COMMUNICATION BETWEEN CAD SYSTEMS AND THE PLC CONTROLLER

Kamil Lysek, Aleksander Gwiazda, Krzysztof Herbuś

Silesian University of Technology, Faculty of Mechanical Engineering, Institute of Engineering Processes Automation and Integrated Manufacturing Systems
Konarskiego 18A, 44-100 Gliwice, Poland

Corresponding author: Kamil Lysek, kamil.lysek@polsl.pl

Abstract: Mechatronics is a science which part are simulation systems, including hardware-assisted simulations. The aim of this article is to describe how to connect a CAD system with a mechatronic module to PLC. The solution of the problem of combining these two modules allows for sending setting values to the virtual PLC controller. This is important because a PLC controller can be combined at a later stage with the robot and virtual control panel, which can be made in CAD system. In summary, the solution of the problem presented is the first step to create a virtual control system for a real object.
Key words: PLC, NX, CAD systems.

1. SIMULATION IN THE NX ENVIRONMENT - MECHATRONICS MODULE

Nowaday, CAx programs are an inseparable part of every design office dealing with more or less advanced simulations. They can be used for calculations (CAE), design (CAD) and manufacturing simulation (CAM). One of the leading programs of this type are Inventor, Solid Edge, Solid Works and NX, which will be used to simulate. An example of the NX possibilities of using the program is presented in Figure 1, [4].

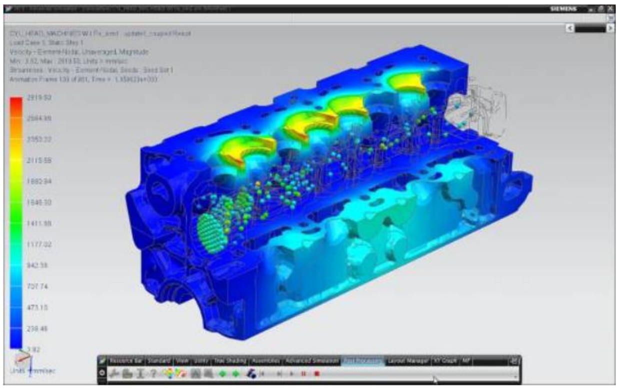

Fig. 1. The example of the capabilities of the NX [5]

The NX program, through the Mechatronics Concept Design (MCD) application which provides tools to simulate the complex movement of mechatronics system interactively. The entire MCD application can be divided into several smaller parts, among which can be distinguished:

- System Engineering;
- Mechanical Concept;
- Simulate;
- Mechanical;
- Electrical;
- Automation;
- Design Collaboration.

The focus of this work should be mainly on Simulate, Mechacial, Electrical and Automation.
In the simulations menu (Figure 2), simulation can be controlled by playback. It allows both simple controlling by start, pause, stop and transition to subsequent operations, or setting the time scale.

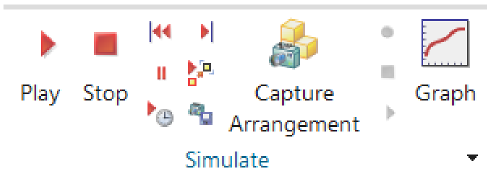

Fig. 2. Simulate menu in MCD

In the mechanical menu (Figure 3) properties of a given object can be declared. Examples of properties can be:

- Rigid Body – defines a rigid body for physics body for simulation. Each element that does not have additional functions is defined as rigid;
- Collision Body – an object that acts as a sensor that detects that another object is touching it.
- Hidge joint – connects objects along an axis of rotation;
- Sliding joint – connects objects along a fixed linear axis;
- Cylindrical joint – connects objects along a rotatable linear axis;

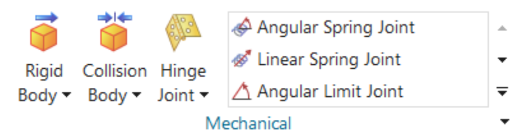

Fig. 3. Mechacial menu in MCD

Examples of electrical properties can be (Figure 4):

- Position control – allows to determine the position of a given object and its speed.
- Speed control – allows to examine the speed of move for a given object.
- Signals – allows to integration with other systems, e.g. OPC UA. They can be assigned to input signals and output from the server.

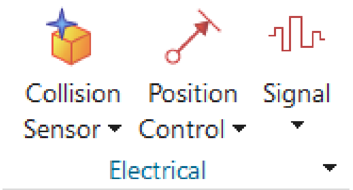

Fig. 4. Electrical menu in MCD

One of the most important functions from the point of view of this article is the signal mapping, located in the Automation menu. It allows the assignment of signals from the OPC server to the simulation and vice versa. It is important to connect signals of compatible types. Ie bool values must also be linked to logical values and numeric type values to other values of this type.

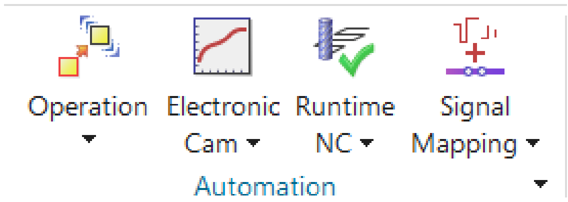

Fig. 5. Automation menu in MCD

An example of signal mapping is shown in Figure 6. First select the server from which the signals will be mapped. Then enter the appropriate address for this server. Then select two signals for mapping and press the map signals button. Signals mapped in this way are visible in the mapped signals tab [6].

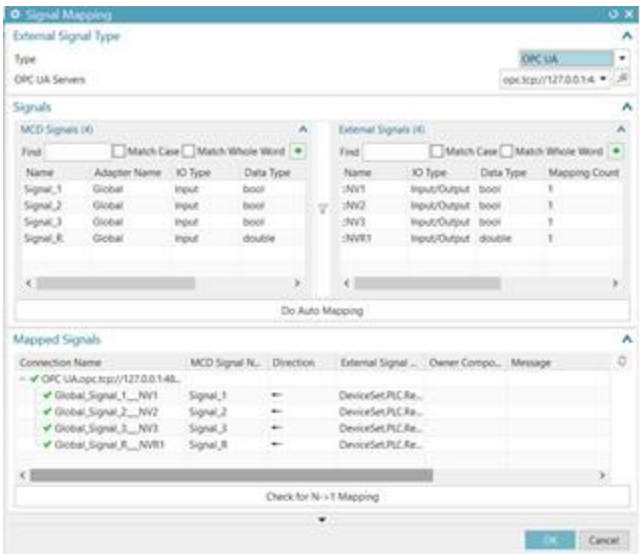

Fig. 6. Signal Mapping windows

2. VIRTUAL COMMISIONING

In order to improve the quality of production while minimizing costs, a number of methods have been created to accelerate the design process and reduce costs by, for example, using virtual commissioning. With the help of such a model can check the correct operation of the entire factory system.

The virtual commissioning scheme is shown in Figure 7. The virtual factory sends signals from virtual sensors to the controller while the controller transmits the setting signal to the actuators.

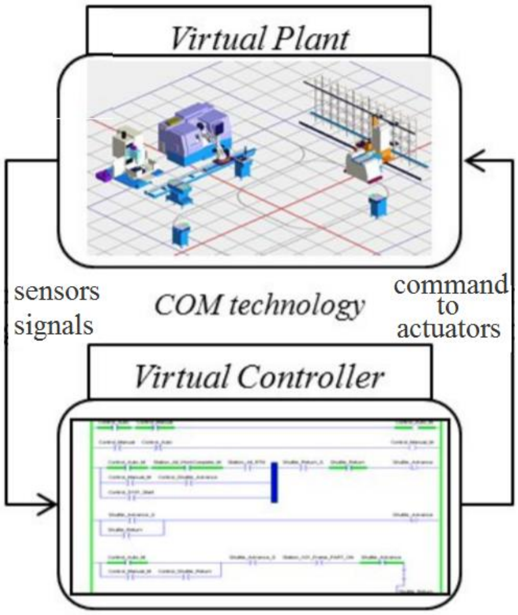

Fig. 7. Virtual commissioning [7]

Using this technique allows to reduce costs and accelerate the implementation of the optimal solution for the factory.

3. PROGRAMMING OF PLC DRIVERS

The PLC is a microprocessor device that is commonly used to control both the operation of individual machines, but also entire technological processes. The IEC 61131-3 standard defines basic concepts related to PLC controllers. On its basis, we can distinguish languages used for programming PLC controllers such as:

- LD – Ladder Diagram – ladder language is based on similarity to contact relay circuits in which contacts and coils can be distinguished. Contacts are input values, while coils are output values. Programming in this language is possible using logic functions. The contacts can be connected with other contacts in series (AND function) or in parallel (OR function). The use of function blocks and arithmetic functions is also allowed, but these are not typical features of the LD language;
- FBD – Functional Block Diagram – one of the

graphic languages in which the scheme of the program is based on signal flow. It uses ready function blocks or procedures;

• ST – Structure Text - one of the text languages. It is used to describe expressions whose description is graphically difficult or impossible. The basic elements of a language are conditional expresions and commands. In ST, variables are defined between the words VAR and END_VAR;

•IL – Instruction List - equivalent to assembler language. Program requirements are expressed by means of successive lines of simple code.

These are the languages typically used for PLCs, it is possible to program them using other programming languages, such as C++ [2, 12].

4. AUTOMATION STUDIO

One of the most popular controllers next to Siemens controllers are those from B&R.

Due to the consistency of the work ecosystem, it was decided to combine the company's drivers with Automation Studio. This program has been supporting connections via the OPC server for many years. In earlier versions, the OPC DA server was used for this purpose, which is shown in Figure 8, [11].

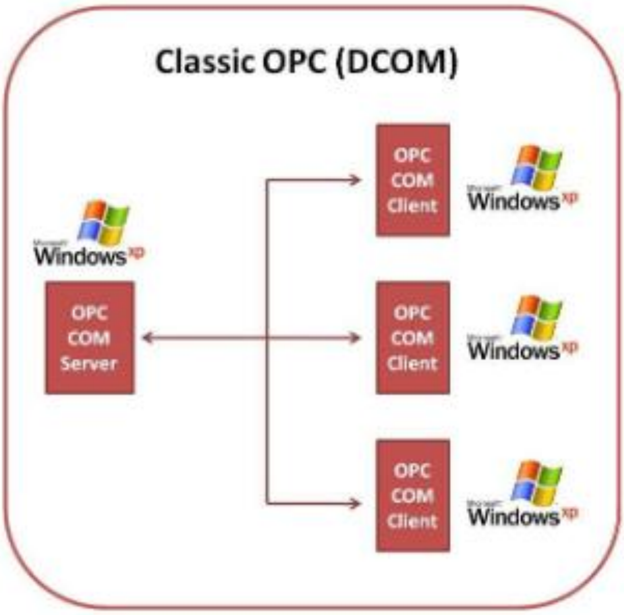

Fig. 8. Communication with OPC DA, [11]

However, he had several disadvantages such as

- dependence from the Windows platform;
- scalability;
- security;
- transmission reliability.

These problems are solved by another OPC UA standard, which was also introduced in the B&R program.

The connection diagram with the use of the OPC UA server is presented in Figure 9.

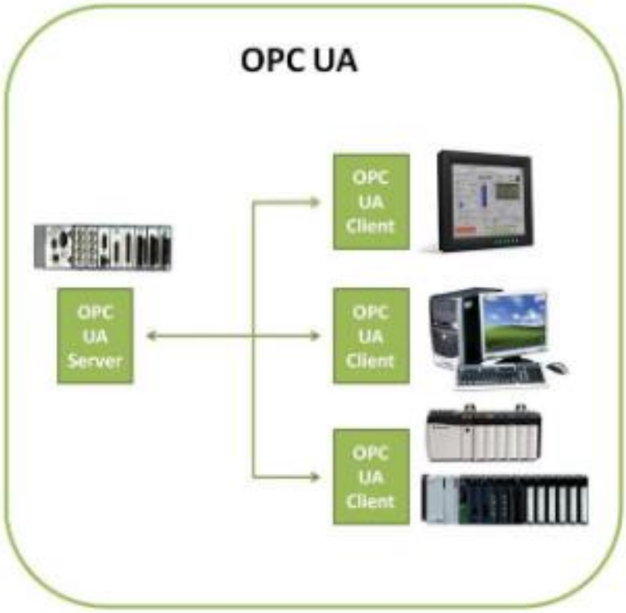

Fig. 9. Communication with OPC UA, [11]

This standard introduces security restrictions such as user authorization, encryption or exchange of digital security certificates [3, 9].

The graphical user interface in Automation Studio is divided into several different areas, each of which has its own task (Figure 10).

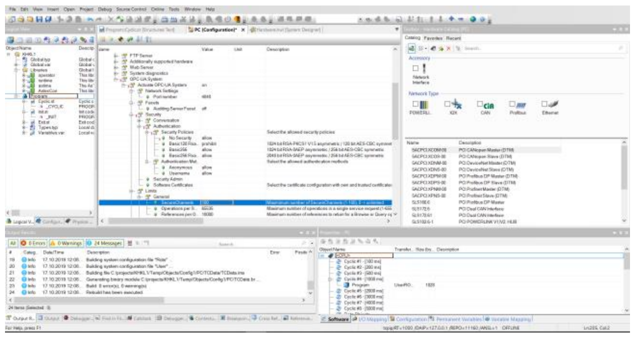

Fig. 10. Automation Studio windows

Project explorer (first from the top left) used to manage and edit program and configuration objects in the project.

In the middle there is a work area with open documents, in the example illustration there is a simple program code.

Toolbox (last from the top left) allows, among others, to add configuration options, hardware including the addition of an OPC server.

First from the bottom left is window displaying program output. For example, there may be error notifications or notificaion about the correct compilation of the program. The sample program did not detect any errors or warnings, while 24 neutral messages were displayed in the window.

The last window is properties. Show configuration options for the currently selected object or hardware module, [10].

5. FANUC ROBOT

The device used for simulation is the model of the six-axis robot ARC MATE 100iB from FANUC. It is built from modules, which are driven by a servo drive. It weighs 135 kilos, and its load capacity allows transporting elements weighing 6 kg with an accuracy of 0.08 mm. This industrial robot is mainly used for cutting and welding. Described robot is shown in the Figure 11 [1, 8].

Fig. 11. Real robot FANUC ARC MATE 100iB in the Silesian University of Technology

The range of motion of the robot and the axis of rotation axis is shown in Figure 12.

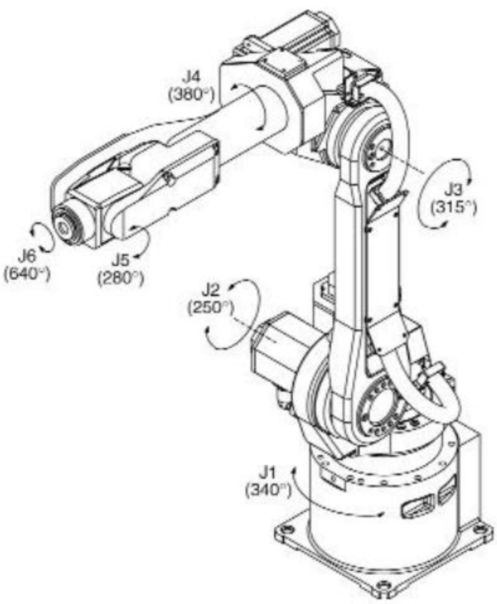

Fig. 12. Robot motion range, [8]

6. PREPARATION FOR SUMULATION

The model used with the marked coordinate axes is shown in Figure 13.

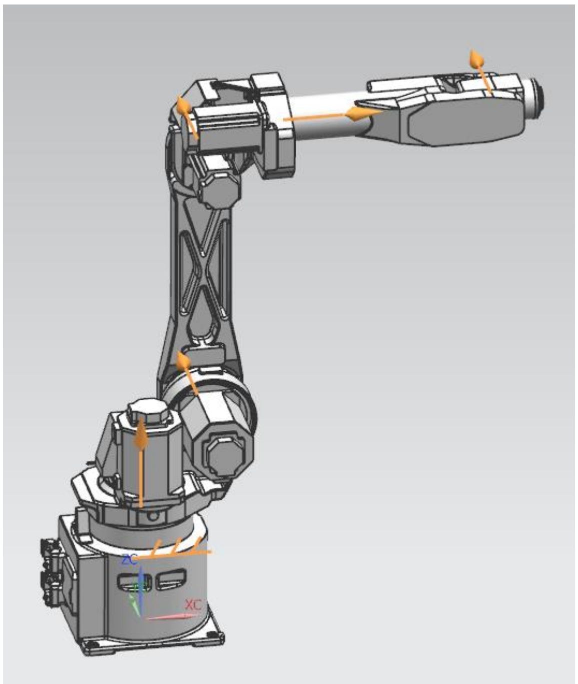

Fig. 13. The model used with the axes

All robot modules are defined as rigid body, as shown in Figure 14.

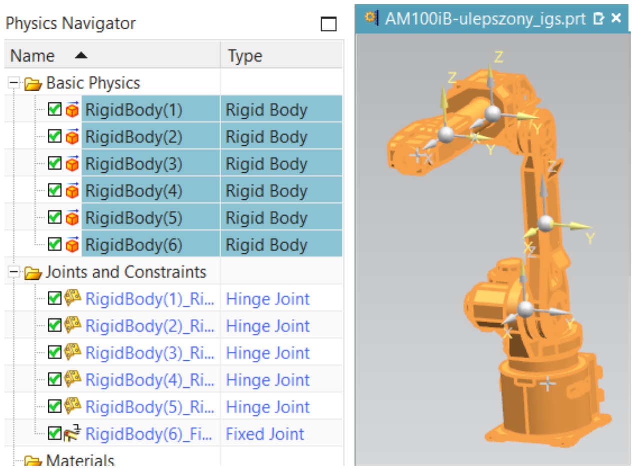

Fig. 14. Rigid body

As it is possible to see all the elements except the base have been connected with hidge joint. The base was of course fixed. The respective rigid body has been numbered from the base.

To control the position of the robot along with the speed of individual member’s position control blocks have been implemented. To enable control, all position control has been set in the NX program to 0.

The next step is to declare the signals that will be connected to the signals of the OPC UA server at a later stage.

The view of formed signals is shown in Figure 15.

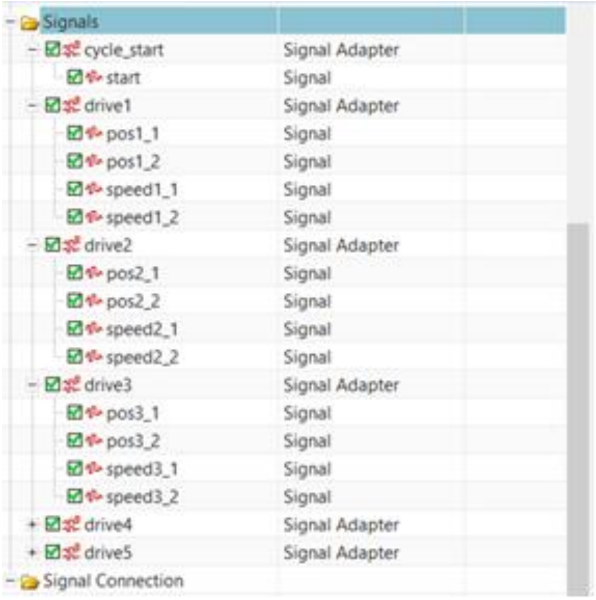

Fig. 15. Declared Signals.

In order to enable the connection of variables from both signals, a connection should be established between the OPC server and the NX program. For this purpose, the ARsim_TCPIP simulation with the IP address 127.0.0.1 has been enabled in the Online Settings menu, as shown in Figure 16.

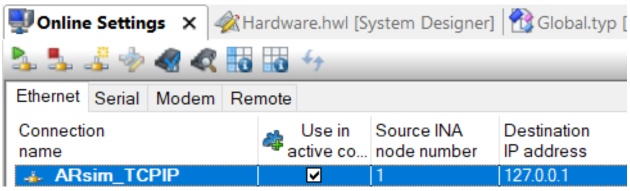

Fig. 16. Turning on the simulation and setting IP address

In addition, it was necessary to turn on the OPC server and determine the port address - the default address of 4840 was left, as shown in Figure 17.

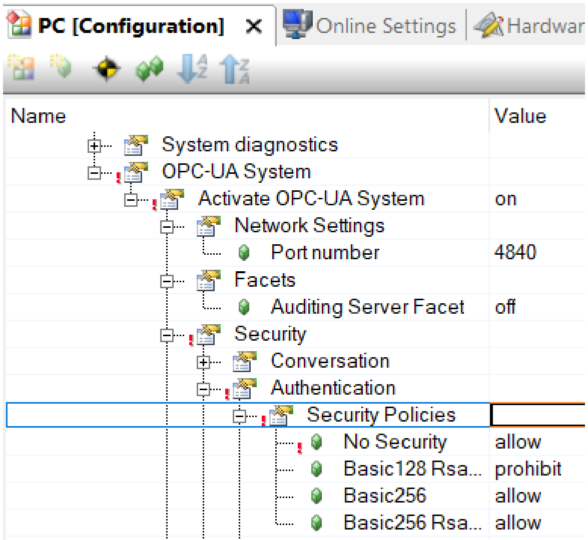

Fig. 17. Turning on the OPC server and setting ports

Then the server was added through the assigned IP address as in Figure 18.

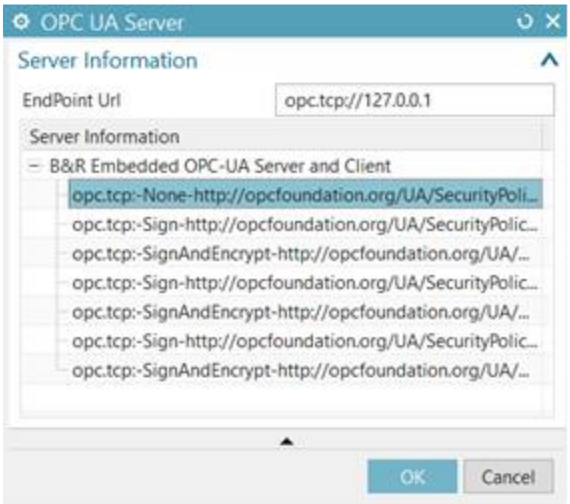

Fig. 18. OPC server selection

When the server was added, it was possible to properly connect the signals, as shown in Figure 19.

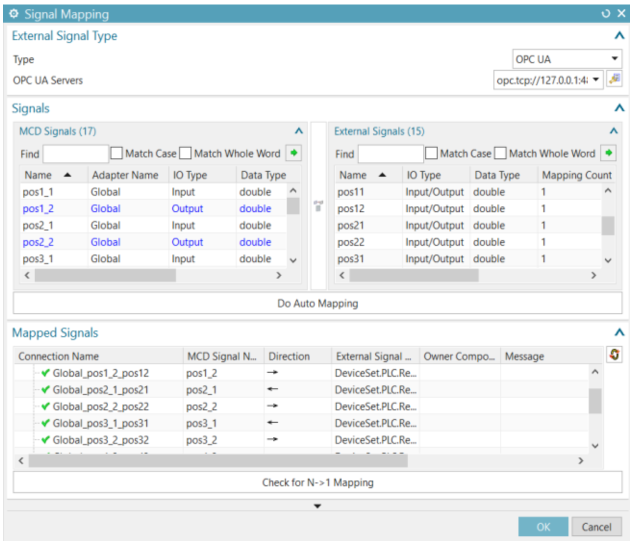

Fig. 19. Mapped signals

The model with all connections can be seen in Figure 20.

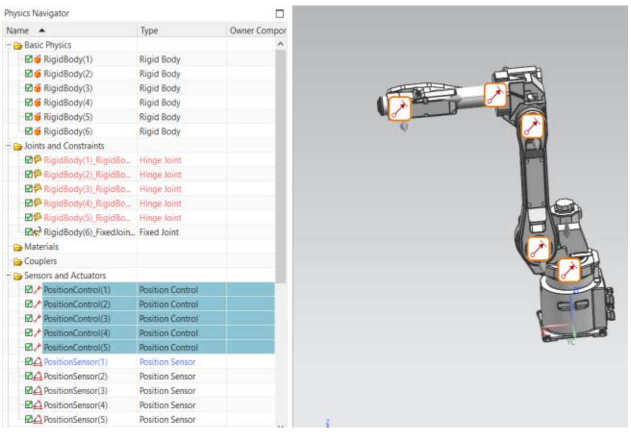

Fig. 20. Joints, constrains, sensors and actuators

7. SIMULATION OF ROBOT CONNECTION WITH THE CONTROLLER

On the basis of the previously presented combination of signals and a program written in the ST language, simulations of robot movement were generated. The robot during the first simulation is shown in Figure 21.

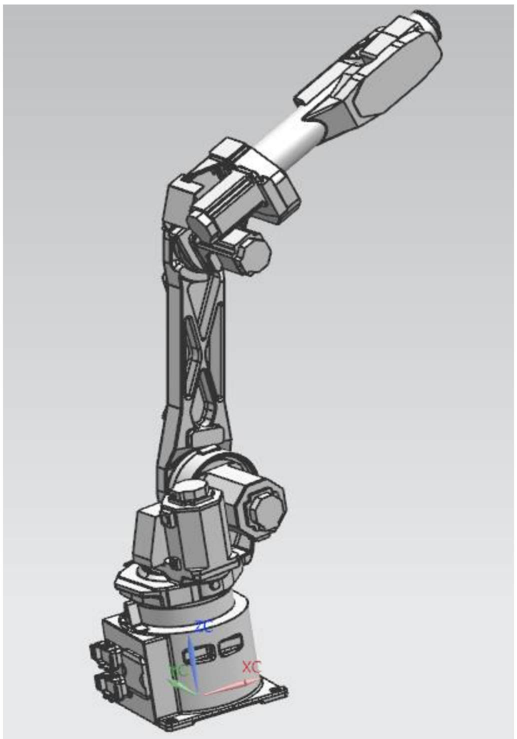

Fig. 21. The Robot during simulation

To confirm the program operation and deeper analysis of the movements that has been made. For this purpose, the robot work cycle was created and shown in Figure 22. The figure shows all the individual works of the robot embedded in the time axis (for a better view the sequence of robot work was divided into equal periods of time and combined into a whole).

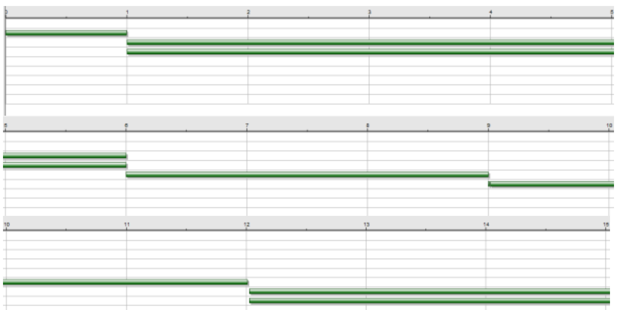

Fig. 22. The robot work cycle

8. CONCLUSIONS

OPC server is good solution for communication between different softwares and devices.
The use of the mechatronics module made it possible to use the robot to carry out the simulation using a pre-programmed (in PLC program) sequence of movements.
The Mechatronic Concept Designer allows to mapping signals with the PLC and other programs via OPC. The resulting mapping enables communication between the OPC server and the simulation object. This means that the simulation objects can be controlled using a PLC (currently virtual) and that the processes that occur during the simulation can be parameters in the control process and that the processes that take place during the simulation can be parameters in the control process.
The next step will be an attempt to communicate with a real PLC and programming a simple control using the JOINT coordinate system.

9. REFERENCES

1. Gwiazda, A., (2015). *Modular industrial robots as the tool of process automation in robotized manufacturing cells*, IOP Conf. Ser. Mater. Sci. Eng. **95,** 012104, pp. 1-6.
2. Kwaśniewski, J., (2008). *Sterowniki PLC w praktyce inżynierskiej*, pp.117-132, BTC, Legionowo,
3. Nalepa, B., Gwiazda, A., Banaś W., (2018) *Study of communication between driver sbRIO9632 and Raspberry Pi 3,* IOP Conf. Ser.: Mat. Sci. Eng., **400**, 052006, pp.1-10.
4. Świder, J., Herbuś, K., (2005). *The idea of modelling aided computer method oriented on the motion analysis,* 7th Intern. Conf. Comp. Integr.Manuf. Int. Manuf. Sys. CIM, pp. 236-241, Gliwice-Wisła.
5. https://3dcad.pl/aktualnosci/5850/kompleksowe-ulepszenia-w-nx-8-z-naciskiem-na-cae.html, Accesed: 12.03.2019.
6. https://ks-iss.com/produkty/nx-mechatronics-concept-designer-nx-cam-robotics/, Accesed: 15.03.2019.
7.https://www.sciencedirect.com/science/article/pii/S2288430014500292, Accesed: 16.03.2019.
8.http://www.mb-siuda.com/spawaniemigmagtigfanuckuka/86, Accesed: 17.03.2019.
9.https://www.automatyka.pl/produkty/opc-ua-nowy-standard-w-automatyce-przemyslowej-34320-2 Accesed: 11.03.2019.
10. https://iautomatyka.pl/co-to-jest-automation-studio-recenzja-oprogramowania-od-br-automation/ Accesed: 02.03.2019.
11.https://www.ni.com/pl-pl/innovations/white-papers/12/why-opc-ua-matters.html, Accesed: 05.03.2019.
12. http://www.zspgrot.edu.pl/www_plc/rodzaje-j-zykow-.html, Accesed: 03.03.2019.

Received: April 10, 2019 / Accepted: December 20, 2019 / Paper available online: December 25, 2019 © International Journal of Modern Manufacturing Technologies