VIRTUAL COMMISSIONING AS PART OF NEW MACHINE IMPLEMENTATION

ALES JANOTA[1], MARIAN HRUBOS[1], PAVOL KUCHAR[1], JURAJ ZDANSKY[1]

[1]Department of Control and Information Systems, Faculty of Electrical and Information Technologies, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia

DOI: 10.17973/MMSJ.2024_12_2024090

ales.janota@uniza.sk

The article provides practical insights into the development of a specialized machine for automating the assembly of components based on customer-specified parameters. The initial design undergoes a risk analysis in accordance with the EN ISO 13849 standard, ensuring safety compliance. For the software aspect, we create sequences and algorithms for the machine's individual components using standardized formats within the TIA Portal software. To expedite programming and real-time debugging, a virtual commissioning approach utilising the fe.screen-sim program is employed. This allows us to address any issues or deficiencies in the virtual model, which mirrors the actual machine, optimizing the time needed for final physical adjustments.

KEYWORDS

Virtual commissioning, automation, simulation, PLC, risk analysis

1 INTRODUCTION

The complexity of modern manufacturing systems is increasing with growing digitization, heightening the demands on design, engineering, and operations. This also creates conflicts among cost, time, and quality requirements [Oppelt 2016]. The concept of Industry 4.0 has brought several transformative advancements to manufacturing and industrial processes. One of them is a digital twin which plays a pivotal role in digitization and optimization. Virtual Commissioning (VC) means performing the tests on the digital twin of a plant instead of on the plant itself. The terms 'virtual commissioning' and 'digital twin' are sometimes used interchangeably, but fundamental differences exist between them (Tab. 1).

Digital Twin	Virtual Commissioning
Focuses on creating a real-time, digital replica of a system or process for continuous monitoring and optimization	Focuses on testing and validating control systems and automation logic before deployment
Is used during the entire lifecycle of the system, from design to operation and maintenance	Is primarily used during the pre-deployment phase (before the physical system is operational)
Is connected to real-time data from the physical system	Often works with simulated data
Monitors performance, optimizes and performs predictive maintenance	Reduces commissioning time and ensures proper control logic

Table 1. Differences between digital twins and virtual commissioning

This paper deals with an integrated VC, where the automation software is continuously tested against a virtual plant model during the overall engineering phase. Deploying all the Industry 4.0 technologies and moving the enterprise to the Smart Factory level is costly, time-consuming and personnel intensive [Barton 2022]. VC enables developers to validate the full operation of new systems before they materialize in the physical environment [Vermaak 2017]. However, VC is still viewed as an additional step in the engineering process. An overview of scientific approaches and use cases for VC is given in [Lechler 2019]. Typical use cases cover, for example, automating the validation tests of the manufacturing tool that holds the parts during the welding operations [Bartz 2021]; VC implementation in industrial painting process [Abassi 2022]; control of a robotic cell [Fernandez 2019]; robotic systems in aircraft industry [Ismail 2023]; scalable production systems in the automotive industry [Kampker 2020]; VC applications in transportation [Novak 2017]; automated guided vehicles [Allmacher 2019]; and others. A comprehensive state-of-the-art summary of the VC topic is given by [Striffler 2023]. In VC, where physical system testing is replaced by digital simulation, modularity plays a crucial role. It serves as a tool for creating flexible and efficient production systems [Svetlik 2013], enabling detailed, step-by-step verification. This approach reduces the risk of errors, minimizes rework, and accelerates the overall deployment of production systems.

Generally, applying VC is easier in the case of discrete event systems, while the number of applications for continuous industrial processes is smaller [Fratczak 2015]. During the commissioning phase of new production systems, this approach can significantly reduce the 'time to market' by 20-50% [Brazina 2020], however, some sources estimate saved time even up to 75% [Fratczak 2015]. Some professional sources attempt to provide an economic justification for VC in the automation industry [Shahim 2016] or quantify economic effectiveness when using a digital twin [Lian 2022].

2 DESIGN

It is crucial for automation production systems to ensure safe, correct, and productive operation. This article presents some practical experience in the design, development, and implementation of a specialized machine that automates the manufacturing process of an electronic unit for controlling air flow in the car (a particular producer may not be intentionally identified because of its privacy regulations). Customer requirements cover placing electronic components into a product's protective casing, and quality control of all inserted parts from both electronic and pressure perspectives. We utilize a modern programming approach through VC with the fe.screen-sim software, linking the project developed in the TIA Portal with a virtual model of the CTMA (Continuous Testing and Model-based Automation) product. This approach streamlines the design and implementation processes of industrial systems by allowing functionality to be simulated and tested in a digital environment. Thus we can identify and debug algorithm errors in real time. The results obtained from comparing the simulated and actual machine cycles serve as a reference point for evaluating the method's applicability to other specific projects.

2.1 Description of the process

The product's core is a Printed Circuit Board (PCB), which handles the control of actuators, processes data from sensors, and communicates via the CAN interface with the car's computer control unit. The PCB cover protects the control unit from external influences and mechanical damage. Additional moulded sealing is applied to eliminate the effects of moisture. To connect the product's control unit to the surroundings, several contact pins need to be installed into the cover – pins for communication with the CAN bus, grounding the PCB, and contacts for controlling the poles of the DC motor.

Our project aims to design, develop, and implement a special monitoring machine used for quality control of components, their subsequent assembly, and final testing of the fully assembled product. The semi-automatic machine requires at least two human operators to insert components and remove the finished product from the conveyor (Fig. 1). This approach emphasizes integrating continuous testing methodologies within model-based automation systems. It allows for ongoing system performance and functionality assessment throughout the development and commissioning phases.

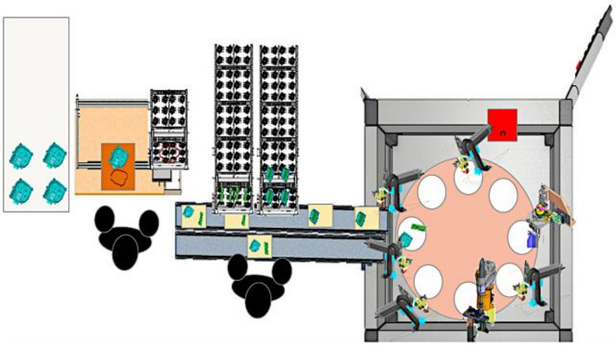


Figure 1. Design of the CTMA station

The machine consists of multiple stations that monitor selected parts of the production process. Each station has a unique code designation and is programmed as a separate block (module):

- Station 0001 (Fig. 2 left) checks the correctness of the inserted components, including the gasket in the sensor cover groove (detecting an error blocks mold production and halts any other operation on it).

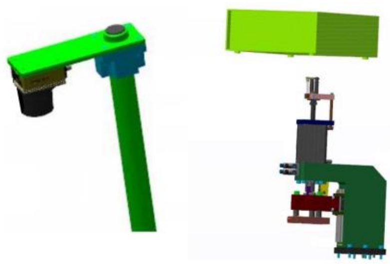


Figure 2. Station 0001 (on the left) and 0002 (on the right)

- Station 0002 (Fig. 2 right) tests the tightness of the sensor cover with the gasket by measuring the differential pressure (depending on the test result, the item is evaluated as either acceptable "Ok" or unacceptable "nOk").
- Station 0003 (Fig. 3 left) can ionize the sensor cover (if excessive static charge is detected on the cover after the tightness test, the station activates the process of removing it using an ionizing device).
- Station 0004 (Fig. 3 right) correctly positions the Printed Circuit Board Assembly (PCBA) and presses it into place. Position tolerance is at least 1.8 ± 0.25 mm, and press force ranges from 400 to 900 N. If the maximum force is exceeded or the final position is outside the tolerance, the item is deemed unacceptable and marked as "nOk" in the system.
- Station 0007 (Fig. 4 left) checks the profile of the assembled product, which is automatically matched with item information from the Manufacturing Audit and Quality System.
- Station 0010 (Fig. 4 right) tests the electronic functionality of the PCBA.

Figure 3. Station 0003 (on the left) and 0004 (on the right)

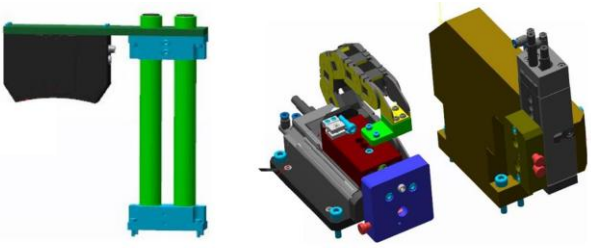


Figure 4. Station 0007 (on the left) and 0010 (on the right)

2.2 Selection of hardware components

An important step in development is the selection of suitable hardware components. The key hardware component of the proposed system is the Programmable Logic Controller (PLC)-based control unit. It must integrate a wide range of subordinate modules and guarantee reliable and safe operation of the entire system. The additional requirement is compatibility with fe.screen-sim software (F.EE 2024a), enabling us to implement virtual commissioning and optimize the development and testing process. The main selected hardware components used in the project are summarized as follows:

- PLC: S7-1500 CPU 1517-3PN [Siemens 2014].
- PCB electrical resistance testing device: Miliohmeter Resistomat 2316 [Burster 2019].
- Cover seal inspection device: Leak Tester 815 [CETA 2017].

2.3 Risk analysis

When designing PLC systems, conducting a thorough risk analysis of the hardware assembly is essential to ensure safe operation in an industrial environment. This analysis is a legal requirement and must be performed to safeguard human life and prevent accidents during the machine's operation. In our case, it is based on [EN 2023]. This standard is a key document for ensuring functional safety in machinery, focusing on the design and integration of safety-related parts of control systems (SRP/CS). It defines general principles for designing control systems that perform safety functions, regardless of the technology or energy source (e.g., electrical, hydraulic, pneumatic). The EN ISO 13849 standard classifies safety levels using a system of Performance Levels (PL), which are categorized from PL 'a' (lowest level of safety) to PL 'e' (highest level of safety). We used the safety calculator software developed by Pilz [Pilz GmbH 2013] which is product-independent. After establishing the project, we configured a new safety function 'F-CPU_Not-Halt Roundtable' with PL equivalent to 'e'. Then we could create a sub-system consisting of particular components (Fig. 5).

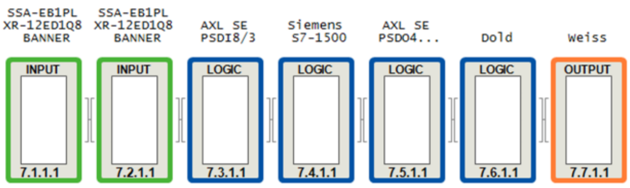
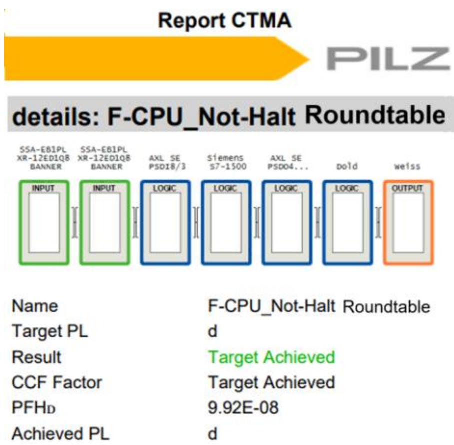


Figure 5. Display of embedded components in the system

The editor displays the components (subsystems) as areas bordered by different colours. The default colour is black, 'Input' subsystem has a green border, 'Logic' has a blue border, and 'Output' has a red border. Each component is configured according to the parameters found in its documentation sheet. It is also necessary to enter the diagnostic coverage value, which measures the effectiveness of the component's self-testing and monitoring mechanisms. For example, two-channel structures with feasibility control, feedback, and position monitoring have a diagnostic coverage of 99%. The result of the risk analysis process is documented by the generated report (Fig. 6).


Report CTMA

PILZ

details: F-CPU_Not-Halt Roundtable

Name	F-CPU_Not-Halt Roundtable
Target PL	d
Result	Target Achieved
CCF Factor	Target Achieved
PFHD	9.92E-08
Achieved PL	d

Figure 6. Generated page from the risk analysis report

3 VIRTUAL COMMISSIONING

Conventional commissioning when debugging a project can be very costly and time-consuming, especially when things don't go according to plans. Software bugs are frequently identified late in the commissioning phase. This delay results in increased costs and the need for increased hours for the programmer who could be working on another project. By modelling the programmed machine in digital form, also known as a digital twin in the initial design phase, the PLC programmer will be able to debug errors and optimize the logic of the algorithms in real-time during programming.

3.1 Software tools

Our project required the usage of several software tools:

FE.SCREEN-SIM

As part of the solution, it is necessary to connect the hardware of the special machine with the logical part and the model in the simulation. For this connection, we used the fe.screen-sim program as the core, which allowed us to connect the simulation model to a real or virtual PLC via the TIA connector software. Within the fe.screen-sim tool, several types of connections can be utilized, either on the local port of the computer or over the network where the simulation program runs on a server to which we connect. However, this model has the disadvantage of allowing the overwriting of the same simulation objects, necessitating the locking of instances we work with. The tool fe.screen-sim is based on a modular software structure to guarantee maximum flexibility. A detailed overview of the individual modules, extensions, interfaces, and software plugins is in [F.EE 2024b].

TIA PORTAL CLOUD CONNECTOR

The Totally Integrated Automation (TIA) Portal is the application that optimizes all operational, machine, and process routines. This tool connects to the TIA Portal project using the TIA Openness API. It exports global variables of outputs, inputs, timers, and variables created by the user to an Excel spreadsheet. They can be inserted into the fe.screen-sim environment as a database of all available variables whose signals will enter the simulation. When exporting, it is necessary to compile the TIA project and have all GSD (General Station Description) files inserted [Siemens 2017].

S7-PLCSIM ADVANCED

This software simulates programmable logic controllers (PLCs), enabling users to test programs without physical hardware. Unlike the basic simulator in the TIA Portal, which only allows local PLC simulations without external communication, the advanced version aims to simulate PLCs as comprehensively as possible. One key advantage is the ability to simulate multiple instances simultaneously, making it ideal for developing multi-layer systems involving several PLCs. It can also create virtual ports for communication with other devices and grants access to simulations for third-party software [Siemens 2016].

3.2 Creating a simulation programme

We used various visualization objects representing parts of simulated environments. Their overview is in Tab. 2. In the first step, it is essential to add the Floor object to the simulation with a collision model to prevent other models from collapsing. Next, the station model (a file with .asm extension) is imported into the simulation environment, and its orientation is adjusted if needed. For easier future work, we modify the visibility of certain objects. Labelling cover objects and setting the visibility of higher-level components (using the 'hide all children' setting) simplifies the simulation process and improves the clarity of hidden objects once logical elements are created (Fig. 7).
Proper simulation functioning requires adding a product model to the environment, separate from the machine assembly (Fig. 8). Since this model will acquire physical properties like mass and a collision model, it should be imported as a payload object. Without doing this, the simulation would not perform any operations in response to the movements of conveyors, tables, and robots.

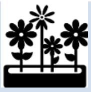
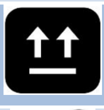
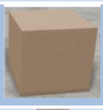
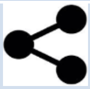
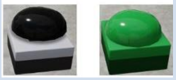
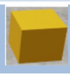
Object	Icon	3D model
Floor		
Decoration		
Payload		
Motion joint		
Sensor		
Button		
Surface		

Table 2. Overview of used visualization objects

Figure 7. Model of the station (after removing the covers)

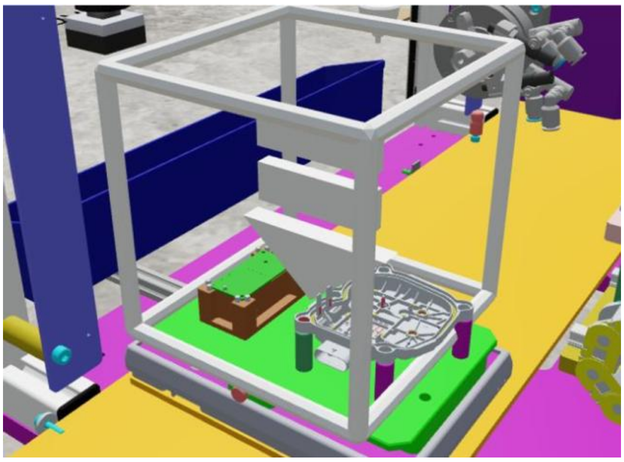


Figure 8. Inserting the cover and PCB of the product into the simulation

The CTMA working station consists of both moving and stationary parts. Since the model is imported as a decorative element, it's necessary to create clusters of moving parts and assign them movement attributes corresponding to the actual movements of the components in the real environment. The 'motion joint' function assigns these attributes, where grouped models performing the same movement are inserted. After the assignment, the movement type—either rotary or linear—is specified (Fig. 9). This setup allows the object to be manipulated later when defining the movement logic.

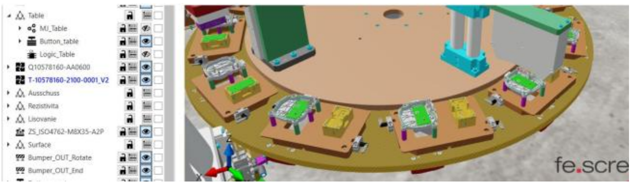


Figure 9. Configuration of rotary table movement

Special attention must be given to the insertion of sensor surfaces. Since conventional sensor models with functional detection surfaces cannot be directly inserted into the simulation, replacing the detected areas with a sensor object is necessary. In the simulation, the dimensions of the sensor's detection space are defined. When an object with a collision model enters this space, the sensor object's tag changes its binary value (Fig. 10).

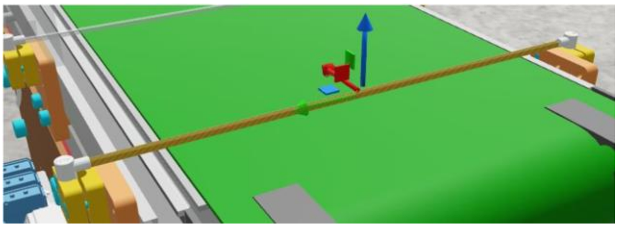


Figure 10. Product detection sensor on the conveyor belt

In the next step, logical objects must be assigned to individual groups to verify the movement logic and ensure the proper functioning of the model. We use a pre-built block from the fe.screen library to define rectilinear and rotational movements. This block is designed with logic for interaction with a PLC program, supporting off, manual, and automatic modes of operation.

The block can also be switched into the simulation mode, allowing for debugging in non-standard situations without causing hardware malfunctions for testing purposes. Another mode available is the staged failure mode, which simulates scenarios such as signal cable or power line failures. This is particularly useful when testing the machine's safety systems to ensure operator protection during an accident.

Below is a description of the block's behavior in automatic mode (Fig. 11):

1. Initialize: Upon initial startup or after a reset, all variables are initialized, and default values are assigned.

2. Control of Movement to the Working Position (*WP*): The program transitions to *Automatic Mode* if manual mode is inactive. When the command to move to the working position is selected and the movement-enabling signal is given, the valve is activated to move the cylinder toward the working position. This movement is controlled by *VelocityWP*, which is calculated based on the maximum cylinder position and the time required to reach the working position.

3. Home Position Control (*IP*): Similar to the working position movement, if the command to return to the home position is selected and the movement enable signal is given, the valve is activated to move the cylinder toward the home position. *VelocityIP* controls this movement.

4. Return to Automatic Mode: Once the cylinder has moved to either the working or home position, the system reverts to automatic mode and awaits further commands.

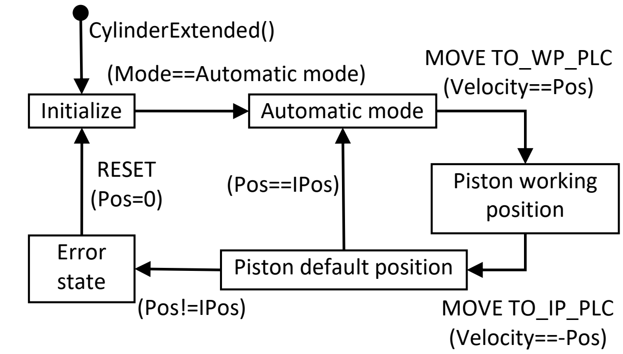


Figure 11. Statechart diagram of the block *Cylinder_Extended*

If a safety door simulation is required, we use a pre-built block called *Door* (Fig. 12). This block adds logic for controlling the safety door with output transferred to the PLC for further processing. The behavior of the *Door* block is as follows:

1. Initialize: Upon the first startup or after a reset, variables are initialized, and default values are set.

2. Door Movement Control: The program monitors the signals from the button and position sensor to control the door's movement. If the button is pressed and the door is not fully open, the door will open outward. If the button is not pressed and the door is not in the fully closed position, the door will close inward. The movement will stop if the door reaches either the fully open or fully closed position.

3. Door Status Detection: The program continuously checks the door's position to determine whether it is fully closed or open. If the door is fully closed, it is indicated as closed. Otherwise, it is marked as open.

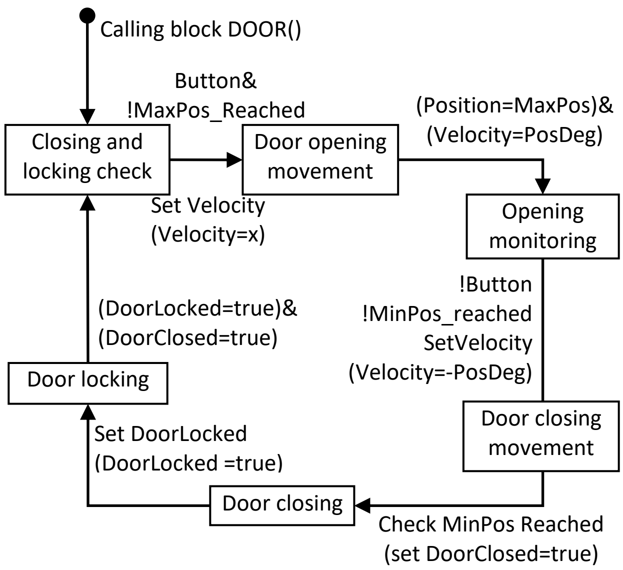


Figure 12. Statechart diagram of the block *Door*

To detect special types of materials, we use the *DetectionFlagToPayload* block (Fig. 13). This block ensures that when a material collides with a sensor object and the material tag matches the sensor tag, the event is recorded, and a logical signal is sent to the PLC for processing. The behaviour of the payload object detection module is as follows:

1. Initialize: Upon first startup or after a reset, variables are initialized, and default values are set.

2. Object Presence Detection: The program monitors the sensor signal to detect the presence of a payload object. If the sensor detects a collision, the object is recorded and removed from the simulation. The payload is then reconstructed from the saved data and placed where the sensor detected it.

3. Object Initialization: After the payload object is reconstructed, its state and position are initialized based on the stored data.

4. Error Detection: If error simulation is enabled, the payload will be placed in a simulated error state and visually represented in a distinct color.

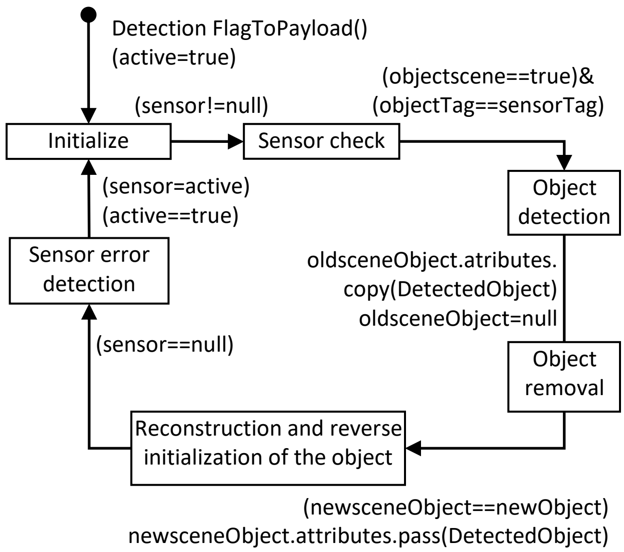


Figure 13. Statechart diagram of the block *DetectionFlagToPayload*

After assigning the logical element to the movement, the next step is to input values for the inputs and outputs of each block attribute (Fig. 14). When controlling movements through the cylinder block, buttons must be created in the simulation to switch between manual and automatic modes. This requires creating a button that writes a logical "1" to the *ManualMode* attribute. If this variable is set to logical "0," the movement will be performed automatically.

Additionally, it is important to input the physical values of the object being moved, as these will determine its behavior in the simulation. The *MovingDistanceStart* and *MovingDistanceEnd* attributes define the cylinder's positions before and after extension. The *TimeMoveToWP* and *TimeMoveToIP* attributes specify the time it takes for the cylinder to extend and retract, respectively.

The block also includes a *ForceFailure* attribute, which simulates random failures during movement, allowing the demonstration of behavior in non-standard situations. This process is applied to all other elements for which movement needs to be simulated.

After creating the logical elements of the simulation, it is necessary to connect individual blocks with signals from the PLC. Since we do not have access to a physical PLC in virtual commissioning, it is necessary to generate a table of input and output signals that will enter the blocks with movements and control their operation (Fig. 15). For the generation, we used the TiaConnector.exe software, which is part of the fe.screen program package.

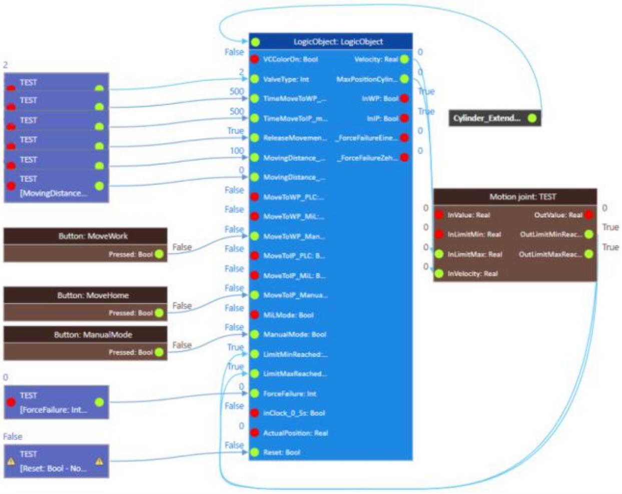


Figure 14. Logical diagram of cylinder movement in simulations

20	ToConnectWord	STD	Word	%MW1004	
21	ToConnectInt	STD	Int	%MW1006	
22	ToConnectDWord	STD	DWord	%MD1008	
23	ToConnectDint	STD	DWord	%MD1012	
24	ToConnectNetProID	STD	Conn_Ouc	%MW1016	
25	ToConnectReal	STD	Real	%MD1018	
26	MobilePanelDirectKeysIn.F1	STD	Bool	%I20.0	(BACK) Function key bit
27	MobilePanelDirectKeysIn.F2	STD	Bool	%I20.1	(NOTIFICATIONS) Function key bit
28	MobilePanelDirectKeysIn.F3	STD	Bool	%I20.2	Function key bit

Figure 15. Sample of generated PLC inputs and outputs

From the generated table, it is possible to create a communication interface block, which runs on the basis of a direct connection and periodically communicates with the TIA software and the PLC Sim v6 simulation software, thus providing updated data directly in the simulation in real time.

After a successful connection, we continued by assigning control signals to the motion control blocks according to the instructions. After connecting, the simulation is fully functional and it is possible to continue programming algorithms and sequences directly in the TIA environment.

During the simulation, we observed variations in cycle times. It is important to recognize that the simulation environment cannot fully replicate real-world conditions, where factors like increased friction in sliding mechanisms due to wear or reduced piston acceleration due to potential valve clogging may occur. For this, we ran multiple simulations, considering optimistic and pessimistic scenarios.

The average cycle times under ideal and simulated conditions with friction on the linear guides and lack of lubrication were 7.45 seconds and 9 seconds, respectively. However, the actual cycle times on the real machine are expected to be longer. This is because the simulated model does not account for all real-world factors – for instance, the model assumes a constant object speed throughout the entire movement, which is unrealistic. The object also must overcome both static and rolling friction under real conditions.

When programming the station in the Tia portal environment, we worked with version Tia portal v17. We leave out the detailed description of individual actions for understandable reasons. Anyway, as part of this process, it was necessary to make settings for message classes and monitoring for ProDiag (must be done before the integration of control panels), optional setting of languages used in visualization, insertion of PLC with control panel, hardware configuration, expansion of program structure, creation data block of variables, programming new error messages in the station and programming the register for product movement between stations.

After commissioning the machine (Fig. 16), we initially achieved a cycle time of 8.5 seconds. However, after optimizing the QR code detection system on the PCB boards, we could reduce the cycle time to 8 seconds. We identified a notable difference comparing these results with the simulated cycle time of 7.5 seconds. This discrepancy is due to several factors, including the absence of friction and the differing conditions between the simulated and real environments.

Figure 16. Seal test station [Belacek 2014]

In the simulations, the pneumatic ram of the press and the air-tightness measurement station operated at a constant speed throughout the cycle. In contrast, in the real machine, acceleration is lower due to drag, resulting in a slight cycle time reduction of 0.35 seconds during each stroke and 0.15 seconds during each fall [Belacek 2014]. Air pressure in the pneumatic system could be increased to compensate for this difference. However, Verifying that all pistons in the station can operate under the increased load is essential.

Considering the complexity of the processes within the cycle, multiple factors, such as potential delays during pressure measurements on the product cover or resistance on the PCB board can influence cycle time. These issues may cause unwanted delays and impact the overall cycle time.

4 CONCLUSIONS

The article presents the functional design results based on the customer's requirements for the CTMA station. The programming and debugging process utilized VC, with final adjustments made directly on the machine after assembly and connection to maximize time efficiency. The project also involved designing equipment to measure the quality of incoming materials and developing program blocks for managing and forwarding the measured data to the central system. A risk analysis was performed using Pilz Pascal software following the EN ISO 13849 standard.

By leveraging VC, we could compare the cycle times of the simulated and the real environments. Although simulation provides valuable insights and helps optimize the system, it's important to note that certain factors can cause deviations from reality. In our case, the discrepancy arose from the piston extension stuttering under load, resulting in a 0.5-second deviation, causing a 6.25% decrease in production efficiency.

Despite this, the comparison between virtual and real systems shows the programmer and the customer advantages, allowing for an analysis of estimated cycle times in production. Similar cycle times can be achieved in real and simulated environments with accurate actuator coefficients and proper optimization. This capability enhances planning and predictable results for the end user.

We could not quantify the time savings we achieved using VC compared to the conventional approach, however, as mentioned earlier in the paper, it should be somewhere between 20 and 50%. VC is expected to become more commonly used in the future.

ACKNOWLEDGMENTS

This research has been supported by the project APVV SK-IL-RD-23-0002: Advanced Localization Sensors and Techniques for Autonomous vehicles and Robots. The authors would like to thank S. Belacek for his valuable inputs in the field of fruitful cooperation with an industrial partner.

REFERENCES

[Abassi 2022] Abassi, A. et al. Virtual Commissioning Implementation of Industrial Painting Process. In: 2022 2nd International Conference on Computers and Automation (CompAuto), Paris, France: August 18-20, 2022, IEEE, pp. 65-68, ISBN 978-1-6654-8194-6, doi: 10.1109/CompAuto55930.2022.00019

[Allmacher 2019] Allmacher, C. et al. Virtual Reality for Virtual Commissioning of Automated Guided Vehicles. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan: March 23-27, 2019, pp. 838-839, ISBN 978-1-7281-1377-7, doi: 10.1109/VR.2019.8797981

[Barton 2022] Barton, M. et al. Identification Overview of Industry 4.0 Essential Attributes and Resource-Limited Embedded Artificial-Intelligence-of-Things Devices for Small and Medium-Sized Enterprises. Applied Sciences, 2022, Vol. 12, No. 11, 5672, ISSN 2076-3417, doi: 10.3390/app12115672

[Bartz 2021] Bartz, P. et al. Manufacturing machine virtual commissioning: Automated validation of the control software. In: 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 2021, pp. 1-5, doi: 10.1109/ICECET52533.2021.9698716

[Belacek 2024] Belacek, S. Design and implementation of CTMA product measuring station management (in Slovak). MSc. Thesis, Žilina: DCIS FEIT UNIZA, 2024.

[Brazina 2020] Brazina, J. et al. Virtual commissioning as part of the educational process. In: D. Maga & J. Hajek, eds., 2020 19th International Conference on Mechatronis – Mechatronika (ME), Prague, Czech Republic: December 2-4, 2020, pp. 1-7, ISBN 10.1109/ME49197.2020.9286613

[Burster 2019] Milliohmmeter RESISTOMAT Model 2316 Datasheet. Burster praezisionsmesstechnik gmbh & co kg Germany, 05/2019 [Online] [cit. 2024-10-10]. Available from https://www.burster.com/fileadmin/user_upload/redaktion/Documents/Products/Data-Sheets/Section_2/2316_EN.pdf

[CETA 2017] Leak tester CETATEST 815 Datasheet (in Czech). CETA Testsysteme GmbH, 02/2017 [Online] [cit. 2024-10-10]. Available from https://cetatest.com/fileadmin/pdf/datenblatt_tschechisch/CETATEST_815_cz_2017_02_14.pdf

[EN 2023] Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design (ISO 13849-1:2023). 2023 [Online] [cit. 2024-10-10]. Available from https://www.cencenelec.eu/news-and-events/news/2023/eninthespotlight/2023-05-25-en-iso-13849-1-2023

[F.EE 2024a] fe.screen-sim 3D simulation with top performance and reliability. F.EE, 2024 [Online] [cit. 2024-10-10]. Available from https://www.fescreen-sim.de/en/

[F.EE 2024b] fe.screen-sim Software modules and structure. F.EE, 2024 [Online] [cit. 2024-10-10]. Available from https://www.fescreen-sim.de/en/modules

[Fernandez 2019] Fernandez, I. A. et al. Virtual commissioning of a robotic cell: an educational case study. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain: September 10-13, 2019, IEEE, pp. 820-825, ISBN 978-1-7281-0303-7, doi: 10.1109/ETFA.2019.8869373

[Fratczak 2015] Fratczak, M. et al. Virtual commissioning for the control of the continuous industrial processes — Case study. In: 2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR), August 24-27, 2015, Miedzyzdroje, Poland: IEEE, pp. 1032-1037, ISBN 978-1-4799-8701-6, doi: 10.1109/MMAR.2015.7284021

[Ismail 2023] Ismail, O. and Fay, A. Optimizing Virtual Commissioning of a Robotic System using Process Mining and Footprints Conformance Checking. In: H. Dörksen at al., eds., 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany: July 17-20, 2023, IEEE, pp. 1-6, ISBN 978-1-6654-9313-0, doi: 10.1109/INDIN51400.2023.10217891

[Kampker 2020] Kampker, A. et al. Virtual Commissioning for Scalable Production Systems in the Automotive Industry: Model for evaluating benefit and effort of virtual commissioning. In: 2020 9th International Conference on Industrial Technology and Management (ICITM), Oxford, UK: February 11-13, 2020, IEEE, pp. 107-111, ISBN 978-1-7281-4306-4, doi: 10.1109/ICITM48982.2020.9080348

[Lechler 2019] Lechler T. et al. Virtual Commissioning – Scientific review and exploratory use cases in advanced production systems. Procedia CIRP, 2019, Vol. 81, pp. 1125-1130, ISSN 2212-8271, doi: 10.1016/j.procir.2019.03.278

[Lian 2022] Lian, T. et al. Design of Virtual Commissioning System for Intelligent Manufacturing Production Line Based on Digital Twin. In: 2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), November 18-20, 2022, Ma'anshan, China: IEEE, pp. 01-06, ISBN 978-1-6654-7368-2, doi: 10.1109/WCMEIM56910.2022.10021445

[Novak 2017] Novak, P. et al. Model-based engineering and virtual commissioning of cyber-physical manufacturing systems — Transportation system case study. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus: September 12-15, 2017, IEEE, pp. 1-4, ISBN 978-1-5090-6505-9, doi: 10.1109/ETFA.2017.8247743

[Oppelt 2016] Oppelt, M. Virtual commissioning in a virtual environment. In: 2016 Petroleum and Chemical Industry Conference Europe (PCIC Europe), Berlin, Germany: June 14-16, 2016, IEEE, pp. 1-8, ISBN 978-1-5090-1966-3, doi: 10.1109/PCICEurope.2016.7604641

[Pilz GmbH 2013] Pilz GmbH. PAScal – The free software tool from Pilz for verifying functional safety. Maschinen und Maschinenbau, 2013 [Online] [cit 2024-10-10] Available from https://maschinenbau.pr-gateway.de/pascal-the-free-software-tool-from-pilz-for-verifying-functional-safety/

[Shahim 2016] Shahim, N. and Møller, C. Economic justification of Virtual Commissioning in automation industry. In: T. Roeder, P. Frazier, R. Szechtman and E. Zhou, eds., 2016 Winter Simulation Conference (WSC), December 11-14, 2016, Washington, DC, USA: IEEE, pp. 2430-2441, ISBN 978-1-5090-4487-0, doi: 10.1109/WSC.2016.7822282

[Siemens 2014] Simatic S7-1500, CPU 1517-3 PN/DP (6ES7517-3AP00-0AB0), Manual. Siemens AG, 12/2024 [Online] [cit. 2024-10-10]. Available from https://cache.industry.siemens.com/dl/files/765/90471765/att_58645/v1/s71500_cpu_1517_3_pndp_manual_en-US_en-US.pdf

[Siemens 2016] Siemens. SIMATIC. S7-1500. S7-PLCSIM Advanced. Function manual. A5E37039512-AA, 09/2016 [Online] [cit. 2024-10-10]. Available from https://cache.industry.siemens.com/dl/files/153/109739153/att_895955/v1/s7-plcsim_advanced_function_manual_en-US_en-US.pdf

[Siemens 2017] Siemens AG. Working with the TIA Portal Cloud Connector. Entry ID: 109747305, V1.0, 05/2017 [Online] [cit. 2024-10-10]. Available from https://support.industry.siemens.com/cs/attachments/109747305/109747305_TIAPortalCloudConnector_DOC_v10_en.pdf

[Striffler 2023] Striffler, N. and Voigt, T. Concepts and trends of virtual commissioning – A comprehensive review. Journal of Manufacturing Systems, December 2023, Volume 71, pp. 664-680, ISSN 0278-6125, doi: 10.1016/j.jmsy.2023.10.013

[Svetlik 2013] Svetlik, J. and Demec, P. Principles of Modular Architecture in the Manufacturing Technology. Applied Mechanics and Materials, Feb. 2013, Vol. 309, Trans Tech Publications, Ltd., pp. 105–112, ISSN 1660-9336, doi: 10.4028/www.scientific.net/AMM.309.105

[Vermaak 2017] Vermaak, H. and Niemann, J. Virtual commissioning: A tool to ensure effective system integration. In: 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), Donostia, Spain: May 24-26, 2017, IEEE, pp. 1-6, ISBN 978-1-5090-5583-8, doi: 10.1109/ECMSM.2017.7945899

CONTACTS:

Ales Janota, prof. Ing. PhD.
University of Zilina, Faculty of Electrical Engineering and Information Technology, Department of Control and Information Systems
Univerzitna 8215/1, 010 26 Zilina, Slovak Republic
+421-41-513 3300, ales.janota@uniza.sk, https://kris.uniza.sk