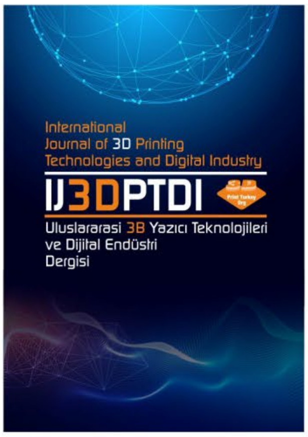

ULUSLARARASI 3B YAZICI TEKNOLOJİLERİ
VE DİJİTAL ENDÜSTRİ DERGİSİ
INTERNATIONAL JOURNAL OF 3D PRINTING
TECHNOLOGIES AND DIGITAL INDUSTRY

ISSN:2602-3350 (Online)
URL: https://dergipark.org.tr/ij3dptdi

AN INDUSTRIAL APPLICATION OF DIGITAL TWIN FOR A SMART FACTORY MODEL USING COPPELIASIM

Yazarlar (Authors): Cüneyt KILIÇ, Kamil ÇETİN*, Mehmet Erdal ÖZBEK

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): *Kılıç C., Çetin K.,Özbek M. E., "An Industrial Application of Digital Twin for A Smart Factory Model Using Coppeliasim" Int. J. of 3D Printing Tech. Dig. Ind., 8(3): 316-325, (2024).*

DOI: 10.46519/ij3dptdi.1452734

Araştırma Makale/ Research Article

Erişim Linki: (To link to this article): https://dergipark.org.tr/en/pub/ij3dptdi/archive

AN INDUSTRIAL APPLICATION OF DIGITAL TWIN FOR A SMART FACTORY MODEL USING COPPELIASIM

Cüneyt KILIÇ[a], Kamil ÇETİN[a,b]*, Mehmet Erdal ÖZBEK[a,b]

[a]Izmir Katip Celebi University, Facult of Engineering and Architecture, Department of Electrical and Electronics Engineering, Izmir, TURKIYE
[b]Izmir Katip Celebi University, Akıllı Fabrika Sistemleri Uygulama ve Araştırma Merkezi, Izmir, TURKIYE

*Corresponding Author: *kamil.cetin@ikcu.edu.tr*

(**Received:** 14.03.2024; **Revised:** 06.11.2024; **Accepted:** 25.11.2024)

ABSTRACT

A digital twin is quickly becoming a necessary component of manufacturing. When combined with Internet of Things (IoT) technology, it is possible to fulfil Industry 4.0's requirement for a digital transformation to create a smart factory. Many IoT devices, such as sensors, detect physical phenomena in their environments, collect data, and communicate. Digital twins that mimic the features and capabilities of real physical IoT devices can also be realized. Digitalization using digital twin requires not only IoT sensors but also software tools for virtualization. However, there are limited number of practical applications for the digitalization of large and complex systems. In this work, a smart factory model is used to develop its digital twin. Because the simulation software of industrial automation system manufacturers is licensed and different systems are incompatible with each other for digital twin, CoppeliaSim digital twin software, which is open source and can work independently of the industrial automation system model, was used in this study. The corresponding physical sensor data is transferred to CoppeliaSim in real time. Moreover, three scenarios were realized by achieving bilateral data transmission and control between the physical and digital twin models. Instant status monitoring presents the performance of digital twinning. While most of the digital twin studies in the literature are carried out either by transferring data only from the virtual system to the physical system or only from the physical system to the virtual system, in this study, simultaneous data exchange was implemented between the real and virtual systems.

Keywords: Digital Twin, Internet of Things, Smart Factory, Industry 4.0, CoppeliaSim.

1. INTRODUCTION

The quality of life and the advantages that technological developments offer to the human beings are seen through the revolutions in industry. The most recent stage of the industrial revolution, known as "Industry 4.0," is centred on deploying cutting-edge technologies to digitalise production processes [1-2]. The traditional manpower-based factories continue to be replaced by smart factories or smart working environments that are digitalized and therefore they are able to reduce the unskilled manpower. The ever-increasing production and efficiency requirements with reduced costs is a reality that every manufacturing company demands. The Internet of Things (IoT) has made it possible for Industry 4.0 to become increasingly digital. IoT connects physical items to the Internet so that they may be managed remotely, making production processes smarter and more efficient [2-3]. IoT technology converts data taken from the physical world into a digital form with developed sensors. The transfer of those data to the virtual environment ensures the digitalization and named as "digital twinning" of the physical world [4]. Using digital twin technology, a synchronized virtual representation of a real-world object is created in a digital environment [1, 4-5]. The object transferred to the virtual environment is a real-time measured data. In this way, it is possible to easily and effectively monitor and analyze the performance of the physical object in the digital environment [6]. The digital twin is widely applied in various industry sectors, including agriculture, electricity generation, healthcare, and smart factory environments [7]. Because it enables free design testing and early problem discovery, a digital twin is the recommended technology for enhancing production integration and quality [8].

While the digital twin embraces big industry manufacturing processes, there are also small-scale system models mentioned in the literature. For example, Park et al. [9] studied about reinforcement learning and the realization of a digital twin of the micro smart factory model. In [10], a digital twin of a real-life painting robot arm is created using the CoppeliaSim simulation software [11]. In [12], another digital twin study carried out in the Gazebo simulation software is related to the localization of the robot [13]. It was also shown that digital twin technology was used in the Gazebo simulation software of an autonomous mobile robots [14]. In the study of Al-Geddawy et al. [15], a digital twin of a low-cost system was realized in a different simulation environment called RoboDK [16]. Similarly, the UR3 robot arm is observed, whose digital twin was realized in CoppeliaSim [17].

As mentioned above, digital twin studies in the literature have increased recently and mostly simulation studies of small-scale systems such as robot arms or one-way control from the simulation to the real system have been provided. The aim of this paper is to raise awareness about the creation of a digital twin of a large-scale factory model while maintaining bidirectional control between physical and simulation systems. Thus, the remaining of this paper is organized as follows: Section 2 explains the working system of the smart factory and its creation in the simulation software. Section 3 shows the communication method between the software and the smart factory. Section 4 presents the experimental studies and results. Section 5 briefly evaluates and discusses the study with the results. Finally, the paper terminates with discussions and future works in Conclusion section.

2. PHYSICAL AND DIGITAL TWIN MODELS OF THE SMART FACTORY

Computer-aided design (CAD) and simulation programs are needed to create a smart production system and to realize a digital twin of this system based on real data. There are many designs and simulation software developed to build a digital twin of the real system in virtual environments. CAD modeling software (e.g., SolidWorks [18], AutoCAD [19], SketchUp [20]) is consulted to create an exact replica of the designs of physical objects in the real world and the designs in the virtual environment [21]. Those environments are generally CoppeliaSim, Matlab [22], Gazebo, RoboDK. In this study, SolidWorks having the most common library, is preferred as the CAD program. The distributed control architecture of the open-source CoppeliaSim robotic simulator software (formerly known as V-REP) is built on an integrated development environment. Each object/model in CoppeliaSim can be independently controlled using an API (Application Programming Interface) client, a plug-in, and an embedded script. CoppeliaSim's versatility and suitability for multi-robot applications stem from these properties. The programming languages Lua [23], Octave [24], Matlab, C/C++, Python, and Java are also supported. According to Rohmer et al. [25], CoppeliaSim is utilised as a digital twin, for fast algorithm development, factory automation simulations, and rapid prototyping and validation. Programmable Logic Controller (PLC) is used to control production lines, motors and robotic systems in smart factory environments.

2.1. Physical Twin of Smart Marble Factory

In this study, we aim to build a digital twin of a marble factory. For this purpose, we use a smart marble factory scale model shown in Figure 1 in our Smart Factory Systems Application and Research Center (AFSUAM) laboratory. In this section, a brief information is given about the working principle of our model. It consists of two production lines and three workstations. The first workstation of the smart factory model, shown in Figure 2, will be used to create its digital twin. The first workstation has conveyors, ovens, temperature sensors, distance sensors, direct current (DC) encoder motors, PLC, relays, microprocessor (MPU) and robot arms. In the first workstation, the slabs of marble are placed on the production line with the help of a robot arm, manually filled with epoxy process to close the cracks on the front surface, mesh and epoxy processes are applied again to the back surface for durability, and drying processes are carried out to dry the applied epoxy. In the rest of the article, the first workstation of the smart factory scale model is simply called a physical twin.

Figure 1. Smart factory scale model in AFSUAM laboratory.

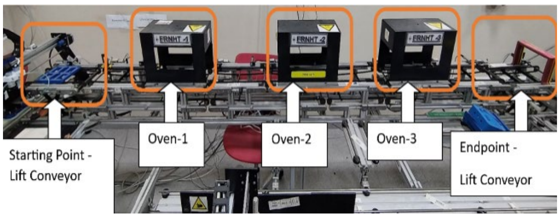

Figure 2. The first workstation of smart factory scale model: Locations of ovens and lift conveyors.

The electrical connection schematics of the motors, drives, sensors, MPUs, PLC and central data acquisition and control unit used in the physical twin of the first workstation of the smart factory model photographed in Figure 1 and Figure 2 is drawn in Figure 3. The 8 DC motors were used for the 8 conveyors. For the sensors, the 3 MAX6675 K-type thermocouples and the 11 IME12-08NPSZC0S model inductive proximity sensors were used, which are efficient and offer high performance. Arduino Uno was used as the MPU and Python was preferred for programming language. Mitsubishi FX5U PLC module with high input output speed was used for data collection and control operations. An API is required to ensure communication between the simulation program CoppeliaSim and the PLC system. For this communication, the Python programming language is used as the API between the personal computer (PC) on which CoppeliaSim is run and the PLC where the smart factory model is controlled. Therefore, a double-sided control of the factory model is provided in the virtual environment with real-time data.

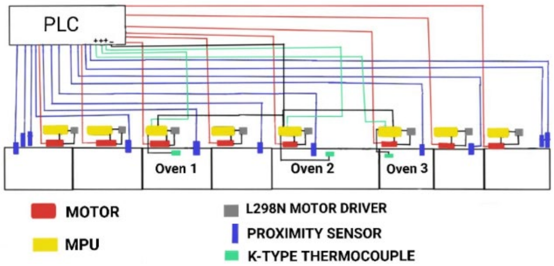

Figure 3. Electrical connection schematics of the equipment in the smart factory scale model.

As shown in Figure 4, the connections of the equipment used in the physical twin are as follows. For the 24V supply voltage of the PLC, the red (+) and blue (-) cables of the SMPS (Switch Mode Power Supply) are connected to the L (+) and N (-) pins of the PLC, respectively. The DC encoder motors used for conveyors have six cables (red, black, green, blue, yellow, white). For the motor power, red (+) and black (-) cables are connected to the out1 and out2 inputs of the motor driver, respectively. The yellow and white cables are for the encoder A and B outputs of the motor, respectively, and these outputs are connected to the digital inputs of the PLC. The blue and green cables are for the encoder supply and ground and are connected to the 5V and ground pins of the MPU, respectively. The enable, in1, and in2 pins of the motor driver are connected to the MPU pins through the brown, green, and purple cables, respectively. The MAX6675 K-type thermocouple has five wires (black, red, yellow, blue, green). The red and black cables are for the 5V supply and ground connections of the MPU, respectively. The yellow, blue, and green cables are for 'sck', ‘cs’ and ‘so’ connections of the MPU, respectively. The temperature values received from the MPU are converted to values in the range of 0-255 in the MPU code, and these values are applied as voltage to the analog input (Built-in Analog Input) of the PLC. The proximity sensor has three cables (black, blue, brown). The connections of the proximity sensor are as follows, brown (+) and blue (-) cables are connected to the (+) and (-) poles of the SMPS, respectively. The cable that transfers proximity sensor data is the black cable, and it is connected to the digital input of the PLC.

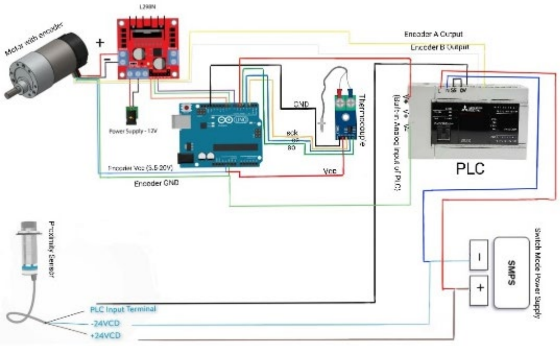

Figure 4. Physical twin connections between sensors, motors/drivers, MPU, PLC input and outputs.

The Mitsubishi FX5U PLC is used to control all subsystems in the physical twin and receive all sensor data and is programmed with the ladder method. As shown in Figure 4 and explained above, the input and outputs of PLC are connected to the sensors and motors/drivers used in the physical twin. The physical twin has a three degree-of-freedom robot arm at the starting and ending points. In addition, there are a total of twelve conveyors, six conveyors on the upward line and four conveyors on the downward line and excluding two lifting type conveyors at the starting and ending points. Lift type conveyors are conveyors that have the ability to move up and down. The operation of the conveyors is carried out by DC encoder motors integrated into the conveyor. DC encoders are electromechanical devices that, when the motor shaft they are linked to moves, generate digital electrical signals. There are three ovens placed on the conveyors in the upward line. These ovens (i.e., FRNHT-1, FRNHT-2, FRNHT-3) are placed on the third, fifth, and sixth conveyors, respectively. The temperature sensors are integrated into the three ovens. The proximity sensors used to detect the marble slabs are integrated into each conveyor.

The operation of the lift conveyors and oven lines of the physical twin is shown in the working diagram given in Figure 5. In this flow diagram, the decision-making process is carried out by checking whether the proximity sensors on the system detect the product or not.

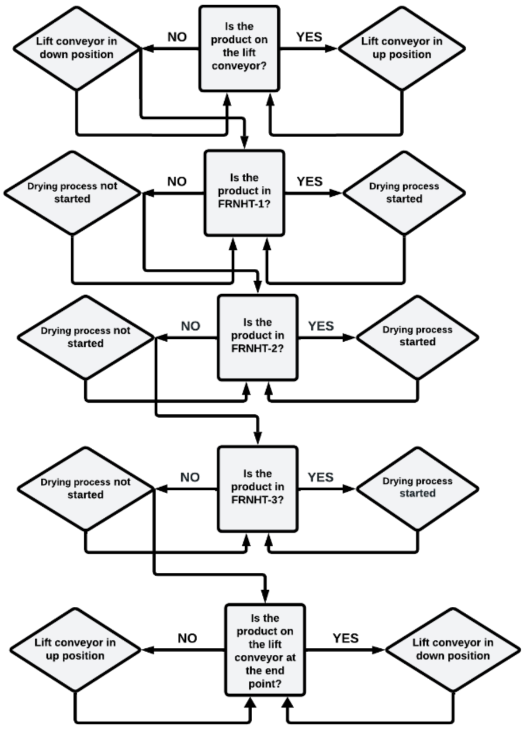

Figure 5. Flow diagram of lift conveyors and oven lines in the first workstation of the smart factory scale model.

The robot arm being at the starting point of the first workstation places the marble block on which the epoxy has been applied and which is desired to be dried, onto the transport tray on the first lift type conveyor. The lift type conveyor is in the down position before the starting moment. Proximity sensors connected to the conveyor move to the up position after detecting the marble placed on the transport tray on the conveyor. Once the proximity sensors detect the product, the motors start, and the conveyor starts moving. When the transport tray leaves the lift conveyor, it is detected by proximity sensors and the conveyor is allowed to return to its down position. When the transport tray moving on the conveyor reaches the oven areas on the third, fifth, and sixth conveyors, respectively, the drying process is started thanks to the proximity sensors. Drying processes take place within the time and temperature specified by the user. After the drying process is completed, the marble comes out as a semi-finished product. When the transport tray reaches the lift conveyor at the end point, the conveyor, which is normally in an up position, comes to a down position as a result of the proximity sensors detecting the semi-finished product. The semi-finished product on the tray is taken with the help of a robot arm and placed on smart transport wagons. After the semi-finished product is picked by the robot arm, the transport tray returns to its starting position via the downward conveyors to place the new product on it.

2.2. Digital Twin of the Physical Twin

In this section, a digital twin of the physical factory model created in the real environment is designed in the simulation (virtual) environment for monitoring and double-sided control. The digital twin of the smart factory is realized using the CoppeliaSim simulation software. All parts of the physical twin in the real environment are drawn on Solidworks and uploaded to CoppeliaSim in Unified Robotics Description Format (URDF) and Standard Triangle Language (STL) format. After the designed drawings are loaded into the simulation environment, the dimensions of the materials used in the physical twin are applied in the simulation environment and the design of the real system is established by combining the drawings. The length of the lift conveyors is 35 cm. The five of the upward conveyors is 35 cm long, the remaining is 70 cm long. Six of the downward conveyors are 70 cm long and the remaining is 35 cm long. The width of all conveyors is 26.3 cm. The ovens are 35 cm deep, 25 cm long, and 17.8 cm wide. The established simulation environment, as shown in Figure 6, shows the locations of the equipment (conveyors and ovens) used in the system, where the physical twin is replicated on the CoppeliaSim simulation software.

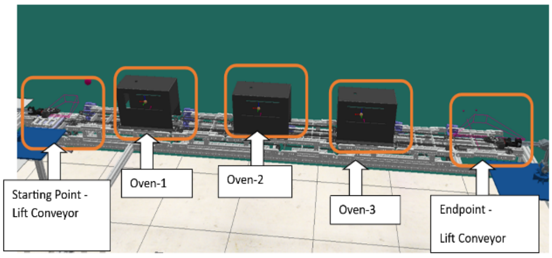

Figure 6. Simulation of the first workstation of the smart factory scale model on CoppeliaSim.

After the installation in the simulation environment, the necessary tools such as sensors and conveyor movement providers are integrated into the system with the help of simulation tools in environment. The tools are written in the Lua programming language, and they are ensured to work according to the requirements in the physical twin.

3. COMMUNICATION BETWEEN THE PHYSICAL AND DIGITAL TWIN

After the installation of the real system in the simulation environment, bidirectional data flow and control must be ensured between the real and virtual systems in order to fully establish digitalization [26]. Firstly, in order to enable data flow between the real and virtual systems, the PC on which the CoppeliaSim simulation runs and the PLC that enables the real system to operate must communicate. An API is written using the Python language so that CoppeliaSim can communicate with the PLC used in the physical twin. Python is a useful programming language that contains the necessary libraries of PLC and CoppeliaSim simulation. As shown in Figure 7, the PLC-PC communication and PC-CoppeliaSim communication are carried out using the necessary software and hardware.

3.1. PLC–PC Communication

Mitsubishi PLC uses GX-Works3 as the program interface. In the PLC part of the communication, the SeamLess Message Protocol (SLMP) communication protocol is used on GX-Works3. In the PC part of the communication, the "HslCommunication" library of the Python language is used. The HslCommunication library is a library that supports industrial communication protocols in the Python programming language [27]. With this method, PLC and PC communication is provided on the software. In terms of hardware, the Ethernet ports are used for maintaining communication between the PLC and the PC.

3.2. PC–CoppeliaSim Communication

For the software-based communication between the PC and CoppeliaSim, the "sim" library of the Python language and the following necessary communication codes are used with sim.simxStart(.);

- connectionAddress
- connectionPort
- waitUntilConnected
- doNotReconnectOnceDisconnected
- timeOutInMs
- commThreadCycleInMs

In the CoppeliaSim, it is communicated with the simRemoteApi.start(connectionPort) code.

With this method, the PC and CoppeliaSim are communicated.

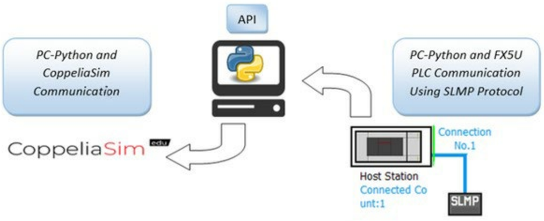

Figure 7. PLC–PC and PC–CoppeliaSim communication diagrams.

Many of the fundamental functional applications of industrial software development, including Mitsubishi PLC software, are integrated with the HSLcommunication Library, an industrial IoT-based computer communication architecture implementation. An example part of the Python code for PLC-PC and PC-CoppeliaSim communication via the HslCommunication plugin can be seen in Figure 8. In this code, first, "sim" library is imported with HslCommunication plugin and Arduino MPU is connected to PC via the serial COM8 port by determining the IP address and baud rate. Then, using the “MelsecMcNet” node of the HslCommunication plugin, the motor velocity data, temperature and proximity sensor values are read from Arduino’s pins to send commands to the client CoppeliaSim.

```
import sim, serial, time
from HslCommunication import MelsecMcNet, SoftBasic
arduino = serial.Serial(port='COM8', baudrate=9600, timeout=.1)

if __name__ == "__main__": melsecNet = MelsecMcNet("192.168.1.10", 4001)
# Connect to CoppeliaSim
clientID = sim.simxStart('127.0.0.1', 19999, True, True, 500, 5)

def send_command(command): arduino.write(f"{command:.0f}".encode())

while True:
    a=sim.simxGetFloatSignal(clientID,"velocity_data1",sim.simx_opmode_oneshot)
    send_command(a[1])
    x1 = melsecNet.ReadInt32("D0")
    x2 = melsecNet.ReadInt32('D5')
    Ps_x0=melsecNet.ReadBool("X0")
    Ps_x1=melsecNet.ReadBool("X1")
    Ps_x2=melsecNet.ReadBool("X2")
    Ps_x3=melsecNet.ReadBool("X3")
    print('Temperature 1,2,3 :')
    printReadResult(x1)
    printReadResult(x2)
    print('Proximity 0,1,2,3 :')
    printReadResult2(Ps_x0)
    printReadResult2(Ps_x1)
    printReadResult2(Ps_x2)
    printReadResult2(Ps_x3)

    if arduino.readline().decode('utf-8').strip():
      sim.simxSetStringSignal(clientID, "velocity", str(line), sim.simx_opmode_blocking)

    time.sleep(1)
```

Figure 8. A part of a long Python code with HslCommunication library for PLC–PC and PC–CoppeliaSim communication.

4. EXPERIMENTAL STUDIES AND RESULTS

After the communication infrastructure between physical and digital twin has been completed, experimental studies are carried out for three different scenario examples to demonstrate the operability of the system. The scenarios performed are listed as follows:

- Controlling the physical twin through the digital twin
- Controlling the digital twin through the physical twin
- Providing simultaneous double-sided control between the physical and the digital twin

4.1. PLC–PC Communication

As an example for the first scenario, as seen in Figure 9, when the proximity sensors used in the digital twin detect the product placed by the robot arm on the first workstation, the DC encoder motors connected to the conveyors in the physical twin are started and then the speed values of the motor are recorded according to the voltage change. The proximity sensors in CoppeliaSim determine that the DC encoder motor in the real system moves or does not move, depending on whether it detects the product or not.

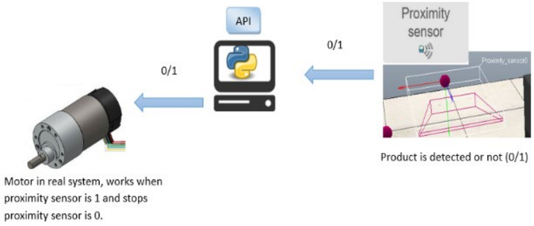

Figure 9. The relationship between the proximity sensors in the digital twin and the motors in the physical twin.

4.2. Controlling the Digital Twin Through the Physical Twin

As an example for the second scenario, as seen in Figure 10, when the proximity sensors used in the physical twin detect the product placed on the lift conveyor with the help of the robot arm, the “True” or “False” values are sent to the digital twin and then the lift conveyors in the simulation environment move or not move according to the “True” or “False” values, respectively. As shown in Figure 11 and Figure 12, the proximity sensor in the real system ensures that the lift conveyor in CoppeliaSim moves or does not move, depending on whether it detects the product or not.

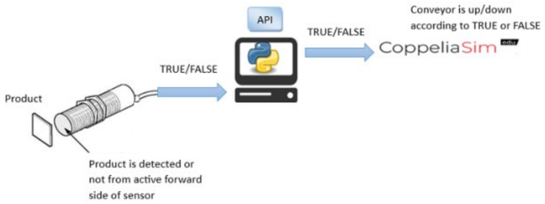

Figure 10. The relationship between the proximity sensors in the physical twin and the lift conveyors in the digital twin.

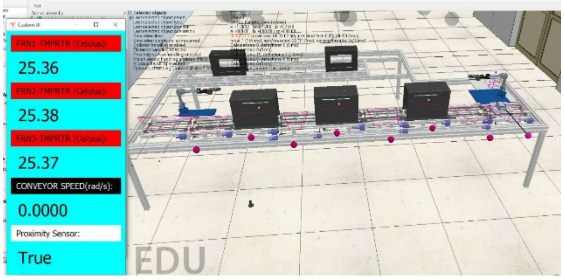
FRN1-TMPRTR (Celsius):
25.36
FRN2-TMPRTR (Celsius):
25.38
FRN3-TMPRTR (Celsius):
25.37
CONVEYOR SPEED(rad/s):
0.0000
Proximity Sensor:
True

Figure 11. The lift conveyor in CoppeliaSim is controlled with the proximity sensors in the real system. In the "True" state, the lift conveyor moves to the upward position.

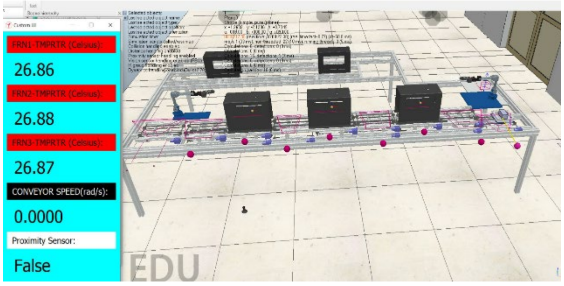
FRN1-TMPRTR (Celsius):
26.86
FRN2-TMPRTR (Celsius):
26.88
FRN3-TMPRTR (Celsius):
26.87
CONVEYOR SPEED(rad/s):
0.0000
Proximity Sensor:
False

Figure 12. The lift conveyor in CoppeliaSim is controlled with the proximity sensors in the real system. In the "False" state, the lift conveyor moves to the downward position.

4.3. Double-Sided Control Between the Physical and the Digital Twin

The physical twin can be controlled via the digital twin and the digital twin can also be controlled via the physical twin, as shown in Figure 13. As an example for the third scenario, the example scenarios applied above are implemented simultaneously in a single scenario. In this scenario, the temperature data obtained from the temperature sensors in the oven lines, the speed data obtained from the DC encoder motor connected to the conveyor, and the data obtained from the proximity sensors used to detect the positions of the marble slabs received from the physical twin are transferred to the CoppeliaSim interface, and the digital twin is operated according to these real data. The interface shown in Figure 14 is created using the "xml" language in CoppeliaSim. On the other hand, the speed values of the motors can be adjusted as desired via the digital twin, and in this way the speeds of the real motors in the physical twin can be controlled. Thus, instant status monitoring and simultaneous double-sided control of the physical and digital twin is provided.

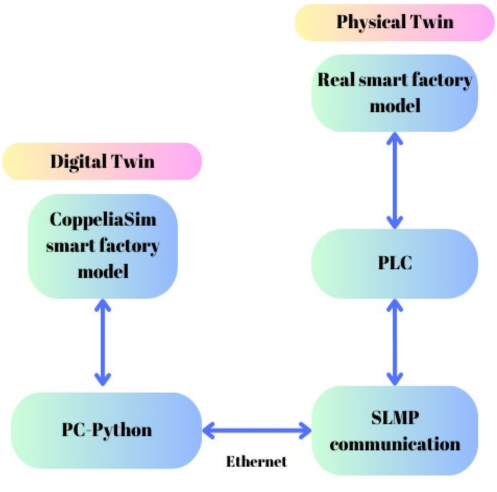

Figure 13. Double-sided control block diagram.

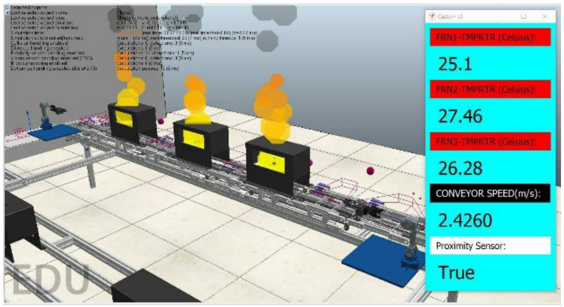
FRN1-TMPRTR (Celsius):
25.1
FRN2-TMPRTR (Celsius):
27.46
FRN3-TMPRTR (Celsius):
26.28
CONVEYOR SPEED(m/s):
2.4260
Proximity Sensor:
True

Figure 14. Implementation of the digital twin in CoppeliaSim with the real oven temperatures, motor speeds and proximity sensors' data coming from the smart factory scale model.

4.4. DC Encoder Motor Data

Via the CoppeliaSim interface, the voltage values in the range of 0-12 Volts are applied to the DC encoder motor used in the physical twin, and the speed data of the motor in radians per second and the current data drawn by motor can be plotted as shown in Figure 15.

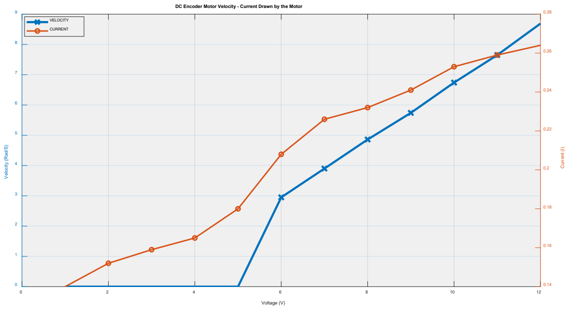

Figure 15. Speed change of the DC encoder motor in rad/s and current drawn by the motor when the voltage is applied in the range of 0-12V.

4.5. Oven Temperature Data

The real temperature values in the oven lines are sent to the digital twin via the physical twin and displayed on the CoppeliaSim interface. As shown in Figure 16, the temperature values recorded every second show the temperature values in the laboratory environment of the smart factory model.

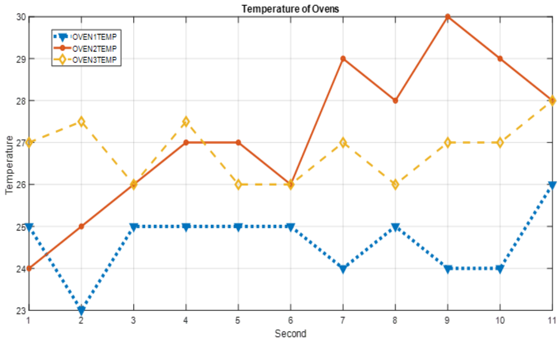

Figure 16. Recorded temperature data of three ovens transferred from the physical twin to the digital twin.

5. EVALUATION AND DISCUSSION

While most digital twin studies in the literature [1-8] are carried out either by transferring data from the virtual system to the physical system only, as we did in the first experimental scenario, or from the physical system to the virtual system only, as we did in the second experimental scenario, we can simultaneously exchange data between the real and virtual systems, as we have shown in the third experimental scenario in this study. To exemplify this, with the two studies we conducted in the third experimental scenario, we simultaneously transferred the motor velocity data from the virtual system to the real system and loaded the temperature data from the real system to the virtual system. As seen in Figure 15, although the DC motor draws a maximum current of 0.18A until 5V voltage is applied, the rotation speed of the motor shaft is 0rad/s as seen from the encoder data. When the voltage is applied to the motor step by step from 5V to 12V, the current consumed increases gradually from 0.14A to 0.28A and the rotation of the motor shaft accelerates from 0rad/s to 9rad/s in approximately linear. According to the different voltage values sent from the CoppeliaSim interface, the movement of the real motor in the physical twin could be observed simultaneously. On the other hand, as seen in Figure 16, the values of three different temperature sensors were measured every second between 23 and 30 centigrade degrees for 11 seconds. The sensor data on the real physical twin was sent to the CoppeliaSim interface and temperature changes could be observed simultaneously. Although it is stated that double-sided data transfer between the physical and digital twins is performed simultaneously, there is a negligible time delay of around a few milliseconds due to the Ethernet communication between the PLC and the PC.

6. CONCLUSION

In this study, a synchronized digital twin of the first workstation of the scale model of the smart marble factory located in AFSUAM has been created in the CoppeliaSim simulation software. The communication of the digital twin with the PLC of the smart factory model and the PC on which the CoppeliaSim program is running has been carried out using an API. In order to test the overall system, three different scenarios have been implemented on the system. According to the results of these scenarios, the control of the real system through the simulation environment, the control of the simulation environment over the real system, and the control of both scenarios as a single scenario were achieved simultaneously between the virtual and the real system. We believe that with this digital twin model and double-sided simultaneous control method we have developed, different system designs for the smart factory model can be realized faster and easier.

The current study is restricted to a single marble factory workstation model. There are several obstacles to overcome before this digital twin can fully capture the scope of the smart factory, with its numerous interconnected workstations and intricate manufacturing procedures. More research is needed to create a digital twin model that fully captures the smart factory because of problems including growing data volumes, network congestion, and processing power demands. In-depth testing to assess the scalability and performance of the digital twin across the board in the smart marble plant can be carried out by addressing more comprehensive scalability testing. This means that further research can be done in the future that calls for the system to manage bigger data amounts, more devices, and more intricate physical interactions. In addition, in the smart factory model, it will be possible to detect and immediately intervene in production malfunctions that may occur in the future, as well as to monitor the real-time status of the real system and perform product performance analysis.

ACKNOWLEDGEMENT

This study was supported by Scientific and Technological Research Council of Turkey (TUBITAK) under the Grant Number 123E288. The authors thank to TUBITAK, İzmir Kâtip Çelebi University Smart Factory Systems Application and Research Center (AFSUAM) and Kur Mermer San. ve Tic. A.S. for their supports.

REFERENCES

1. Kumas, E., & Erol, S., "Endustri 4.0’da Anahtar Teknoloji Olarak Dijital Ikizler", ", Politeknik Dergisi, Vol. 24, Issue 2, Pages 691-701, 2021.

2. Kalsoom, T., Ahmed, S., Rafi-ul-Shan, P. M., Azmat, M., Akhtar, P., Pervez, Z., & Ur-Rehman, M., "Impact of IoT on Manufacturing Industry 4.0: A new triangular systematic review", Sustainability, Vol. 13, Issue 22, Pages 12506, 2021.

3. Dilibal, S., and Şahin, H., "Isbirlikci endüstriyel robotlar ve dijital endustri.", International Journal of 3D Printing Technologies and Digital Industry Vol. 2, Issue 1, 86-96, 2018.

4. Tao, F., Zhang, H., Liu, A., & Nee, A. Y., "Digital twin in industry: State-of-the-art", IEEE Transactions on industrial informatics, Vol. 15, Issue 4, Pages 2405-2415, 2018.

5. Tao, F., Xiao, B., Qi, Q., Cheng, J., & Ji, P., "Digital twin modeling", Journal of Manufacturing Systems, Vol. 64, Issue 1, Pages, 372-389, 2022.

6. Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., & Calinescu, A., "Digital twins: State of the art theory and practice, challenges, and open research questions", Journal of Industrial Information Integration, Vol. 30, Issue -, Pages 100383, 2022.

7. Liu, M., Fang, S., Dong, H., & Xu, C, "Review of digital twin about concepts, technologies, and industrial applications", Journal of Manufacturing Systems,Vol. 58, Issue 1, Pages 346-361, 2021.

8. 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, Vol. 60, Issue 1, Pages 119-137, 2021.

9. Park, K. T., Son, Y. H., Ko, S. W., & Noh, S. D., "Digital twin and reinforcement learning-based resilient production control for micro smart factory", Applied Sciences, Vol. 11, Issue 7, Pages 2977, 2021.

10.Chancharoen, R., Chaiprabha, K., Wuttisittikulkij, L., Asdornwised, W., Saadi, M., & Phanomchoeng, G., "Digital twin for a collaborative painting robot, Sensors", Vol. 23, Issue. 1, Pages 17, 2022.

11. Coppelia Robotics "Robot simulator CoppeliaSim: create, compose, simulate, any ...", https://www.coppeliarobotics.com/, February 10, 2024.

12. Ren J., Nalpantidis, L. And Andersen N. A. &. Ravn., O. "Building Digital Twin of Mobile Robotics Testbed Using Centralized Localization System," in 11th International Conference on Control, Mechatronics and Automation (ICCMA), Pages 139-145, Grimstad, 2023.

13.Open Robotics, "Gazebo", https://gazebosim.org/home, January 20, 2024.

14. Stączek, P., Pizoń, J., Danilczuk, W., & Gola, A., A digital twin approach for the improvement of an autonomous mobile robots (AMR's) operating environment-A case study. Sensors, Vol. 21, Issue 23, Pages 7830, 2021.

15. Al-Geddawy, T., "A digital twin creation method for an opensource low-cost changeable learning factory," Procedia Manufacturing, vol. 51, no. 1, Pages 1799-1805, 2020.

16. RoboDK Inc., "RoboDK: Simulator for industrial robots and offline programming", https://robodk.com/, October 15, 2023.

17. Pires, F., Cachada, A., Barbosa, J., Moreira, A. P. and P. Leitão, "Digital Twin in Industry 4.0: Technologies, Applications and Challenges," in IEEE 17th International Conference on Industrial Informatics (INDIN), Pages 721-726 Helsinki, Finland, 2019.

18. Dassault Systèmes, " SOLIDWORKS | 3D CAD Design Software & PDM Systems", https://www.solidworks.com/, February 21, 2024.

19. AutoDesk, "AutoCAD", https://www.autodesk.com/, February 21, 2024.

20. Trimble Inc., "SketchUp: 3D Design Software | 3D Modeling on the Web", https://www.sketchup.com/, January 25, 2024.

21. Magrin, C. E., Del Conte, G. And Todt, E., "Creating a digital twin as an open-source learning tool for mobile robotics," in Latin American Robotics Symposium (LARS), Brazilian Symposium on Robotics (SBR), and Workshop on Robotics in Education (WRE), Pages 13-18, Natal, 2021.

22. MathWorks, "Mathematical computing software for engineers and scientists", https://www.mathworks.com/, February 28, 2024.

23. PUC-Rio, "The Programming Language Lua", https://www.lua.org/, February 28, 2024.

24. Eaton, J. W. "GNU Octave ", https://octave.org/, February 22, 2024.

25. Rohmer, E., Singh, S. P., & Freese, M., "V-REP: A versatile and scalable robot simulation framework," in IEEE/RSJ international conference on intelligent robots and systems, Pages 1321-1326, Tokyo, 2013.

26. Nascimento, F. H., Cardoso, S. A., Lima, A. M. and Santos, D. F., "Synchronizing a collaborative arm's digital twin in real-time," in Latin American Robotics Symposium (LARS), Brazilian Symposium on Robotics (SBR), and Workshop on Robotics in Education (WRE), Pages 230-235, Salvador, 2023.

27. Richard Hu, "HslCommunication", https://pypi.org/project/HslCommunication/, February 25, 2024.