ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20240423112303

Subject review
Received: 2024-04-23, Accepted: 2024-04-30

Development and Control of Virtual Industrial Process using Factory IO and MATLAB

Goran Munđar*, Miha Kovačič, Uroš Župerl

Abstract: In today's rapidly evolving business landscape, the strategic adoption of virtual manufacturing methods has emerged as a key driver for companies seeking to streamline operations and expedite product launches in a cost-effective manner. This progressive approach involves the creation of a synthetic and interconnected environment, empowered by advanced software tools and systems, including Virtual Reality and Simulation technologies, tailored to optimize industrial processes. Our methodology employs a unique combination of two simulation software tools: Factory I/O for process development and MATLAB for control program implementation. Furthermore, we explore the use of the Modbus TCP/IP communication protocol as the framework for seamless interaction between these software tools during simulation. This research presents practical insights into the transformative potential of virtual manufacturing, showcasing its real-world application in enhancing operational efficiency and agility within industrial settings.

Keywords: Factory I/O; MATLAB; Modbus TCP/IP; simulation technologies; virtual manufacturing

1 INTRODUCTION

In the contemporary era of industrial advancement, the shift towards digitalization and virtual integration marks a pivotal transformation in manufacturing. The advent of Virtual Manufacturing (VM) embodies this transition, heralding a paradigm where operational efficiencies are enhanced and product development cycles are significantly expedited. This paper explores the strategic deployment of VM through a methodology that integrates software tools Factory I/O and MATLAB. While the use of TCP/IP protocol is widely recognized in various communication setups between hardware and software systems, this research delves into its tailored application within the realm of virtual manufacturing simulations, an area less explored and ripe for innovation [1].

VM emerges as a critical enabler for agility and cost-effectiveness in product launches, pivotal for reducing time-to-market in today's dynamic business landscape [2]. By leveraging Virtual Reality (VR) and Simulation technologies, VM allows for the creation of detailed digital twins, enabling predictive analytics, process optimization, and precise decision-making [3]. The integration of Factory I/O with MATLAB, underpinned by the Modbus TCP/IP protocol, though recognized in various technological applications, is leveraged in unique ways specific to virtual manufacturing scenarios. This combination not only exemplifies VM's practical application but also underscores its potential for scalable implementations. To support our approach, we reference similar applications across various industries, demonstrating how TCP/IP methodology improves operational efficiency and agility [4], [5]. In the automotive sector, TCP/IP is extensively utilized to streamline communication across global manufacturing sites, enhancing coordination and efficiency. Parziale et al. [6] explore its broad applications, highlighting its essential role in automotive manufacturing. Da Silva and Shih [7] discuss the use of TCP/IP in automotive automatic manufacturing systems through an ES/CPS architecture that improves the efficiency and reliability of data communications within production lines. Steffen et al. [8] provide insights into an IP-based network architecture for in-car systems, underscoring TCP/IP's potential to support innovative applications. Similarly, Bohuslava et al. [9] examine how TCP/IP facilitates the dynamic control of robotic cells in automotive manufacturing, playing a critical role in adopting Industry 4.0 technologies.

As the manufacturing sector navigates the complexities of digital transformation, our research provides valuable perspectives on leveraging conventional tools in novel configurations, supporting the evolution toward more dynamic and cost-effective manufacturing processes. This comprehensive analysis underscores the transformative potential of integrating established technologies in novel ways, enriching the discourse on virtual manufacturing and its capabilities within the industrial sector.

2 METHODOLOGY

In our study, we employ a comprehensive methodology to explore the integration of VM tools, specifically focusing on Factory I/O and MATLAB. The primary objective is to showcase how these integrations can significantly improve operational efficiency and agility in industrial settings. In the selection of simulation tools, Factory I/O is chosen for its versatility in simulating real-world industrial processes and its compatibility with a diverse range of control systems. It serves as the primary environment for the development and visualization of manufacturing processes [10]. MATLAB is selected for its advanced computational capabilities, particularly in algorithm development, data analysis, and visualization. Within our methodology, MATLAB is utilized to implement control logic and algorithms that effectively manage the simulated manufacturing processes within Factory I/O [11].

As for the communication protocol, Modbus TCP/IP is identified as the preferred choice due to its widespread adoption in industrial applications and its support for both Factory I/O and MATLAB. This selection is based on the necessity for a reliable and efficient means of data exchange between the simulation tools.

Within the scope of our research, we have developed two distinct systems to demonstrate the capabilities of our methodology. The first system, referred to as the test system, was specifically designed for showcasing the Modbus TCP/IP communication protocol. This system serves as a comprehensive illustration of the effective exchange of data between Factory I/O and MATLAB, highlighting the reliability and efficiency of the chosen communication protocol. Additionally, we have created a second system, which we term the industrial system, to further underscore the versatility of our approach. This industrial system emulates a more complex scenario, involving the assembly of two components into a single part. Through this system, we demonstrate that MATLAB can effectively handle intricate control programs, showcasing its proficiency in algorithm development, data analysis, and visualization within the context of more sophisticated manufacturing processes.

3 TEST SYSTEM DEVELOPMENT

In this chapter, we demonstrate the development of our test system, dedicated to demonstrating the effectiveness of the Modbus TCP/IP communication protocol. We detail the design and implementation of the system, offering insights into its structure, functionalities, and the successful outcomes achieved through seamless data exchange between Factory I/O and MATLAB.

3.1 System Development in Factory IO

In the software tool Factory I/O, three components were utilized for the development of the test system:

- Start Button,
- Light Indicator,
- Electric Switchboard.

The arrangement of these components within Factory I/O is depicted in Fig. 1. Specifically, the "Start Button" component serves as the digital input for the test system, the "Light Indicator" represents the digital output, and the "Electric Switchboard" is employed to mount the components to prevent them from floating in the air.

3.2 Definition of Communication Protocol

To establish communication between the virtual system created in Factory I/O and the control program developed in Matlab, we utilized the Modbus TCP/IP communication protocol. In Factory I/O, we defined the communication mode as "Modbus TCP/IP Server". The key steps for establishing communication involved configuring the IP address, port, Slave ID, network adapter, and specifying the type and number of inputs or outputs. The communication configuration in Factory I/O is illustrated in Fig. 2.

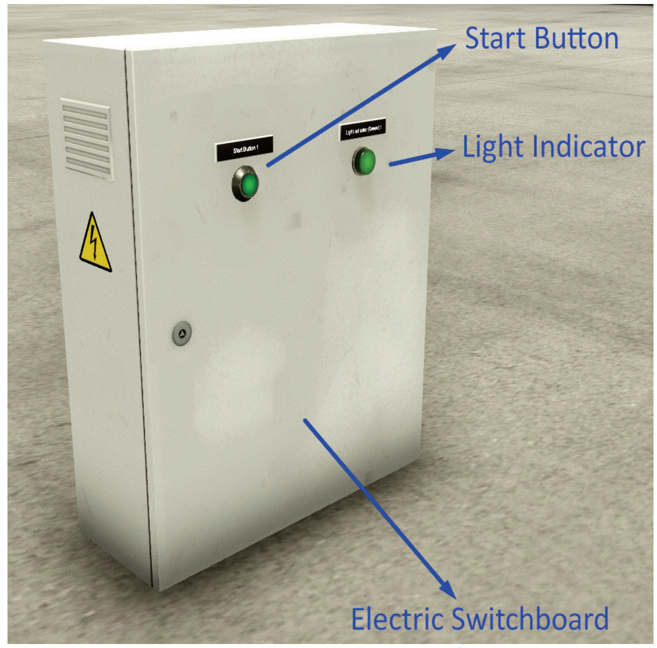

Figure 1 Test system in Factory IO

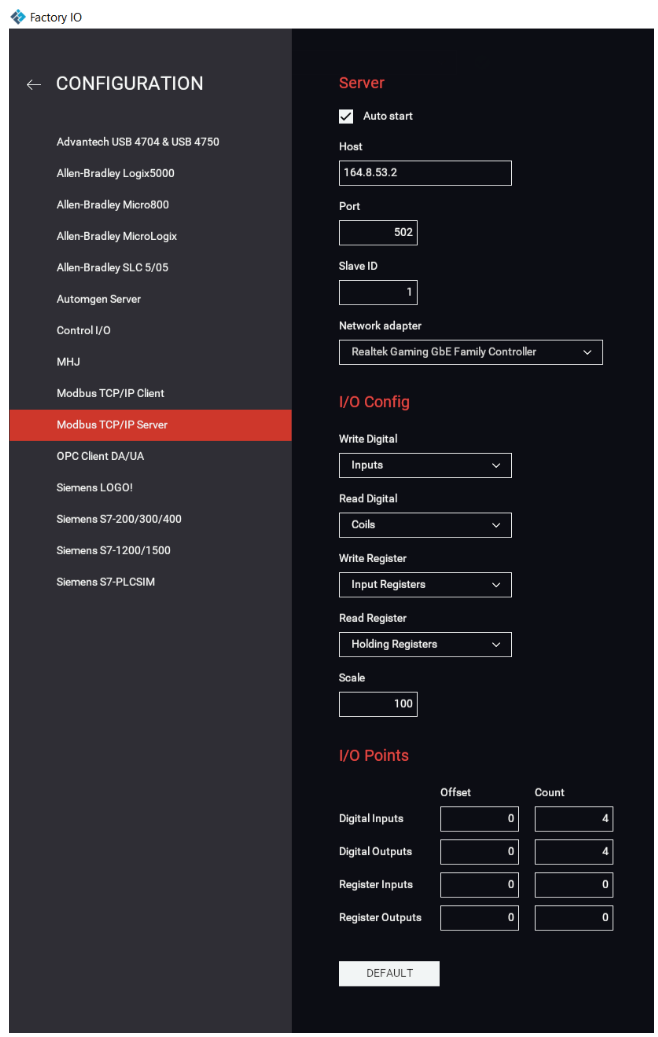

Figure 2 Communication configuration in Factory IO

3.3 Definition of I/O Component Addresses

After the successful configuration of the communication protocol, it is necessary to determine the addresses of the digital input and digital output. The digital input "Start Button" was defined as "Input 0", and the digital output "Light Indicator" was defined as "Coil 0". The described connection is shown in Fig. 3.

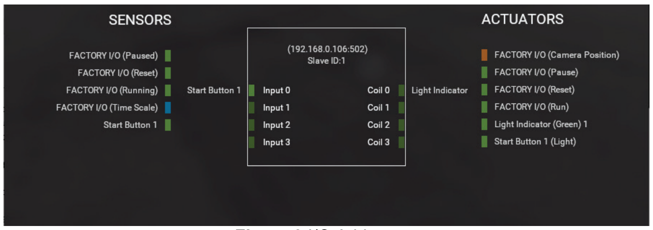

Figure 3 I/O Addresses

3.4 Development of a Control Program in MATLAB and Simulation

To effectively devise a control program within the MATLAB software tool, integration with the "Industrial Communication Toolbox" [12] was imperative. This toolbox facilitates connectivity to Modbus servers and offers diverse functionalities, including the creation of interface objects and functions for data reading, writing, and manipulation.

For the establishment of Modbus communication, the initial step involves creating an interface object using the "modbus" function. Essential data such as "Transport", "Device IP Address", and "Port" must be inputted for proper functionality. Upon successful communication setup, a control program is developed utilizing "read" and "write" functions. The control program follows a sequence: first, the Modbus connection is configured using the Modbus function. Then, in an iterative loop, the status of the Start button in Factory I/O is continually assessed through the read function. Upon detecting a pressed Start button, the write function is employed to activate the "Light Indicator" in Factory I/O. The outlined control program for the test system is provided below:

```
% MODBUS TCP connection with Factory IO
Server
m = modbus("tcpip",'192.168.0.106',502)

% Infinite loop
while true
    %read "button" state
    inputReg0 = read(m,"inputregs",1)

    %turn light indicator on/off
    if (inputReg0 == 1)
        write(m,"coils",1,1)
    else
        write(m,"coils",1,0)
    end
end
```

Post the development of the control program, we conducted a simulation of its operation. Initiation involved running the simulation in Factory I/O followed by launching the control program in MATLAB. The simulation executed successfully, allowing us to toggle the "Light Indicator" on and off by pressing Start button in Factory IO.

4 PRODUCT ASSEMBLY SYSTEM

In this chapter, the creation of a more complex virtual system for product assembly and the development of a control program to manage the developed system will be described.

4.1 Technological Requirements

The primary function of the developed system is to transport two components (bases and covers of the product) along conveyor belts to a workstation. Upon reaching the workstation, the components undergo positioning to the appropriate location. Subsequently, utilizing a two-axis manipulator, the product covers are grasped using a pneumatic gripper and transferred to the product bases, where the assembly takes place. Following a successful assembly, the completed product exits the workstation, making room for new components to undergo the assembly process.

4.2 Development of the Virtual System in Factory I/O

In the creation of the described virtual system in Factory I/O, various actuators, sensors, and structural components were employed. The actuators utilized include:

- 2 Belt Conveyors with a length of 4 m
- Two-Axis Pick & Place manipulator
- 2 Right Positioning pneumatic manipulators
- Digital Display
- 3 Light Indicators.

Ensuring the successful operation of the system required the incorporation of sensors. The following sensors were integrated into our system:

- 3 Diffuse Sensors for detecting elements on the conveyor belts
- 8 limit switches – 2 installed on each pneumatic actuator
- 2 sensors for detecting movement in individual axes of the two-axis manipulator
- 3 control buttons for system operation
- Emergency switch for immediate shutdown.

In addition to actuators and sensors, the constructed system includes structural components such as sensor mounts, an electrical enclosure, an enclosure mount and safety fences. While these components do not directly affect the system's operation, they contribute to a more aesthetically pleasing and illustrative representation of how such a system might appear in a real industrial environment.

Finally, components for element generation and removal were added to the system. Two Emitter components were employed for element generation—one for generating product bases and the other for generating covers of the final products. The system also includes a Remover component, responsible for deleting assembled products when they reach the end of the conveyor belt.

Fig. 4 illustrates the complete virtual industrial system created in the Factory I/O software tool.

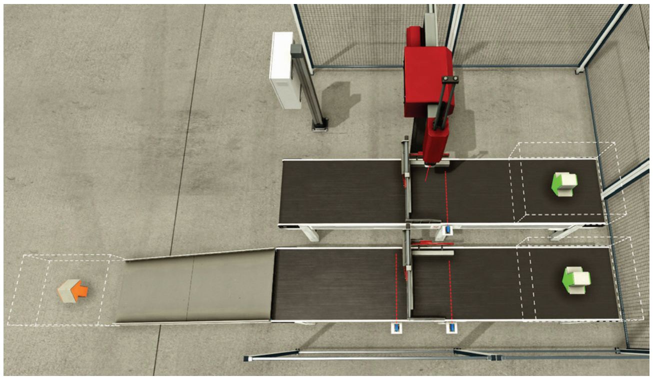

Figure 4 Virtual industrial system

4.3 Definition of I/O Addresses

The addresses of actuators and sensors were defined using the same approach as outlined in the test system described in Chapter 3.3. A total of 16 digital inputs, 12 Coils (digital outputs), and 1 Output Holding Register were employed. The Output Holding Register serves the purpose of controlling the digital display, responsible for monitoring the number of assembled products. Fig. 5 presents the definition of all utilized inputs and outputs in the Factory I/O software tool.

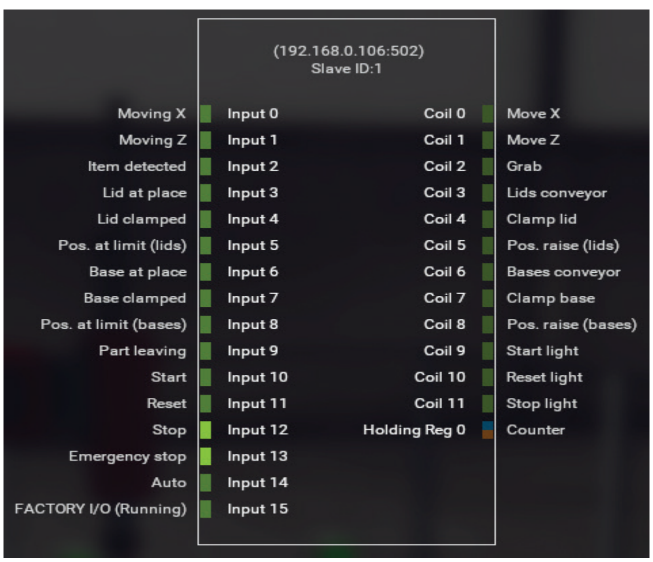

Figure 5 Definition of I/O Addresses

4.4 Development of Control Program and Simulation

The development of the control program in MATLAB followed a similar methodology as outlined in the test system described in Chapter 3.3. To construct the control program, we utilized the "modbus" function for establishing the Modbus TCP connection, the "read" function for retrieving sensor values from Factory I/O, and the "write" function for activating and deactivating actuators.

Once the control program was meticulously crafted, a comprehensive simulation of the system's functionality was conducted. This simulation aimed to validate the effectiveness and efficiency of the control logic in managing the virtual industrial system. The successful execution of the simulation affirmed the reliability of the control program, showcasing its capability to respond appropriately to dynamic inputs and successfully manipulate the virtual components within Factory I/O. System during simulation is shown on Fig. 6.

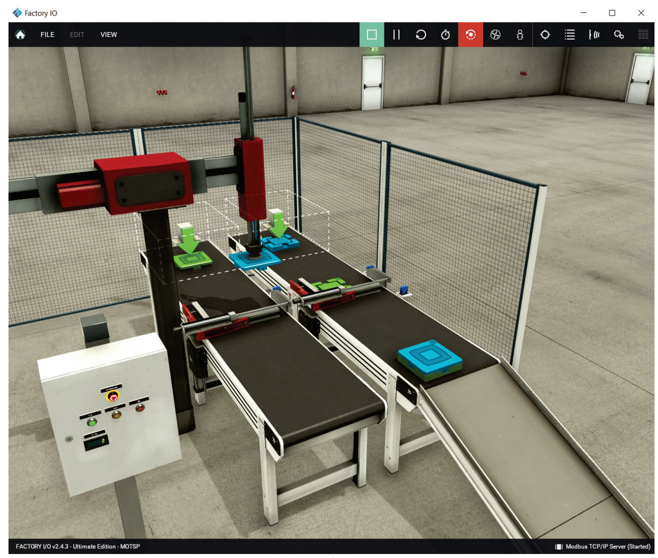

Figure 6 Simulation

5 RESULTS

The simulation of both the test system and the more complex industrial assembly system proved to be highly successful. The Modbus TCP communication protocol demonstrated sufficient speed, ensuring the smooth operation of the simulations without any hindrance.

Following the successful simulations, we conducted further tests to evaluate the performance of communication and control programs at higher simulation speeds. Factory I/O allows simulation speeds to be adjusted from 0.1x to 4x. We systematically tested the functionality at all speeds. The simulation operated seamlessly and could be effectively controlled even at significantly elevated speeds. The results indicated that both the communication protocol and the control program maintained robust functionality, affirming their efficacy even in dynamic and high-speed simulation environments.

Furthermore, the real-time monitoring capabilities of the control program were evaluated during the simulations, showcasing its ability to react promptly to changes in the virtual environment. This responsiveness is a key attribute for real-world applicability, demonstrating the adaptability of the developed system to dynamic manufacturing scenarios.

In summary, the results highlight the successful implementation and robust performance of the proposed virtual manufacturing framework, validating its capability to handle intricate communication and control tasks, even in dynamic and accelerated simulation scenarios.

6 CONCLUSION

This work has presented the process of developing a virtual industrial system using the Factory I/O software tool and creating a control program within MATLAB. The seamless communication between these programs was established through the Modbus TCP communication protocol, selected for its widespread adoption and ease of integration, despite the availability of more secure protocols, which could be considered in future enhancements to improve cybersecurity measures.

MATLAB, as a powerful programming environment, was chosen over traditional PLCs for its superior computational capabilities, enabling more complex simulations and sophisticated data analysis. This decision allows for straightforward control of the constructed systems within Factory I/O, showcasing the potential of high-level software tools in reducing the time and costs associated with physical prototyping. Furthermore, the integration of MATLAB enhances the system's flexibility and scalability, proving essential for addressing complex industrial challenges that PLCs alone might not efficiently manage.

The procedural framework outlined in this work showcases how digital twins of real systems and corresponding control programs can be efficiently developed. This methodology streamlines the creation of control programs, simultaneously reducing costs by facilitating simulation-based testing without the need for physical components. This not only minimizes the potential expenses associated with damage to real system elements but also mitigates the risk of harm to personnel.

The developed system is amenable to further enhancements. MATLAB offers extensive extensions, such as the Simulink® PLC Coder™ [13], enabling the creation of control programs in the Ladder programming language within MATLAB. While automatically generated Ladder diagrams from Simulink might require optimization to remove extraneous elements, they provide a foundational code that can be refined for practical application in PLCs. This could potentially facilitate easier learning of programming and control program development in educational settings, where programmers may lack sufficient experience to test their control programs on real systems.

In essence, the presented approach demonstrates the viability of employing virtual manufacturing frameworks for the design and testing of control programs, providing a valuable alternative to traditional methods. The combination of Factory I/O and MATLAB, coupled with the potential for further expansion, presents a robust solution for the development, testing, and educational application of control systems in diverse industrial contexts.

Acknowledgements

This research was funded by the Slovenian Research and Innovation Agency; grant number L2-3173.

7 REFERENCES

[1] Swales, A. (1999). Open modbus/tcp specification. *Schneider Electric*, *29*(3), 19.
[2] Soori, M., Arezoo, B. & Dastres, R. (2023). Virtual manufacturing in industry 4.0: A review. *Data Science and Management*. https://doi.org/10.1016/j.dsm.2023.10.006
[3] Salah, B., Abidi, M. H., Mian, S. H., Krid, M., Alkhalefah, H. & Abdo, A. (2019). Virtual reality-based engineering education to enhance manufacturing sustainability in industry 4.0. *Sustainability*, *11*(5), 1477. https://doi.org/10.3390/su11051477
[4] Munđar, G. (2022). Development and control of a virtual industrial process for assembling computers. *IOP Conference Series: Materials Science and Engineering, 1268*(1), p. 012017. https://doi.org/10.1088/1757-899X/1268/1/012017
[5] Wang, B. (2020). Research and Development of Virtual Manufacturing. In *International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019: Applications and Techniques in Cyber Intelligence, 7*, 1523-1529. Springer International Publishing. https://doi.org/10.1007/978-3-030-25128-4_188
[6] Parziale, L., Liu, W., Matthews, C., Rosselot, N., Davis, C., Forrester, J. & Britt, D. T. (2006). TCP/IP tutorial and technical overview.
[7] Shih, I. J. L. & da Silva, G. C. (2018). Proposal of ES/CPS Architecture for Automotive Automatic Manufacturing Systems. *Journal of Production and Automation (JPAUT) ISSN 2595-9573*, *1*(1), 19-34.
[8] Steffen, R., Bogenberger, R., Hillebrand, J., Hintermaier, W., Winckler, A. & Rahmani, M. (2010, May). Design and realization of an ip-based in-car network architecture. *The 1st International ICST Symposium on Vehicular Computing Systems*.
[9] Bohuslava, J., Martin, J. & Igor, H. (2017). TCP/IP protocol utilisation in process of dynamic control of robotic cell according industry 4.0 concept. *The IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000217-000222. https://doi.org/10.1109/SAMI.2017.7880306
[10] Factory I/O – Next-Gen PLC Training. https://factoryio.com/
[11] Matlab, S. (2012). Matlab. *The MathWorks, Natick, MA*.
[12] Industrial Communication Toolbox. https://www.mathworks.com/products/industrial-communication.html
[13] Simulink® PLC Coder™ User's Guide. https://www.mathworks.com/help/pdf_doc/plccoder/plc_st_ug.pdf

Authors' contacts:

Goran Munđar, M.Sc. Mech. Eng.
(Corresponding author)
University of Maribor, Faculty of Mechanical Engineering,
Smetanova 17, 2000 Maribor, Slovenia
goran.mundjar@um.si

Miha Kovačič, PhD, Associate Professor
Štore Steel d.o.o.
Železarska 3, 3220 Štore, Slovenia
University of Ljubljana, Faculty of mechanical engineering,
Aškerčeva cesta 6, 1000 Ljubljana, Slovenia
miha.kovacic@store-steel.si

Uroš Župerl, PhD, Associate Professor
University of Maribor, Faculty of Mechanical Engineering,
Smetanova 17, 2000 Maribor, Slovenia
uros.zuperl@um.si