

VIRTUAL COMMISSIONING OF A ROBOTIC CELL USING TECNOMATIX PROCESS SIMULATE

Nicolae-Adrian DUMITRAȘCU¹,
Alexandru DINCĂ², Nicolae PREDINCEA³,

Rezumat. *Conceptul de Virtual Commissioning (VC) cunoscut în România ca punerea în funcțiune virtuală, a devenit o etapă esențială în dezvoltarea și simularea proceselor de fabricație automate și robotizate. Prin crearea copiei digitale a echipamentelor din producție, companiile au astfel posibilitatea de a testa mai multe scenarii complexe cu roboți și echipamente complicate, reducând în acest mod timpul fizic de punere în funcțiune. Această lucrare prezintă conceptul punerii în funcțiune virtuale și descrie toate resursele necesare realizării cu succes a acestei etape.*

Abstract. *The concept of Virtual Commissioning (VC) has become one of the most essential phases in the development and simulation of automated manufacturing processes. By creating a digital copy of the real system installment, companies have the ability to test out more complex scenarios with robots and complicated mechatronics design, greatly decreasing the startup time of the plant and shortening the product's time to the market, while increasing the overall quality at the same time. This paper presents the core concepts of VC and all the required resources and technologies that are involved in the development of a state-of-the-art manufacturing process.*

Keywords: Industry 4.0, Virtual Commissioning, Process Simulate

1. Introduction of Virtual Commissioning

In current trends, the leading aspect that drives the competitive market of the manufacturing business is being first to launch a product to the market. In order to have short launch time it is important that all phases of the manufacturing process, from the initial design to the physical installation of the plant. The last step, in particular, can be greatly aided by using powerful tools such as Virtual Commissioning, as it allows a company to sort out any bugs in the PLC programme before even connecting to any real equipment.

Also, due to the fact that manufacturing processes tend to be extremely complex, dealing with tensor hundreds of different types of devices constantly communicating with each other, it is of the utmost importance to use a technology that permits high flexibility and great precision for the models it can create and simulate.

¹PhD Student, Eng., Digital Manufacturing Manager, ADA Computers, Bucharest, Romania (e-mail: adrian.dumitrascu@adacomputers.ro).

²Eng., Virtual Commissioning Consultant, ADA Computers, (alexandru.dinca@adacomputers.ro).

³PhD, Eng., Prof., Department of Machines and Manufacturing Systems, University "Politehnica" of Bucharest, Romania.

During each stage of the process development, changes are bound to happen to the initial design, putting the companies in a position where they have to decide on whether or not to implement the specified changes. By knowing exactly the impact of each and every change, companies no longer have to wait on complex studies and are able to make a decision much more rapidly. Having a digital representation of the real process and factory greatly helps in detecting problems and testing which scenario or solution is better to implement; all in a very timely manner.

With the present technologies it is possible to make major changes and to adapt the automation as the decisions are being made. A good example can be taken from the automotive industries, as we can see that nowadays, a new model for a car is released every one or two years. The same plant is being used when transitioning from one model to the new one and in short time, both models are manufactured in the same plant at the same time.

As new challenges emerge for every manufacturing process, companies are looking towards a standardization for the technological solution available. An out of the box, plug and play solution is preferred and highly sought after and any feature or tool that will optimize and streamline the manufacturing process will gain a competitive edge. Of course, there is no standard solution for every problem, but as technology has advanced greatly in the past 15 years, it is possible now to simulate any type of device from the factory, including and not limited to industrial robots, to a degree that any difference between the real and digital model can be easily neglected.

What was considered impossible or extremely difficult 15 years ago in terms of Robotic Simulation and Control, now has become an industry standard and is an essential part for any automated process, big or small.

If we analyze the conventional tools that are still being used in digital manufacturing, we notice that they do not create any benefit other than seeing how the completed model will look like, thus diminishing the advantages from a production point of view. By using Virtual Commissioning, companies can see how the plant will react to different situation, emergencies and test out the behavior of the mechatronics as an assembly, rather than an individual element. Furthermore, VC has become one of the most important tools for the development of a correct and efficient PLC (Programmable Logic Controller) code.

System Control Engineers now have the possibility to test out immediately their code, without connecting to a real robot or a factory device. As a result of writing the PLC code and testing it as the process itself is being developed, it is possible to decrease the launch time of a plant to such a degree that renders conventional methods of simulation and commissioning practically obsolete.

The objective of this paper is to present and evaluate the concept of Virtual Commissioning with all the resources and technologies involved, also by providing an extensive model that can be used in a real manufacturing plant.

2. Virtual Commissioning concepts

This chapter will focus on the basic concepts of VC and the underlying tools and technologies utilized in the industry

2.1. Virtual Commissioning concept

The fundamental aspect regarding of VC is the digitalization of every device from the plant so that, in the end the virtual model will work exactly the same as the real one. This is also one of the basic principles of what is now known as Industry 4.0, having a digital representation of the plant and commissioning that model before and alongside the real one. With our current capabilities, we have the option to test out everything with software, this means we can simulate both the process with Robots and field devices and actual PLC. Another option is to run the simulation on the PC and to use a real PLC to develop the code and run the simulation. The latter variant is preferable as it allows for better validation of the PLC program and the hardware configuration, as well as the effectiveness of the plant devices (motor drivers, encoders, sensors etc.) or Human Machine Interfaces (HMI) panels.

Knowing the basic principles that make up the VC concept, various configurations for implementing VC can be distinguished:

- Software-in-the-Loop and Model-in-the-Loop. By using this method, everything has a digital representation, so that the Plant has had a very precise 3D Model. That model will be controlled by an emulated PLC. Using this method of testing has the advantage of being relatively cheap and fast to deploy. Also, by not using any real equipment, several safety related issues are avoided altogether (such as robot collisions or Human injuries).
- Hardware-in-the-loop method tries to emulate or use as much of the hardware as possible, including PLC controller, auxiliary modules and other field devices. This has the advantage of creating a environment setup identical or real close to the actual plant and permits writing the PLC code by testing on the actual hardware. The disadvantages present themselves as high cost of implementation, due to the fact that industrial equipment are very expensive, low flexibility when changes occur to the design and also a high risk of injuries and accidents become a factor.
- Hybrid Simulation: Makes use of a 3D model of the plant with virtual robots controllers (RCS - they are developed by the Robot Manufacturer) and

connects the simulation to a real PLC. In this way we can run the code on the exact model of the PLC with the Hardware configuration that is going to be used, and any changes that are being made to the Plant design can be easily adapted into the 3D model. Also, field devices such as encoders and motor drivers can be included in the Hardware configuration by using equipment such as Simulation UNIT.

2.2. Prerequisites for Virtual Commissioning

In order to implement the Virtual Commissioning phase correctly it is important to have the following prerequisites from both the mechanical and the controls department:

- A detailed layout of the plant and a correct description of all the resources used;
- Comprehensive 3D simulation for every type of device and Robots that are going to be used. This means that is absolutely necessary to a 3D model with the correct geometry and kinematics, the robots should have all the operations finalized with the interference zones defined and all other safety measures implemented into the model;
- The sequence of the operation must be completed and defined as efficiently as possible;
- The input and output signals must be correctly mapped and defined;
- The hardware configuration of the PLC and its auxiliary equipment must be determined and should be available to the Controls Engineer as either Physical or emulated devices;

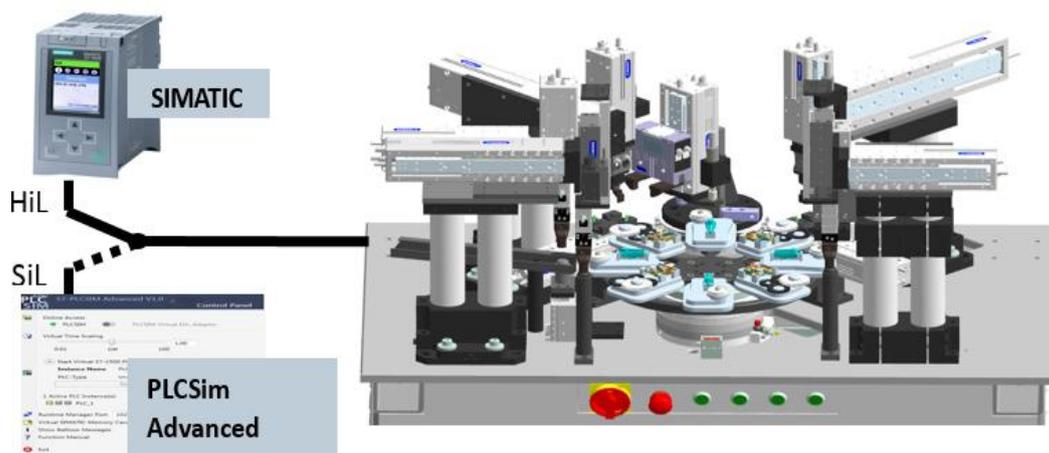


Fig. 1. Prerequisites in Virtual Commissioning.

2.3. Virtual Commissioning in Process Simulate

In the past decade, there has been an immense interest from the developers of Robotic Simulation Software to increase the quality and performance of their products and to create more complex features and tools that would benefit a more comprehensive and precise digital model.

Such efforts stand at basis of the Industry 4.0 concept and have made Virtual Commissioning a must have tool in the development of a manufacturing plant.

Although there are software developers who are working on different software solution, one of the most advanced applications is **Process Simulate** from the Tecnomatix package, part of the Siemens PLM division.

It has a lot in common with the Robcad software solution, but focuses a lot more on the ability to create smart objects and to control those ones using signals. This capability allow the user to run the PLC code either on a virtual Siemens PLC and connect directly with Process Simulate or to run the logic on a real PLC (from any make) and connect with Process Simulate via OPC Server.

Virtual Commissioning implemented with Process Simulate greatly aids with the reduction of costs, time and other resources involved in real commissioning on the plant, by working with fully proven Robot Virtual Controllers, PLC programs and by aiding in the seamless information exchange between Manufacturing, Simulation and Control Engineers. Also, Process Simulate benefits from being very well integrated in the Tecnomatix Software Package, especially with Teamcenter.

Since Process Simulate has proven the most comprehensive software solution, since it has become an industry standard in the automotive industry from the entire world, the next chapters of this paper will be presented using examples from Process Simulate.

3. Virtual Commissioning environment

This section will present in detail all the elements that are required in order to conduct the Virtual Commissioning phase of a project.

3.1. Process Simulate VC prerequisites

Even though Process Simulate requires the same conventional prerequisites in order to enable Virtual Commissioning, new technologies have been introduced so that the simulation is more realistic.

- The **CEE** (Cyclic Event Evaluator) is used for event-based simulation within a mechanical system simulation.

- **Material flow** support by the creation of appearances of parts assemblies based on events within the simulation sequence.
- **Sensors** support to allow for proper control of event triggers according to real system behavior.
- **Smart Components** to allow for the creation and storage of logic blocks within standard components to ease the process of connecting those components up within the simulation and to the PLC.
- **ESRC** (Emulated Specific Robot Controller) modules or teach pendants for robotic programming, allowing for the creation and execution of robotic programs in the native language of the machine, including logic.
- **Device operation** automatically creates an operation-end signal. This signal is used as default condition in the transition between one operation and the successor operation.

3.2. Cyclic Event Evaluator (CEE)

The CEE (see Figure 2) is a feature imbedded to the core functionality of Process Simulate. Its purpose is to run the simulation based on a sequence of predetermined operation, end to look for events such as triggers, signal and state changes in order to transition from one operation to another. Unlike a time based operation, an event based has the support for variants or different branches of operations and can alternate through all the possibilities by executing multiple cycle of the simulation. A key aspect of the CEE is that after the current flow of operations has ended it resets the simulation to its initial position.

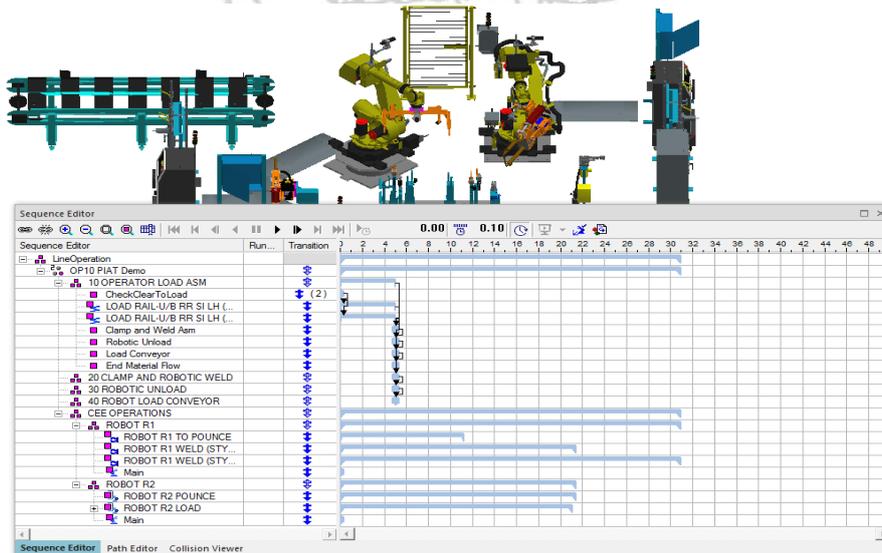


Fig. 2. Event-based simulation in Process Simulate.

3.3. Material flow simulation

It is used to integrate a 3D model of the working part into the sequence of operations. The main concept is to have a separate instance of the part called “appearance”, which will act as a copy to the original part. By using the material flow, we can simulate the appearance of a part or assembly into our simulation when it's needed and it makes it disappear once all the manufacturing operations have ended. It gives the user the ability to test out different styles of parts and to see how it would affect the production of a plant as a whole (see Figure 3).

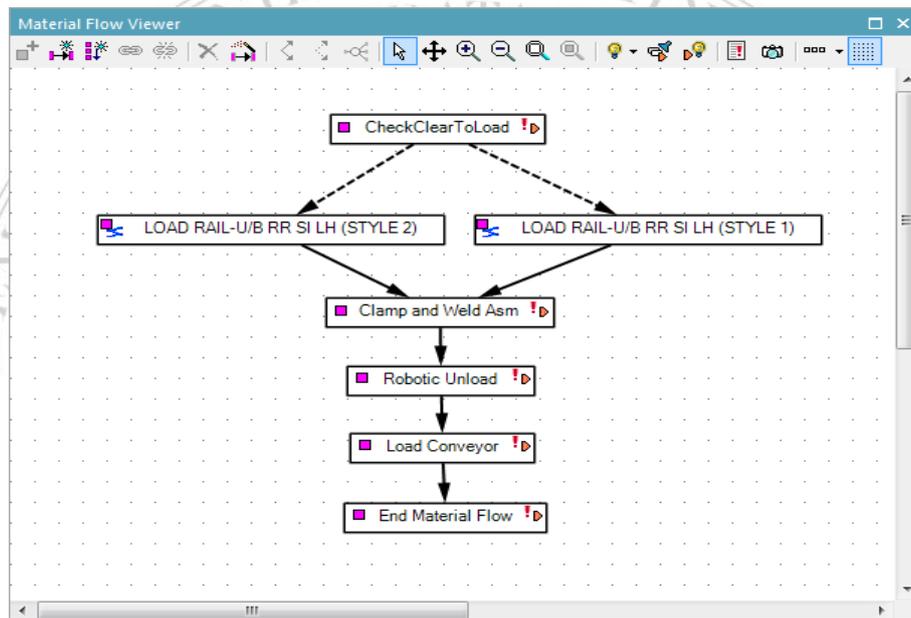


Fig. 3. Material Flow viewer in Process Simulate.

3.4. Sensors

Just like in real life, sensors in Process Simulate detect changes that happen in the 3D environment. As such, the sensors that can be used are the following:

- **Proximity and Photoelectric Sensors.** They are used for detecting the presence of a 3D resources or part in regard to the sensor's detection range. They can also be used for security measures as a way to stop the activity of a device or a robot if it gets to near to another equipment or protection fence.

When defining a sensor in Process Simulate (see Figure 4) the user has the ability to choose a 3D representation, a position that is the most suitable in the plant model environment and the specific parts or resources that the sensor can detect. The sensor is activated when one or more predefined elements enter its detection range.

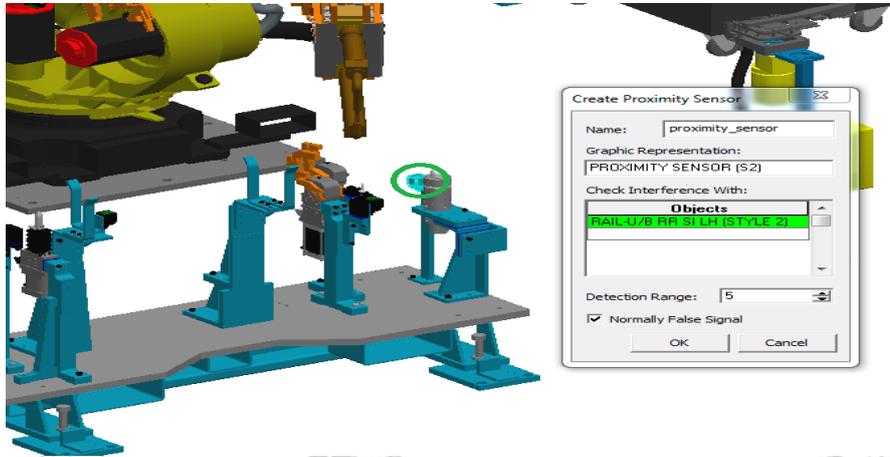


Fig. 4. Defining a proximity sensor in Process Simulate.

- **Property Sensors** allow for the identification of parts based on specific properties, such as barcode, weight, temperature, color and type and others. The properties are predefined by the user and are projected onto the part using the “Create Property Projector” command.
- **Joint Value Sensors** have the ability to detect the state of any given joint from a device and when that joint is in a determined tolerance range a signal can be activated. This capability mimics the functionality of some industrial equipment that send a feedback to the PLC when they are in a certain state. As such, it is very important to have the kinematics correctly defined for the device so that the PLC receives the correct information from the Simulation (see Figure 5).

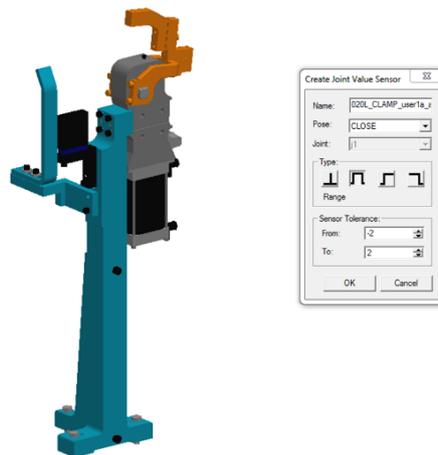


Fig. 5. Defining a joint value sensor in Process Simulate.

3.5. Smart devices

A smart device relies on the graphical representation of a resource, but adds elements of logic in order to control the kinematics of the resource. As a result the user has the ability to change the values of certain sensors, speed and acceleration or the distance travelled by a resource.

Those changes can be triggered by signals that are used in the PLC program. A great feature of this functionality is that the logic of the device is stored inside the associated JT file of the 3D model.

Using smart devices contributes to making the simulation more efficient and precise. The functionality of such smart components is implemented a “Logic Block”.

In Process Simulate, the Logic Block acts like the brain of the component. It detects changes to specific signals and based on those signals it can trigger actions tailored to the device (opening or closing a clamp, for example). More than this, report the state of the device by using Joint Value Sensors and sending those signals to the PLC (see Figure 6).

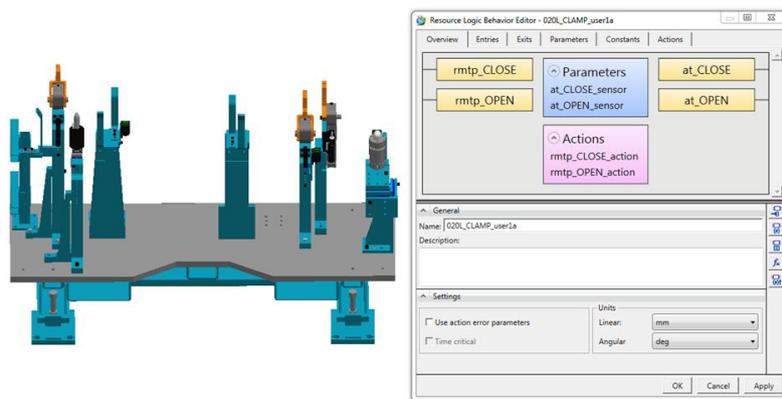


Fig. 6. Defining a logic block in Process Simulate.

As illustrated in the Figure above, a Logic Block accepts input signals and output signals. The input signals are used to trigger the actions, which in our case make the clamps open or close. The parameters are made out of Joint Value Sensors which detect the state of the device.

When a Parameter detects the specified value of the joint, it changes the corresponding signal to the True value; it keeps it to False if otherwise.

Logic Blocks work very closely with PLC logic, as the inputs for the Logic Block are outputs from the PLC and the exits from the Logic Blocks are inputs for the PLC (see Figure 7).

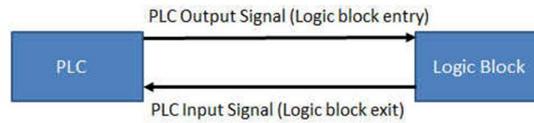


Fig. 7. Signal exchange between a PLC and a Logic Block.

Since the PLC also works with other type of signals, the Logic Block has the support for more complex signals as well. The following types can be used: BOOL (1 Bit), BYTE (8 Bits), INT (16 Bits), DINT (32 Bits), WORD (16 Bits), DWORD (32 Bits), Real (32 Bits). This means that a Logic Block can simulate the behavior of any kind of device, making them very flexible and versatile in terms of simulating factory equipment.

3.6. Emulated specific robot controller (ESRC)

As its name implies it emulates a Robot Controller for a distinct robot make and model. Although Process Simulate has a Default Controller which can drive any resource defined as a robot, it's functionality is limited it is intended to be a general tool. By using a Controller made by a Robot Manufacturer for its own product, the user gains access to more precise motion curves for the robot and specific commands that the real controller can execute. As a result, the user will notice a significant difference operation times when he is running the simulation with a Robot Controller, the robot with the ESRC being much faster. Knowing this, it becomes extremely important to have the right Robot Controller in order to properly validate the simulation (see Figure 8).

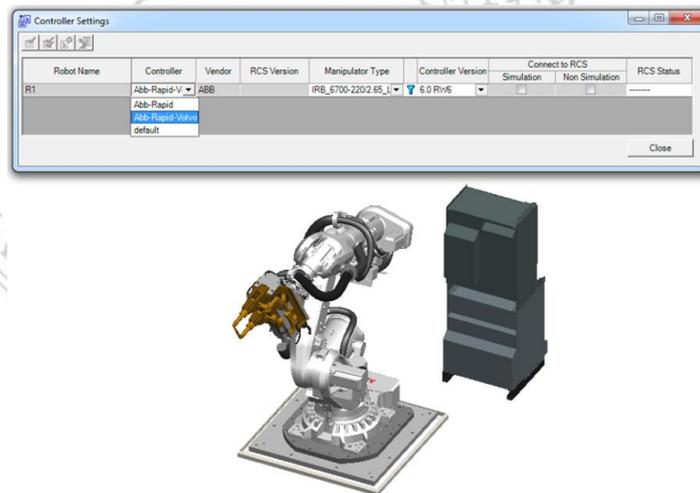


Fig. 8. Settings the controller for a robot.

Other than precise motion, the user also has access to more advanced instructions called OLP (Off-Line Programming) commands. They are custom made for each robot through what is called “XML Customization”.

At its core the Customization is a “.xml” file which contains the real robot command in its specific controller language and also a representation for Process Simulate in a more user-friendly appearance.

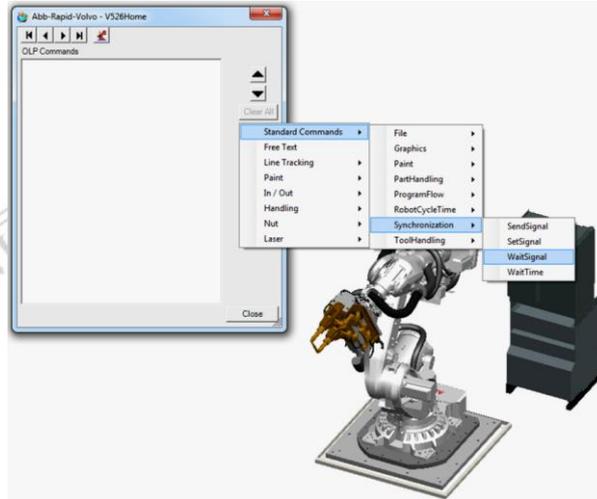


Fig. 9. Using the OLP commands.

OLP commands can be introduced at any location of the robotic operation and after the robot arrives at that location, firstly the controller goes through every command and only after the subroutine was executed, the robot will move to the next location. This functionality alongside the customization profiles introduced with the previously mentioned, gives the user the ability to use the information from the Robotic Operations and OLP Commands and to export a program that will work with the **real** controller and robot. More than this, the user can import the real programs that run on the physical controller.

Process Simulate automatically generates an operation with all the OLP commands at the correct locations. This is extremely useful when a company wants to digitalize an existing factory. In this way, they can have the correct and precise robot programs, with all the parameters included, overcoming the need to recreate them from scratch.

3.7. Signal Viewer and Signal Mapping

One of the most essential parts for Virtual Commissioning is to have each and every signal correctly defined and mapped with the appropriate address. Signals are not only important to the Event-Based Simulation, but they constitute the basis on which the communication is made with the PLC. Because running an Event-Based Simulation requires constant exchange of information between the 3D Model and the Control System, Process Simulate has a very intuitive interface that greatly aids the management of signals, called “Signal Viewer”, as shown in the Figure 10.

Signal Name	Type	Address	IEC For	PLC Connector	External Connection	Resource
TT_to_180dgr	BOOL	38.3	Q38.3	<input checked="" type="checkbox"/>	KEPServer	● 105_471
TT_to_Ddgr	BOOL	38.4	Q38.4	<input checked="" type="checkbox"/>	KEPServer	● 105_471
TT_at_180dgr	BOOL	0.3	I0.3	<input checked="" type="checkbox"/>	KEPServer	● 105_471
TT_at_Ddgr	BOOL	0.4	I0.4	<input checked="" type="checkbox"/>	KEPServer	● 105_471
R110_ReqFromRob	INT	4	I4	<input checked="" type="checkbox"/>	KEPServer	● R1
R110_AckToRob	INT	36	Q36	<input checked="" type="checkbox"/>	KEPServer	● (2)
OR_PrgmToRob	DINT	32	Q32	<input checked="" type="checkbox"/>	KEPServer	■ Robot_comm
105_2Y24S	BOOL	43.6	Q43.6	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y24R	BOOL	43.7	Q43.7	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y23S	BOOL	43.4	Q43.4	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y23R	BOOL	43.5	Q43.5	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y22S	BOOL	43.2	Q43.2	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y22R	BOOL	43.3	Q43.3	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y21S	BOOL	43.0	Q43.0	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y21R	BOOL	43.1	Q43.1	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A
105_2Y14S	BOOL	42.6	Q42.6	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y14R	BOOL	42.7	Q42.7	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y13S	BOOL	42.4	Q42.4	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y13R	BOOL	42.5	Q42.5	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y12S	BOOL	42.2	Q42.2	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y12R	BOOL	42.3	Q42.3	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y11S	BOOL	42.0	Q42.0	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2Y11R	BOOL	42.1	Q42.1	<input checked="" type="checkbox"/>	KEPServer	● 105_721_42A
105_2S24S	BOOL	12.3	I12.3	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A_UNIT03
105_2S24R	BOOL	11.3	I11.3	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A_UNIT03
105_2S23S	BOOL	12.5	I12.5	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A_UNIT05
105_2S23R	BOOL	11.5	I11.5	<input checked="" type="checkbox"/>	KEPServer	● 105_721_41A_UNIT05

Fig. 10. Signal Manager in Process Simulate.

The Signal Viewer manages all the signals that are generated inside the Process Simulate project. Here, the user has the possibility to create and delete signals, rename existing ones, choose the type, give them an address and choose if the signal is connected to the PLC or not. Not all signals require a connection; many of them are used only internally in the Simulation. Also, the user can see the Resources that a signal is associated with.

3.8. Virtual Commissioning environment setup

The final steps for creating a Virtual Commissioning environment are to have the correct setup prepared. In Chapter 2.1 there were mentioned three types of simulations. In all cases, the user has to use the same components, even though they are simulated or consisting of physical equipment.

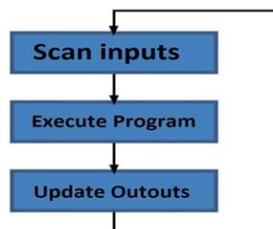


Fig. 11. Basic PLC Program execution cycle.

The required components are as follows:

- **PLC (Programmable Logic Controller).** It is used to control the Simulation as well as the real Factory. All PLCs, no matter the brand, work with the same principle of cyclic operation. They analyze the inputs, then they run the programme according to the information received and lastly, they generate the outputs. As such it becomes very important to send the correct information to the PLC in order to have the functionality desired (see Figure 11).

- **PLC Code.** It's represented by the instructions that make up the PLC code.

- **HMI - Human Machine Interface.** This is the bridge between the Operator and the plant. Its purpose is to have dedicated buttons and commands so that the operator can control the behaviour of the plant.

- **OPC Server.** It's used for the real-time exchange of signal between all the devices connected. As such, the communication between the PLC, 3D Model and other devices is managed by the OPC Server. This element becomes essential because it allows the connection of any type of PLC or device, ensuring the flexibility of the configuration. The OPC server/client comes in two mainly used variants:

- **OPC DA (Data Access).** It provides real-time communication with any number of PLCs using the specific manufacturer communication protocol. It has proven to be a robust and reliable connection and for each signal, the OPC Client associates three attributes: value, timestamp and the quality of the value.

- **OPC UA (Unified Access).** It's very similar to OPC DA, the difference being that is more modern, more secure, cross-platform and uses an integral information model that can be used with any type of equipment that supports OPC UA, no matter the brand. Also, it can be used in conjunction with other technologies that use more complex programming languages such as C++, Java or Python.

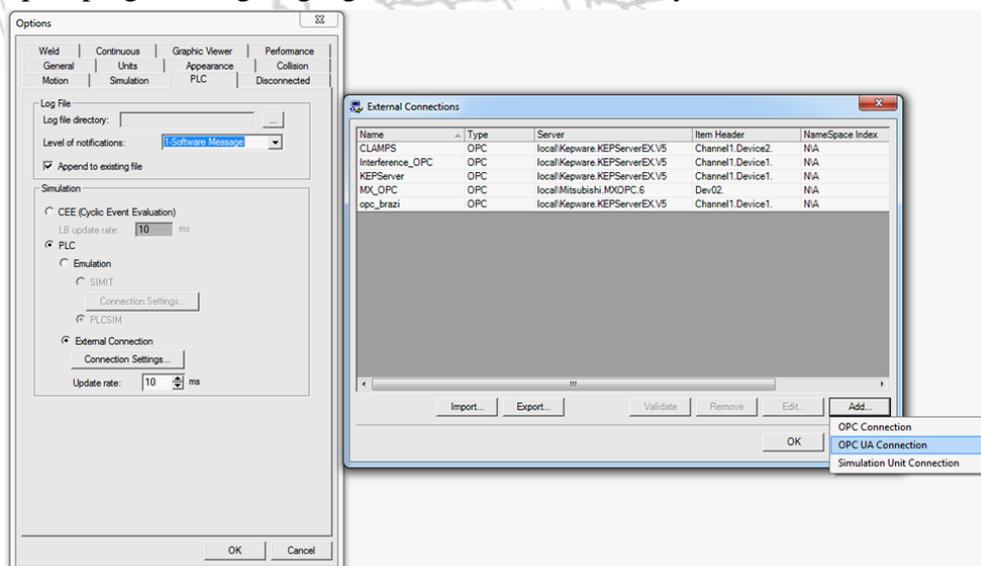


Fig. 12. Setting the external connection option for signals in Process Simulate.

Process Simulate has the ability to establish both types of connections, as illustrated in the Figure 12.

Another consideration regarding Process Simulate's connection capabilities is that of PLCSIM by Siemens. By using the Tia Portal or Simatic Manager, the user has the ability to create a virtual PLC and download the program to that virtual device. Because Process Simulate has access to the same protocol as the Siemens PLC it can connect directly to the Emulated PLC, without the need of a OPC Server. The advantage is that the communication is much faster, without any delays and the disadvantage is that the user is limited to only use Siemens Technologies.

The Figure 13 presents a basic schematic that can be referenced to any Virtual Commissioning configuration.

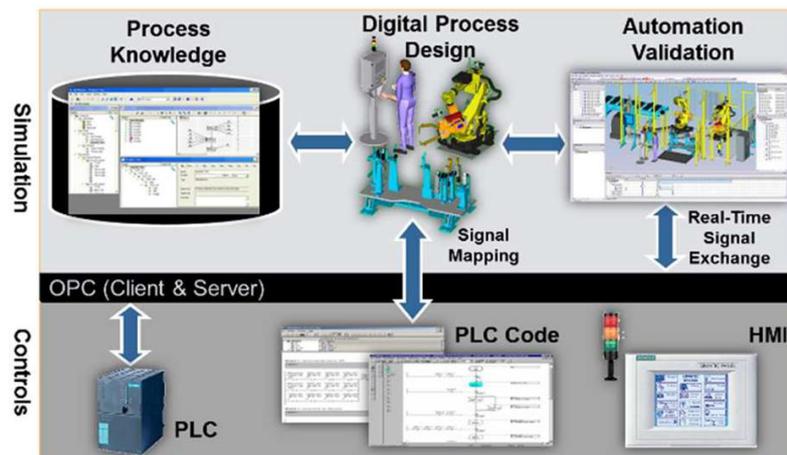


Fig. 13. Virtual Commissioning configuration.

By using all the mentioned elements, a simulation run with Virtual Commissioning can reach up to 100% accuracy. As such, when the real factory is commissioned, it is expected to run exactly as the simulation does.

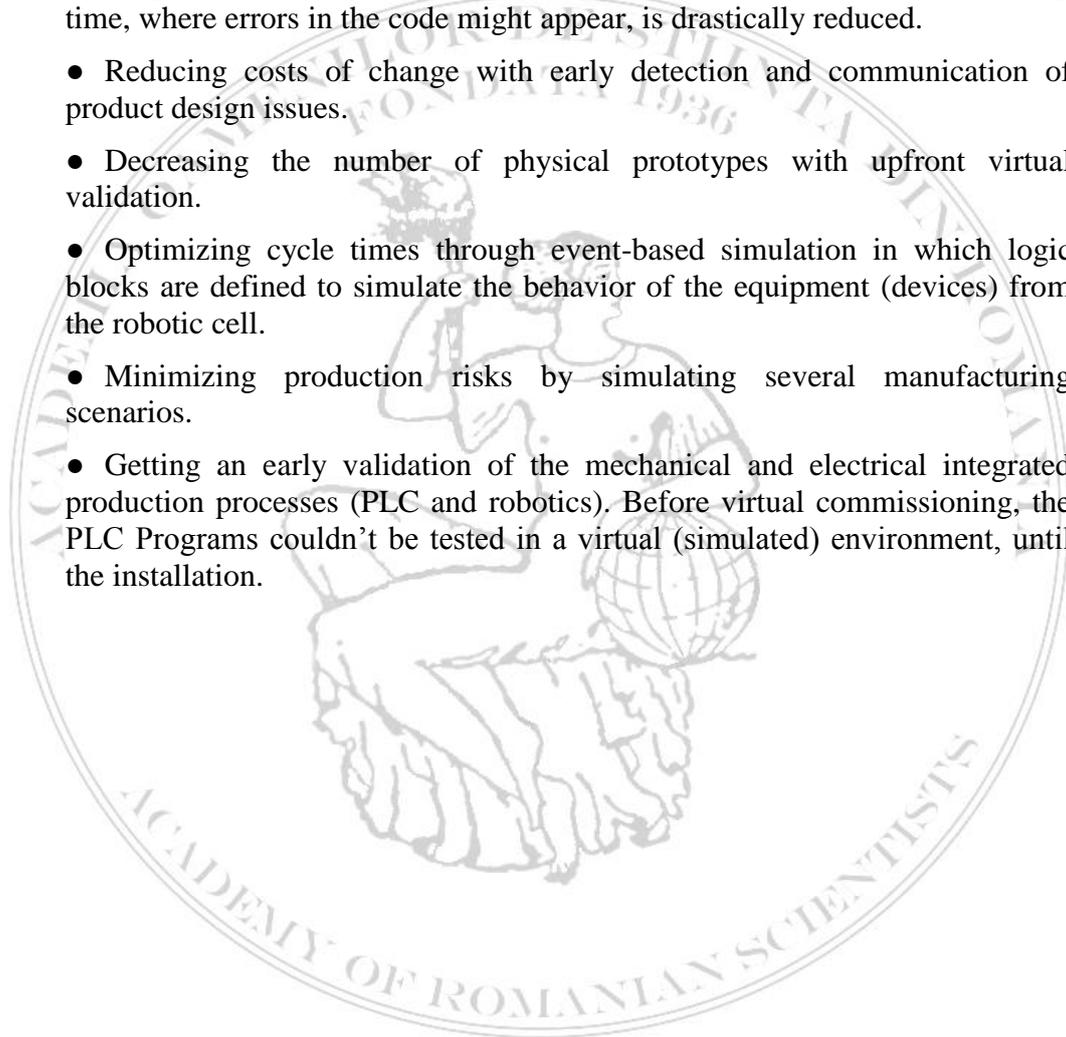
4. Contributions

In the recent years, the consumer has created the need for faster and more efficient productions. A good example is from the automotive industry, where car manufacturers are expected to release a model with several variations such as left or right handed, hard roof, sunroof, convertible, four seats, coupe and after some time to release a facelift model. With traditional commissioning executing this objecting would have been very difficult and time consuming, but with virtual commissioning manufacturers have the ability to optimize the process, see the effects of changes in real time and to validate the entire process before connecting to any actual physical devices.

5. Conclusions

The main benefits of the proposed environment setup of Virtual Commissioning, which provides a reliable process for validation of robotic manufacturing systems prior to physical installation, are:

- Decreasing ramp-up time which affects the total installation time up to 20-30 percent. Having validated all the programs through VC, the ramp-up time, where errors in the code might appear, is drastically reduced.
- Reducing costs of change with early detection and communication of product design issues.
- Decreasing the number of physical prototypes with upfront virtual validation.
- Optimizing cycle times through event-based simulation in which logic blocks are defined to simulate the behavior of the equipment (devices) from the robotic cell.
- Minimizing production risks by simulating several manufacturing scenarios.
- Getting an early validation of the mechanical and electrical integrated production processes (PLC and robotics). Before virtual commissioning, the PLC Programs couldn't be tested in a virtual (simulated) environment, until the installation.



REFERENCES

- [1] J. Stark, *Product Lifecycle Management – 21st Century Paradigm for Product Realisation*, Springer, **2004**.
- [2] H. Bley and C. Franke, *Integration of product design and assembly planning in the digital factory*, CIRP Annals, vol. 53, no. 1, pp. 25–30, **2004**.
- [3] R. Cooper, *Winning at New Products: Accelerating the Process from Idea to Launch*, Perseus Book Group, **2001**.
- [4] S. Dominka, F. Schiller, and S. Kain, *Hybrid commissioning - from Hardware-in-the-Loop simulation to real production plants*, Proceedings of the 18th IASTED International Conference on Modeling and Simulation (MS '07), pp. 544–549, Montreal, Canada, **2007**.
- [5] Zheng Liu, Nico Suchold, Christian Diedrich, *Virtual Commissioning of Automated Systems, Chapter 7* (CC BY, **2012**)
- [6] Siemens PLM Virtual Commissioning. Web p., <https://www.plm.automation.siemens.com/en/products/tecnomatix/launch-production/virtual-commissioning/>

