Virtual Commissioning of Industrial Robot through Software-in-the-Loop Strategies for Cyber-Physical Systems

Roman Ružarovský, Richard Skýpala, Nikolas Blahušiak, Ján Šido, Martin Csekei
Slovak University of Technology
Faculty of Materials Science and Technology in Trnava/ Institute of Production Technologies
Jana Bottu 25, 917 24 Trnava, Slovakia
{roman.ruzarovsky, Richard.skypala,xblahusiak, jan.sido, martin.csekei}@stuba.sk

Tibor Horák, Peter Střelec
Slovak University of Technology
Faculty of Materials Science and Technology in Trnava/ Institute of Applied Informatics, Automation and Mechatronics
Jana Bottu 25, 917 24 Trnava, Slovakia
{tibor.horak, peter.strelec}@stuba.sk

***Abstract**. This paper considers the virtual commissioning of an industrial robot. An industrial robot with peripherals is formed as a Cyber-Physical System. Virtual commissioning uses the Software-in-the-Loop method, where a co-simulation interfaces the emulation of a virtual controller from ABB RobotStudio with the digital twin of the industrial robot application in Tecnomatix Process Simulate through the VRC server. The objective was to provide a functional interconnection between the simulation model of a robot workstation and the control software for the industrial robot with peripherals. The result of this interfacing is to replace the RCS system and test the functionality of the proposed VRC system on a case study.*

Keywords. Virtual Commissioning, Industrial Robot, Software-in-the-Loop, Co-Simulation

1 Introduction

The digitalisation and virtualisation of industrial processes have made significant advances in recent years, transforming traditional production methods and processes. (Gašpar et al., 2023; Sharma et al., 2023) Digitisation enables the conversion of physical processes into digital form. Virtual models and simulations provide a detailed view of the functionality and behaviour of systems before they are deployed, allowing us to identify and correct potential problems in the early stages of development.

Virtual Commissioning (Pullnig, 2023) is the key tool in this process. It uses advanced software tools to create faithful digital twins of industrial equipment and systems that can be tested and optimised in detail in a simulated environment. This approach reduces the risks associated with physical installation and commissioning and significantly reduces the time and cost required to implement new technologies.

Industrial enterprises must integrate cyber-physical systems (CPS) that combine physical and digital components into a single whole as part of their digital transformation. (Jazdi, 2014) Such systems enable comprehensive real-time analysis and optimisation of manufacturing processes, thereby improving their performance and reliability. Virtual commissioning using Software-in-the-Loop (SiL) strategies is essential in this context. It enables realistic testing and debugging of systems, leading to smoother integration and higher quality of the resulting solutions.

Software in the Loop (SiL) is one of several methods used for virtual commissioning. SiL is the most effective method for testing and validating control software in a simulated environment where the software interacts with virtual models of equipment. (Ayed et al., 2017)It's particularly advantageous when physical components are not available or when their use is costly or impractical. SiL enables engineers to detect and fix software bugs before deployment in real operations, significantly reducing implementation costs and risks.

In addition to SIL, there are other methods such as Hardware-in-the-Loop (HiL) and Model-in-the-Loop (MiL). (Shylla et al., 2023) HiL integrates physical components into a simulated environment, allowing real devices to be tested under the control of simulated software. MIL, in turn, allows software testing using models of mathematical and physical systems. Each method has its own specific advantages and is suitable for different stages of development and testing.

SiL is particularly suitable for projects where detailed and repeated testing is required without the risk of damaging expensive physical equipment. It provides flexibility and the ability to iterate quickly, which is crucial for developing complex systems in a dynamically changing industrial environment.

There are two main ways of programming industrial robots. (Kelemen et al., 2018; Vagas & Romancik, 2023) One is online programming, which is ideal for small projects in SMEs where it is easy to manipulate products without a large number of peripherals, or in the case of robotic devices for arc welding, where workflows and products are often changed flexibly. In this case, the programmer connects to the real controller of the industrial robot, configures the communication with the peripherals and directly creates the program. (Kohrt et al., 2013; Marcinko et al., 2024)

The second way is the offline programming method (OLP, OLRP). OLP is the general term for offline programming, which is used for industrial robots. (Bilancia et al., 2023; Vocetka et al., 2020)OLRP is a method for programming industrial robots through software that includes a virtual controller. Offline Robot Programming (OLRP) is often used by integrators and manufacturers of whole robot solutions. This solution is an introduction to Virtual Commissioning. It allows the simulator to program, deploy and reprogram its solution several times without the use of a physical device. The program can be debugged on a virtual model. By programming robot functions in a virtual, offline environment (on a computer), OLRP eliminates the need to take robots out of production. This significantly increases productivity and the bottom line. The industrial market has a number of software solutions. You have two options: an OEM solution from the robot manufacturer (ABB RobotStudio, Fanuc RoboGuide, KUKAsim, Motosim, etc.) or an agnostic solution (RoboDK, Tecnomatix Process Simulate, Delmia, Robomaster, Visual Components, etc.).

2 Related Work and Background

The issue of virtual commissioning has been addressed by a number of authors. These contributions focus on several areas, on the creation of simulation models (J. Wang et al., 2023), behavioural models (Strahilov & Damrath, 2015; Süß et al., 2016), or for case studies (Ugarte et al., 2022). Most of the literature focuses on the PLC systems themselves, which is understandable given that these are the main focus of research in the field. However, if we are talking about IRC systems and virtual commissioning, the situation is different. The authors (Ben Ayed et al., 2017; Shylla et al., 2023) dealt with this issue, but used Open Source solutions such as ROS and Gazebo. Authors (Wallner et al., 2023) were interested in the concept of a robotic cell, but primarily with signals for the machine and not for the robot.

Before we deploy virtual commissioning for industrial robots, we must define and characterise the relevant terminologies and technologies.

- RC/ IRC: Robot Controller/ Industrial Robot Controller. Robotic controllers are an essential part of industrial robots. The robot's controller is a computer system that connects to the robot to control its movements and movements of the robot's arm. The controller also controls the end-effector and prevents interference from occurring within the robots work area. All industrial robot must be paired with a controller in order to operate. (Khatib & Siciliano, 2008)
- PLC: PLC stands for Programmable Logic Controller. They are the gold standard for controlling automated systems in industries. They are the pinnacle of control systems, and they are now replacing hard-wired logic relays at a large scale. (Bolton, 2006)
- Co-simulation: Co-simulation is an emerging enabling technique, where global simulation of a coupled system can be achieved by composing the simulations of its parts. (Gomes et al., 2018; Li et al., 2014)
- Emulation: Emulation is the use of an application, program, or device to imitate the behaviour of a different program or device. We use a PLCSim for the emluation of the real PLC or Virtual Controller ABB for industrial robot e.g. (McGregor, 2002; Mykoniatis & Harris, 2021)
- Behavioral model: Behavioral Model is specially designed to make us understand behavior and factors that influence behavior of a System. In our case, it is the components of a manufacturing system and a mathematical description of the behaviour of these components. (Stegmaier et al., 2023)
- RRS/ RCS: Realistic Robot/ Controller Simulation. This interface enables a standardized integration of motion software for robot controllers with simulators. Hence, the motion software of any robot manufacturer can be integrated into any simulation system. Today the RCS-Interface is the world-wide de-facto standard for precise simulation of robot motion behavior. (ROZITEK, 2023)
- VRC: Virtual Robot Controller. The VRC-Interface Specification aims at integrating robot controller software almost completely into simulators. Furthermore it supports the over-all data flow during the development and maintenance of installations with robots and automates several technical integration aspects. (Fraunhofer Institute for Production Systems and Design Technology, 2024)

When performing virtual commissioning with ABB robots, a connection between Process Simulate and RobotStudio can be created and used to simulate a complete virtual robot control (VRC) system. Process Simulate is designed to read and write information from and to RobotStudio. This includes joint values, signal values and other relevant data. This allows the simulation to visualise and interact with the VRC.

2.1 Defining the problem

The integration with the robot will be very tight thanks to the OEM solution. The advantage of this solution is that we use directly the virtual controller from the specific robot we want to use, including it's functionalities. However, these software are limited to direct programming of the robot and do not have other functionalities like large agnostic solutions. When integrating robots into production workplaces, it is necessary to maintain a precise cycle time. In this case, we need to have built-in RCS (Realistic Controller Simulation) modules to provide accurate cycle times. 100% accurate cycle times are important for certain industries such as automotive. Another concern is safety, collisions, dealing with singularities, etc.

The implementation process (calibrating the virtual world with the real world) is an extremely important part of the solution. OLRP is of little value if it does not guide the robot to exactly the right coordinates in the real world. The OLRP is a program generated for the robot via a postprocessor that contains the basic trajectories and motions of the robot, and often times we know from experience that the online roboticist will fine-tune the program on the jobsite or via an OEM robotics solution so that the robot interacts with the peripherals.

2.2 The aim of the work

The objective is to establish a seamless link between the robotic workstation simulator and the industrial robot controller, integrating inputs and outputs. This will leverage the benefits of both solutions. The outcome of this integration will be the replacement of the RCS system and the testing of the proposed VRC system in a real-world case study.

3 Materials and Methods

We will use two methods to link and verify functionality. The first method is the co-simulation method (Hasmukhbhai et al., 2021), which will create a joint simulation of the robotic workstation on two simulators. The first one will provide the simulation of the robotic workstation in the Tecnomatix Process Simulate environment. It is necessary to create a full model of the workplace with the industrial robot and its peripherals and to prepare the model for Virtual Commissioning. This means that it is necessary to create the necessary signals for the control of the industrial robot and the input-output commutation with the pefiferia such as the robotic tool. On the other hand, it is necessary to create a simulation model for the ABB RobotStudio industrial robot with the necessary specification and the created virtual controller. The time synchronization in the co-simulation between the simulators will be taken care of by the co-simulation algorithm itself. (Sadjina et al., 2019; L. Wang et al., 2023) The interactions between these subsimulators are synchronized only at discrete communication points. The sub-simulators can run in parallel, which can speed up simulations of the whole system. Models can be efficiently developed in parallel and can be easily reused. Ability to include hardware, software, and human in the loop relatively easily and consistently. Due to its features and benefits, co-simulation is best used for complex and heterogeneous cyber-physical systems.

4 Development of the Virtual Commissioning Model

The model workplace will be developed in four basic steps. The first step is to create a model robotic workstation in the Tecnomatix Process Simulate environment. The second step is to create behavioural models for the individual peripherals and the industrial robot. We then distribute the individual control and command signals created for the individual connections to the virtual controllers, using the Software-In-the-Loop (SiL) method in Virtual Commissioning. The last step is to connect the virtual controller from the ABB RobotStudio environment via the VRC server and test the co-simulation.

4.1 Simulation model of a robotic workplace

To operate an industrial robot and its workplace and create a control programme for the robot and its controller using the OLRP method, you have to model the workplace. This is done using CAD, where you create fixture models, end-effector models and other components if they are not already in the database.

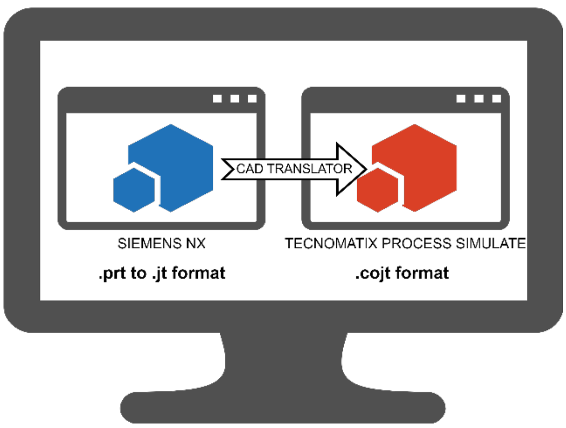

Figure 1. General procedure for the creation of models for simulation

The manufacturer of the single-label markers provides models of industrial robots. These models are ready for simulation in both types of software, whether OEM or agnostic large solution. If we have the models available, we need to translate them via CAD translator into the necessary format for the simulation. In our case it is the *.jt format, Fig. 1.

The entire simulation model of the workplace is designed to verify the collision-free nature of the workplace, the reaches of industrial robots, and to generate robot trajectories and motions for a specific manufacturing application. The simulation model works with a virtual controller and MOP (Motion Planner) that can control robot motion including individual axis, speed and zone configurations. The entire robotic program can be translated through the postprocessor, where it is necessary to have a virtual robot controller installed for ABB robots. This can be characterized as offline programming of industrial robots, Fig. 2.

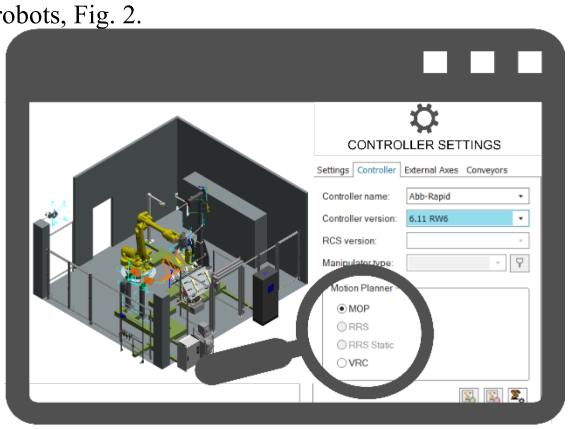

Figure 2. MOP controller setup for the robot

4.2 Behavioural model for VC

The entire simulation model can be upgraded to a higher level and prepared for virtual commissioning of the system. Virtual commissioning is the process of using virtual simulation technology to commission - design, install or test control software with a virtual model of a robotic workstation before connecting it to the real system. The created workstation can be used to simulate real robot control code and program in combination with emulated control hardware (SiL) or in combination with real hardware (HiL) in a realistic virtual commissioning environment to verify complete system function before production starts.

The kinematic models created for the implementation of the robotic workplace simulation and offline programming of the industrial robot are the basis for the implementation of the behavioral models for Virtual Commissioning. Virtual commissioning of a robotic system brings together three important parts: the digital model (sometimes referred to as the digital twin), the controller code that controls motion and responds to sensor feedback, and the development environment that enables them to run together. The digital twin needs a behavioural model, according to our research. Models that represent the temporal and logical behavior of a production device in relation to the corresponding control system are referred to as behavioral models. (Skýpala & Ružarovský, 2022; Striffler & Voigt, 2023) Behavioural models are modelled in a separate simulation and modelling tool, which is linked to the control at the input and output level. From a management perspective, behavioural models represent a 'black box'. The behaviour models also serve as a link between the control system and the extended geometry/kinematics mode. (Kiefer et al., 2009) In our case, we created a behavioral model in the Tecnomatix Process Simulate LB environment, where we defined Entries, Exits, Parameters and Constants for the individual kinematic models, Fig. 3.

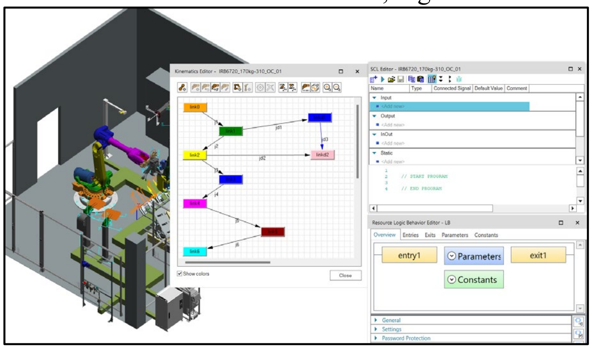

Figure 3. Kinematic model of robot and creation of behavioural model

Based on the nature of the motion and behavior of the model, we generated inputs and outputs for model control, individual variables and data types. The same had to be done for the robot control except for its peripherals. As can be seen, we thus generated 4 inputs and 5 outputs, Fig. 4.

InterfaceName	RobotInternalName	I_Q	TYPE	Comment
HX165_hub40_startProgram	startProgram	Q	BOOL	(Starting Program)
HX165_hub40_programNumber	programNumber	Q	BYTE	(Program Number)
HX165_hub40_emergencyStop	emergencyStop	Q	BOOL	(Program Emergency Stop)
HX165_hub40_programPause	programPause	Q	BOOL	(Program Pause)
HX165_hub40_reduceSpeed	reduceSpeed	Q	BYTE	(Reducing Speed)
HX165_hub40_programEnded	programEnded	I	BOOL	(Ending Program)
HX165_hub40_mirrorProgramNumber	mirrorProgramNumber	I	BYTE	(Mirror Program Number)
HX165_hub40_errorProgramNumber	errorProgramNumber	I	BOOL	(Error Program Number)
HX165_hub40_robotReady	robotReady	I	BOOL	(Robot Ready)

Figure 4. Created signals for robot control

The behaviour model is thus ready for Virtual Commissioning. In our case, we have currently prepared the model only for communication between the virtual controller emulator and the robotic workstation simulation model, in order to verify the functionality of co-simulation and to use the proposed VRC system instead of the MOP or RCS system. The latter needs to be installed and run. If the robot name in Process Simulate and the controller name in RobotStudio do not match, a new string parameter called OLP_VRC_NAME must be created in Process Simulate on the corresponding robot instance and set to the controller name in RobotStudio. When Process Simulate attempts to find a matching VRC instance for the robot, it will attempt to do so using the value of the parameter in addition to the robot name. In this case, either the <RobotName> in Process Simulate or the <value OLP_VRC_NAME> must match the name of the VRC instance.

4.3 Virtual controller emulation

As mentioned, the simulation model must be prepared to communicate with the VRC instance. A new folder RobotParameterEditor was created in the Abb-Rapid folder at \eMPower\Robotics\OLP\.

```
<RobotController Name="Abb-Rapid">
   <RoboticParams>
      <Param Name="OLP_VRC_NAME" ValueType="String" />
   </RoboticParams>
</RobotController>
```

Figure 5. Created a folder for communication

At the same time set other parameters for communication. Communication is done on localhost via communication port 8523 and the service runs on Tecnomatix VRC Server ABB Real Time. Each robot instance in Process Simulate (and therefore each VRC instance) requires a unique port. An emulation of the virtual controller is then created in ABB RobotStudio. In RobotStudio, the program or module must exist as a global procedure nested under the robot's TASK1. The operation can be completely empty (no OLP commands or locations) or it can have nested information underneath, but the robot only executes the code stored in RobotStudio. We can use the Process Simulate Upload Programs and Download to Robot commands to exchange programs between Process Simulate and RobotStudio.

4.4 Testing the functionality of co-simulation

Tecnomatix Virtual Robot Controller Server ABB Real Time provides communication between Process Simulate and RobotStudio VRC. The server allows you to execute robot programs using the ABB VRC, which is part of RobotStudio, while connecting to Process Simulate for simulation. For proper functioning, several parameters need to be set for the server to run on localhost, the parameters need to be set on the simulation side of the VRC and the same on the ABB virtual emulator, and an important point is to have the same name for both controllers on both sides.

5 Results

The communication between the two controllers was successful.

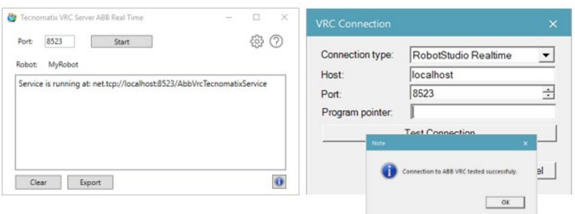

Figure 6. Successful communication via the configured VRC server

We sent several data between the two controllers in two software, the robot moved according to the instructions we entered through the RAPID control program. (ABB Robotics, 2004) We monitored the response, but it was not measured, we could monitor it currently through the ABB Real Time application. The delay was felt just when the 3D simulation model was complex and the problem was on the hardware performance side.

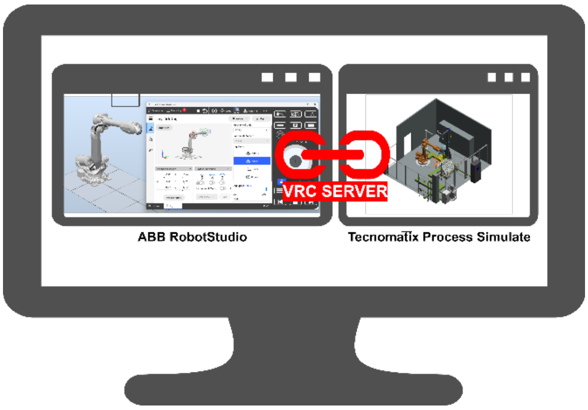

Figure 7. Communication between the emulated virtual controller and the simulated model via VRC

Overall, the testing was successful, but future testing needs to be done. There is no guarantee that all the different parts of the joint simulation will work well together. So far there is no general and easy to use co-simulation solution (software or algorithm). This is mainly due to technical problems and the fact that it is very difficult to define a universal method that works for all kinds of situations and requirements.

Co-simulation in most cases requires sufficient knowledge at the system level and potentially also knowledge of the individual models. It is not easy to determine where and how to partition the system into sub-models, leading to a trade-off between modularity, accuracy and performance. This has far-reaching implications for sub-model development, collaboration and ease of use.

It is also important to consider how to define links between sub-models. In addition, it is necessary to define how computational causality between sub-models is defined.There is a wide range of co-simulation algorithms to choose from, ranging from very simple (which may be slow and inaccurate) to quite complex (which may be difficult to tune and may have other requirements). The simplest co-simulation algorithms support only constant communication intervals, so the user must choose the correct interval for a given simulation.

5 Conclusion and Future Work

Our project involved co-simulation between a robot simulation model and a virtual controller emulator. Testing was carried out using the Virtual

Commissioning and Software-in-the-Loop methods. Virtual Commissioning is the best way to create a digital twin of the system, thus enabling testing and verification of functionality prior to its physical deployment. This involves the creation of detailed models of the behaviour of individual components and their interactions, thus reducing the risk of errors and increasing reliability.

We created behavioural models for this project to simulate the robot's motion in detail, including its movements, speed, acceleration and interaction with the environment. These models included mathematical representations of the control algorithms that govern the robot's movements. For example, the simulations included controlling the robot's arm motion, its interaction with objects, and its responses to various sensory inputs.

The virtual controller emulator has been designed to faithfully mimic the logic and responses of a real controller. This included the processing of control commands, synchronization with the robot simulation model, and consideration of time delays that may arise in real control.

Testing using the Software-in-the-Loop method allowed us to simulate the control algorithms in a software environment before implementing them in a physical controller. This approach provided the ability to thoroughly verify and optimize the control algorithms in a controlled virtual environment, minimizing the risk of errors when deployed in a real system. Preliminary testing results indicate that the implementation is functional, however, more detailed testing will need to be performed in the future for full confirmation. It is also necessary to validate the communication at the signal level through a PLC emulation such as PLCSim Advanced. This would confirm the relevance of deploying the Virtual Commissioning method for industrial robots and their peripherals as cyber-physical systems.

Acknowledgments

This research was funded by the Scientific Grant Agency of the Ministry of Education, Science, Research, and Sport of the Slovak Republic and the Slovak Academy of Sciences, grant number VEGA 1/0391/24 Research and development of sensor calibration methodology for diagnostic equipment for automated and robotic systems.

References

Ayed, M. Ben, Zouari, L., & Abid, M. (2017). Software In the Loop Simulation for Robot Manipulators. *Engineering, Technology & Applied Science Research*, *7*(5), 2017–2021. https://doi.org/10.48084/ETASR.1285

Ben Ayed, M., Zouari, L., & Abid, M. (2017). Software In the Loop Simulation for Robot Manipulators. *Engineering, Technology & Applied Science Research*, *7*(5), 2017–2021. https://doi.org/10.48084/ETASR.1285

Bilancia, P., Schmidt, J., Raffaeli, R., Peruzzini, M., & Pellicciari, M. (2023). An Overview of Industrial Robots Control and Programming Approaches. *Applied Sciences 2023, Vol. 13, Page 2582*, *13*(4), 2582. https://doi.org/10.3390/APP13042582

Bolton, W. (2006). Programmable Logic Controllers. *Programmable Logic Controllers, Fourth Edition*, 1–292. https://doi.org/10.1016/B978-0-7506-8112-4.X5018-9

Fraunhofer Institute for Production Systems and Design Technology. (2024). *Realistic Robot Simulation (RRS) - Fraunhofer IPK*. https://www.ipk.fraunhofer.de/en/expertise-and-technologies/automation/realistic-robot-simulation-rrs.html

Gašpar, G., Dudak, J., Sladek, I., Zuziakova, I., & Meravy, T. (2023). Utilization of Open-Source in Upgrading Technological Devices for Industry 4.0 readiness. *Central European Conference on Information and Intelligent Systems; Varazdin*. https://www.proquest.com/docview/2876106571?pq-origsite=gscholar&fromopenview=true&sourcetype=Conference%20Papers%20&%20Proceedings

Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2018). Co-Simulation. *ACM Computing Surveys (CSUR)*, *51*(3). https://doi.org/10.1145/3179993

Hasmukhbhai, S., Le Henaff, P., Schiffer, C., Krammer, M., & Benedikt, M. (2021). View of Accurate Robot Simulation for Industrial Manufacturing Processes using FMI and DCP Standards. *Proceedings of the 14thInternational Modelica ConferenceSeptember 20-24, 2021*. https://www.ecp.ep.liu.se/index.php/modelica/article/view/247/207

Jazdi, N. (2014). Cyber physical systems in the context of Industry 4.0. *Proceedings of 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2014*. https://doi.org/10.1109/AQTR.2014.6857843

Kelemen, M., Virgala, I., Lipták, T., Miková, L., Filakovský, F., & Bulej, V. (2018). A Novel Approach for a Inverse Kinematics Solution of a Redundant Manipulator. *Applied Sciences 2018, Vol. 8, Page 2229*, *8*(11), 2229. https://doi.org/10.3390/APP8112229

Khatib, Oussama., & Siciliano, Bruno. (2008). *Springer Handbook of Robotics*.

Kiefer, J., Ollinger, L., & Bergert, M. (2009). Virtuelle Inbetriebnahme - Standardisierte Verhaltensmodellierung mechatronischer Betriebsmittel im automobilen Karosserierohbau.

Atp Magazin, *51*(07), 40–46. https://doi.org/10.17560/ATP.V51I07.2000
Kohrt, C., Stamp, R., Pipe, A. G., Kiely, J., & Schiedermeier, G. (2013). An online robot trajectory planning and programming support system for industrial use. *Robotics and Computer-Integrated Manufacturing*, *29*(1), 71–79. https://doi.org/10.1016/J.RCIM.2012.07.010
Li, W., Zhang, X., & Li, H. (2014). Co-simulation platforms for co-design of networked control systems: An overview. *Control Engineering Practice*, *23*(1), 44–56. https://doi.org/10.1016/J.CONENGPRAC.2013.10.010
Marcinko, P., Semjon, J., Jánoš, R., Svetlík, J., Sukop, M., & Ondočko, Š. (2024). Analysis of the Methodology for Experimental Measuring of the Performance Criteria of the Laser-Using Collaborative Robot's Path Accuracy. *Applied Sciences 2024, Vol. 14, Page 1414*, *14*(4), 1414. https://doi.org/10.3390/APP14041414
McGregor, I. (2002). The relationship between simulation and emulation. *Winter Simulation Conference Proceedings*, *2*, 1683–1688. https://doi.org/10.1109/WSC.2002.1166451
Mykoniatis, K., & Harris, G. A. (2021). A digital twin emulator of a modular production system using a data-driven hybrid modeling and simulation approach. *Journal of Intelligent Manufacturing*, *32*(7), 1899–1911. https://doi.org/10.1007/S10845-020-01724-5/TABLES/3
Pullnig, M. (2023). *Virtual Comissioning: Process Optimization*. https://www.3dfindit.com/en/engiclopedia/virtual-commissioning
ROZITEK. (2023). *The ultimate guide to Robot Control System (RCS) - Rozitek*. https://rozitek.com/the-ultimate-guide-to-robot-control-system-rcs/
Sadjina, S., Kyllingstad, L. T., Rindarøy, M., Skjong, S., Esøy, V., & Pedersen, E. (2019). Distributed co-simulation of maritime systems and operations. *Journal of Offshore Mechanics and Arctic Engineering*, *141*(1). https://doi.org/10.1115/1.4040473/366241
Sharma, M., Raut, R. D., Sehrawat, R., & Ishizaka, A. (2023). Digitalisation of manufacturing operations: The influential role of organisational, social, environmental, and technological impediments. *Expert Systems with Applications*, *211*, 118501. https://doi.org/10.1016/J.ESWA.2022.118501
Shylla, D., Jain, A., Shah, P., & Sekhar, R. (2023). Model in Loop (MIL), Software in Loop (SIL) and Hardware in Loop (HIL) Testing in MBD. *2023 4th IEEE Global Conference for Advancement in Technology, GCAT 2023*. https://doi.org/10.1109/GCAT59970.2023.10353323
Skýpala, R., & Ružarovský, R. (2022). The Role of a Behavioural Model for the Virtual Commissioning of Robotic Manufacturing Systems. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, *30*(50), 45–52. https://doi.org/10.2478/RPUT-2022-0006
Stegmaier, V., Eberhardt, T., Schaaf, W., Jazdi, N., & Weyrich, M. (2023). A behavior model for Digital Twins of vacuum suction cups. *Procedia CIRP*, *118*, 958–963. https://doi.org/10.1016/J.PROCIR.2023.06.165
Strahilov, A., & Damrath, F. (2015). Simulation of the behavior of pneumatic drives for virtual commissioning of automated assembly systems. *Robotics and Computer-Integrated Manufacturing*, *36*, 101–108. https://doi.org/10.1016/J.RCIM.2015.01.001
Striffler, N., & Voigt, T. (2023). Concepts and trends of virtual commissioning – A comprehensive review. *Journal of Manufacturing Systems*, *71*, 664–680. https://doi.org/10.1016/J.JMSY.2023.10.013
Süß, S., Hauf, D., Strahilov, A., & Diedrich, C. (2016). Standardized Classification and Interfaces of Complex Behaviour Models in Virtual Commissioning. *Procedia CIRP*, *52*, 24–29. https://doi.org/10.1016/J.PROCIR.2016.07.049
Ugarte, M., Etxeberria, L., Unamuno, G., Bellanco, J. L., & Ugalde, E. (2022). Implementation of Digital Twin-based Virtual Commissioning in Machine Tool Manufacturing. *Procedia Computer Science*, *200*, 527–536. https://doi.org/10.1016/J.PROCS.2022.01.250
Vagas, M., & Romancik, J. (2023). Calibration of an intuitive machine vision system based on an external high-performance image processing unit. *Proceedings of the 2023 24th International Conference on Process Control, PC 2023*, 186–191. https://doi.org/10.1109/PC58330.2023.10217606
Vocetka, M., Huňady, R., Hagara, M., Bobovský, Z., Kot, T., & Krys, V. (2020). Influence of the Approach Direction on the Repeatability of an Industrial Robot. *Applied Sciences 2020, Vol. 10, Page 8714*, *10*(23), 8714. https://doi.org/10.3390/APP10238714
Wallner, B., Trautner, T., & Bleicher, F. (2023). Functionality test methodology for virtual commissioning of reconfigurable manufacturing systems. *Procedia CIRP*, *118*, 56–61. https://doi.org/10.1016/J.PROCIR.2023.06.011
Wang, J., Niu, X., Gao, R. X., Huang, Z., & Xue, R. (2023). Digital twin-driven virtual commissioning of machine tool. *Robotics and Computer-Integrated Manufacturing*, *81*, 102499. https://doi.org/10.1016/J.RCIM.2022.102499
Wang, L., Wang, Z., Kendall, P., Gumma, K., Turner, A., & Ratchev, S. (2023). Deep Dynamic Layout Optimization of Photogrammetry Camera Position

Based on Digital Twin. *IEEE Transactions on Automation Science and Engineering*. https://doi.org/10.1109/TASE.2023.3323088