International Journal
Information Technology Applications (ITA)
Volume 11, Number 2, December 2022

THE PROGRAMMABLE VIRTUAL MODEL OF THE MECHATRONIC SYSTEM USING NX 12 ENVIRONMENT

Filip Žemla, Ján Cigánek

Abstract:

The paper deals with the design and implementation of the programmable virtual 3D model of the mechatronic system. The paper offers an analysis of a programmable logic controllers, as well as an overview of Industry 4.0, specifically an analysis of the concept of a digital twin and technologies for the creation of a digital twin. The design of the digital twin of the mechatronic system consists of two steps. The first step is the creation of the virtual 3D model, the second step is the creation of an interactive communication between the virtual model and programmable logic controller. The resulting digital twin of the high bay warehouse is compared with a real laboratory model. The task is to design and create a digital twin based on a real model that will simulate its behavior as accurately as possible.

Keywords:

PLC, digital twin, Industry 4.0, virtual 3D model, NX 12, mechatronic system.

ACM Computing Classification System:

Information systems, Information Systems applications, Process control systems.

Introduction

Nowadays, we can observe that the term digital twin is becoming more and more popular in all industries. We can find its use when solving problems in real objects or processes. In the standard commissioning of automated machines and equipment, there are usually many unexpected and unforeseen problems that extend the time of commissioning with increasing costs. This is because the PLC programmer can only fully verify the program after the technology is physically connected.

The digital twin solves problems such as eliminating errors in the control software, mechanical damage during the implementation of the device or late detection of the problem in the earlier stages of the project. It allows the programmer to verify the functionality of the prototype, which was created by another designer. The programmer can also test the control program and the functionality of the entire system already in the development phase and have the program tuned during commissioning itself. This will prevent mechanical damage and further downtime caused by traditional commissioning.

A digital twin is a real-time virtual representation of a physical object or process. Using a digital twin, we can observe the movement of products and other objects in real time on a virtual model. They can then be analyzed and the best possible solution evaluated. The virtual model of the device can be used by anyone, anywhere in the world. Through it, the user can monitor individual processes of automated lines and detect errors remotely without being physically present. A digital twin can be a digital replica of an object in the physical world, e.g. a production line, a car, or even larger objects like buildings, entire cities. There are many applications and use cases for the digital twin, it is a very active area of research and innovation.

Programmable Logic Controller (PLC) is considered as a fundamental component of automation in industry. PLC is an industrial digital computer that is relatively small and adapted to control production processes in real time, such as production lines, robotic devices or any activity requiring high reliability [1]. Looking at the function of a PLC, it can be defined as a device that receives multiple inputs or signals from a physical system. It can receive information from connected sensors or input devices, then process them and trigger outputs based on pre-programmed parameters. Such devices can be networked with other PLC and SCADA systems [2].

1 Industry 4.0 and Digital Twin

The industry is currently exposed to a significant technological change called Industry 4.0. This term was first mentioned by the German Federal Government in 2011. It arose as a result of the development of Internet and the idea of connecting the systems of the physical real world with the virtual world. The main task of the change is to achieve fully automated production, capable of autonomously adapting to external influences, such as changes in demand. This can be achieved by connecting all production equipment in one global network with autonomous exchange of information [3].

Industry 4.0 or the fourth industrial revolution represents a global trend in automation and industrial processes, with the use of modern intelligent technologies, towards the exchange of data in production processes and the complete automation of machines. Industry 4.0, sometimes referred to as IIoT (Industrial Internet of Things) or intelligent manufacturing, combines physical production and operations with intelligent digital technology, machine learning and big data. It creates a more integrated and better connected ecosystem for companies focused on manufacturing and supply chain management. The term Industry 4.0 also includes four key aspects: cyber-physical systems (CPS), Internet of Things (IoT), Internet of Services (IoS) and smart factories.

CPS represent a significant evolutionary step. They are physical devices with built-in tools for digital data collection in real time, their processing and distribution, they are interconnected using the Internet. The combination of CPS, high-performance software and special user interfaces, integrated into digital networks, creates a completely new world of system functionality.

IoT is a concept referring to the connection between physical objects such as vehicles, machines and other objects with embedded electronics, software, sensors connected to a network. IoT enables the collection and exchange of data. Connected objects can be controlled remotely, through existing network infrastructures, and create opportunities for further direct integration of the physical world into computer systems.

The Internet of Services (IoS) represents an infrastructure using the Internet as a medium for offering and selling services. As a result, services become tradable goods. IoS provides the business and technical basis for advanced business models focused on the provision and use of services.

A smart factory uses technologies, solutions and approaches within Industry 4.0. It is a concept derived from IIoT, assuming the production environment as a fully automated and intelligent network of systems, enabling the management of equipment, machines and logistics chains in the production plant, without human intervention. A smart factory is a place where data is exchanged not only between production tools and machines, but also between all elements in the production technology chain.

Virtual Commissioning (VC) is the simulation and modeling of the production system for the purpose of developing and testing the system before physical commissioning. Engineers simulate processes before turning on the physical system, allowing them to verify that everything is working. VC uses a virtual model, which is an accurate 3D simulation of mechanical, electrical and control systems, to verify the physical functions of the system before implementation.

VC technology and applications have been developed to significantly reduce or even eliminate the physical process, shorten the required time and ultimately deliver significant cost savings. VC does not have to wait for all the hardware to start up. VC can begin before the programmer has any hardware available. Because all the components exist in the software model, engineers can start testing in the digital space.

Today we use two virtual commissioning methods, Software in Loop (SiL) and Hardware in Loop (HiL). Both methods work with a virtual model of the device, as a copy of the real device on which the control program is tested. Both methods result in real-time simulation. The virtual model simulates all processes and thus generates input and output signals, which are subsequently converted into digital and analog form and transferred to the control program [4].

In the Hardware in Loop (HiL) method, the automated device is converted into a virtual form. A real PLC station with appropriate communication interfaces is subsequently used to control this model. This method allows the programmer to verify the control program in a complex and precise way, and with its results to be closer to reality. The disadvantages of this method are the need for real hardware, i.e. PLC with the associated I/O interface and PC with a running virtual model.

The principle of the Software in Loop method is based on simulations of both the virtual device and the controlling PLC. The developed control algorithms run in a simulation environment on the development computer, in simulation or in real time, depending on the requirements. The programmer thus can test the control program without the need for additional hardware. It allows testing the software before initializing the hardware prototypes, which significantly speeds up the development cycle.

Over the years, the definition of a digital twin has evolved, but the basic idea has remained the same: a dynamic virtual software-generated representation of the corresponding physical systems and processes. Since the introduction of this concept, the amount of data and information that can be transferred between products has increased. With a virtual model, it is possible to simulate the behavior of various states and thus evaluate its performance. These models can be simplified so that only the geometry or other characteristics match, thus reducing their size and speeding up the simulation calculation. Such simplified models allow to simulate complex systems and their behavior in real time, with acceptable computing power [5].

There are several software tools from different manufacturers for virtual commissioning or digital twin. The leading manufacturer of simulation software is the Siemens company, with its products NX Mechatronics concept designer and Tecnomatix Process Simulate, which contain all the necessary tools for the complex simulation requirements of the project. Another suitable environment for virtual commissioning is Emulate 3D from Rockwell Automation.

2 Case Study

For the practical part of the paper, a kit from Fischertechnik called Automated High-Bay Warehouse was chosen (Fig.1). This is a laboratory model in which a pallet with a workpiece is automatically transported by a conveyor and placed in a position in the warehouse using a manipulator. The model consists of three main parts: a conveyor with a color sensor, a manipulator that moves in 3 axes, and a warehouse where pallets with workpieces are stored. There are also limit switches, sensors and encoders that protect the manipulator from mechanical damage.

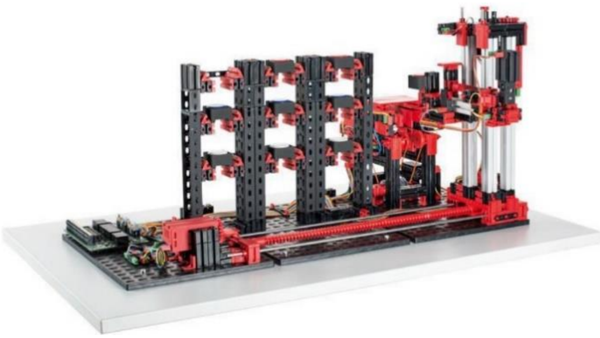

Fig.1. Laboratory model of High-bay warehouse.

VC of our high-bay warehouse was carried out using several environments: NX 12 and its MCD module with the running virtual model; TIA Portal V15.1 to configure individual PLC modules of Siemens S7-1500, and to create the control program for the high-bay warehouse and its visualization on HMI panel KTP700 Basic PN from Siemens; S7-PLCSim Advanced V3.0 environment to create a virtual instance of the PLC S7-1500 and to upload the created control program from the TIA Portal.

A. Creation of virtual 3D model

For creating a 3D model, we chose the NX 12 program from Siemens in the CAD module. The NX program is Computer-aided design (CAD), Computer-aided manufacturing (CAM) and Computer-aided engineering (CAE) software for detailed modeling of machines, from individual parts to large assemblies. In addition to modeling, it is possible to perform calculations, simulations, analyses, creation of drawing documentation, programming, etc. All these areas are interconnected, which increases the efficiency of the entire solution [6].

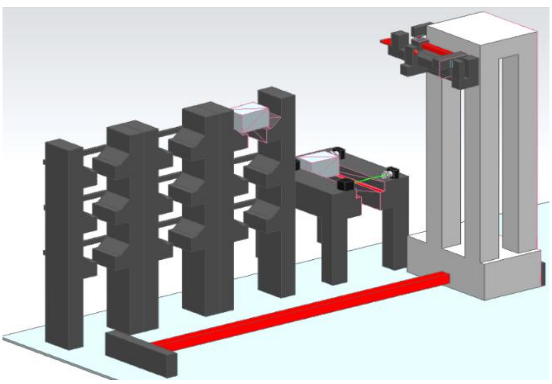

Fig.2. Virtual model of High-bay warehouse.

When creating the 3D model, we used real data from the Fischertechnik kit. With the help of a caliper, we obtained all characteristic and necessary dimensions by measuring, in order to maintain proportionality between the real model and the virtual model. We tried to simplify the individual details of the kit as much as possible, in order to speed up the subsequent simulations and simplify the modeling of the virtual model.

Our 3D model consists of several parts, which we will later combine into a single unit. The basic element is the manipulator, composed of three parts. The first part is the base that allows the manipulator to move up and down along Y axis. The other two parts of the manipulator are used for loading or unloading the pallet. One part is used for picking a pallet, thanks to which the manipulator can move forward and backward along Z-axis.

Another separate part related to the manipulator is the rail along the X axis. This is used to allow the manipulator to move left and right.

An equally important element is the warehouse, which we modeled as a separate component. In the warehouse, it was necessary to model 9 positions for pallet storage.

The last larger part is the conveyor, used to transport the pallet for the manipulator. The last step was the modeling of the limit switches and the sensors located on the conveyor, which consist of two separate components. In NX 12 environment, we combined individual functional elements and components into the resulting virtual model.

B. Creation of digital twin

The creation of the digital twin also took place in the NX 12 environment, in the Mechatronics Concept Designer (MCD) module. We uploaded the completed 3D CAD model of high-bay warehouse to the MCD application. We tried to revive the individual functional elements of 3D model in a virtual environment. MCD software is a module of the NX program that enables 3D modeling and interactive simulation of automation-related multi-body physics concepts. It allows creating a mechatronic 3D CAD model of an existing device, with the aim of mapping the physical and kinematic properties of the device. The simulation technology in MCD software is based on the game physics engine, based on simplified mathematical models, bringing the physical behavior of the real world into the virtual world.

The first step of the conversion was defining the movement coupling of individual functional elements. We assigned physical properties to the individual elements, simulating the real behavior of the body (option to choose manual assignment of the object's weight, location of the center of gravity, moment of inertia in individual axes or the choice of automatic assignment of values, based on the 3D model of the given part). Collision properties defining the mutual interaction of the bodies had to be assigned to the individual bodies.

The high-bay warehouse model contains four end sensors, two light sensors and drives ensuring movement in three axes, and the corresponding encoders. For all these elements it was also necessary to create a simulation in the virtual reality. For rectilinear movement of the conveyor, Transport Surface function was used, which will create a surface allowing objects to be moved in the selected direction and speed. Speed Control function was used to simulate the manipulator drive in all directions. Next, we created a simulation of the individual sensors. The Collision Sensor function was used for two light barriers on the conveyor to detect the intersection of the green beam with another object.

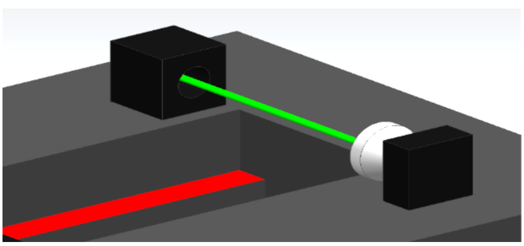

Fig.3. Light sensor (barrier) of virtual model.

It was also necessary to create collision sensors for the end sensors placed on the manipulator. These sensors detect the presence of the manipulator in the extreme position and prevent mechanical damage or collisions. Collision Sensor function ensures that the manipulator stops if it touches the limit switch.

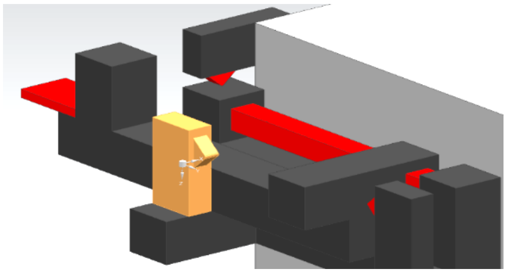

Fig.4. Limit switch of virtual model.

The laboratory model contains two encoders on the X and Y axes, which ensure the mapping of the current position of the manipulator. To simulate the encoders, we used Position Sensor function to measure the value of the displacement of the body on the given axes. The value of the encoders must be zero when the limit sensors are switched on, because the limit switches serve as reference values.

C. Connection to digital twin

In order to connect the virtual PLC and our digital twin, it was necessary to define the signals that ensure the connection between the virtual model and PLC. The creation of signals for PLC took place in NX 12 environment, in MCD module. Individual signals were created using Signal function. The output from PLC is the input in MCD, and conversely the input from PLC is the output in MCD. The names of the signals should be identical in MCD and in TIA Portal, as the Signal Mapping function will automatically allow to map the signals from MCD with the signals from PLC.

Bool or Double signals created in this way must then be connected to the corresponding functions controlling the functional elements of the laboratory model. We functionalized this connection using Runtime Expression tab, in which the behavior of the model can be defined during the simulation run. The signal connected to Speed Control function can control its activity, speed value, acceleration, simulating load conditions in the X, Y and Z axes.

To make the sensors functional during the simulation, Runtime Expression was used again. We chose the "triggered" option, because when the light beam is crossed or the limit switch is pressed, the signal value changes. Thus, we will be able to send the value from the sensor to the PLC.

D. Creation of PLC algorithm

The creation of the control algorithm took place in TIA Portal V15.1 environment. When creating a project, it is necessary to select correct PLC module and CPU type. Our kit consists of an S7-1500 series PLC with a CPU 1516-3 PN/DP from Siemens. Digital and analog inputs and outputs from a real lab model had to be added to CPU.

We also used memory variables to start specific sequences or part of the program, respectively to store a value. PLC program is stored in organizational blocks (OB). The organizational block forms the interface between the operating system and the program that is called cyclically by the system.

At first, we created "Safety" OB, which serves to safely stop the drives in extreme positions and prevents mechanical damage of individual parts. Here we used information from limit switches and pulse counting from encoders on individual axes. We created the algorithm for the reference of the manipulator in OB "Main", which ensures the introduction of the manipulator to the reference extreme position on the X and Y axis, and the initialization of individual auxiliary variables.

The automatic mode is used to automatically place the pallet in the warehouse. After placing the pallet on the conveyor and selecting a specific position in the warehouse using HMI panel, the manipulator will automatically store the pallet. It means the transfer of the pallet from the conveyor to the desired position in the warehouse. If the given position is already occupied, the manipulator unloads (moves) the pallet from the selected position to the conveyor.

The selection of the position is made through HMI screen (Fig.5), where the user can select one of 9 positions using a button. The user can thus decide which position he wants to load or unload. Loading or unloading from the position by the manipulator depends on the color of the button. If the color of the button is green, the user can store the pallet in the given position. If the pallet is red, it can be removed from the given position. We have created an OB called "HMI to PLC" for data transfer after pressing a button from HMI to PLC. Now we can identify whether the user wants to unload or store the pallet in the selected position.

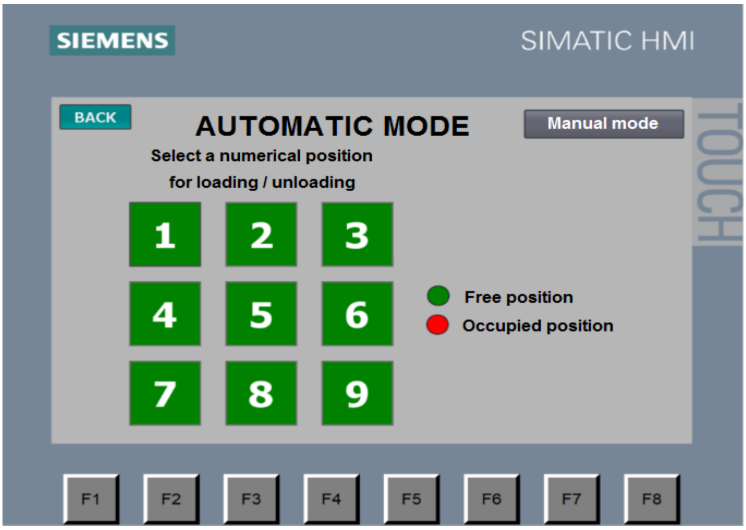

Fig.5. HMI for automatic mode.

After choosing the position, it is necessary to create an OB to map positions in the warehouse and to enter instructions for the manipulator: to load or to unload. For individual rows and columns, it was necessary to find out their individual coordinates. When moving the manipulator to a specific position, we use encoders to monitor the value on the X and Y axes.

In order to match the value in the virtual model and the value from the encoder, it was necessary to convert the measured pulses to milimetres. For the calculation, we used the length of the movement path in the X and Y axis, and pulses count of the real encoders. When the manipulator starts to move, we add or subtract value in the auxiliary variable according to the direction of movement, which we determine using the input signals.

In the case of the virtual model, we used Position Sensor function, which automatically measures the displacement value of the manipulator. At the start of the simulation, it was necessary to set the manipulator, to the starting positions, in order to activate the limit switches.

We send the value of the position from the virtual encoder directly to the PLC, using the connection of the signal and the auxiliary variable.

Next OB serves for unloading or loading a pallet to the selected position. The algorithm consists of four consecutive steps: a) arrival of the manipulator in front of the correct position, b) insertion of the manipulator into the warehouse, c) displacement of the manipulator along y axis, d) extension of the manipulator from the warehouse.

The manual mode is used for manual control and testing of the functionality of the manipulator, conveyor and individual functional elements. We can track the position of the manipulator on X and Y axes and we were able to map individual positions in the warehouse and to create an automatic mode. We can test the functionality of the safety stop of the drives, if the limit switch is turned on or if the position of the manipulator exceeds a critical limit. The movements of the manipulator in all directions are controlled using the buttons on HMI. In manual mode, the manipulator can be set to the reference position using the "Initialize" button.

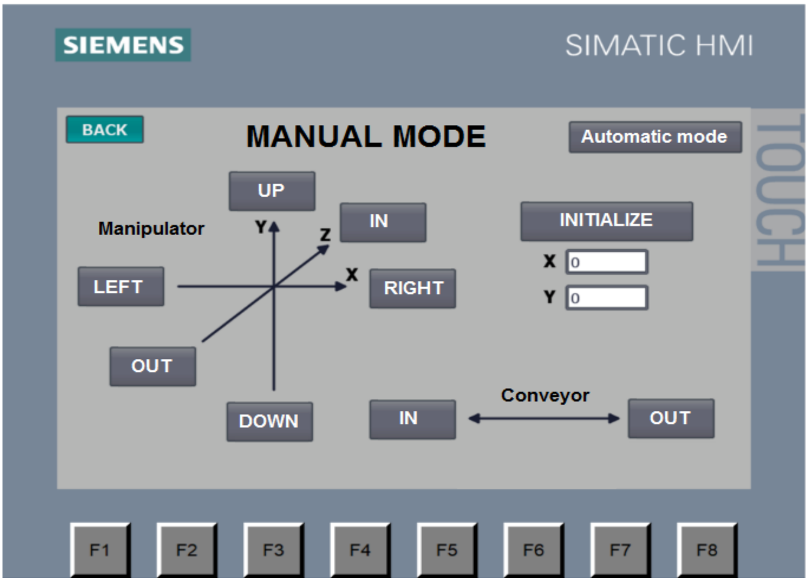

Fig.6. HMI for manual mode

E. Virtualization of PLC

PLC virtualization was executed in PLCSim Advanced V3.0, which allows creating a PLC instance replacing a real PLC. When creating a virtual PLC, in the user interface (Fig.7) we selected the PLCSIM option in Online Access option. We created an instance called my_PLC. In PLC type menu, Unspecified CPU 1500 option was selected, and then Start button was pressed. This step creates an instance with the specified name that represents the created virtual PLC.

Uploading a program created in TIA Portal environment to a PLC instance is the same as uploading to a real PLC, using Download to device option in TIA Portal. The dialog window offers the option of choosing the device to which the program should be recorded. In the line Type of the PG/PC interface we selected the bus type PN/IE, in the next line PG/PC interface we chose PLCSIM as the connection adapter. The last setting in Connection to interface/subnet line was Direct at slot '1 X1' option. After configuration, using Start search button, we found the created instance called my_PLC. After selecting the given device and pressing the load button, the program will be uploaded to this instance.

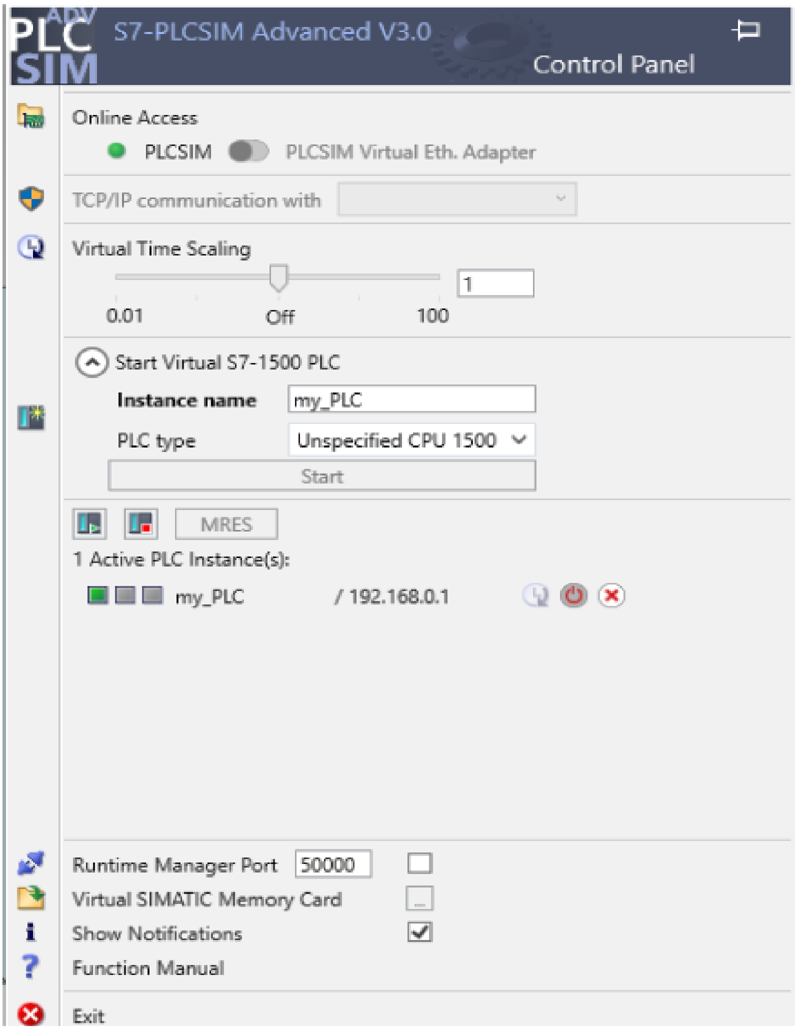

Fig.7. PLCSim Advanced.

F. Connection of virtual model and PLC

The connection of the signals of the virtual model and the signals in the PLC (running in PLCSim Advanced) was realized through the MCD module of NX 12 environment. In Automation tab, we selected External signal configuration function, which is used to upload PLC signals to MCD. In this dialog box, we selected PLCSIM Adv tab.

After selecting the PLC instance, all input and output signals of the virtual PLC are displayed in the lower part of the dialog window. When choosing IOMDB in Show line, memory type tags (auxiliary variables) will also be displayed.

Signal mapping function (Fig.8) was used to match the signals from PLC to the signal of the virtual model in MCD. In Signals section, the signals created in the virtual model are displayed on the left side, and the imported signals from the virtual PLC are on the right side.

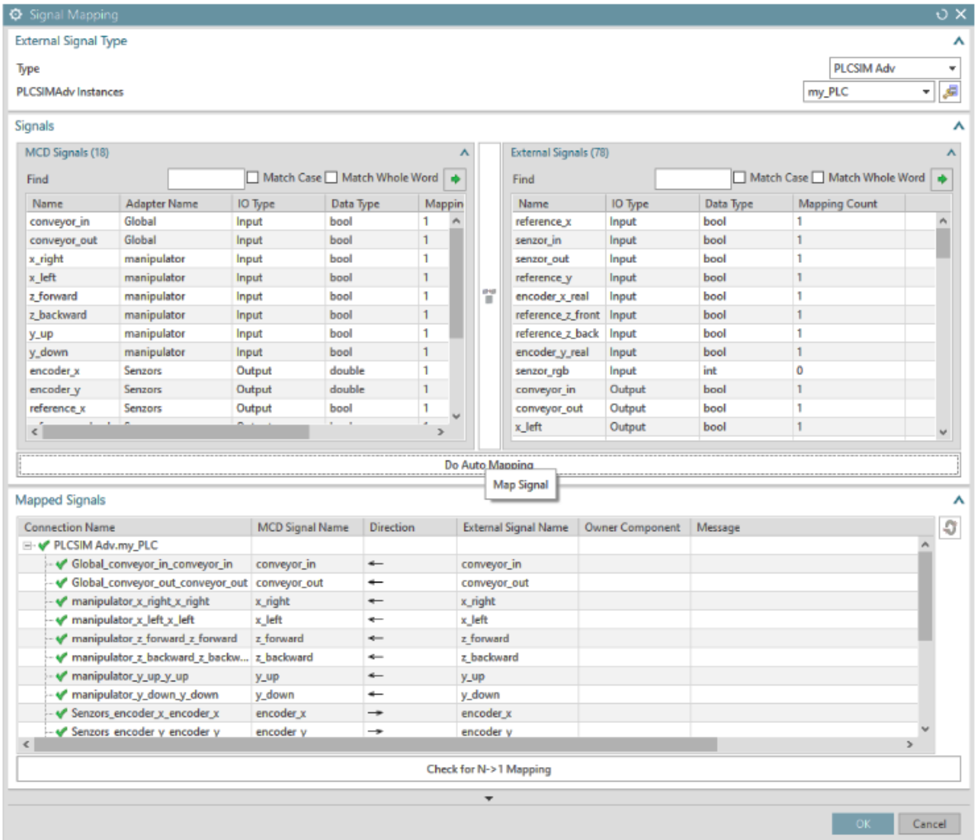

Fig.8. Signal connection between virtual model and virtual PLC.

To create a connection between individual signals, it is necessary to select one signal from the right side and its corresponding signal from the left side. We connect both signals using the central button. Several conditions must be met when connecting signals. The signals must be of the same data type and opposite I/O type. Therefore, the signals for controlling the drives were created as Input, and the sensor signals were defined as Output.

G. Simulation results

Starting the virtual model in NX MCD environment is possible only after starting the virtual PLC with the downloaded program. In NX, we launch the virtual model using the Play button. To control the high-bay warehouse, it is necessary to start the virtual HMI touch panel in TIA Portal. We verified the functionality of the virtual model using the manual mode, where we tested the control of individual manipulator drives. The course of the simulation was smooth and the behavior of the virtual model corresponded to the behavior of the real kit.

Finally, we verified using manual mode the functionality and control of individual drives in all axes of the manipulator, as well as conveyor drives. After verifying the functionality of all elements, a successful manipulator reference was made.

Using the automatic mode, we tested the loading of the pallet and its subsequent unloading from a random position in the warehouse (Fig.4).

After testing the functionality of the virtual model, we were able to compare the real and virtual model. In the case of comparing the functionality of PLC algorithm on a real and virtual model of a high-bay warehouse, we evaluate the behavior as identical. Individual positions in the warehouse were served by the manipulator as expected and without any problems in both models. All functional elements in the virtual model corresponded to the behavior in the real kit.

Conclusion

The realization of the digital twin was preceded by the creation of a 3D model. We created the virtual model in the CAD module of NX 12 environment. We tried to simplify the individual details of the virtual model as much as possible, in order to speed up subsequent simulations and simplify modeling. Subsequently, we uploaded the created 3D model to MCD module, where the conversion to a digital twin was executed. We tried to bring the individual functional elements of the virtual model to life in a virtual environment. It resulted in a programmable virtual model of the high-bay warehouse, where its inputs and outputs corresponded to the real kit.

PLC control algorithm consists of Automatic mode and Manual mode. With the help of the created virtual model and PLC program, a virtual commissioning was carried out, which verified the functionality of the model and the PLC algorithm. After the testing, we agreed the behavior of the virtual model and the real kit.

The virtual model of the high-bay warehouse could be improved by a more complex version of 3D model. At the same time, by adding more components to the virtual model, we can create more complex systems, for example by connecting with a robotic arm and other Fischertechnik kits to create an automated line in the virtual reality.

Acknowledgement

This work has been supported by the Slovak Grant Agency VEGA 1/0107/22, KEGA 039STU-4/2021 and Scientific Grant APVV-21-0125.

References

[1] A. Goel, “40 Important PLC Projects for Engineering Students”, Engineering, 2018. https://engineering.eckovation.com/plc-based-final-year-projects/

[2] T. Achyut, “PLC – Architecture/Hardware”. PLCAUTONETICS, 2020. https://www.plcautonetics.com/plc-architecture/

[3] J. Kruger, “Why are there still so many jobs?”, Siemens, 2020. https://new.siemens.com/global/en/company/stories/industry/why-are-there-still-so-many-jobs.html.

[4] J. Dzinic, Y. Charlie, „Simulation-based verification of PLC programs”, Department of Signals and Systems, Chalmers University of Technology. 2013.

[5] M. Grieves, “Digital Twin: Manufacturing Excellence through Virtual Factory Replication”. ResearchGate, 2015. https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication.

[6] B. Lyndon, “Machine design software generates open source Mechatronics program code”, Automation, 2003.

Authors

Ing. Filip Žemla
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
filip.zemla@icloud.com
Currently a student of doctoral studies at Slovak University of Technology in Bratislava. The main focus of his studies is oriented to virtualization and optimalisation of modern manufacture processes. His main skills are SCADA systems, database systems and front-end programming.

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk
He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professors at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.