Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-22 (2018) 50–55

Interactive Robot Programing Using Mixed Reality

M. Ostanin * **A. Klimchik** **

* *Innopolis University, Innopolis, 420500, Russia (e-mail: m.ostanin@innopolis.ru).*
** *Innopolis University, Innopolis, 420500, Russia (e-mail: a.klimchik@innopolis.ru).*

Abstract: The number of robots in industry is increasing every year, they are used to automate various processes. It is necessary to create a system for intuitive and effective interaction between robot and human. This paper represents a system for interactive programming of industrial robots based on Mixed Reality (MR). Microsoft HoloLens glasses were used for immersion in MR. By the developed system, a user without programming skills can assign a task to the manipulator. The main system parts are intuitive geometric path planning, time-optimal trajectory planning, and simulator based on MR. Via MR users are able to interact with robots and the spatial environment through the highly intuitive interface. The system comprises two KUKA robot with different kinematic models, with and without redundancy. The works show features and advantages of the MR-based system comparing to Augmented and Virtual Reality. Experiments on three different path cases show the system performance.

Keywords: Mixed Reality, Robot programming, Industrial robots, HoloLens

1. INTRODUCTION

Robots are often used to perform programmed tasks aimed at improving the efficiency and quality of the work, as well as replacing people in dangerous and boring tasks. A survey of the robot programming system was presented by Biggs and Macdonald (2003). According to that survey, a robot programming system can be divided into different methods, one group of them is Programming by Demonstration (PbD). PbD includes gesture, voice and vision interfaces, that help the user to program robot without programming skills. One of the PbD challenges is intuitive and effective human-robot interaction (HRI). HRI in industrial robotics is mainly limited to finding the best ways to reconfigure or program robots. Typically, HRI refers to the process that path description of the task in the sequence of movements of the robot, which corresponds to the robot's capabilities and work requirements. In particular, this includes path planning, retrieving specific points, and creating robots.

A recent study of the problems of HRI by Nee and Ong (2013) suggests that interfaces augmented reality (AR) can enrich the process of interaction by manipulating the robots. Moreover, there exists a mixed reality (MR), which plunges the user into a virtual environment(VE) deeper than AR. The usage of MR in robotic systems preserves the capabilities of virtual reality(VR) based systems. In addition, the entire production environment does not need to be replicated, and users can instantly perceive and interact with the geometric information associated with the planned paths.

The term mixed reality was originally introduced in a paper by Milgram and Kishino (1994). This paper presents the concept of the virtual continuum and focused on the categorization of taxonomy applied to displays. Since then, the application of MR goes beyond displays but also includes environmental input, spatial sound, and location. MR creates a space in which both physical and virtual elements co-exist. That enables elements in one world to react directly to what is happening into another via direct data communication.

In this research, we use Microsoft HoloLens as a holographic device, that provides the ability to place digital content in the real world as if it was really there. This device has the following limitations:

- displays the holograms only at a distance from 1.25 to 5m from the user, which makes impossible to attach a virtual object directly to the hand;
- rough construction of the spatial mapping, which leads to errors in the interaction of virtual objects with the real environment (RE);
- the policy of interaction is only through a limited number of gestures, adding motion controller would improve the accuracy of the system, and adding their gestures could improve the interface.

However, these limitations are not critical for establishing the system and assessing its capabilities.

1.1 Motivation

Via MR users are able to interact with robots and the spatial environment through highly intuitive interaction interfaces, such as gestures and virtual tools, and perceive instantaneous feedback through optical tracking devices. The interaction with a virtual robot model, instead of the real robot, makes the operator job safer.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.11.517

The aim of the work is to develop an MR-based system capable of performing robot programming and task planning in an efficient and intuitive manner. The research investigates the potential of MR in robotics applications and the ways that MR can add the values in solving classic robotics problems, particularly in robot EE trajectory planning, such as the planned trajectories can be translated directly into the robot controller.

1.2 Contribution

The first contribution is the interactive programming system prototype which shows high potential and possibility for usage in our days. This prototype provides an understanding of the possibilities and the prospects of such a system. We designed various virtual objects and developed a methodology for geometric path planning, furthermore, we implemented a virtual simulator based on MR.

The second theoretical contribution of that paper is a new Mixed Reality interface based on Microsoft HoloLens. The interface is designed in such a way that each system feature has its own understandable virtual representation and interaction scenario. It gives a possibility to program robots without programming skills.

1.3 Overview

To address these issues the remainder of the paper is organized as follows. Related works are observed in Section 2. Section 3 describes the system hardware and software parts. Section 4 presents the methodology: MR interface and trajectory planning. The experiment results are shown in Section 5. Section 6 provides a discussion and analysis of the MR. Section 6 summarises the main contributions of the paper and presents directions for our future work.

2. RELATED WORKS

In the original MR definition, AR and Augmented Virtuality (AV) are seen as special instances of MR (Milgram and Kishino (1994)). In AR, virtual objects are projected onto the physical environment, while in AV, physical objects are incorporated into a virtual environment. Basically, in manufacturing AR system provides additional information to the employee by applying the illustration to the video stream, such as tips for assembling or operation, the robot state, a safe zone for working, and so on, there are few related works Michalos et al. (2016); Makris et al. (2016); Rentzos et al. (2013). There are also studies aimed at the analyzing system of the robot programming using AR (RPAR-I and RPAR-II systems) by Chong et al. (2009); Fang et al. (2009, 2012); Gaschler et al. (2014). These papers present methodology for manipulator trajectory planning by means of demonstration with the help of a marker traced by the camera and planning a path without collisions by specifying the free volume by the same marker.

In the work of Honig et al. (2015), the authors describe the benefits of robotics for MR over AR. The first highlighted advantage is a simplification of debugging. The shared physical and virtual space in MR reduces the gap between simulation and implementation. The second benefit is unconstrained additions to the robots, users can easily modify the virtual robot. The third advantage is that the MR eliminates safety risks. The fourth benefit is spatial flexibility. By this feature, a group of users can do experiments with robots together and remotely. Spatial flexibility gives the possibility to implement a scaled environment. In the study performed by Rosen et al. (2017) the HRI interfaces were compared, namely, without visualization, visualization through the display and MR. The results demonstrate the potential benefit of MR to communicate with the robot. Cousins et al. (2017) developed the MR interface for HRI based on Oculus Rift DK2 and LEAP Motion, limitation of that work is the webcam usage for workspace displaying. HoloLens combines functions of Oculus Rift DK2, LEAP Motion, a webcam is not needed. Guhl et al. (2017) presented the potential system architecture for programming industrial robotics based on HoloLens.

The developing system is based on the MR, thus it expands the above-described AR system capabilities. Our system has special features which allow MR to simulate the virtual robot movements in a real environment and analysis automatically its workspace. The user may build and change a trajectory using an interactive and intuitive interface with instant visual feedback. The architecture of the system is designed in such a way that it would be easy to integrate new robot models. Our research uses well-known methods for planning and trajectory optimization, the result is a comprehensive system for interactive robot programming based on MR.

3. SYSTEM DESCRIPTION

The system consists of physical and virtual objects. Physical objects are a MR glasses HoloLens, two robots from KUKA company with the different kinematic model (KR 10 r1100 (Agilus) and the IIWA LBR 14), robot controllers, a desktop computer and a real working environment, Fig. 1. The virtual objects of the system are robot's models, the menu, the virtual map of workspace and interaction tools. The virtual workspace map is built using HoloLens and it repeats the geometric parameters of the real world.

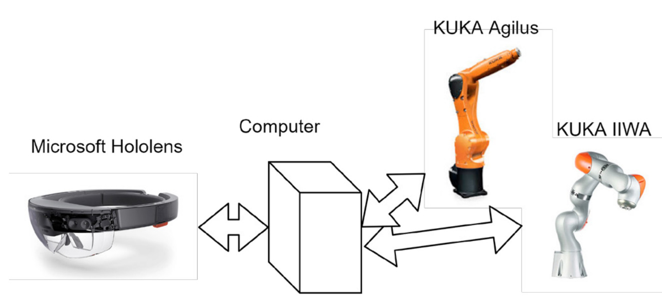

Fig. 1. System Hardware

To create a system, based on HoloLens, we used game engine Unity3D and Mixed Reality Toolkit-Unity (MRTK). MRTK contains scripts, special components and sample applications for accelerated program development for HoloLens. It is important to note that projects for HoloLens are built on the basis of Universal Windows Platform framework, this imposes restrictions on the using various third-party libraries.

Virtual robot models KR Agius and IIWA were taken from the ros-industrial /Kuka-experimental package. The geometric model of each link is represented as a separate model and accurately follows the real geometric parameters and features of the link. Each component is exported to a format .FBX, this is necessary for Unity. The virtual robot is assembled from these individual components. The value of joints angles in the real manipulator corresponds to the rotation of one virtual link relative to the other. The rest of the virtual models were taken from the MRTK examples or created in Unity.

Main software blocks are shown in Fig 2. There are 4 separate parts: Application Manager, Geometrical Path Planner, Trajectory Planner and Simulations. App Manager controls all blocks. Geometrical Path Planner block sends to Trajectory Planner software block the points sequence, each point contains Cartesian position and orientation in yaw-pitch-roll angles. After trajectory planing we obtain joint's velocity and time array which is applied to the robot simulator.

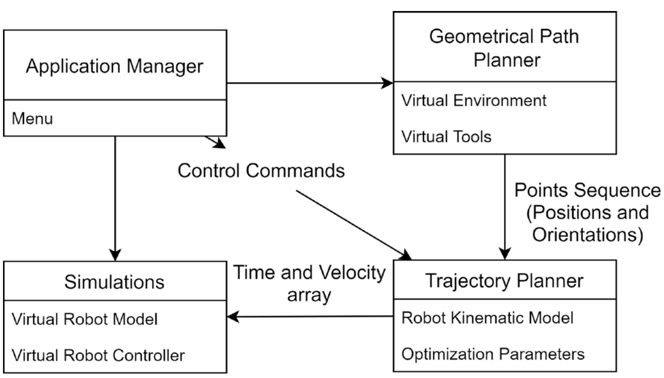

Fig. 2. System Software Blocks

Human interaction with a virtual robot occurs using glasses through gestures, virtual menus and virtual instruments. To program the robot for different tasks, there are different interaction scenarios. For example, for point to point (PTP) operation, the operator specifies the starting and finishing position for the manipulator, selects the configuration for the robot for these 2 positions. After the task formation, the geometric movement path is planned and the trajectory is planned to take into account the robot kinematic model and its limitations, as joints limits, speed, and acceleration. Then the simulation is performed and if an operator is satisfied with the result of the simulation, the resulting trajectory is sent to the real manipulator.

4. SYSTEM METHODOLOGY

4.1 Human – virtual robot interaction

In HoloLens interaction is carried out through gestures:

- air tap, it is a simple click;
- tap and hold, more complex click interactions when combined with arm movement;
- manipulation. it used to move hologram by 1:1 reacts to the user's hand movements.

The click area is determined by the cursor that is located in the center of the screen and rotates with the user's head. By that cursor a person interacts with virtual objects.

The first object that meets the user is the menu. The menu helps the user to select the manipulator model and set up the system, shows robot state, control trajectory creating a process and etc. The next important virtual object is a spatial map of the world. It gives us the opportunity to analyze the RE, for example, to find a free space for movement planning. It also allows the user to specify a point directly on the surface of real objects. A virtual goal point object is used to specify a frame which describes the transformation matrix to which the manipulator end-effector (EE) should come, it sets the goal position and orientation. The operator has the ability to draw a path for the EE movement, it is carried out through a virtual pointer which attaches to the user's hand and repeats its movement.

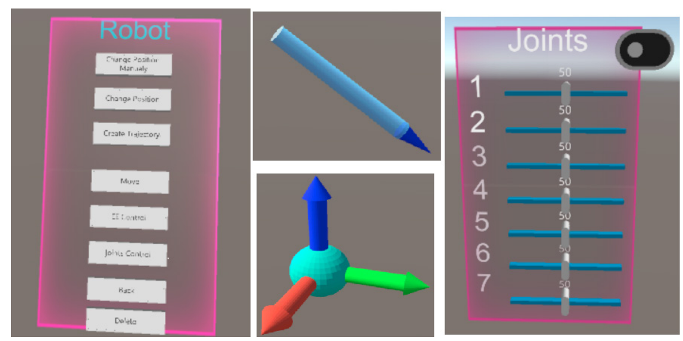

Fig. 3. Virtual tools: Main menu, Pointer, Goal Point, Joints information menu

4.2 Robots kinematics models

Configuration the robot is presented as a vector $q = [q_1, q_2, ..q_n]$, where n is the number of degrees of freedom(DoF) of the manipulator. For KR Agilus DOF is 6, and for IIWA is 7. Finding the trajectory of the robot occurs by solving the forward and inverse kinematic problem. The product of transformation matrices is used to solve the direct kinematics. Inverse kinematics of a robot without redundancy is considered analytically using Pieper's method, and for a manipulator with redundancy Saturation in Null Space (SNS) method is used (Flacco et al. (2015)). SNS algorithm base on velocity control, joints velocity $\dot{q}$ can be obtained from:

$$\dot{q}_{SNS} = \dot{q_N} + J^{+} \cdot (s \cdot \overline{x} + J \cdot \dot{q_N}) \quad (1)$$

where J^+ is the pseudo inverse of the Jacobian, a task scaling factor s (0, 1] and a null-space velocity vector $\dot{q_N}$.

Both methods take into account joints limits and have the option of choosing a more preferred configuration based on the previous position. In addition, the operator can choose the configuration of the robot for a certain position of EE possible options. Along the path specified by the operator, there may be unreachable points, for example, outside the workspace, in that case, the system will indicate where is needed to make a change.

4.3 Tools and Grippers

The system has the capability to choose an additional virtual tool or gripper for the robots. Tools are universal and can be applied to each virtual robot model. All tools take into account their geometric model and unique features. The kinematic model of the robot varies depending on the selected tool.

4.4 Geometrical Path planning

The geometry of the path is specified with the help of goal points. These points can be positioned and oriented in two ways. The first, in this course the point normal to the plane of the object. The second, using gestures and menus, the user sets the position and orientation independently and manually. Next, a user creates the second and subsequent points.

The path between 2 goal points can be carried out in one of the following ways:

- Point to point(PTP) without any constraints in the Cartesian space. Each point corresponds to one joint configuration. The robot will move from the beginning configuration to the final;
- Line. That line is presented as a sequence of via points which are visible for the users and each step presents as PTP motion.
- Arc. For settings the arc user should choose 2 goal points which set start and end frame, then user set the third point without orientation. Arc presents similarly to the line.
- User path. This is the specific option when users can draw the path between two points using the special virtual tool (pointer).
- Without collision path. The system finds the shortest path in the Cartesian space.

After settings, it is illustrated as points sequence, fig. 4, except for the case with PTP.

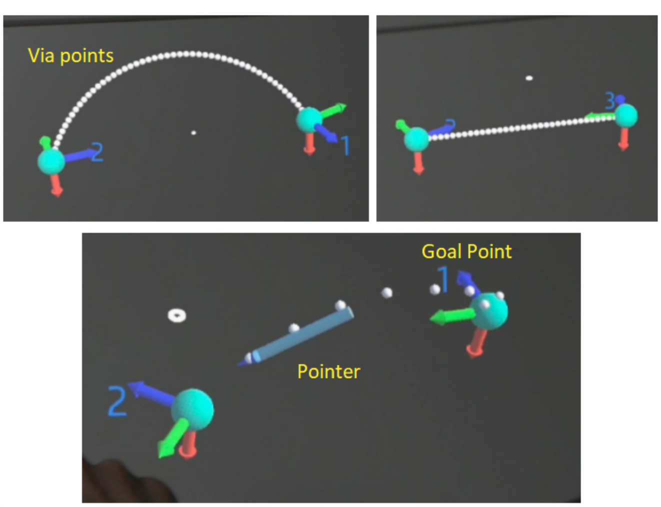

Fig. 4. Lines between 2 goal points: Arc, Line, User path with pointer.

To find the shortest path in the Cartesian space, a weighted graph is constructed. Each vertex corresponds to a free space cube between two goal points and edges connect the nearest cubes. The edge weight is the distance between the cubes centers. The construction of these cubes occurs in the space between two goal points. If the cube captures an obstacle, it is not considered. The shortest path in this graph corresponds to the desired geometric path between the points. If the path is not found, the analyzed space is increased.

As a result, the path is obtained as a sequence of goal points, which are connected by different lines. If the user does not specify a path, PTP is used by default.

4.5 Path modification

The user can change the desired section of the path by deleting the existing part and creating a new one, or by inserting additional goal points into the path. During path planning, some sections are needed to set with high accuracy. In that case, the user can scale the desired area and then more specify accurately the path in the right place. It is important to note that scaled part of the path is displayed together with the scanned map fragment of the RE. This is necessary for the correction to take place taking into account the working place.

Interface for path scaling contains 2 cubes, the small and the big ones, Fig. 5. The small cube is located in a place where the user wants to change the path. After that virtual objects inside the cube are displayed inside the big cube with 2 or more high scale. The user can interact with scaled objects and all changes automatically applied to the path with real scale.

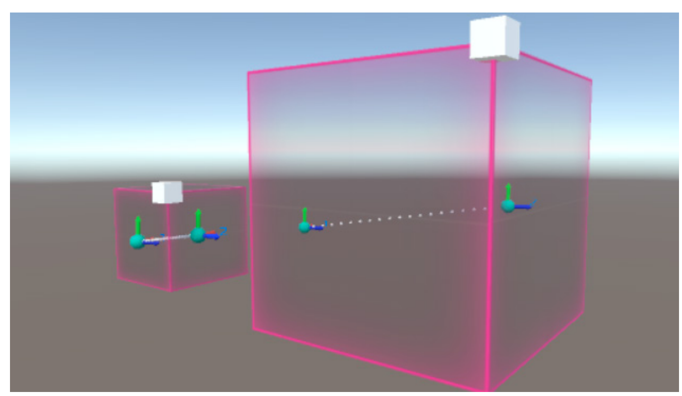

Fig. 5. Path scaling.

4.6 Time Optimal Trajectory planning and Simulation

Before planning a trajectory and simulation of motion, it is necessary to accurately set virtual robot position in place of the real one. This is needed for an accurate transition from the Cartesian Space to Joint Space. Then the trajectory planning takes place under each robot separately. For minimum-time trajectory planning we realize a practical approach which is described by Kim et al. (2010), that method allows smoothing of the trajectory near control points. The size of the smoothing is set by the user. In addition, a more optimal path for the manipulator can be obtained by changing the discretization of the via points generation during the path geometry planning stage using the following optimization equation:

$$(\alpha_1 \cdot \int_0^T |q - q_{real}| + \alpha_2 \cdot T) \rightarrow min \tag{2}$$

where T is total time, α_1 and α_2 are optimization coefficients. In the case of an automatic path search between 2 goal points, the discrete optimization methodology is used, that approach described in work by Ostanin et al. (2018). The space between the points is divided into cells and a weighted graph is constructed between two points through the obtained cells, then the problem of finding the minimal path is solved. By setting different weight functions, the optimal path can be found from different points of view: time, displacement in the Joint space or the wasted energy.

To simulate the motion the robot controller was created, which applies the joint velocity to virtual robot links. Velocity and time array are obtained from trajectory planning stage. The resulting working cell can be copied and transferred to a new place. The application makes it easy to replace the robot model and plan the trajectory again.

5. EXPERIMENTS

To demonstrate the capabilities of the system, we considered 3 different cases: pick and place operation, circular trajectory and rectangular trajectory.

Case 1. PTP with obstacle avoidance

The start and end position of the EE were assigned using goal points in a such a way that there is an obstacle in the space between these 2 positions. As a result, the system builds a shortest path for a workaround. All stages are shown in the Figure 6. Firstly, the user walks around the working place and scan it with HoloLens. Then, the user sets the goal points whose orientation can be defined manually or as normal to the object at that point. On the next step the user chooses a point connection as the shortest path in the Cartesian space. Finally, the user runs the robot simulation.

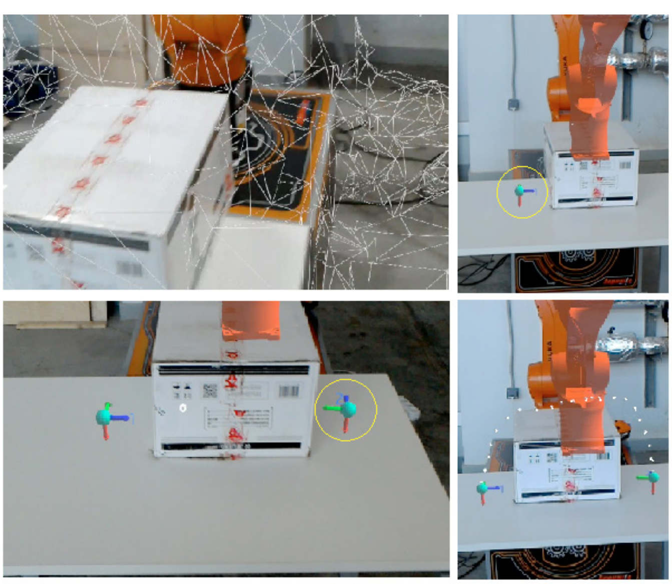

Fig. 6. PTP with collision avoidance. Scanning, Goal points setting, Path planing.

Case 2. Move along the Circle

Figure 7 shows the steps for the circle trajectory planning. The circle consists of 2 arcs. Each arc contains 2 control points, and additional 3rd and 4th points to form the arcs. The action sequence is the same as in the case 1, but for the goal point connection the user chooses Arc.

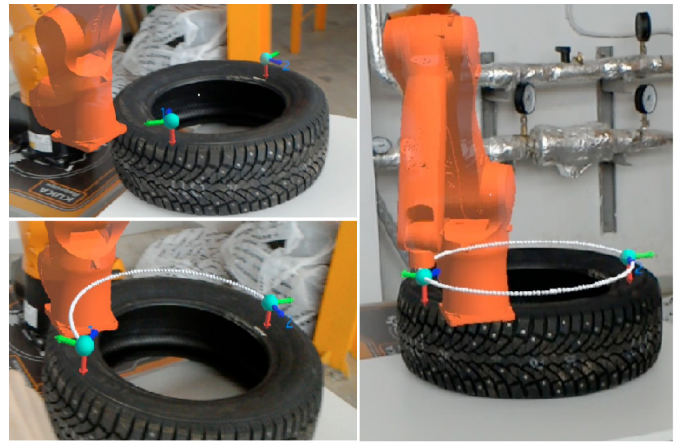

Fig. 7. Circle curve. Goal points setting, First arc setting, Simulation.

Case 3. Rectangle curve

For the passage of the Rectangle curve it is required to set 4 corner points and connect them by straight lines. Figure 8 shows key steps of this process.

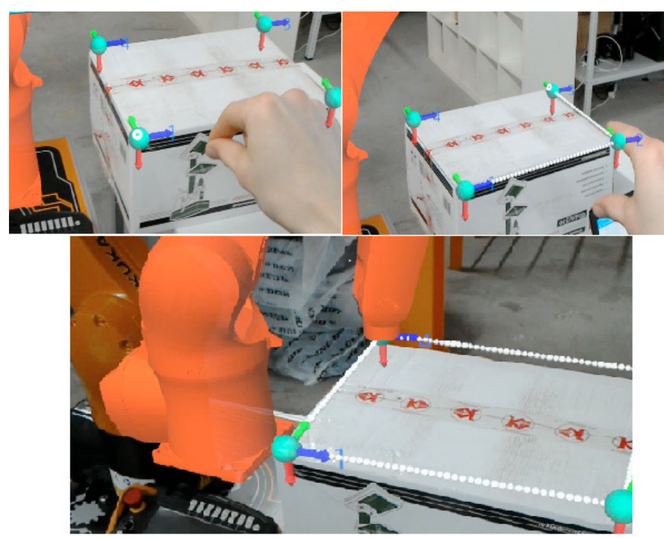

Fig. 8. Rectangle curve. Goal points setting, Lines setting, Path planing.

6. DISCUSSIONS

To highlight the advantages of MR let us show its comparison with AR and VR solutions. The first difference is the ability of mapping of real space into virtual space. It is impossible to deal with a real space in VR and it can be at a low level in AR. The shared space in the VR and AR applications is related to virtual components only. Simulation of virtual robots in the real environment is possible only in MR, theoretically, it is possible to realize in AR, but without interaction with the real workspace. In our study, we realized trajectory scale feature, resizing of the virtual and real part that is possible in MR only, in VR and AR it is possible to resize only virtual objects.

The summary of performed comparison analysis is given in Table 1.

Table 1. Comparison Digital Reality Features

Feature	MR	AR	VR
Mapping VE and RE	Yes	Yes/No	No
Sharing space	Yes	Only VE	Only VE
Simulation of Virtual in Real Space	Yes	Yes, but without interaction with RE	No
Scale	Yes	Only VE	Only VE
Obstacle avoidance	Yes	Additional device to scanning RE	Avoiding of VE
Real Robot Control	Yes	Yes	No

7. CONCLUSIONS

In this paper, a system for intuitive robot programming based on MR was proposed. We developed a methodology for planning the EE geometric path including orientation. Technologies of optimal trajectory planning by time and displacement are applied, as well as discrete optimization methodology for automatic path planning. The possibilities of the system for programming of the industrial robots based on MR in comparison with the AR and VR systems are analyzed, the advantages of MR are highlighted. Geometrical path planning is based on the goal points representing the EE locations. Goal point can be connected in different ways: PTP, line, arc, use path and the shortest path without collision. The path can be modified and specified using scaling feature. The system performance was shown on three different experiments.

7.1 Future works

In the future, we are planning to extend the potential of the system and realize several new features. One of the most important facilities is a multiuser mode, which will provide the users with the possibility to cooperate while assigning the robot task and observe the robot working process in the virtual environment. It is expected to integrate VR, AR and MR devices, which will provide different functionality depending on the gadget capabilities. For example, with VR glasses only the virtual part will be shown, and with a smartphone a picture from the camera on the glasses HoloLens will be visible. Further, the system can be expanded by the positioning table and multi-robots capabilities, which will require group trajectory planning for common workspace.

ACKNOWLEDGEMENTS

This work has been supported by the Russian Ministry of education and science with the project "Development of anthropomorphic robotic complexes with variable stiffness actuators for movement on the at and the rugged terrains" agreement: No14.606.21.0007, ID: RFMEFI60617X0007.

REFERENCES

Biggs, G. and Macdonald, B. (2003). A Survey of Robot Programming Systems. *Proceedings of the Australasian conference on robotics and automation*, 1–3.

Chong, J.W., Ong, S.K., Nee, A.Y., and Youcef-Youmi, K. (2009). Robot programming using augmented reality: An interactive method for planning collision-free paths. *Robotics and Computer-Integrated Manufacturing*, 25(3), 689–701.

Cousins, M., Yang, C., Chen, J., He, W., and Ju, Z. (2017). Development of a mixed reality based interface for human robot interaciotn. *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, 27–34.

Fang, H., Ong, S., and Nee, A. (2012). Robot Path and End-Effector Orientation Planning Using Augmented Reality. *Procedia CIRP*, 3, 191–196.

Fang, H., Ong, S.K., and Nee, A.Y.C. (2009). Robot Programming Using Augmented Reality. *2009 International Conference on CyberWorlds*, 13–20.

Flacco, F., De Luca, A., and Khatib, O. (2015). Control of Redundant Robots under Hard Joint Constraints: Saturation in the Null Space. *IEEE Transactions on Robotics*, 31(3), 637–654.

Gaschler, A., Springer, M., Rickert, M., and Knoll, A. (2014). Intuitive robot Tasks with augmented reality and virtual obstacles. *Proceedings - IEEE International Conference on Robotics and Automation*, 6026–6031.

Guhl, J., Tung, S., and Kruger, J. (2017). Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft HoloLens. *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–4.

Honig, W., Milanes, C., Scaria, L., Phan, T., Bolas, M., and Ayanian, N. (2015). Mixed reality for robotics. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem, 5382–5387.

Kim, J., Kim, S., Kim, S., and Kim, D. (2010). A practical approach for minimumtime trajectory planning for industrial robots. *Industrial Robot: An International Journal*, 37(1), 51–61.

Makris, S., Karagiannis, P., Koukas, S., and Matthaiakis, A.S. (2016). Augmented reality system for operator support in humanrobot collaborative assembly. *CIRP Annals - Manufacturing Technology*, 65(1), 61–64.

Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö., and Chryssolouris, G. (2016). Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation. *Procedia CIRP*, 41, 370–375.

Milgram, P. and Kishino, F. (1994). A Taxonomy of Mixed Reality Visual-Displays. *Ieice Transactions on Information and Systems*, E77d(12), 1321–1329.

Nee, A. and Ong, S. (2013). *Virtual and Augmented Reality Applications in Manufacturing*, volume 46. IFAC.

Ostanin, M., Popov, D., and Klimchik, A. (2018). Programming by Demonstration Using Two-Step Optimization for Industrial Robot. *16th IFAC Symposium on Information Control Problems in Manufacturing, 11-13 June 2018, Bergamo, ITALY.*

Rentzos, L., Papanastasiou, S., Papakostas, N., and Chryssolouris, G. (2013). *Augmented Reality for Human-based Assembly: Using Product and Process Semantics*, volume 46. IFAC.

Rosen, E., Whitney, D., Phillips, E., Chien, G., Tompkin, J., Konidaris, G., and Tellex, S. (2017). Communicating Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays. 1–16.