Visualization of Control Processes and Code Validation

Monika Rybczak, Natalia Wenta, Dawid Trzciński

Abstract: The article contains an overview of articles related to the description of control process visualization. It provides short information on how to visualize the production line based on two programming environments: Factory IO and Inventor together with Matlab/Simulink. The analysis of these two environments concerns control of a virtual 3D object from a real PLC. Both virtual production line projects are based on control from the S7-1214 DC/DC/DC controller. Currently, there is a need to validate the program code or control process which has been done using several commercially available programs.

Keywords: Validate Process Technology, S7-1200, Factory Io, Visualization of The Control Process.

I. INTRODUCTION

Controlling a technological process in industry is a very broad issue. Commissioning process is described in many articles especially in last few years. This article divides technological process control into several steps:

1) Establishment of assumptions, guidelines, etc. between the involved parties (company, customer)
2) Production line design - electrical diagrams
3) Production line design - diagrams of individual 3D CAD control objects
4) Software project - PLC program (Programmable Logic Controller)
5) Visualization of the control process - HMI (Human Machine Interface) or SCADA (Supervisory Control and Data Acquisition)
6) Management of the communication network between elements of the production line (PLC, HMI, Sensors, executive elements)
7) Collecting data from line elements including input, output and intermediate signals
8) Design verification for the whole process
9) Diagnostics of the entire production process
10) Data storage, remote control via the Internet.

Each of these steps crosses to the next and depends on the others. Many terms and names for individual process control steps are described in [6] and [9], where the authors specify more precisely what is the object (plant), what is electrical and mechanical control. Due to the complexity of individual steps, this section will present those articles that are directly related to controller program and virtual production line. Authors [8] describe systems like PEMS and they include PLC (Programmable logical control) HMI (Human Machine Interface) and RC (Robot Control), who propose names such that control level zero is a PLC and HMI and the next level is control of an object designed in a 3D modeling environment such as AutoCAD or similar. Authors write about Automation Model Building using simulation programs like PLCSIM in Siemens. The reason this article focuses on the virtual production line is because verification of the entire production system is very important and consists of diagnostics at many levels of work of individual elements. For example for a Siemens PLC [20] the way to verify the program is to run PLCSIM (simulator) program which can be done without the need to connect a physical controller [2]. This is a good start to monitor the individual steps of the program, but you cannot really see how it affects the executive elements working in real conditions. Manufacturers allow the construction of machines and actuators mounted on production lines in 3D technology. In publication [3] the authors have implemented 3D models available from website level, using the X3D standard (the successor of VRML Virtual Real Management Language standard). There are also tools for verifying control algorithms from the level of the control program, where the object is a virtual 3D object [10]. At the same time, visualization of control processes cannot be confused with data archiving system through HMI (Human Machine Interface) operator panels or SCADA (Supervisory Control And Data Acquisition) systems. New technologies allow the emergence of new and separate development directions of visualization. Among controller manufacturers, Siemens offers a solution that is an intermediate step between project and installation of the system in the production facility. This step is implemented thanks to the NX MCD environment, it is a professional environment used for industrial solutions proposed by Siemens [16]. The article [4] shows that there are tools for verifying algorithms controlled from the level of the driver program to a virtual 3D object. Another interesting platform is the DELMIA solution proposed by its manufacturer [17], [12]. It provides a virtual production line simulation that can be programmed using PLCs from various manufacturers. Two programming environments will be presented in this article: Matlab [18] and Factory IO [19] available at Maritime University in Gdynia, by means of which verification of program code is similar to verification on real objects.

Manuscript received on November 20, 2021.
Revised Manuscript received on November 25, 2021.
Manuscript published on November 30, 2021.
***** Correspondence Author

Monika Rybczak*, Department of Ship Automation, Gdynia Maritime University, Gdynia, Poland. Email: m.rybczak@we.umg.edu.pl

Dawid Trzciński, Department of Ship Automation, Gdynia Maritime University, Gdynia, Poland. Email: 44756@student.umg.edu.pl

Natalia Wenta, Department of Ship Automation, Gdynia Maritime University, Gdynia, Poland. Email: 43410@student.umg.edu.pl

II. STEPS OF PROGRAM CODE VALIDATION

Currently, the programmer no longer executes the program lines themselves. He must know more, he needs to know the object he is programming. This knowledge helps with the creation of control algorithms and 3D drawings help to understand the technological process. However, time and financial issues matter a great deal in business. Hence, companies are trying to enable programmers to verify the technological process virtually before it is installed on the production line. It turns out that visualization of control processes can be done on four planes as shown in figure 1.

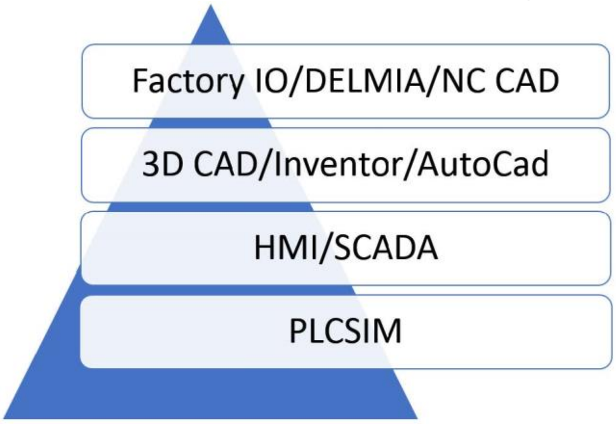

Fig. 1.Pyramid of verification process steps

The first layer concerns the verification of program code lines. Depending on the manufacturer, for Siemens, whose environment is TIA Portal, program code simulation is possible thanks to PLCSIM. The second layer, known as HMI, allows for reading, writing or remote control of the production process. The third layer, described as 3D CAD, allows the design of 3D objects. These allow the programmer to become familiar with the object which in turn affects the creation of algorithms to control the entire technological process. The latest, fourth, layer is verification of the simulation of the control process for 3D objects that have their dynamics based on real object physics. This is no longer an ordinary animation path, it is a simulation of the object's dynamic trajectory based on the written control algorithm on the PLC. The fourth layer is necessary during tests performed for ordinary technological processes. It is necessary during the work of scientists who write new equations for optimization or partial specific control used in industry [5]. Validation of the control algorithm on the virtual object allows the programmer to see some inaccuracies during design, and allows the scientist to perform many tests without having to replace the materials of the tested object during the tests. General block diagram of program code validation method is shown in figure 2. According to the author it can be done in two different ways, the simple way, in a lab, as shown in figure 3 or the professional way, where a lot of complex programming tools are needed, as shown in figure 4. The following sections of the article show short characteristics of two programming environments, with the help of which it was possible to validate the controlled object and Siemens PLC.

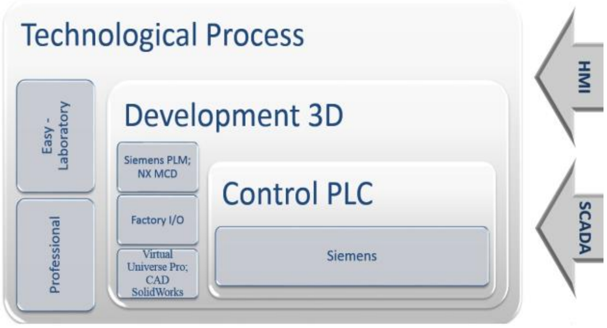

Fig. 2.Technology process - General schemat

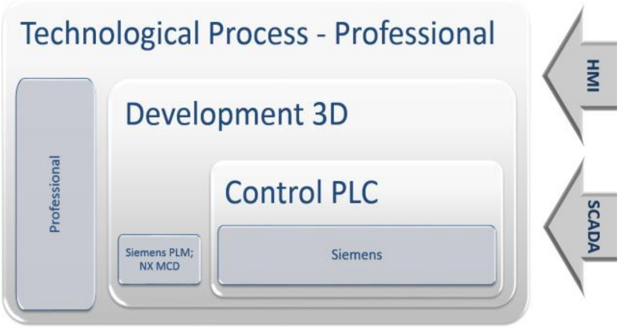

Fig. 3.Technology process - Laboratory schemat

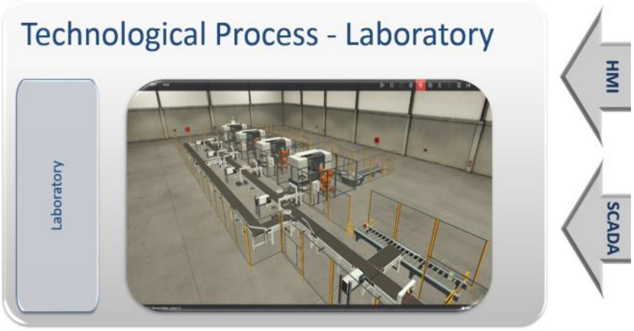

Fig. 4.Technology process - Professional diagram

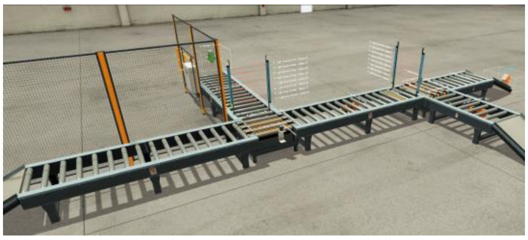

Fig. 5. Production line controlled by Siemens S7-1214DC controller, visualization done in Factory I/O

III. VALIDATION OF THE CONTROL PROCESS USING FACTORY I/O ENVIRONMENT

This section will briefly discuss the environment proposed by Real Games. Factory I/O software is used to simulate industrial facilities in real time. The program offers over 80 elements, including conveyors, robots, pushbuttons, signaling devices, electrical switchboards, etc.

These can also be controlled by real controllers, supported brands include Siemens, Allan Bradley and Schneider. As for programming languages, they comply with the IEC 61131-3 standard dedicated to PLC controllers [15]. In addition, the tool is equipped with its own Control I/O programming language. The graphic detail of objects allows for real behavior of programmed objects and failures that can occur on them. The elements are made in 3D technology taking into account the dynamics of real objects [11]. The program is equipped with several ready-made scenarios with written programs. Only two elements are needed to run a simulation: Factory I/O installation file and a control element which can be either a PLC, its simulator or Control I/O simulator. The Factory I/O website has all the necessary installation files (also a 30-day free demo version) and configuration files for controllers. Visualization shown in picture 5 shows the implementation of control for a parcel sorter, the control element is a Siemens PLC controller from S7-1214 family, programmed in TIA Portal V15 environment. The project uses 10 inputs, including start, stop, emergency (safety) button and information from two capacitive sensors and a laser sensor reading the height of sorted boxes. In picture6 selected two executive elements are shown: a light curtain and a chain conveyor necessary for sorting boxes. The 16 outputs control: five chain conveyors, five conveyor belts, one box feeder and three ramps, additionally one emergency lamp and one stop state signal lamp.

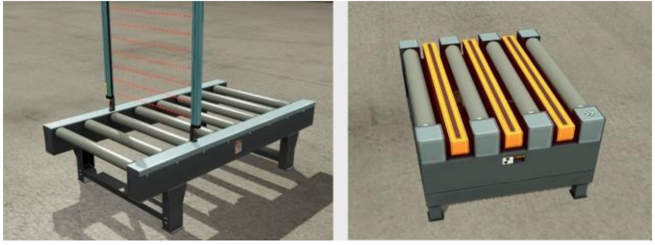

Fig. 6. Elements of Factory I/O software - light curtain: reads the height of the box through laser beams (left) - chain conveyor: uses the rollers and chains to send boxes in one of four directions (front, back, right, left).

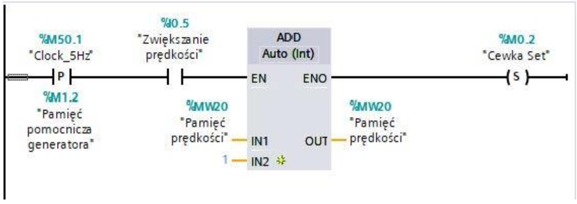

Fig. 7. Part of the program written in LAD language, used in the S7-1214DC/DC/DC

Visualization of the production process is associated with the logic of the S7-1214 DC/DC/DC controller as shown in drawing7 After completing the appropriate configuration steps, you can go to the dynamic part and put the production line in motion according to the program algorithm introduced to the PLC and associated with the 3D environment.

IV. VALIDATION OF THE CONTROL PROCESS USING INVENTOR ENVIRONMENT

This section will briefly discuss the environment proposed by AutoCad. The design of the production line using a programmable logic controller and 3D software involves the design of a virtual production line model consisting of roller tables where the basic sliding element is a roller conveyor. The project was based on the Matlab environment in which the object's dynamics were designed. Picture 8 shows a block diagram of the tools used to visualize the production line control process.

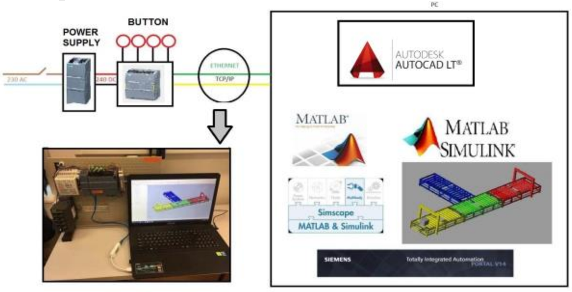

Fig. 8.Block Diagram Of Programming Tools Used: Matlab Environment with Simscape

The steps of creating the project are presented below. Design assumptions are:

1) Production line design - presented in a realistic version in Inventor environment. The facility has control stations with optical sensors.
2) Object movement performed by combining Simulink (Simscape Multibody) programs - including roller control blocks and TIA Portal V14, this gives control over the movement of the entire process.
3) Communication between the real object or controller - S7-1214 DC/DC/DC, and the virtual object using TCP/IP protocol based on the Simscape Multibody libraries.

After completing the production line in a 3D environment, the Matlab environment had to be properly configured with the S7-1214 DC/DC/DC controller. Figure 9 presents a general diagram with the s-function written to perform communication between the PLC and the Matlab-Simulink [7] environment. The implementation of these three steps allows validation of a virtual production line, whose work depends on the program written on the PLC. The effect of visualization of the process itself is shown in drawing 10. It was designed using Simulink library, Simscape Multibody. Initial stage was the design of the roller table in Inventor 3D environment and determination of all axes necessary for later design stages.

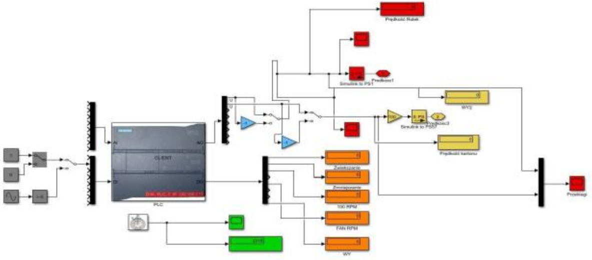

Fig. 9. Control block for S7-1214 controller written in s-function needed for PLC com munication in TCP/IP protocol with Simulink softwar

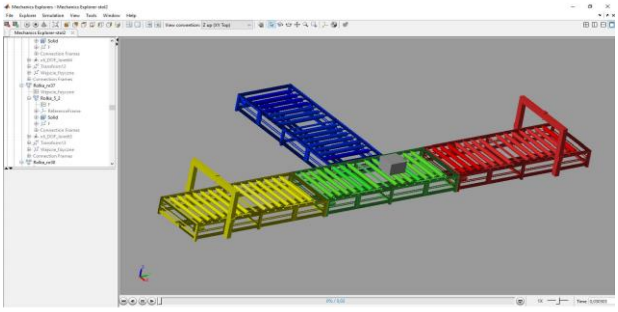

Fig. 10. Production line controlled by S7-1214 DC controller, visualization done in In ventor

V. CONCLUSIONS

Control algorithms are verified on real objects in the production facilities, which is often not beneficial to working conditions due to the short installation time and lack of indirect verification. The purpose of this form of visualization, by linking with the logic of PLC programs, allows gaining experience and skills needed to program real industrial facilities. Both the Factory I/O and Matlab / Simulink / Inventor environments allow such operations without costs and risk of damaging expensive equipment. At present, the need to validate the entire technological system is visible already at the stage of the simulated project, which will significantly affect the time of installation on a real object in the production hall. Despite not being professional tools proposed by well-known manufacturers, both presented environments are available at Maritime University in Gdynia and can be used by students as an intermediate step between programming and working in industry. Visualization created using the Matlab / Simulink / Inventor environments proved that despite the programmer's significant impact on the physics of the 3D object, unfortunately the complexity of the stages of the technological process was too high. In contrast, Factory IO environment is an easy to use tool created for scientists, students or individual customers who can study complex control algorithms written in controller programming environment. Programming and testing a technological process using Matlab / Simulink takes significantly much more time than the same work done using Factory I/O. The reason being mainly the fact that 3D objects need to be created in Matlab / Simulink by the user compared to a library of ready to use objects in Factory I/O. Siemens, one of leading manufacturers of industrial systems, on their website [22] show advantages of using 3D programming environments, with dynamic control and programming languages in accordance with IEC 61131, for control algorithm testing. These advantages include reducing risk, reducing costs of implementing industrial processes or detecting faults in these processes on early stages. According to the authors, virtual validation of the technological process seems to be an interesting issue and a new thread in the process of managing automation systems in industry.

ACKNOWLEDGMENT

Funding: The project is financed in the framework of the program by the Ministry of Science and Higher Education called "Regionalna Inicjatywa Doskonałości" in the years 2019–2022 (project number: 006/RID/2018/19; the sum of financing: 11,870,000 PLN).

REFERENCES

1. Michalski A., Starzyński J., Wincenciak S.: 3D Approach to Design the Excitation Coil of an Electromagnetic Flow Meter, IEEE Trans. Instrumentation and Measurement, 51(4), pp. 833–839, 2002.
2. Jose Calderon, Antonio; Gonzalez, Isaias: SOFTWARE-IN-THE-LOOP APPROACH FOR AU TOMATION AND SUPERVISORY SYSTEMS EDUCATION, 12TH International Technology, education and development conference, INTED Proceedings, pp. 4241–4245, 2018
3. EunHee Kim, Jinsang Hwang, Gyeong-June Hahmc, Jae Hyun Lee. 3D CAD model visualization on a website using the X3D standard, Computers in Industry 70, pp. 116—126, 2015.
4. Wang, J., Dai, C., Shi, K., Qin, R.: Research on Rigid Body Motion Tracing in Space based on NX MCD. In IOP Conference Series: Materials Science and Engineering Vol. 320, No. 1, pp. 012012, 2018
5. Li, G., Long, X., Zhou, M. (2019). A new design method based on feature reusing of the nonstandard cam structure for automotive panels stamping dies. Journal of Intelligent Manufacturing 30(5), 2085-2100, 2019 C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
6. Choi, S., Kim, B. H., Do Noh, S. A diagnosis and evaluation method for strategic planning and systematic design of a virtual factory in smart manufacturing systems. International Journal of Precision Engineering and Manufacturing, 16(6), pp. 1107-1115, 2015
7. Jacek Hryniewicz Uk lad sterowania poziomu cieczy w zbiorniku o zmiennym przekroju Engineering Work ZUT, 2011
8. Liu, Z., Suchold, N., Diedrich, C. Virtual commissioning of manufacturing systems a review and new approaches for simplification, Virtual Commissioning of Automated Systems. In Automation. IntechOpen, 2012
9. Hossain, M., Semere, D. T. Virtual control system development platform with the application of PLC device. In Proceedings of the International MultiConference of Engineers and Computer Scientists Vol. 2, pp. 13 - 15, 2013
10. Vermaak, H., 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), pp. 1 - 6, 2017
11. Podgórski K., Jałoszyńska Z., Cybulska D.: Symulacja procesów sterowania z wykorzystaniem sterownika PLC, Poszerzamy Horyzonty, Tom X, pp.392-399, 2018.
12. Ogbemhe, J., Mpofu, K., Tlale, N., Ramatsetse, B. Application of robotics in rail car manufacturing learning factory: A case of welding complex joints. Procedia Manufacturing 31, 316-322, 2019
13. O. Salamon and A. Heidari, Virtual commissioning of an existing manufacturing cell at Volvo Car Corporation using DELMIA V6 Master Thesis, Department of Signals and Systems, Chalmers University of Technology, Sweden, 2012.
14. Xiong, J., Zhu, G., Huang, Y., and Shi, J. A User-Friendly Verification Approach for IEC 61131-3 PLC Programs Electronics 9(4), pp 572-588, 202
15. Norma IEC 61131-3 about language PLC [web page] https://webstore.iec.ch/publication/4552/ [Accessed on 28 Dec 2019]
16. Platform NX CAM Siemens [web page] http://siemens-nx-cam.com/ [Accessed on 10 January 2020]
17. Platform Delmia [web page] https://www.3ds.com/ [Accessed on 10 January 2020]
18. Company Matlab [web page] https://www.mathworks.com/ [Accessed on 20 January 2020]
19. Company Factory IO [web page] https://factoryio.com/ [Accessed on 20 January 2020]
20. Company Siemens [web page] https://new.siemens.com/global/en/products/ /automation/systems/industrial/plc.html/ [Accessed on 20 January 2021].
21. Company Siemens 2 https://automatykaonline.pl/Z-branzy/Siemens-dostarcza-/ /technologie-dla-fabryki-Volkswagen-Crafter, 2020}/ [Accessed on 20 March 2021].
22. Company Siemens 3 \ https://www.youtube.com/watch?v=lfG86yVFqCA\ [Accessed on 20 September 2021]

AUTHORS PROFILE

Monika Rybczak, She received her Ph.D. degree in UMG from Maritime University in Gdynia, Poland, in 2016. She work in laboratory Programmable Logical Control, Visualization Process and Industrial Computer Network. Gdynia, Poland and his research interests

Natalia Wenta, Completed engineering studies, currently in the process of master's studies on UMG. Student of the HMI research club: https://www.facebook.com/Ko%C5%82o-Naukowe-HMI-1983523395219225/ Works at the Remontowa Shipyard.

Dawid Trzciński, Currently in process of engineering studies. Student of the HMI research club: https://www.facebook.com/Ko%C5%82o-Naukowe-HMI-1983523395219225/.