Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 100 (2021) 133–138

www.elsevier.com/locate/procedia

31st CIRP Design Conference 2021 (CIRP Design 2021)

# Virtual reality environment for industrial robot control and path design

Theodoros Togias, Christos Gkournelos, Panagiotis Angelakis, George Michalos, Sotiris Makris*

*Laboratory for Manufacturing Systems and Automation, Department of Mechanical Engineering and Aeronautics, University of Patras, Patras, 26504, Greece*

* Corresponding author. Tel.: +30-261-091-01600; fax: +30-261-099-7314. E-mail address: makris@lms.mech.upatras.gr

**Abstract**

The current market trends have focused on personalization, where products based on the same platform can have different variations so they can satisfy multiple market segments. Trying to keep up with these trends, industries have introduced hybrid Human Robot Collaborative stations that offer short throughput times, along with the flexibility to process different tasks. For this kind of flexibility to be achieved in production, a concept is examined for remotely reprograming industrial robots. As a result, a fully automated assembly line or a production station, could be repurposed to handle different products or processes, using the currently installed equipment with minor adjustments. The concept that is described in this paper presents a teleoperation – based method for process design and control of industrial robots utilizing Virtual Reality. The main aim is to reduce the time and effort required for repurposing the robot operation without the physical presence of a robot operator at the shop floor. A case study is presented to demonstrate the above described concept.



*Keywords:* Virtual Reality; Flexible manufacturing; Robot manipulation, Robotic cell

## 1. Introduction

As industry aims at manufacturing of custom products [1] and shifts from mass production towards mass customization, the need for flexible production stations has become apparent. Such production stations need to be able to handle different variations of a part. Fully automated robotic stations are capable of processing a variety of parts and perform multiple activities however hardware limitations (available tooling and peripheral equipment) and the complexity of reprogramming [2] limit the overall level of reconfigurability. For this reason, research has focused on production concepts that can increase flexibility. An example is the case of Human Robot Collaborative (HRC) stations where human operators work together with cobots to perform a variety of tasks that require human cognition. Robot to robot cooperation is a similar case where two or multiple robots can work together to perform a shared task. Furthermore, robotic cells with the appropriate equipment installed (e.g. tool changers, vision systems etc.) are able and perform different processes [3] and manipulate more product variants. The use of multiple robots with many tools seems to be the solution to flexibility, however this increases exponentially the complexity of programming such a robotic cell [4]. Another approach would be to enhance the currently deployed automated systems, with AI algorithms and neural network decision capabilities, so that the variations in the parts to be detected and the process plan to be automatically generated. However these systems are not fully error-prone [5], as changes in the environment and the input data can have a negative effect in the accuracy of the results from a Deep Neural Network. Furthermore, more trust needs to be developed for these systems in order for them to be able to act autonomously without human supervision. The benefit of reconfigurable robotic cell as well as AI enhanced robot cognition, is that both solutions can be implemented at an already existing setup with minor and low-cost intervention. On

2212-8271 © 2021 The Authors. Published by Elsevier Ltd.

10.1016/j.procir.2021.05.021

the other hand, for an HRC cell, an investment should be made for ensuring safety and the integration of collaborative robot.

Teleoperation of robotic resources is a common technique that can be applied in several cases such as performing operations and handling of parts in hazardous environments, (e.g. deep sea and space exploration missions), as well as applications in agriculture where a robot is controlled to perform a standardized task in a remote field [6][7][8][9]. A teleoperation system can be divided in three main components a master system, a slave system and the connection pipeline that allow these systems to exchange information [10]. Variations of this architecture could be found, with multiple master or slave components[11]. These variations can have benefits depending on the scenario and the hardware used at the teleoperation system implemented. The slave system contains the robotic resources while the master hosts the interface to control the robots. The effectiveness of a teleoperation system can be roughly defined based on three criteria: i) Stability ii) Transparency iii) Task performance [10]. Even though teleoperation systems can be effective in many cases, several challenges can be found in the implementation of such a system. From a technical perspective, network bandwidth and latency are main handicaps of teleoperation systems, as it can affect the stability and the transparency of the solution. Another important challenge of teleoperation is the user experience. As conventional interfaces displayed through a monitor are not sufficiently capable to provide to the user an accurate representation of the remote environment. This challenge extends also to the methods utilized by the user when trying to point out a position in the remote environment, reducing the task performance of the overall system.

Virtual Reality (VR) is a technology that has proven its feasibility in the manufacturing sector. With the use of VR engineers aim to immerse the user in a simulated environment, to have a close to real experience and better understanding of the situation presented [12]. The understanding of the environment where the robot operates is a common challenge in teleoperation systems [13][14]. Additionally, complex control interfaces incur an increased cognitive load from the user in order to perform a task [15][16]. Research has shown that utilization of immersive interfaces and multi-view interfaces for teleoperation can lead to better user performance [17][18][19]. This is a result of both better environment understanding and more natural interaction that VR offers.

In this paper the design of a teleoperation system is presented, that aims at increasing the flexibility of an existing workstation by allowing effortless, intuitive programming of the robotic resources through a VR interface. In section 2 the general approach of this work is presented, while in section 3 a more accurate description of the implementation is given. Finally, two industrial case studies are presented in section 4 as a proof of concept and evaluation of the proposed approach.

## 2. Approach

The design of a VR teleoperation system is presented, that focuses on providing the required tools for easy process design and control over the robots that exist in a production station. This framework allows multiple actors to be immersed in a production station and reconfigure the robotic production process, by defining new positions and actions. After the newly designed procedure is validated in simulated environment, it can be pushed to the robot controller for actual execution of the process. The realistic virtual environment can allow easy positioning and localization to a user that is immersed in it. The control interface is as simple as placing virtual objects in space, and therefore interaction with the robot can be less complex than the traditional teaching methods (e.g. axis jogging). Usually found interfaces in teleoperation replicate the robot Teach Pendants, by offering separate buttons that can control the different joints of the robot or control the tool position in separate cartesian axis [15]. Other interfaces can utilize a 3D mouse or IMU sensors attached to different variations of devices to capture the position and angles in space for the end effector [16]. Another technique commonly used for robot control is teaching by demonstration (TBD) [20], where the user interacts directly with the end effector of robot to teach a path to the robot for it to follow or replicate. The described technique has similarities with TBD, as the user manipulates a virtual end effector to showcase to the robot its final position. The difference between TBD is that in this case only the final position is set from the user. In the discussed approach, virtual reality technology and automated path planning of robotic manipulators are integrated in the simulation environment to enable indirect human robot control. The planner and the limits of the robot can be modified depending on the hardware and purpose this tool will be implemented for. In this way the user can spend less time to design a complex sequence of actions for the robot to follow, in comparison to teaching all the in between waypoints. The method presented could provide tools of assistance for the engineer or robot programmer as follows:

1. Continuous feedback from simulation is provided, for every different action planed for the robot in a safe and immersive way through the VR interface.
2. Automated collision avoidance calculation and automatic localization of real parts through vision system.
3. By enabling remote connection, this framework provides an environment for remote collaboration of multiple actors.

### 2.1. Intuitive Virtual Reality interface

The design of the VR interface is based on a Head Mounted Display hardware as visualization means, while the user can use remote controllers to interact with the VE. Teleoperation VR interfaces are usually found as two implementations: i) egocentric where the user perceives the world from the robot point of view, and ii) robocentric were the user moves freely in the Virtual Environment (VE) as an observer [21]. The VR interface is designed based on robocentric perspective. The VE provides a realistic view of the remote shop floor environment where the user can move freely around with the use of the joystick on the hand controllers of the HMD. Implementations like the one described minimize the requirements in terms of room clearance and setup time, making VR technology more serviceable and accessible to unfamiliar users. With the buttons of the HMD controllers actions can be selected from a

Graphical User Interface (GUI). After selecting an action, color coded arrows are spawned representing the position of the action in space and the user can manipulate these arrows with natural movements. The VR interface provides an intuitive method to program a complete sequence for the robot with the method of code blocks visualized as 3D objects in the virtual space. Every building block defines an action for the robot and contains preprogramed messages to the simulation environment. The user can set a chain of building blocks to define a complete procedure for the robot to execute. This concept requires all the digital I/O of the robotic systems to have the ability to be mapped into this building block architecture. Thus, all the robotic components existing in the workstation, are able to be reprogrammed through this interface. A VR simulation as an interface for robot control allows for close inspection and natural manipulation of virtual objects, without the risk of being close to a real potentially harmful environment. Through this immersive interface the effort and time required from an engineer to reprogram a robot could be minimized since a human could act with no fear of injury or damaging equipment.

*2.2. Simulation and robot control*

One of the aims of the proposed method is to serve as a useful tool that can easily be integrated with several industrial robots. For ensuring the compatibility across many robot brands, this approach includes libraries and frameworks for robot simulation, motion planning and controlling that are commonly used in industry and research. For providing a usable tool, this method focuses on ordinary activities of a robotic cell programmer. The dual arm robot that is used in this paper has two different arms and on each of them different tools are mounted at different orientations relatively to the robot base. Assembly parts on the other hand have also their own unique manner to be manipulated.

At first the engineer designs the robot process by setting positions in the cartesian space through VR interface. The process is transmitted to the robot simulation which checks the path for possible collisions and generates the robot trajectory and control schemes. The robot simulation is connected with the VR environment and thus, the user can have immediate feedback regarding the new robot process. Only after validation of the process by the engineer, the physical robot will proceed with execution. This form of near real-time control can overcome challenges of real time teleoperation which would require additional compensation, such as:

i. Jerkiness of the robot due not steady hands of the user or tracking losses that may occur.
ii. Requirement for a high-speed network.

As the speed of the transmitted packages from the user to the physical robot is not something that has to be considered. An implementation with a TCP connection would be beneficial in this case of near real-time communication of the robot with the user. Even though TCP connection may offer bigger delay in comparison with a UDP connection, it ensures delivery of an extensive robot process without package loss.

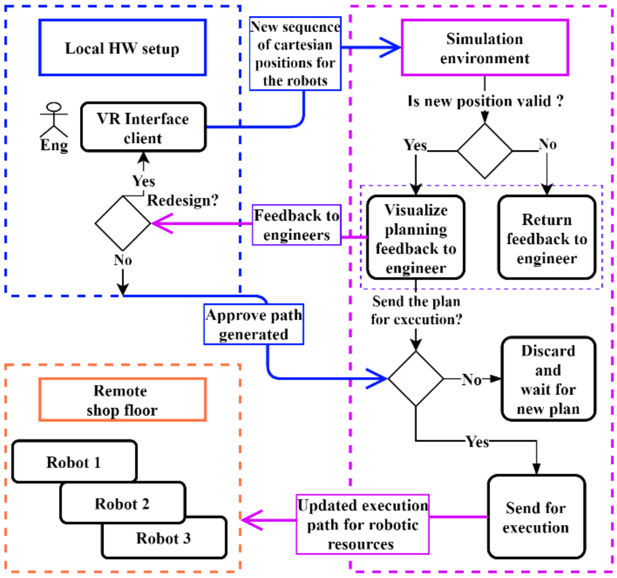


Fig. 1 Robot programming information flow

As mentioned, in this proposed method the engineer does not have real time control of the physical robot. Instead, this closed loop of process design, execution in simulated environment and immersive visualization of the robot motion can provide a real like experience to the engineer. The trajectories that the simulation generated, are calculated from a motion planner integrated to the simulation. The simulation planner is build with the use of a motion planning framework that provides tools for environment specification, collision detection of the robot with the defined environment, visualization as will as integrated motion planning algorithms for the calculation of the robot kinematics. The motion planning algorithm selected in the implemented work is the Open Motion Planning Library (OMPL).

The advantage of this approach regarding robot programming, is the reduced time required to program a complete robot procedure. Even though in some cases many iterations of replanning and simulation execution could occur, these interactions can happen rapidly. The action flow of the robot programming is presented in Fig. 1. The robot simulation and control module can also host supplementary sensory equipment that will further assist the engineer during the design of a process. A vision system will perform automatic localization of the parts that are to be manipulated and thus, precise positioning of the robot by the engineer is not required. For example, for a pick and place process, the engineer only has to specify the part that the robot has to grasp. The localization and grasping of the part will be performed automatically by the robot. The vision system will also be utilized for the accurate representation of the environment and the different components in the simulation environment.

## 3. Implementation

*3.1. System Configuration*

For the software architecture implementation, Unity was used for the e visualization of the simulation environment and the VR interface. In order to be compatible with a big range of robots, the robot connection and control functionality of this system is based on ROS [23]. ROS acts as a middleware to establish connection with the hardware devices. Moreover, the robot control functionalities are also developed in ROS. The integration of the VR interface with the robot and vision system modules are achieved through ROS bridge and this section focuses on the integration of motion control from the simulation to the VE (Fig. 2). For Unity to connect to the ROS bridge server, ROS Sharp plugin was used since it enables the modeling of all messages implemented in ROS. Through ROS bridge pipeline information retrieved from production shop floor can be acquired by engineers remotely, as all the machines data are presented to the VR interface.

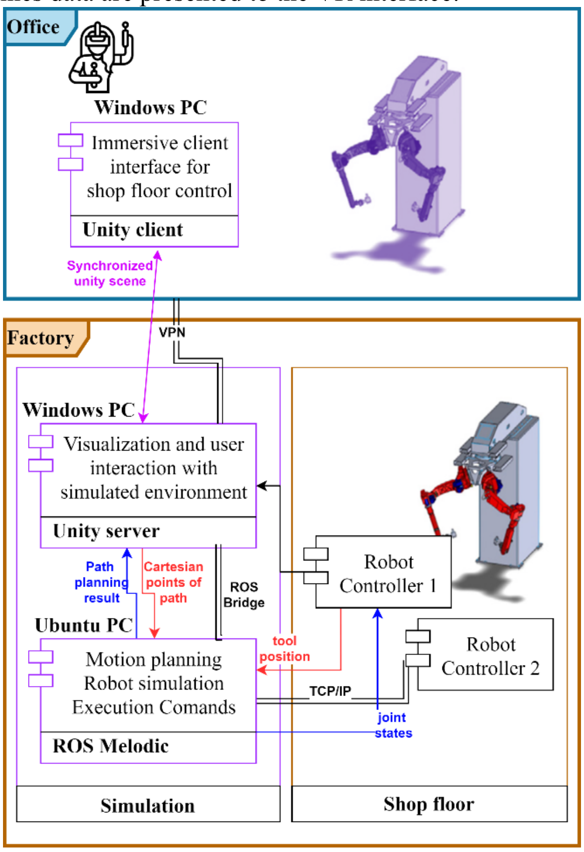


Fig. 2. Overall architecture, communication pipelines and message exchange between sub modules

The selected hardware architecture makes use of Oculus Rift which is a commercial VR HMD device with Touch controllers that can enable intuitive interactions inside the VE. A medium spec laptop running Windows is handling the VR interface. The user can interact with the VR interface through the HMD device, by utilizing a VPN connection. Through this secure pipeline the VR interface can be connected to a windows PC that exists in the remote location and is purposed to act as a server for different VR interface clients. The VE where the users are immersed is synced through the server PC, allowing for synced update of the environment between users. Also, the latency of the individual clients will not affect the communication with the simulated robot in ROS. The server PC acts as middleware to control the federation of data to the simulated robot, so that smooth transactions of information to and from the simulation can be achieved. The simulation PC is running the Ubuntu 18.04 OS, so that it can host complex environments and calculate kinematic motions for multiple robots. It able to transmit/receive information to/from the robot controller through a TCP/IP interface. The manipulator that was selected for this implementation is the Comau dual arm manipulator. The main reason that a dual arm robot was selected is that a dual arm robot can perform a wide range of processes, regarding handling and assemble. Also, the control of a dual arm robot can be of high complexity as it is similar with control of two individual manipulators.

### *3.2. VR Interface & Simulation Environment Implementation*

As described at section 3.1, the simulation environment instance is executed at a PC located in the shop floor. The simulation environment is directly connected and exchange continuously information with the robot simulation and control module developed in ROS. The above-described instance of the simulation is also responsible for the communication with the VR interface of the engineer by providing server capabilities. The VR interface instance is the client to the simulation environment. It acts as a window for the user to the simulation environment and the actual situation at the shop floor. For the development of this communication channel the Mirror Networking [24] an open-source API for Unity was used. This networking APIs is developed to support Massively Multiplayer Online (MMO) games where synchronization and low latency between large numbers of players are important. This kind of network capabilities are also required for a teleoperation system that aim to support multiple user instances. Regarding the design of the simulation environment, initially an empty warehouse was designed as a base environment that can host various workstations. With the utilization of functions provided from ROS Sharp[25] the robot URDF and .stl models were imported from ROS to Unity. The shop floor environment around the robot was designed in CAD software to match the dimensions of the physical environment. After finalizing the design of the cell the 3D models were imported in Unity (Fig. 3).

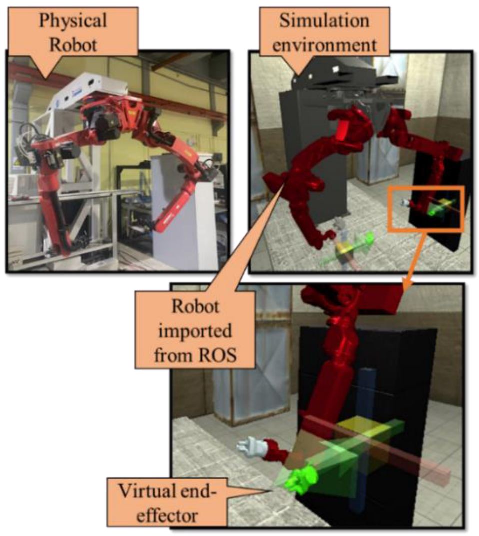


Fig. 3. Real setup replicated in simulation environment and virtual end effector

To enable support for the HMD, the Oculus libraries for Unity were used. The implemented interface makes use of the touch controllers to manipulate two virtual end effectors (one for each arm of the robot) and set the new desired position. After setting a new position, the user with the press of the buttons at the top of the controllers can edit a command at the position the virtual end effectors are located. These commands can be simple waypoints that the robot will follow in the new program, or actions that make use of the equipment installed at the robot flanges, for example actions could trigger the robot grippers to open or close.
All the commands that the user can set correspond to a ROS message that will be send to the simulation through ROS actions. Through the implemented ROS drivers, the simulation can control the physical robot position along with the different I/O that exist at the robot controller, to enable functionalities such as pick and place. After setting a series of commands, the user can select "plan" of the new robot sequence, in order to have a validation from the simulation and a preview. User interface panels located at the wrist of the user, display information regarding the status of the plan, status of execution, etc. (Fig. 4). At the end of the process design, after the sequence has been validated from the simulation, the user can send the new program for execution from the actual robot through the use of the ROS Drivers.

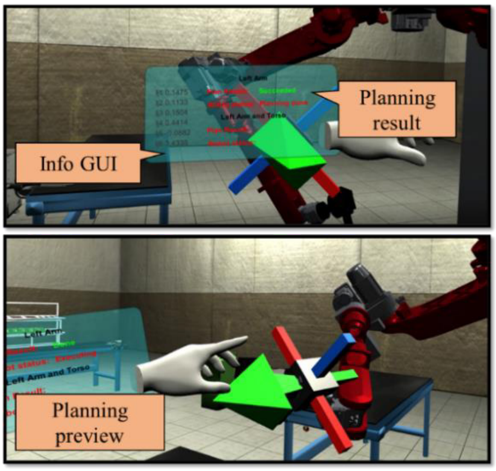


Fig. 4. Wrist user interface during planning of a new action

### 3.3. Robot Control

Robot programming and control will be served through the ROS interfaces and libraries. MoveIt [26] motion planning framework is utilized, in order to have an accurate simulation of the robot motions that matches with the execution on the physical setup. MoveIt provides a collision free robot trajectory plan, following a configuration procedure that includes robot's description and environment planning scene. Planning scene is configured in the same manner with the VR environment and represents the physical setup of the robotic cell. The use of Transform frames (tf) library enables the easy and quick generation of cartesian trajectories for different assembly operations. Tf efficiently keeps track of the relationship of multiple coordinate frames which in this work are the assembly objects, tools and the two robot arms. The actual control of the dual arm robot is handled from a custom driver, developed based on the proposed interface guidelines from ROS community. This way the presented implementation can accommodate a wide range of robots, that are working in this manner. The robot driver is running on a Linux machine and is deployed as a ROS node which is able to interact easily with the motion planning node and the tf tree. The communication with the robot is established over a dedicated ethernet network through TCP connection.
The accuracy of the control of the robot is achieved through a URDF which has been calibrated specifically for every different robot used. Through this way unique deviations of each robot can be calculated from the calibration process that the manufacturer implements on each robot. After configuring the URDF to match the physical position, it can be trusted for planning accurate cartesian goals with the ROS driver. Detailed description of the robot control driver will be presented in future work as it is not the focus of this paper.

## 4. Case study

The software needs to prove its feasibility regarding the flexibility it provides and its performance in comparation with conventional methods of robot reprogramming. A number of assembly processes were selected driven from the automotive industry so that the developed methodology of robot path design to be applied and tested. The first case is focusing on a car's differential assembly. This case requires mainly pick and place actions, and a screwing process where the robot grabs an electric screwdriver to fasten a set of bolds at the differential casing.

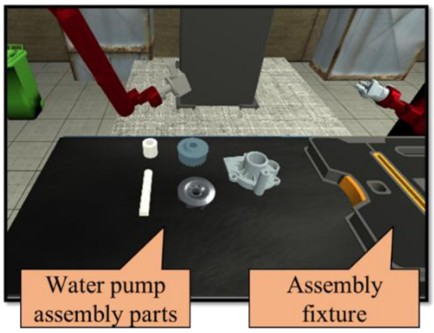


Fig. 5. Water pump assembly setup

The second case study refers to an assembly of different products. Two variations of water pumps were modeled as Fig. 5 shows. This assembly process includes pick and place actions from the robot. Initially the robot grasps the water pump fan and shaft and place them at fixtures. After that it assembles the rest of the components. The differences between the pump models are the shape of the outer casing as well as an extra bearing that is included in one of the variants. The realistic representation of the VE assist the user on pointing out with precision the start and end points of the desired trajectories, and in conjunction with the automated path planning provided from the simulation, the effort and time required for the design of the robot path could be reduced. In continuation of this work, these two case studies will be used to measure the user experience and the other metrics such as design time in comparison to other methods.

## 5. Conclusions & Future work

This paper describes the design and implementation of a teleoperation based system, that utilizes VR to provide the user with good understanding of the shop floor environment as well as intuitive interfaces for reprogramming of the robotic resources. The benefits that this system aims, is at enhancing the flexibility of a currently installed robotic cell in a cost-effective way and with minimum hardware alternations. Through this proposed setup a typical robotic cell that performs assembly operations could rapidly be reprogramed to handle different assembly variations. A VR interface has been implemented as an offline simulation of the remote process, providing near real time control of the robotic resources to the user. Through this simulated environment users can have a clear image of the situation in the remote workplace, set new positions in a natural intuitive way and also collaborate online in defining and detailing the same task.

As next steps, an evaluation of the developed system from engineers and robot operators will take place. Through the evaluation the user experience will be measured based on the feedback from the participants. Furthermore, the performance of reprogramming the robotic cell will be measured and will be compared with conventional techniques. Future developments will mainly involve the finalization of the network capabilities and the implementation of the AI enhanced vision system for the automatic grasp of various parts. Finally, a market and cost analysis will be performed, comparing the proposed setup with other alternatives that provide this kind of flexibility in an assembly station.

## Acknowledgements

This work has been partially funded by the European Institute of Innovation and technology (EIT) research project MirrorLabs and the European project "SHERLOCK Seamless and safe human-centred robotic applications for novel collaborative workshops" (Grand Agreement: 820689) (www.sherlock-project.eu).

## References

[1] Chryssolouris G. Manufacturing Systems: Theory and Practice. Springer Science & Business Media; 2006.

[2] Michalos G, Kaltsoukalas K, Aivaliotis P, Sipsas P, Sardelis A, Chryssolouris G. Design and simulation of assembly systems with mobile robots. CIRP Annals. 2014 Jan 1;63(1):181-4

[3] Aivaliotis P, Michalos G, Makris S. Cooperating robots for fixtureless assembly: modelling and simulation of tool exchange process. International Journal of Computer Integrated Manufacturing. 2018 Dec 2;31(12):1235-46.

[4] Papakostas N, Alexopoulos K, Kopanakis A. Integrating digital manufacturing and simulation tools in the assembly design process: A cooperating robots cell case. CIRP journal of manufacturing science and technology. 2011 Jan 1;4(1):96-100.

[5] Marino DL, Grandio J, Wickramasinghe CS, Schroeder K, Bourne K, Filippas AV, Manic M. Ai augmentation for trustworthy ai: augmented robot teleoperation. In2020 13th International Conference on Human System Interaction (HSI) 2020 Jun 6 (pp. 155-161). IEEE

[6] Lunghi G, Prades RM, Di Castro M. An Advanced, Adaptive and Multimodal Graphical User Interface for Human-robot Teleoperation in Radioactive Scenarios. InICINCO (2) 2016 Jul 29 (pp. 224-231).

[7] Feng G, Li W, Zhang H. Space Robot Teleoperation Experiment and System Evaluation Method. In2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) 2018 May 25 (pp. 346-351). IEEE.

[8] Adamides G, Katsanos C, Parmet Y, Christou G, Xenos M, Hadzilacos T, Edan Y. HRI usability evaluation of interaction modes for a teleoperated agricultural robotic sprayer. Applied ergonomics. 2017 Jul 1;62:237-46

[9] Berthet-Rayne P, Leibrandt K, Gras G, Fraisse P, Crosnier A, Yang GZ. Inverse kinematics control methods for redundant snakelike robot teleoperation during minimally invasive surgery. IEEE Robotics and Automation Letters. 2018 Mar 7;3(3):2501-8.

[10] Kebria PM, Abdi H, Dalvand MM, Khosravi A, Nahavandi S. Control methods for internet-based teleoperation systems: A review. IEEE Transactions on Human-Machine Systems. 2018 Nov 15;49(1):32-46.

[11] Shahbazi M, Atashzar SF, Patel RV. A systematic review of multilateral teleoperation systems. IEEE transactions on haptics. 2018 Mar 22;11(3):338-56.

[12] Mourtzis D. Simulation in the design and operation of manufacturing systems: state of the art and new trends. International Journal of Production Research. 2020 Apr 2;58(7):1927-49.

[13] Rentzos L, Vourtsis C, Mavrikios D, Chryssolouris G. Using VR for complex product design. InInternational conference on virtual, augmented and mixed reality 2014 Jun 22 (pp. 455-464).

[14] Almeida L, Menezes P, Dias J. Improving robot teleoperation experience via immersive interfaces. In2017 4th Experiment@ International Conference (exp. at'17) 2017 Jun 6 (pp. 87-92). IEEE

[15] Kent D, Saldanha C, Chernova S. A comparison of remote robot teleoperation interfaces for general object manipulation. InProceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction 2017 Mar 6 (pp. 371-379).

[16] Yang C, Luo J, Pan Y, Liu Z, Su CY. Personalized variable gain control with tremor attenuation for robot teleoperation. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2017 May 8;48(10):1759-70.

[17] Hedayati H, Walker M, Szafir D. Improving collocated robot teleoperation with augmented reality. InProceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction 2018 Feb 26 (pp. 78-86).

[18] Oh Y, Parasuraman R, McGraw T, Min BC. 360 vr based robot teleoperation interface for virtual tour. InProceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI) 2018.

[19] Chen J, Glover M, Yang C, Li C, Li Z, Cangelosi A. Development of an immersive interface for robot teleoperation. InAnnual Conference Towards Autonomous Robotic Systems 2017 Jul 19.

[20] Solanes, J.E., Muñoz, A., Gracia, L., Martí, A., Girbés-Juan, V. and Tornero, J., 2020. Teleoperation of industrial robot manipulators based on augmented reality. The International Journal of Advanced Manufacturing Technology, 111(3), pp.1077-1097.

[21] Rosen E, Whitney D, Phillips E, Ullman D, Tellex S. Testing robot teleoperation using a virtual reality interface with ROS reality. InProceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI) 2018 Mar (pp. 1-4).

[22] Aschenbrenner D, Fritscher M, Sittner F, Krauß M, Schilling K. Teleoperation of an industrial robot in an active production line. IFAC-PapersOnLine. 2015 Jan 1;48(10):159-64.

[23] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY. ROS: an open-source Robot Operating System. InICRA workshop on open source software 2009 May 12 (Vol. 3, No. 3.2)

[24] Mirror Networking – Open source networking for Unity (https://mirror-networking.com/)

[25] ROS-Sharp (https://github.com/siemens/ros-sharp)

[26] MoveIt Motion Planning Framework (https://moveit.ros.org/