count=20&sortBy=effectiveDate&sortDir=DESC&start=0&tag=19467>, last accessed 2022/02/24.
 21. Qiu, W., Yuille, A.: UnrealCV: Connecting computer vision to unreal engine. In: *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*). pp. 909–916. Springer Verlag (2016). https://doi.org/10.1007/978-3-319-49409-8_75.
 22. GitHub - FreeOpcUa/python-opcua: LGPL Pure Python OPC-UA Client and Server, <https://github.com/FreeOpcUa/python-opcua>, last accessed 2022/02/25.