A New Approach to Gesture Based Real-Time Robot Programming Using Mixed Reality

Luisa Hornung, Simon Lawo, and Christian Wurll

Karlsruhe University of Applied Sciences
luisa.hornung@hs-karlsruhe.de
christian.wurll@hs-karlsruhe.de

Abstract. While being increasingly used in larger industry companies, industrial robots have not yet prevailed in smaller enterprises. Not least, this is due to the time-consuming programming and the requirement for robotics experts. An intuitive control and programming concept can decisively reduce the need for expert knowledge. Using modern rendering software and innovative visualization frameworks, a gesture-based programming approach was developed at Karlsruhe University of Applied Sciences. Here, the user creates the robot program by virtually executing and chaining robot poses and gripper instructions. An evaluation shows the advantages of the developed concept and a comparison with competing methods.

Keywords: Robot Programming; Virtual Robot; Mixed Reality; 3D-Engine; Small and Medium-sized Enterprises

1 Introduction

Industrial robots are mainly deployed in large-scale production, especially in the automotive industry. Today, there are already 26.1 industrial robots deployed per 1,000 employees on average in these industry branches. In contrast, Small and Medium-sized Enterprises (SMEs) only use 0.6 robots per 1,000 employees [1]. Reasons for this low usage of industrial robots in SMEs include the lack of flexibility with great variance of products and the high investment expenses due to additional peripherals required, such as gripping or sensor technology. The robot as an incomplete machine accounts for a fourth of the total investment costs [2]. Due to the constantly growing demand of individualized products, robot systems have to be adapted to new production processes and flows [3]. This development requires the flexibilization of robot systems and the associated frequent programming of new processes and applications as well as the adaption of existing ones. Robot programming usually requires specialists who can adapt flexibly to different types of programming for the most diverse robots and can follow the latest innovations. In contrast to many large companies, SMEs often have no in-house expertise and a lack of prior knowledge with regard to robotics. This often has to be obtained externally via system integrators, which, due to high costs, is one of the reasons for the inhibited use of robot systems. During the initial generation or extensive adaption of process flows with industrial robots, there is a constant risk of injuring persons and damaging the expensive hardware components. Therefore, the programs have to be tested under strict safety precautions and usually in a very slow test mode. This makes the programming of new processes very complex and therefore time- and cost-intensive.

The concept presented in this paper combines intuitive, gesture-based programming with simulation of robot movements. Using a mixed reality solution, it is possible to create

a simulation-based visualization of the robot and project, to program and to test it in the working environment without disturbing the workflow. A virtual control panel enables the user to adjust, save and generate a sequence of specific robot poses and gripper actions and to simulate the developed program. An interface to transfer the developed program to the robot controller and execute it by the real robot is provided.
The paper is structured as follows. First, a research on related work is conducted in Section 2, followed by a description of the system of the gesture-based control concept in Section 3. The function of robot positioning and program creation is described in Section 4. Last follow the evaluation in Section 5 and conclusion in Section 6.

2 Related Work

Various interfaces exist to program robots, such as Lead-Trough, Offline or Walk-Trough programming, Programming by demonstration, vision based programming or vocal commanding. In the survey of Villani et al. [4] a clear overview on existing interfaces for robot programming and current research is provided. Besides the named interfaces, the programming of robots using a virtual or mixed reality solution aims to provide intuitiveness, simplicity and accessibility of robot programming for non-experts. Designed for this purpose, Guhl et al. [5] developed a generic architecture for human-robot interaction based on virtual and mixed reality. In the marker tracking based approach presented by [6] and [7], the user defines a collision-free-volume and generates and selects control points while the system creates and visualizes a path through the defined points. Others [8], [9], [10] and [11] use handheld devices in combination with gesture control and motion tracking. Herein, the robot can be controlled through gestures, pointing or via the device, while the path, workpieces or the robot itself are visualized on several displays. Other gesture and virtual or mixed reality based concepts are developed by Cousins et al. [12] or Tran et al. [13]. Here, the robots perspective or the robot in the working environment is presented to the user on a display (head-mounted or stationary) and the user controls the robot via gestures. Further concepts using a mixed reality method enable an image of the workpiece to be imported into CAD and the system automatically generates a path for robot movements [14] or visualizing the intended motion of the robot on the Microsoft HoloLens, that the user knows where the robot will move to next [15]. Other methods combine pointing at objects on an screen with speech instructions to control the robot [16]. Sha et al. [17] also use a virtual control panel in their programming method, but for adjusting parameters and not for controlling robots. Another approach pursues programming based on cognition, spatial augmented reality and multimodal input and output [18], where the user interacts with a touchable table.
Krupke et al. [19] developed a concept in which humans can control the robot by head orientation or by pointing, both combined with speech. The user is equipped with a head-mounted display presenting a virtual robot superimposed over the real robot. The user can determine pick and place position by specifying objects to be picked by head orientation or by pointing. The virtual robot then executes the potential pick movement and after the user confirms by voice command, the real robot performs the same movement. A similar concept based on gesture and speech is persued by Quintero et al. [20], whose method offers two different types of programming. On the one hand, the user can determine a pick and place position by head orientation and speech commands. The system automatically generates a path which is displayed to the user, can be manipulated by the user and is simulated by a virtual robot. On the other hand, it is possible to create a path on a surface by the user generating waypoints. Ostanin and Klimchik [21]

introduced a concept to generate collision-free paths. The user is provided with virtual goal points that can be placed in the mixed reality environment and between which a path is automatically generated. By means of a virtual menu, the user can set process parameters such as speed, velocity etc.. Additionally, it is possible to draw paths with a virtual device and the movement along the path is simulated by a virtual robot.
Differently to the concept described in this paper, only a pick and place task can be realized with the concepts of [19] and [20]. A differentiation between movements to positions and gripper commands as well as the movement to several positions in succession and the generation of a program structure are not supported by these concepts. Another distinction is that the user only has the possibility to show certain objects to the robot, but not to move the robot to specific positions. In [19] a preview of the movement to be executed is provided, but the entire program (pick and place movements) is not simulated. In contrast to [21], with the concept presented in this paper it is possible to integrate certain gripper commands into the program. With [21] programming method, the user can determine positions but exact axis angles or robot poses cannot be set.
Overall, the approach presented in this paper offers an intuitive, virtual user interface without the use of handheld devices (cf. [6], [7], [8], [9], [10] and [11]) which allows the exact positions of the robot to be specified. Compared to other methods, such as [12], [13], [14], [15] or [16], it is possible to create more complex program structures, which include the specification of robot poses and gripper positions, and to simulate the program in a mixed reality environment with a virtual robot.

3 Mixed Reality Robot Programming System

In this section the components of the Mixed Reality Robot Programming System are introduced and described. The system consists of multiple real and virtual interactive elements, whereby the virtual components are projected directly into the field of view using a Mixed Reality (MR) approach. Compared to the real environment, which consists entirely of real objects and virtual reality (VR), which consists entirely of virtual objects and which overlays the real reality, in MR the real scene here is preserved and only supplemented by the virtual representations [22]. In order to interact in the different realities, head-mounted devices similar to glasses, screens or mobile devices are often used. Figure 1 provides an overview of the systems components and their interaction.
The system presented in this paper includes KUKAs collaborative, lightweight robot LBR iiwa 14 R820 combined with an equally collaborative gripper from Zimmer as real components and a virtual robot model and a user interface as virtual components. The virtual components are presented on the Microsoft HoloLens. For calculation and rendering the robot model and visualization of the user interface, the 3D- and physics-engine of the Unity3D development framework is used. Furthermore, for supplementary functions, components and for building additional MR interactable elements, the Microsoft Mixed Reality Toolkit (MRTK) is utilized.
For spatial positioning of the virtual robot, marker tracking is used, a technique supported by the Vuforia framework. In this use case, the image target is attached to the real robot's base, such that in MR the virtual robot superimposes the real robot. The program code is written in C♯.
The robot is controlled and programmed via an intuitive and virtual user interface that can be manipulated using the so-called Airtap gesture, a gesture provided by Microsoft HoloLens.

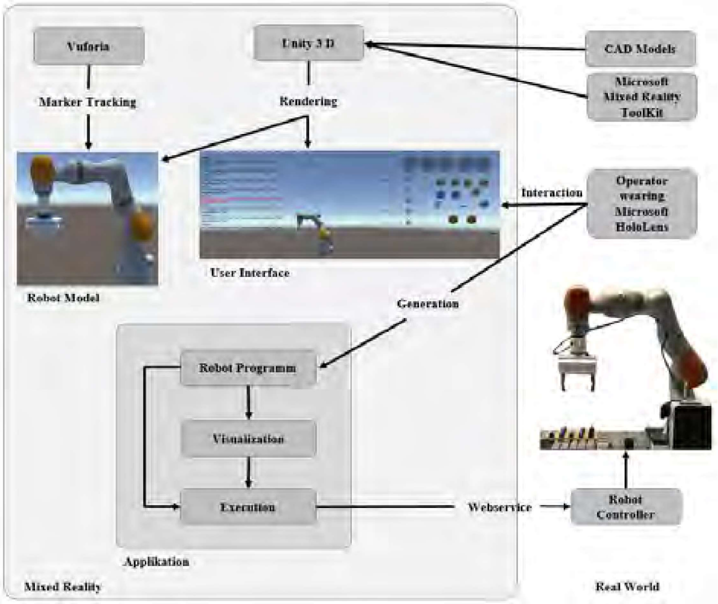

Fig. 1. Diagram of the systems components and their interaction.

3.1 Virtual Robot Model

To ensure that the virtual robot mirrors the motion sequences and poses of the real robot, the most exact representation of the real robot is employed. The virtual robot consists of a total of eight links, matching the base and the seven joints of iiwa 14 R820:

– the base frame,
– five joint modules,
– the central hand and
– the media flange.

The eight links are connected together as a kinematic chain. The model is provided as open source files from [23] and [24] and is integrated into the Unity3D project.
The individual links are created as GameObjects in a hierarchy, with the base frame defining the top level and are limited similar to those of the real robot. The CAD data of the deployed gripping system is also imported into Unity3D and linked to the robot model.

3.2 User Interface

The canvas of the head-up displayer of the Microsoft HoloLens is divided into two parts and rendered at a fixed distance in front of the user and on top of the scene. At the top left side of the screen the current joint angles (A1 to A7) are displayed and on the left side the current program is shown. This setting simplifies the interaction with the robot as the informations do not behave like other objects in the MR scene, but are attached to the Head Up Display (HUD) and move with the user's field of view. The user interface, which consists of multiple interactable components, is placed into the scene and is shown at the right side of the head-up display.
At the beginning of the application the user interface is in "Clear Screen" mode, i.e. only the buttons "Drag", "Cartesian", "Joints", "Play" and "Clear Screen" and the joint angles at the top left of the screen are visible. For interaction with the robot, the user

has to switch into a particular control mode by tapping the corresponding button. The user interface provides three different control modes for positioning the virtual robot:

- Drag Mode, for rough positioning,
- Cartesian Mode, for Cartesian positioning and
- Joint Mode, for the exact adjustment of each joint angle.

Figure 2 shows the interactable components that are visible and therefore controllable in the respective control modes.

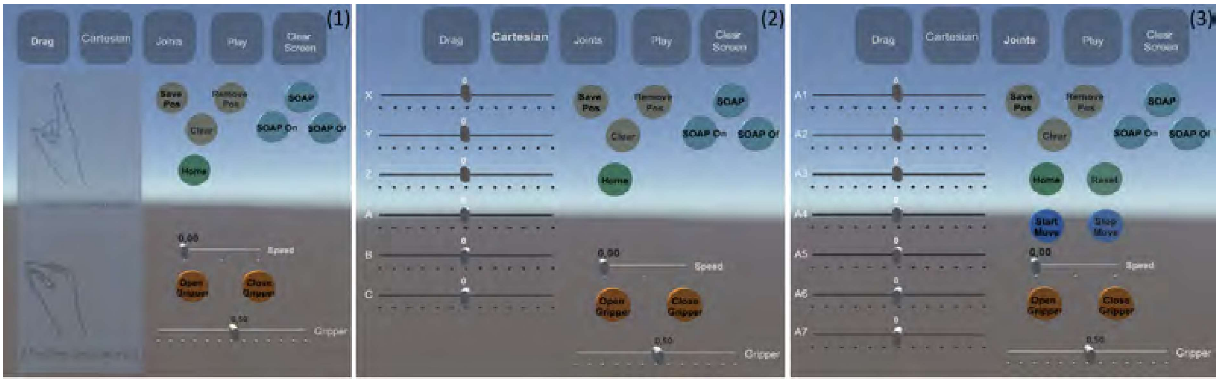

Fig. 2. User interface control modes and visible interactable elements: (1): Drag Mode; (2): Cartesian Mode; (3): Joints Mode.

Depending on the selected mode, different interactable components become visible in the user interface, with whom the virtual robot can be controlled. In addition to the control modes, the user interface offers further groups of interactable elements:

- Motion Buttons, with which e.g. the speed of the robot movement can be adjusted or the robot movement can be started or stopped,
- Application Buttons, to save or delete specific robot poses, for example,
- Gripper Buttons, to adjust the gripper and
- Interface Buttons, that enable communication with the real robot.

4 Usage

This section focuses on the description of the usage of the presented approach. In addition to the description of the individual control modes, the procedure for creating a program is also described. As outlined in Section 3.2, the user interface consists of three different control modes and four groups of further interactable components. Through this concept, the virtual robot can be moved efficiently to certain positions with different movement modes, the gripper can be adjusted, the motion can be controlled and a sequence of positions can be chained.

4.1 Control Modes

Drag By gripping the tool of the virtual robot with the Airtap gesture, the user can "drag" the robot to the required position. Additionally, it is possible to rotate the position of the robot using both hands. This mode is particularly suitable for moving the robot very quickly to a certain position.

Cartesian This mode is used for the subsequent positioning of the robot tool with millimeter precision. The tool can be translated to the required positions using the Cartesian coordinates X, Y, Z and the Euler angles A, B, C. The user interface provides a separate slider for each of the six translation options.The tool of the robot moves analogously to the respective slider button, which the user can set to the required value.

Joints This mode is an alternative to the Cartesian method for exact positioning. The joints of the virtual robot can be adjusted precisely to the required angle, which is particularly suitable for e.g. bypassing an obstacle. There is a separate slider for each joint of the virtual robot. In order to set the individual joint angles, the respective slider button is dragged to the required value, which is also displayed above the slider button for better orientation.

4.2 Programming the Robot

To program the robot, the user interface provides various application buttons, such as saving and removing robot poses from the chain and a display of the poses in the chain. The user directs the virtual robot to the desired position and confirms using the corresponding button. The pose of the robot is then saved as joint angles from A1 to A7 and one gripper position in a list and is displayed on the left side of the screen. When running the programmed application, the robot moves to the saved robot poses and gripper positions according to the defined sequence. For a better orientation, the robots current target position changes its color from white to red. After testing the application, the list of robot poses can be sent to the controller of the real robot via a webservice. The real robot then moves analogously to the virtual robot to the corresponding robot poses and gripper positions.

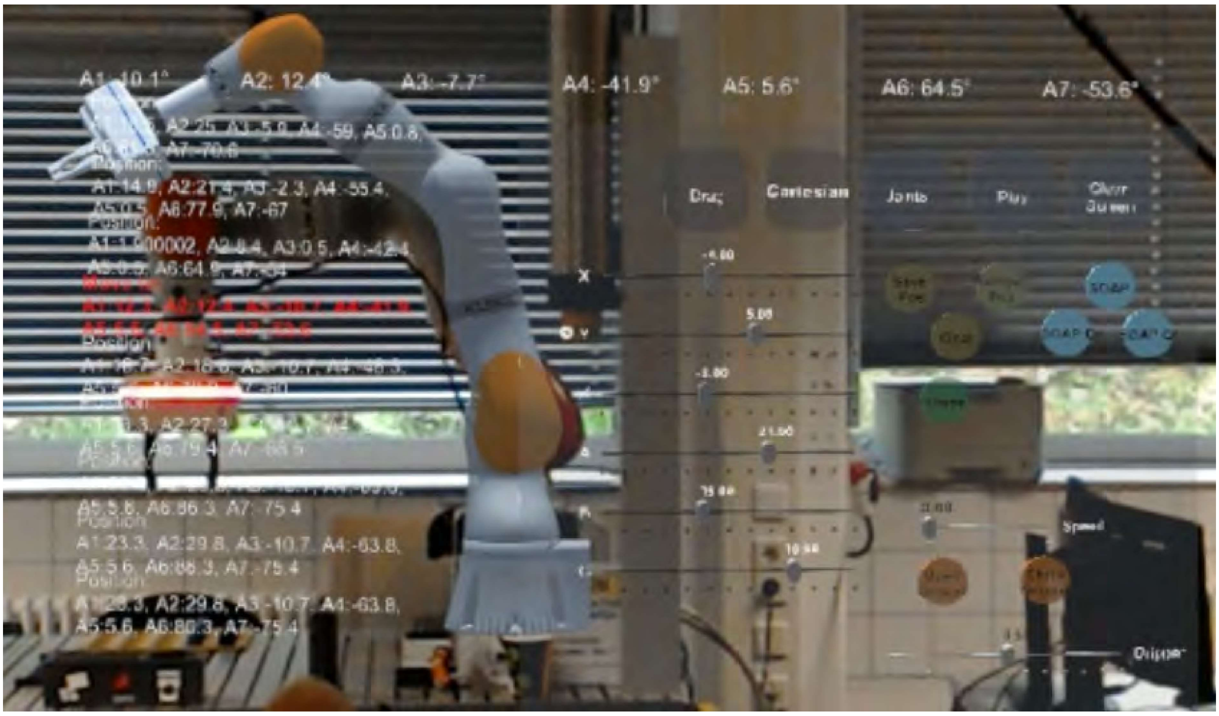

Fig. 3. The user's view in the mixed reality environment.

5 Evaluation of the Programming Concept

The purpose of the evaluation is how the gesture-based control concept compares to other concepts regarding intuitiveness, comfort and complexity. For the evaluation, a study was conducted with seven test persons, who had to solve a pick and place task with five different operating concepts and subsequently evaluate them. The developed concept based on gestures and MR was evaluated against a lead through procedure, programming with Java, programming with a simplified programming concept and approaching and saving points with KUKA SmartPad. The test persons had no experience with Microsoft HoloLens and MR, no to moderate experience with robots and no to moderate programming skills. The Questionnaire for the Evaluation of Physical Assistive Devices (QUEAD) developed by Schmidtler et al [25] was used to evaluate and compare the five control concepts. The questionnaire is classified into five categories (perceived usefulness, perceived ease of use, emotions, attitude and comfort) and contains a total of 26 questions, rated on an ordinal scale from 1 (entirely disagree) to 7 (entirely agree).
Firstly, each test person received a short introduction to the respective control concept, conducted the pick and place task and immediately afterwards evaluated the respective control concept using QUEAD.

5.1 Results of the QUEAD

Figure 4 provides an extract of the results of the study.

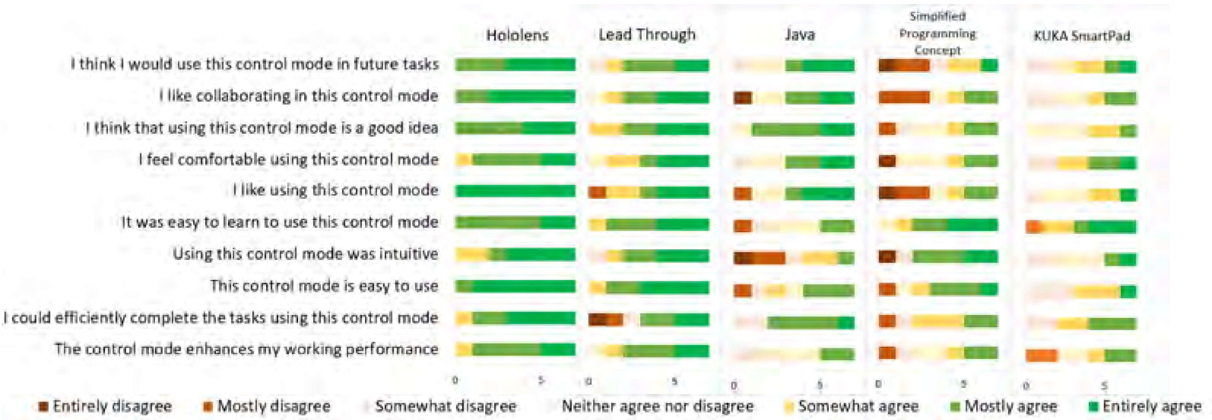

Fig. 4. Extract from the results of the study to compare the control concepts (from left to right): gesture-based control concept, Lead Through, Java, Simplified Programming Concept, KUKA SmartPad.

All test persons agreed that they would reuse the concept in future tasks (3 mostly agree, 4 entirely agree). In addition, the test persons considered the gesture-based concept to be intuitive (1 mostly agree, 4 entirely agree), easy to use (5 mostly agree, 2 entirely agree) and easy to learn (1 mostly agree, 6 entirely agree). Two test persons mostly agree and four entirely agree that the gesture-based concept enabled them to solve the task efficiently and four test persons mostly agree and two entirely agree that the concept enhances their work performance. All seven subjects were comfortable using the gesture-based concept (4 mostly agree, 2 entirely agree).
Overall, the concept presented in this paper was evaluated as more comfortable, more

intuitive and easier to learn than the other control concepts evaluated. In comparison to them, the new operating concept was perceived as the most useful and easiest to use. The test persons felt physically and psychologically most comfortable when using the concept and were most positive in total.

6 Conclusion

In this paper, a new concept for programming robots based on gestures and MR and for simulating the created applications was presented. This concept forms the basis for a new, gesture-based programming method, with which it is possible to project a virtual robot model of the real robot into the real working environment by means of a MR solution, to program it and to simulate the workflow. Using an intuitive virtual user interface, the robot can be controlled by three control modes and further groups of interactable elements and via certain functions, several robot positions can be chained as a program. By using this concept, test and simulation times can be reduced, since on the one hand the program can be tested directly in the MR environment without disturbing the workflow. On the other hand, the robot model is rendered into the real working environment via the MR approach, thus eliminating the need for time-consuming and costly modeling of the environment.

The results of the user study indicate that the control concept is easy to learn, intuitive and easy to use. This facilitates the introduction of robots and especially in SMEs, since no expert knowledge is required for programming, programs can be created rapidly and intuitively and processes can be adapted flexibly. In addition, the user study showed that tasks can be solved efficiently and the concept is perceived as performance-enhancing.

Potential directions of improvement are: Implement various movement types, such as point-to-point, linear and circular movements in the concept. This makes the robot motion more flexible and efficient, since positions can be approached in different ways depending on the situation. Another improvement is to extend the concept with collaborative functions of the robot, such as force sensitivity or the ability to conduct search movements. In this way, the functions that make collaborative robots special can be integrated into the program structure. A further approach for improvement is to engage in a larger scale study.

7 Acknowledgement

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) and the European Social Fund (ESF) within the program "Future of work" (02L17C550) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

References

1. Rickert, M., Perzylo, A.: Industrieroboter für kmu. flexible und intuitive prozessbeschreibung. (2016) 1
2. Mueller, R., Franke, J., Henrich, D., Kuhlenkötter, B., Raatz, A.: Handbuch Mensch-Roboter-Kollaboration. (2019)
3. Perzylo, A., Somani, N., Profanter, S., Rickert, M., Knoll, A.: Toward efficient robot teach-in and semantic process descriptions for small lot sizes. (jul 2015) 1

4. Villania, V., Pinib, F., Lealib, F., Secchia, C.: Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. (feb 2018)
5. Guhl, J., Nguyen, S.T., Krüger, J.: Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft hololens. (2017)
6. Fang, H., Ong, S., Nee, A.: Robot programming using augmented reality. (2009)
7. Fang, H., Ong, S., Nee, A.: Robot path and end-effector orientation planning using augmented reality. (2012)
8. Lambrecht, J., Krüger, J.: Spatial programming for industrial robots based on gestures and augmented reality. (oct 2012)
9. Lambrecht, J., Kleinsorge, M., Rosenstrauch, M., Krüger, J.: Spatial programming for industrial robots through task demonstration. (dec 2012)
10. Abbas, S.M., Hassan, S., Yun, J.: Augmented reality based teaching pendant for industrial robot. (oct 2012)
11. Gaschler, A., Springer, M., Rickert, M., Knoll, A.: Intuitive robot tasks with augmented reality and virtual obstacles. (june 2014)
12. Cousins, M., Yang, C., Chen, J., He, W., Ju, Z.: Development of a mixed reality based interface for human roboter interaction. (2017)
13. Tran, N., Rands, J., Williams, T.: A hands-free virtual-reality teleoperation interface for wizard-of-oz control. (2018)
14. Foit, K.: Mixed reality as a tool supporting programming of the robot. (2014)
15. Rosen, E., Whitney, D., Phillips, E., Chien, G., Topkin, J., Konidaris, G., Tellex, S.: Communicating robot arm motion intent through mixed reality head-mounted displays. (2020)
16. Akan, B., Ameri, A., C, B., Cürüklü, Asplund, L.: Intuitive industrial robot programming through incremental multimodal language and augmented reality. (2011)
17. Sha, X., Jia, Z., Sun, W., Hao, Y., Xiao, X., Hu, H.: Development of mixed reality robot control system based on hololens. (2019)
18. Materna, Z., Kapinus, M., Beran, V., Smrz, P., Zemcik, P.: Interactive spatial augmented reality in collaborative robot programming: User experience evaluation. (2018)
19. Krupke, D., Steinicke, F., Lubos, P., Jonetzko, Y., Görner, M., Zhang, J.: Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction. (oct 2018)
20. Quintero, C.P., Li, S., Pan, M.K., Chan, W.P., der Loos, H.M.V., Croft, E.: Robot programming through augmented trajectories in augmented reality. (2018)
21. Ostanin, M., Klimchik, A.: Interactive robot programming using mixed reality. (2018)
22. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. (1994)
23. ROS-Industrial: Experimental packages for kuka manipulators within ros-indus-trial
24. Siemens: Ros-sharp. rosbridgeclient und urdfimporter
25. Schmidtler, J., Bengler, K., Dimeas, F., Campeau-Lecours, A.: A questionnaire for the evaluation of physical assistive devices (quead). (oct 2017)