

# The application of virtual reality in programming of a manufacturing cell

Halldor Arnarson<sup>1</sup>, Bjørn Solvang<sup>2</sup> and Beibei Shu<sup>3</sup>

**Abstract**—Programming of industrial robots and manufacturing equipment in general requires product specific expertise towards all members in a manufacturing cell. Typically, old and new equipment is present in the same setup and several experts are often involved in the operation of these systems. In order to decrease the level of complexity in programming of manufacturing equipment this paper investigates the use of virtual reality (VR), to create a common programming platform for typical members in a manufacturing setup. A two-way digital twin is created where all robots can be programmed through the same human-machine-interface (HMI). This cyber-physical system (CPS) allows for simulations, testing and safety checks before all programs are converted and downloaded to the respective units.

## I. INTRODUCTION

Humans and industrial robots are the core elements of industry 4.0. While robots are designed to be strong, fast and experts on repetitive work, humans sit on all the experience and knowledge. Efficient communication between humans and robots, in a manufacturing cell or factory, is crucial in order to reach the full potential of industry 4.0.

Using virtual reality (VR) to interact with a virtual robot can be a cost-effective method for acquiring robot programming expertise. Being able to test trajectories in VR removes risks and increases safety in industrial facilities [1].

VR can be defined as a “synthetic or virtual environment which gives a person a sense of reality” [2]. With the rapid growth of VR technology it has become more affordable and capable at the same time [3].

As the industry of today is moving from mass production towards more personalized and small sized batch production [4]. This means more adjustments to robot programs, which can be time-consuming and expensive when developing complex tasks. High mix and low volume production require frequent re-programming of robots, making it difficult for small and medium enterprises (SMEs) to invest in robots [5]. Using VR for robot programming of complex tasks can give the operator a more straightforward and intuitive tool [6]. For instance when painting, grinding, cleaning, handling and assembly operations [7].

VR is an excellent tool for configuration of digital twin systems [7]. These systems can program an industrial robot

in a VR environment and then execute the program on the physical robot [8, 9].

VR programming is not only limited to digital twin systems but can also be used for cyber-physical systems (CPS). CPS aims to integrate the physical world with the information age [10], where VR can be used as an interface between the user and CPS. The user can simulate and interactively explore the CPS with VR [11].

An example of combining VR with a CPS is from [12], where an old physical robot and human is connected to a virtual environment. Then the collision detection in the virtual environment is used to detect if there is a collision between the human and robot. This again allows for human-robot collaboration (HRC).

Another example when VR is used in a CPS system is found in [13] where VR is used with human intention recognition (IR) algorithm to perform robot-assisted welding. The result shows that using VR and IR to preform welding increases effectiveness and performance and removes humans from labor-intensive and dangerous work.

When using VR with a CPS or digital twin there has to be a software/program to connect the VR headset and the 3D environment. Most of the researchers working with VR are using the Unity game engine to develop the VR simulation [4, 14, 15]. It is also possible to connect the Robot Operating System (ROS) [16] to the Unity engine and allow for VR functionality for the virtual environment created in ROS [17].

However, more manufacturing and robotics simulation software are starting to include VR, such as Visual Components [18], RoboDK [19] and ABB RobotStudio [20].

Manufacturing software includes extensive libraries with robots and other manufacturing machinery compared to Unity. In the more specialized software there is also built-in advanced kinematics models and simple tools for programming robots and machines. From such point of view it can be more efficient to use manufacturing software, rather than the general Unity game engine.

ABB RobotStudio has built-in functionality for programming of robots. It is possible to create paths where speed and orientation are selected, as well as editing already created paths for fine-tuning [21]. However, since RobotStudio is made by ABB, it is only supported by ABB robots [22]. It will be more challenging to use this software in a CPS system that contains a mixture of different robots and machines out of the ABB product range.

Another example is RoboDK, which has added VR functionality into their software [23]. RoboDK is often used for offline programming of robots [24] and has an extensive

<sup>1</sup>Halldor Arnarson is with Department of Industrial Engineering, UiT The Arctic University of Norway, Narvik Norway [halldor.arnarson@uit.no](mailto:halldor.arnarson@uit.no)

<sup>2</sup>Bjørn Solvang is with Department of Industrial Engineering, UiT The Arctic University of Norway, Narvik Norway [bjorn.solvang@uit.no](mailto:bjorn.solvang@uit.no)

<sup>3</sup>Beibei Shu is with Department of Industrial Engineering, UiT The Arctic University of Norway, Narvik Norway [beibei.shu@uit.no](mailto:beibei.shu@uit.no)

library from various brands [25]. It should be noted that RoboDK is used for programming of small systems with one to three robots and we don't find it particular suitable in larger CPS systems.

Visual Component has an extensive library with almost all modern robots as well as conveyors, feeders and different varieties of manufacturing machines. In the latest update of Visual Components (version 4.2), there has been added an interactive virtual reality function [18]. This update provides jogging of the robots and interacting with objects in virtual reality.

The default interactions are very simple and do not allow for programming of a integrated complex system. However, there is a python API for advanced users [26] that adds more functionality.

There are advantages and disadvantages to the simulation software mentioned above. However for this project, Visual Components is selected since it supports almost all modern robots and works well with large systems.

As mentioned above there has been developed VR-controlled CPS. However "the future of manufacturing requires interaction between humans and multiple types of robots, and between physical and virtual scenarios" [27]. Thus in this paper we will explore VR control and programming of a system with various robots from different manufacturing brands while also using IIoT. First this paper will present our experimental setup and go through how the VR model is created and how it works. Afterwards we explain the connections between the physical system and the virtual model. Finally we present a few demonstrations, on both component and system level.

## II. SYSTEM SETUP

As seen in figure 1, the experimental setup is divided into two parts - a physical and a virtual environment. The physical part includes the components listed in table I and covers a medium and a small industrial robot, an assembly (SCARA) robot, an automated intelligent vehicle (AIV), and a rotary table. All these machines were connected to an Industrial Internet of Things (IIoT) server using the Open Platform Communications Unified Architecture (OPC UA) [28]. OPC UA is an IEC standard (IEC 62541) [29], that can run on Windows, Apple OSX, Android, Linux and more [28]. It is also supported by SCADA tools [30] and simulation programs, such as Visual Component [31].

The second part in our setup is the interactive virtual environment. This includes a virtual model of the laboratory with all the machines mentioned above. The virtual environment has been developed in Visual Components, which has built-in connectivity functionality. Such functionality was used to connect the virtual environment to the OPC UA server as a client. This allows for easy paring of variables inside the simulation with variables in the OPC UA server.

To run the simulation in VR, the program Visual Components Experience [32] was used and it connects to the Visual Components simulation through a local IP address. All the software components used in this setup are listed in table I.



Fig. 1. Experimental system, components and connectivity.

The setup is compatible with any VR headset that supports Steam VR [33]. During the testing, a wireless Oculus Quest headset [34] was used.

## III. VISUAL COMPONENTS VR

Visual Components is a powerful 3D manufacturing simulation software that can be used in design and validation of production systems. It has an open architecture, and simulations are easy to customize. It is possible to import CAD models to the simulation and there is a Python API that can be used to control the simulation [35].

### A. Virtual model

A replica model of the laboratory has already been created in Visual Components [36], as shown in figure 2. This model includes all the machines listed in table I.

### B. Control interface

To be able to program the machines in VR, a control interface or Human Machine Interface (HMI) was created. An interactive HMI has been added in the latest update, as

TABLE I  
ROBOTS/MACHINES AND SOFTWARE SETUP.

Robot/machine name	Description
KUKA KR30-3	6 axis robot arm
KUKA DK400-V	Rotary table
Nachi MZ07	6 axis robot arm
Scara Adept 604-S	4 axis robot arm
MiR100	Automated intelligent vehicle (AIV)
Software	Function
Visual Components	Simulation and monitoring
Visual Components Experience	To use VR in Visual Components
OPC UA standard	Network server
Python	To create control programs

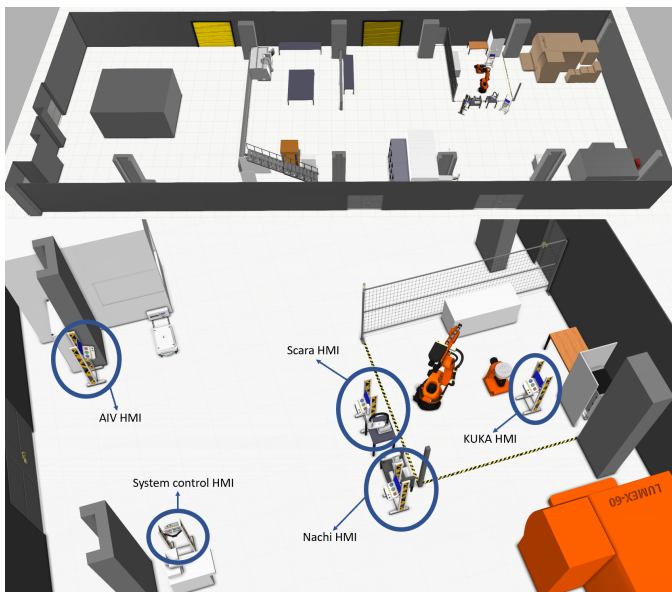


Fig. 2. Upper part: The overall virtual model of our laboratory. Lower part: Key-components and their initial placement.

shown in figure 3. The HMI can be customized and buttons can easily be added or removed.

The HMI was used to program each of the machines listed in table I, with a point to point (PTP) motion. Currently, it is possible to add and store eight positions to the path of each machine. However, more buttons can easily be added, in order to increase the number of points, as well as to create more advanced functionality.

### C. Programming in VR

To add functionality to the buttons, a Python script was added to each of the HMI interfaces. The Python code can save, delete, run the program, as well as transfer the program from Visual Components to the OPC UA server.

The functionality of the buttons is listed below:

- Blue button: Save and add current position (pose) to path
- Green button: Execute the program in the virtual environment
- Red button: Delete position

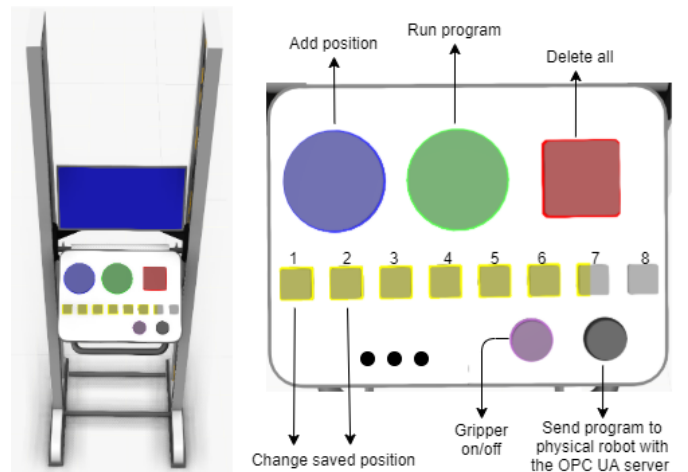


Fig. 3. The left side of the figure shows the HMI interface, which can be used with VR. The right side is a close-up of the buttons and a description of what the buttons do.

- Yellow buttons: Each of them represents a saved position. These are used to check or change the sequence of the PTP motion.
- Black button: Transfer and RUN. Sends the program to the OPC UA server and runs the physical device
- Purple button: Gripper on/off

There is a total of five machines in our setup. There is one HMI table for each of the NACHI, SCARA, and AIV. The KUKA robot and rotary table are usually operated together and thus have a common HMI table. Figure 2 shows the placement of the HMI tables with the robots.

The right-hand VR controller was used to interact with the machines. The robot arms and rotary table can be jogged by using the right trigger on the VR controller. As seen in figure 4 there is a coordinate system at the end of the controller and this coordinate system was used when jogging the robots in VR.

### D. AIV programming

By pressing the blue button the user starts the programming of the AIV. Next, they navigate in the Visual Components model and leave position markers (yellow bars) each time the A button is pressed. Finally, they move the AIV to each position marker. Figure 5 shows a programming sequence of the AIV.

### E. System programming

As mentioned above, all the robots were connected to the same OPC UA server. This made it possible to program all the machines in one sequence, and for this, we added an extra HMI. It works similarly to the AIV interface, by first pressing the blue button to start the programming. Then the corresponding HMI interface for the different machines is used to add new positions. Finally, the program is executed by pressing the green button and the robots will move to its locations in the same order as the positions where added.

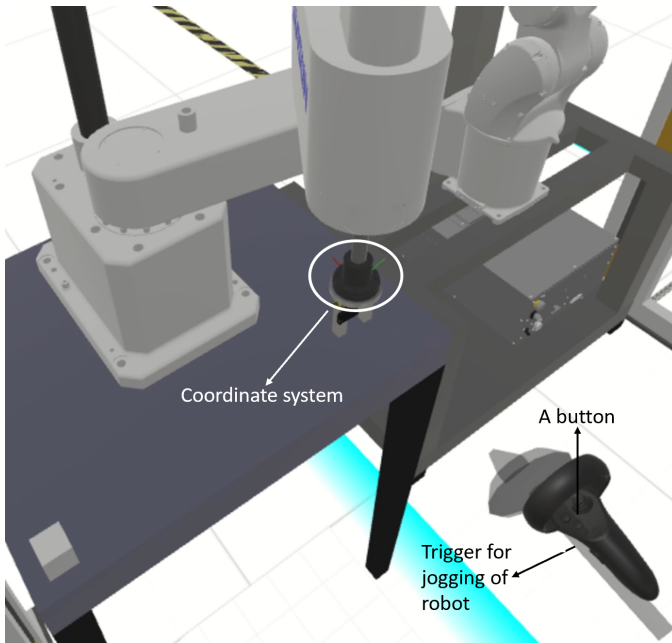


Fig. 4. The figure displays the buttons on the controller and coordinate system used for the jogging of the robots.

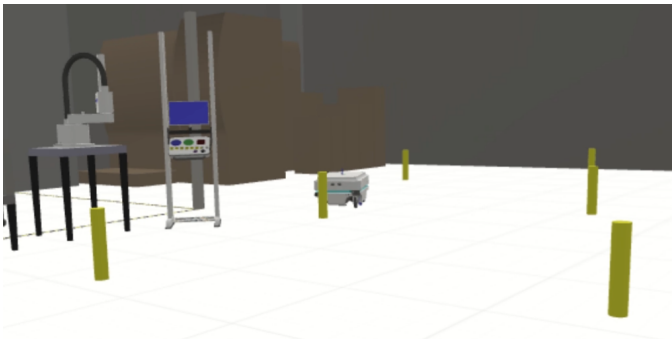


Fig. 5. Illustration of how the AIV is programmed, where the yellow marker/cylinder represents the positions the AIV will move to.

#### F. Connection to physical robots

The robots were connected to an OPC UA server, which allowed for control and monitoring of all components.

Two variables have been added for each of the HMI interfaces in the OPC UA server, as can be seen in figure 6. The variable that ends with "\_positions", is where all the positions are stored and the variable that ends with "start\_stop" is used for starting and stopping program execution of the physical robot.

When the black button is pressed on the HMI interface in Visual Components the positions created in VR are saved to the OPC UA server.

To run the program on the physical robot the variable that ends with "start\_stop" has to be set to "1". When the program is finished executing, the variable will automatically be set to "0". The program created in VR can be executed multiple times on the physical robot.

To simplify the execution of the program, an OPC UA client Android app called "DataFEED OPC UA Client" [38]

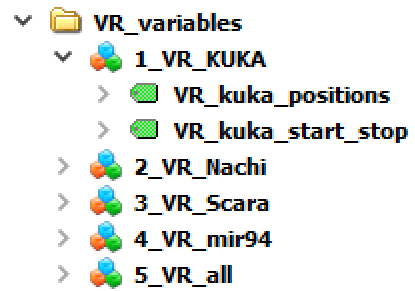


Fig. 6. A screenshot from the the OPC UA server, showing the variables used to send data from Visual Components to the OPC UA server. The program UA Expert [37] is used to view the data in the OPC UA server.

is used to modify the "start\_stop" variable. This makes it possible to program the system in VR, and then start the program on the physical system with an android phone.

#### IV. DEMONSTRATION

In order to demonstrate the VR programming methodology, five videos have been created, as described in table II. This includes separate videos for the NACHI, SCARA, AIV, and the KUKA system (robot and rotary table), and finally a video that showcases programming of the whole system. It should be noted that only the joint positions are sent to the OPC UA server. The speed and details on how the robot should move are not sent, and movements in the virtual environment can somewhat differ from the movement on the actual robot.

TABLE II

THE TABLE LISTS UP THE FIVE VIDEOS THAT HAVE BEEN CREATED AND INCLUDES A LINK TO THE VIDEOS.

Videos from the demonstration	
Description:	Link to video:
Nachi robot	<a href="https://youtu.be/2xoRl1Op5J0">https://youtu.be/2xoRl1Op5J0</a>
Scara robot, with pick and place	<a href="https://youtu.be/tHzgOLw00pk">https://youtu.be/tHzgOLw00pk</a>
AIV	<a href="https://youtu.be/cotE8bOWKKc">https://youtu.be/cotE8bOWKKc</a>
KUKA robot and rotary table	<a href="https://youtu.be/ZSeYazksYNE">https://youtu.be/ZSeYazksYNE</a>
System programming	<a href="https://youtu.be/dbj3Lk8qGMs">https://youtu.be/dbj3Lk8qGMs</a>

Programs were tested multiple times without any errors. There were no latencies on the SCARA and NACHI robots. However, the AIV is not as responsive, because it has to calculate a new trajectory each time a new position is sent.

When the KUKA robot and rotary table are programmed in VR it is not possible to execute the movement of both machines simultaneously. This is a software problem in Visual Components. Thus, when the program runs on the physical robots both machines move simultaneously.

#### V. CONCLUSION

Typically, manufacturing equipment and in particular industrial robots have their own controllers and vendor-specific programming languages. It is challenging and time-consuming to learn how to operate all the different controllers, and to create programs where all components collaborate. In this paper, we investigate the use of VR as a

method for programming and interacting with robots in a typical manufacturing setup. From our experience we believe that VR simplifies the programming process, as the same interface and programming approach can be used for all of the manufacturing equipment.

All models used in our setup were taken from the standard library in Visual Components, and with the built-in OPC UA connectivity the process of implementing VR programming, on both digital and physical robots, was simplified. It should be noted that our system was developed with Visual Components ver. 4.2 that includes the first version of interactive VR tools. In the future, we expect new VR functionality to be added and the programming of robots made even more simple. It should also be noted that using a wireless VR headset with built-in cameras makes our setup more flexible. The robots can be programmed from anywhere as long as the computer is connected to the network with the OPC UA server. Using VR for programming of robots increases safety since the operator can test and execute programs without being in the same room as the robots. We believe VR can be an efficient tool to avoid collisions and alert for dangerous situations in a manufacturing environment.

#### ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825196.

#### REFERENCES

- [1] Luis Pérez et al. "Industrial robot control and operator training using virtual reality interfaces". In: *Computers in Industry* 109 (2019), pp. 114–120. ISSN: 0166-3615. DOI: <https://doi.org/10.1016/j.compind.2019.05.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0166361518308546>.
- [2] Sankar Jayaram, Hugh I Connacher, and Kevin W Lyons. "Virtual assembly using virtual reality techniques". In: *Computer-Aided Design* 29.8 (1997). Virtual Reality, pp. 575–584. ISSN: 0010-4485. DOI: [https://doi.org/10.1016/S0010-4485\(96\)00094-2](https://doi.org/10.1016/S0010-4485(96)00094-2). URL: <http://www.sciencedirect.com/science/article/pii/S0010448596000942>.
- [3] Christoph Anthes et al. "State of the Art of Virtual Reality Technologies". In: Mar. 2016. DOI: 10.1109/AERO.2016.7500674.
- [4] Andrea [de Giorgio] et al. "Human-machine Collaboration in Virtual Reality for Adaptive Production Engineering". In: *Procedia Manufacturing* 11 (2017). 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy, pp. 1279–1287. ISSN: 2351-9789. DOI: <https://doi.org/10.1016/j.promfg.2017.07.255>. URL: <http://www.sciencedirect.com/science/article/pii/S2351978917304638>.
- [5] S.K. Ong et al. "Augmented reality-assisted robot programming system for industrial applications". In: *Robotics and Computer-Integrated Manufacturing* 61 (2020), p. 101820. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2019.101820>. URL: <http://www.sciencedirect.com/science/article/pii/S0736584519300250>.
- [6] Ali Ahmad Malik, Tariq Masood, and Arne Bilberg. "Virtual reality in manufacturing: Immersive and collaborative artificial-reality in design of human-robot workspace". In: *International Journal of Computer Integrated Manufacturing* (Nov. 2019). DOI: 10.1080/0951192X.2019.1690685.
- [7] Andrzej Burghardt et al. "Programming of Industrial Robots Using Virtual Reality and Digital Twins". In: *Applied Sciences* 10.2 (2020). ISSN: 2076-3417. DOI: 10.3390/app10020486. URL: <https://www.mdpi.com/2076-3417/10/2/486>.
- [8] H. Yap et al. "Virtual Reality Based Support System for Layout Planning and Programming of an Industrial Robotic Work Cell". In: *PLoS ONE* 9 (2014).
- [9] Tianyu Zhou, Qi Zhu, and Jing Du. "Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction". In: *Advanced Engineering Informatics* 46 (2020), p. 101170. ISSN: 1474-0346. DOI: <https://doi.org/10.1016/j.aei.2020.101170>. URL: <http://www.sciencedirect.com/science/article/pii/S1474034620301415>.
- [10] K. Zhou, Taigang Liu, and Lifeng Zhou. "Industry 4.0: Towards future industrial opportunities and challenges". In: *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. 2015, pp. 2147–2152.
- [11] D. Gorecky et al. "Human-machine-interaction in the industry 4.0 era". In: *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*. July 2014, pp. 289–294. DOI: 10.1109/INDIN.2014.6945523.
- [12] Raúl Crespo, René García, and Samuel Quiroz. "Virtual Reality Application for Simulation and Off-line Programming of the Mitsubishi Movemaster RV-M1 Robot Integrated with the Oculus Rift to Improve Students Training". In: *Procedia Computer Science* 75 (2015). 2015 International Conference Virtual and Augmented Reality in Education, pp. 107–112. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2015.12.226>. URL: <http://www.sciencedirect.com/science/article/pii/S187705091503687X>.
- [13] Q. Wang et al. "Virtual Reality Robot-Assisted Welding Based on Human Intention Recognition". In: *IEEE Transactions on Automation Science and Engineering* 17.2 (2020), pp. 799–808.
- [14] Vladimir Kuts et al. "Synchronizing physical factory and its digital twin through an IIoT middleware: a case study". In: *Proceedings of the Estonian Academy of Sciences* 68 (Oct. 2019), p. 364. DOI: 10.3176/proc.2019.4.03.
- [15] Christos Maragkos, George-Christopher Vosniakos, and Elias Matsas. "Virtual reality assisted robot programming for human collaboration". In: *Procedia Manufacturing* 38 (2019). 29th International Conference on Flexible Automation and Intelligent Manufacturing ( FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing, pp. 1697–1704. ISSN: 2351-9789. DOI: <https://doi.org/10.1016/j.promfg.2020.01.109>. URL: <http://www.sciencedirect.com/science/article/pii/S2351978920301104>.
- [16] ROS. URL: <https://www.ros.org/>. (accessed: 16.08.2020).
- [17] R. Codd-Downey et al. "From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation". In: *2014 IEEE International Conference on Information and Automation (ICIA)*. 2014, pp. 932–936.
- [18] Visual Components. *VISUAL COMPONENTS 4.2*. URL: <https://academy.visualcomponents.com/lessons/introduction-to-interactive-vr/>. (accessed: 10.04.2020).
- [19] Alex Owen-Hill. *Yes, Virtual Reality Now Works With Offline Programming*. URL: <https://robotdk.com/blog/virtual-reality-offline-programming/>. (accessed: 22.03.2020).
- [20] engineering.com. *White Paper: Now You Can Program Robots in Virtual Reality*. URL: <https://new.engineering.com/story/white-paper-now-you-can-program-robots-in-virtual-reality>. (accessed: 17.06.2020).
- [21] Radovan Holubek, Roman Ružarovský, and Daynier R. D. Sobrino. "An Innovative Approach of Industrial Robot Programming Using Virtual Reality for the Design of Production Systems Layout". In: *Advances in Manufacturing II*. Ed.

by Justyna Trojanowska et al. Cham: Springer International Publishing, 2019, pp. 223–235.

- [22] Ramzi Mehrez et al. “offline programming and development of an intelligent vision system with the KUKA robot”. In: Mar. 2019.
- [23] RoboDK. *Virtual Reality*. URL: <https://robodk.com/doc/en/Virtual-Reality.html#VRactions>. (accessed: 22.06.2020).
- [24] Pollák, Martin et al. “Application of industrial robot in 5-axis milling process”. In: *MATEC Web Conf.* 299 (2019), p. 04004. DOI: 10.1051/mateconf/201929904004. URL: <https://doi.org/10.1051/mateconf/201929904004>.
- [25] Beata Jakubiec. “APPLICATION OF SIMULATION MODELS FOR PROGRAMMING OF ROBOTS”. In: *SOCIETY. INTEGRATION. EDUCATION. Proceedings of the International Scientific Conference 5* (May 2018), p. 283. DOI: 10.17770/sie2018vol1.3214.
- [26] Visual Components. *VISUAL COMPONENTS 4.2*. URL: [https://www.visualcomponents.com/wp-content/uploads/2019/11/Experience\\_Guide\\_1.5.0\\_Update.pdf](https://www.visualcomponents.com/wp-content/uploads/2019/11/Experience_Guide_1.5.0_Update.pdf). (accessed: 19.06.2020).
- [27] Luis Pérez et al. “Digital Twin and Virtual Reality Based Methodology for Multi-Robot Manufacturing Cell Commissioning”. In: *Applied Sciences* 10 (May 2020), p. 3633. DOI: 10.3390/app10103633.
- [28] OPC Foundation. *Unified Architecture*. URL: <https://opcfoundation.org/about/opc-technologies/opc-ua/>. (accessed: 22.03.2020).
- [29] OPC Foundation. *OPC Unified Architecture Interoperability for Industrie 4.0 and the Internet of Things*. URL: <https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN.pdf>. (accessed: 22.03.2020).
- [30] *Cross-Platform OPC UA Functionality*. URL: <https://inductiveautomation.com/ignition/modules/ignition-opc-ua/>. (accessed: 13.08.2020).
- [31] Visual Components. *Connect a Remote OPC UA Server*. URL: <https://academy.visualcomponents.com/lessons/connect-a-remote-opc-ua-server/>. (accessed: 22.03.2020).
- [32] Visual Components. *VISUAL COMPONENTS EXPERIENCE*. URL: <https://www.visualcomponents.com/visual-components-experience/>. (accessed: 05.04.2020).
- [33] URL: <https://store.steampowered.com/steamvr/>. (accessed: 03.08.2020).
- [34] *Our first all-in-one gaming headset*. URL: [https://www.oculus.com/quest/?locale=en\\_US](https://www.oculus.com/quest/?locale=en_US). (accessed: 03.08.2020).
- [35] Visual Components. *VISUAL COMPONENTS 4.2*. URL: <https://www.visualcomponents.com/products/visual-components/>. (accessed: 10.04.2020).
- [36] H. Arnarson, B. Solvang, and B. Shu. “The application of open access middleware for cooperation among heterogeneous manufacturing systems”. In: *2020 3rd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS)*. 2020, pp. 1–6.
- [37] Unified Automation GmbH. *OPC UA Clients – Downloads*. URL: <https://www.unified-automation.com/downloads/opc-ua-clients.html>. (accessed: 30.03.2020).
- [38] Softing Industrial Automation GmbH. *DataFEED OPC UA Client*. URL: <https://apkcombo.com/datafeed-opc-ua-client/com.Softing.AndroidOPCUAClient/>. (accessed: 22.03.2020).