In: 2024 12th International Conference on Control, Mechatronics and Automation (ICCMA), 11-13 November 2024, London, UK
DOI:10.1109/iccma63715.2024.10843879

The development of an augmented reality gesture control human-robot interface

Akhil Hariharasudhan
Centre for Robotics and Assembly
Cranfield University
Cranfield, UK
akhil.hariharasudhan@smccl.co.uk

Gilbert Tang
Centre for Robotics and Assembly
Cranfield University
Cranfield, UK
g.tang@cranfieldl.ac.uk

Phil Webb
Centre for Robotics and Assembly
Cranfield University
Cranfield, UK
p.f.webb@cranfield.ac.uk

Abstract— **There is an increased need for intuitive methods to control and interact with collaborative robot systems that are driven by industries such as manufacturing that involve complex robot operations. Traditional control and programming approaches that involve handheld devices can be both cumbersome and potentially hazardous for operators, as unexpected movements or malfunctions may result in dangerous situations, they also restrict operators' movements and hinder their ability to respond quickly to changing situations, ultimately slowing down overall operations. The implementation of cutting-edge interface technology such as Augmented Reality (AR) and gesture control can revolutionise robot systems and propel companies towards Industry 4.0. However, a significant gap exists in the realm of user-friendly and dependable AR-based interfaces that seamlessly integrate with robot systems, guarantee safe and precise operations, and reduce the likelihood of operator errors and accidents.**

This paper demonstrates the benefits of developing and deploying AR gesture interfaces to empower robotics operators to control and interact with manipulators in more natural and efficient manners. This interface could represent a substantial advancement in addressing the problems presented by conventional control methods, ushering in a new era of robot system control and interaction in complex industrial settings.

An experimental approach was developed to investigate the feasibility and effectiveness of using AR to control robot systems. A comparative experiment to evaluate the effectiveness and usability of an Augmented Reality (AR) gesture interface, developed using HoloLens 2 and UR16e robot in contrast to the conventional teach pendant control method. The study aims to provide valuable insights into the utility and user-friendliness of the AR gesture interface for robot system control from users' perspectives. The devices have been compared using participants who have engaged in a series of tasks involving robot movement, manipulation, and interaction.

Keywords—Augmented Reality, Gesture Interface, Human-Robot Interaction, collaborative robot.

I. INTRODUCTION

In industries that involve complex robot operations, such as manufacturing, there is a need for more intuitive and efficient methods to control collaborative robot systems. The traditional methods of using teach pendants can be cumbersome and may pose safety risks to operators. Additionally, there is a growing demand for leveraging emerging technologies like augmented reality (AR) to enhance control and interaction with robot systems. However, there is a lack of a user-friendly and reliable AR-based interface that seamlessly integrates with collaborative robot systems, ensuring safe and accurate control while minimising the risk of operator errors and accidents. Thus, there is a need to design and develop an augmented reality gesture interface that enables manipulator operators and programmers to control robot systems more intuitively and efficiently, enhancing safety, productivity, and user experience in industrial environments.

Developing a natural and intuitive method of interaction between humans and robots by fusing AR technologies with gesture-based interfaces can improve the user experience. It enables more natural and intuitive user interaction with robots. Operators can communicate with the robot by using hand gestures and movements rather than touchscreens or teach pendants. This approach reduces the learning curve and enhances the accessibility of robot systems, making them more user-friendly for a wide range of users, including those with limited technical expertise.

The incorporation of AR improves users' perception of the robot's activities and provides real-time visual feedback. Users can view virtual features overlaid on the actual surroundings, which gives them a deeper grasp of the robot's capabilities and intents. The visual input can increase user confidence in the robot system and make it possible for people and robots to work together more successfully.

Furthermore, the creation of an AR reality gesture interface creates opportunities for cutting-edge uses in a variety of fields. For instance, in industrial settings, employees can communicate with robots via gestures, allowing them to direct and oversee the robot's movements without the need for in-depth instructions or specialised knowledge.

II. RELATED WORKS

User interaction plays a pivotal role in the adoption of robotic technology within industries. The user interface must be intuitive, replacing traditional control methods and providing an immersive experience. One approach employs deep learning-based object detection, using RetinaNet to estimate real object positions in the physical world. Users control the robot through head gestures for translation and rotation, with the digital twin computing inverse kinematics based on the object's pose [5]. This method outperformed joint-based or end-effector methods, resulting in faster task completion with fewer voice commands.

Another method uses Leap motion sensor and Kinet V2 to track hand movements and obtain human coordinates. This data is processed through the Kalman Filter to teleoperate the robot in real-time[3]. Hand gestures, like palm outstretched and fist closed, control the robot's movements. Separating the robot's tasks into two threads enhanced user interaction, allowing the manipulation of robot parameters, task instructions and shutdown procedures via gestures and voice commands [2]. Hand coordinates tracked using HoloLens 2, are translated to robot language through translators like RAPID for ABB robots. Users define the robot type, coordinate system and workspace recording the paths using Air Tap gesture [7]. This method ensures alignment between the robots and HoloLens 2's coordinates, enabling sage robot interaction within a virtual workspace. Another approach classifies into static, dynamic and composed categories using IMUs [4]. The robot task manager processes these gestures and provides visual and speech feedback. A UWB positioning system maintains proximity control. While intuitive, this method had a longer interactive process and complex composed gestures.

Multiple control options offer flexibility. Manual control moves the holographic robot end effector, which mirrors the real robot. Automatic control allows users to position the robot and execute safe motions. However, the accuracy of motion plan previews may vary due to unity model angles [8]. Accurate robot pose registration was achieved using a 2D marker for alignment between the virtual environment and the real world. A SLAM algorithm in HoloLens enabled two pick-and-place methods. The heading and commit, allow users to select objects with their fingers [1]. The heading and commit method proved more precise, faster and less demanding than point and commit.

To ensure the manipulator creates a safe trajectory to the target position, there should be a system which allows the robot to move to the goal position without colliding with the surroundings or failing. A model for service-oriented architecture between HoloLens and UR5 industrial robots to enable intuitive pick-and-place activities was proposed [1]. This structure enables simple integration, component exchange, and scalability to several workstations and gadgets. The pick-and-place functionality of this method can be completed by a human operator dragging and dropping recognised objects in the required location which creates the required path for the robot to move. Another approach calculates the robot's inverse kinematics with respect to the virtual object and transmits the derived kinematic information to the robot, enabling more efficient HRI than traditional conventional direct manipulation[5]. The most frequently used approach was calibrating the Mixed Reality – Head Mounted Display (HMD) coordinate system to the ROS coordinate system and then defining the target using gestures which will be sent to ROS Reality to compute a motion plan [6]. The poses that the manipulator needs to reach are sent via ROS Reality and ROS Reality enables HoloLens to publish and subscribe topics to a ROS network using Web Sockets which also sends feedback to the HMD. This enables the operator to inspect the path created consequently preventing any potential collisions or failure.

This paper focuses on the development of an AR gesture interface to enhance user interaction and experience with robot systems. AR enriches users' perception of robot activities increasing confidence and facilitating successful human-robot collaboration. The goal is to seamlessly integrate gesture recognition and real-time communication for improved engagement, control accuracy, and user experience. The project contributes to AR technology in human-robot interaction, providing an efficient means of robot. In addition, the project also focuses on usability, performance and ethical considerations in alignment with industry standards, ensuring that any images or videos captured while using the application will only be saved with the explicit consent of users and will be deleted before the end of the month. This work lays the foundation for future research and development in AR interfaces for interactive robots.

III. System Architecture

The developed system architecture consists of three layers as illustrated in Fig.1. The first layer is the AR application development layer using the Unity and Mixed Reality Toolkit (MRTK) for the HoloLens2. The middle layer is the backend of the application which is the development of the inverse kinematic algorithm and communication protocol. The final layer is the conversion of the commands to URScript commands to control the Universal Robot UR16e robot arm.

The set of predefined gestures from MRTK 2 will be used to control the digital twin. This is then used to derive the joint values in degrees and send them to the URScript using the Real Time Data Exchange protocol.

A. Robot digital twin

The digital twin of the UR16e robot was designed in Fusion360, and the link lengths and the frame assignments from the universal robot were used to create the Computer Aid Design model. This will be used to help the user visualise the robot in the HoloLens 2 as illustrated in Fig.2.

The CAD model had to be exported to Blender where the parent-child link of each joint is defined and exported as a game object into Unity.

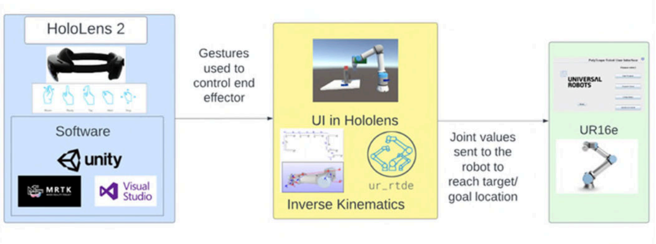
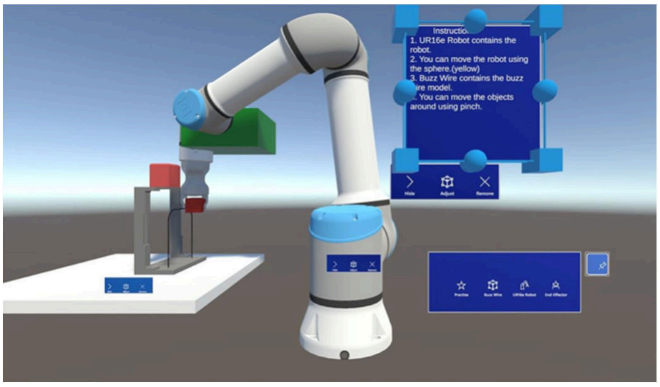


Fig. 1. System architecture



Instructio
1. UR16e Robot contains the robot.
2. You can move the robot using the sphere.(yellow)
3. Buzz Wire contains the buzz ire model.
. You can move the objects around using pinch.

Fig. 2. Augmented reality user-interface

B. *Robot control methods*

Controlling the UR16e using AR involves overlaying digital interfaces onto the physical world, allowing users to interact with the robot through intuitive and immersive methods. The two control methods implemented to manipulate the movement of UR16e are virtual sliders for joint control and Cartesian movement control.

The first method is Virtual Sliders for Joint Control in the AR interface that corresponds to each joint of the robot. Users can adjust the position of these sliders to control the angles of individual robot joints. This is useful for direct control over joint angles and can provide a clear representation of the robot's configuration. The virtual slider control method is similar to the teach pendant. Pinch gestures can be used to control each augmented slider, this will highlight the joint the slider belongs to and update the position when moving the slider in the digital twin. The angle of the specific joint is sent using MoveJ command in RTDE protocol.

In the Cartesian movement control method, users can manipulate the end effector of the robot directly in the AR environment. By moving their hands or other AR interaction devices, users can control the position and orientation of the robot's end-effector. This approach simplifies control by focusing on the robot's tool position rather than individual joint angles.

The existing Cartesian algorithm encountered significant issues due to its utilisation of the MoveL command for robot movement, allowing users control over X, Y, Z, and rotational parameters rx, ry, rz in both positive and negative directions. The MoveL command, inherently designed for linear movements, resulted in inaccuracies and unpredictability in the UR16e's positioning. By relying on MoveL, the algorithm facilitated trajectories that the robot was not optimally designed to execute, causing motor jitter and potential damage. The robot's natural kinematics and joint configurations were not effectively leveraged, leading to suboptimal paths and increased difficulty in achieving desired poses. To address this, a more suitable alternative involves transitioning to the MoveJ command. Unlike MoveL, MoveJ allows for controlled joint movements, enabling precise and predictable positioning of the robot. This approach aligns with the UR16e's joint-based kinematics, ensuring smoother and more accurate trajectories.

Inverse kinematics involves calculating the joint angles required to achieve a desired end effector position. In an AR control scenario, users would specify the desired position and orientation in the AR interface, and the system's software would automatically compute the joint angles necessary to achieve that configuration. This method is user-friendly as it abstracts away the complexities of joint-level control.

To ensure safety, joint limits and singularities must be addressed. Safety constraints are implemented to validate that the calculated joint angles fall within the acceptable range defined by the robot's specifications. Additionally, measures are taken to prevent singularities, where certain configurations in the robot's kinematic chain could impede its ability to reach a position or result in unstable inverse kinematics solutions. This is mitigated by configuring the controllable cube in a way that avoids enabling the UR16e to enter a singularity state, which can be achieved through constraint scripts integrated into the Unity application.

Fig.2 shows the green cube the operator can use to control the robot. The cube is near the end effector and is restricted to reach near the singularity of the robot. The green cube is attached to predefined scripts, “NearInteractionGrabbable” and “ObjectManipulator. The green cube can be moved using pinch gesture, this will send the coordinates and orientation of the cube as the goal position to the inverse kinematic algorithm. Then from the inverse kinematic algorithm, all the angles required by the joints are extracted. This will update the position of the digital twin. Afterwards, the angles are converted to degrees which are then sent to The Real-Time Data Exchange (RTDE) protocol. Finally, the URScript which will have updated values of the joint positions will be sent to the robot system. As a result, the robot will also update to the same position as the digital twin.

C. *RTDE Client*

The RTDE interface provides a way to synchronise external applications with the UR controller over a standard TCP/IP connection, without breaking any real-time properties of the UR controller. This functionality is among others useful for interacting with fieldbus drivers e.g. Ethernet/IP, manipulating robot I/O and plotting robot status e.g. robot trajectories.

A setup process and a synchronisation loop make up the two stages that make up the RTDE functionality. The client oversees configuring the synchronisation variables upon connecting to the RTDE interface. The client can be instructed to write to and read from any combination of input and output registers. To achieve this the client transmits a setup list of specified input and output fields that must be included in the actual data synchronisation packages to accomplish this. The RTDE operates on port 30004.

Where read_input_float_register is the angle at degrees for each joint in the UR16e. The HoloLens 2 application communicates with the UR16e robot using the RTDE protocaol, utilising the ServoJ command to control the joint angles in real-time. The gain parameter is set to 300 which adjusts the robot’s response speed and lookahead_time influences response smoothness. Through multiple iterations, optimal values for lookahead_time, gain and time were determined to ensure stable and safe UR16e control via the digital twin.

D. Calibration process

The calibration process is a crucial stage when using the app as a new user. Calibration ensures that the virtual elements align accurately with the physical environment. Proper calibration prevents misalignment issues, where virtual objects might appear detached from their intended real-world context.

When the user opens the app in HoloLens 2, they first have to go through the calibration process. HoloLens 2 uses eye-tracking technology to improve the user experience when interacting and seeing the virtual environment. It ensures accurate eye-tracking and assists the user with hand tracking, hologram alignment and comfort.

E. Elements in the user interface of AR application

The elements of a user interface (UI) in an app are the visual and interactive components that users interact with to navigate, perform actions and access information within the application. These elements play a critical role in ensuring that the app is intuitive, user-friendly and visually appealing.

The AR UI has a menu with four options:

1. Practise: creates a digital twin of UR16e that can be controlled offline for practising sessions.

2. Buzz Wire: projects the virtual buzz wire game for better visualisation during the task.

3. UR16e Robot displays the digital UR16e which is connected to the real robot system.

4. End-effector: The end-effector can be visualised which will assist users in completing the task.

An instruction block in the interface describes the functionality of each button on the menu and the control technique to the user. The AR application has two stages, where the first stage allows the user to learn about the different gestures that can be used to control the robot and other augmented objects. Participants can control the UR16e using the green cube shown in Fig.2 and learn the controlling method before working with the robot system. The second stage consists of the same scene, but it is connected to the real robot.

IV. EXPERIMENTAL SETUP

A "buzz the wire" game was set up to compare the developed AR gesture control system against a conventional touch screen teaching pendant. 20 participants (13 male, 7 female) were involved in the experiment where they had to control the UR16e robot using both the Teach Pendant and HoloLens 2 to complete a task. Among the participants, over 50% had prior experience with robot manipulators, while the rest were inexperienced. All participants were university students with diverse academic backgrounds, ensuring a varied and comprehensive assessment of the AR application's performance compared to the Teach Pendant, catering to both experienced and novice users.

To ensure fairness, 70% of participants had no prior experience with AR/VR devices, enhancing the accuracy of the comparison. This approach minimised differences in relevant characteristics, aside from AR/VR experience, across the participant pool. Additionally, qualitative data, including feedback and suggestions, were collected to gain insights into user perceptions and experiences with the AR application.

A. Conventional control

The Teach Pendant is a handheld device used to manually control and program industrial robots, including the UR16e robot. It provides a user interface that allows operators, technicians, and programmers to interact with the robot, teach it new tasks, and control its movements.

Participants using the Teach Pendant could move the robot in a way that mimics manual operation, which is particularly helpful for accurately positioning the robot. Using the Teach Pendant effectively requires familiarity with the robot's manual, control interface, and programming capabilities. There are two ways users can move the joints in the UR16e:
Move Tool: enables Cartesian movements of the robot.
Move Joint: control each joint of the UR16e.

B. AR control

AR devices rely on sensors, such as cameras, to map and track the physical environment, enabling the system to comprehend real-world spaces and objects. In the AR application, a 3D model of the UR16e robot and its workspace serves as the virtual representation, facilitating interactions.

Real-time inverse kinematics algorithms calculate the joint angles necessary for the robot to reach specified poses, aligning with AR tracking data from pinch gestures. As users manipulate the virtual robot through the AR interface, the algorithm continuously updates joint angles, synchronising with the user's actions.

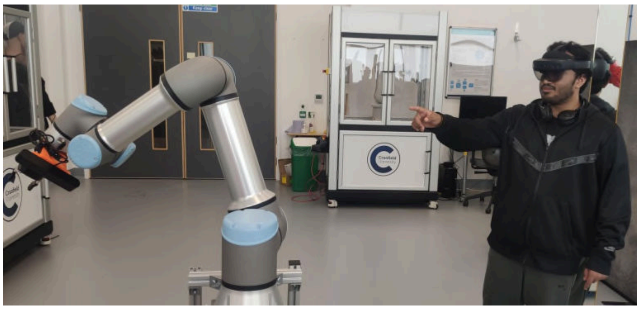


Fig. 3. Augmented reality application in experiment

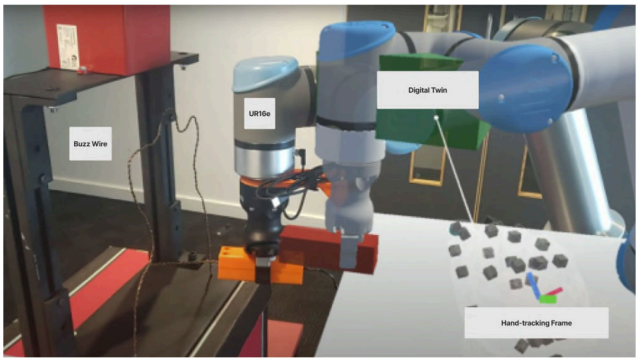


Fig. 4. Virtual and real robots

V. RESULTS

When comparing the AR application to the industry-standard Teach Pendant, a significant observation is the variation in ease of learning and operation. According to the data (Fig.7), 38% of participants with prior experience using manipulators found the Teach Pendant easy to use. However, a substantial portion (36.1%) reported difficulties in learning and operating it. Moreover, more than 60% of participants considered the Teach Pendant to be inaccurate for their tasks (Fig.8). In contrast, the AR application demonstrated clear advantages, with 95.2% of participants rating it as more intuitive and user-friendly than the Teach Pendant, as shown in Fig. 5. The AR application's strengths lie in the visualisation and intuitiveness. The integration of a digital twin into the AR interface enhanced participants' ability to visualise, control, and operate tasks effectively. This not only improved accuracy but also made the application more user-friendly and less stressful, as reflected in the data.

The AR application, despite being an experimental system under development, showed promising results. Participants were able to learn and operate the robot using the AR app in a relatively short time, with the majority completing tasks at a faster pace. This is supported by Fig.6, where 85.7% of participants found the AR app to be more efficient and effective for task performance compared to the Teach Pendant, which was preferred by only 14.3%.

Accuracy in manual control is a key factor where the AR application demonstrated a notable advantage. While the Teach Pendant struggled with accuracy where over 60% of participants found it lacking, the AR app's use of real-time inverse kinematics algorithms allowed for continuous adjustments based on hand gesture tracking data. This resulted in a higher accuracy level as shown in Fig.9. A significant contributing factor to the results is user fatigue. The Teach Pendant, weighing 1.6 kg, led to fatigue among users, particularly those who took longer to complete tasks. The AR application, being a contactless solution, eliminated weight-related fatigue, contributing to a more comfortable user experience.

Overall, the experiment indicates that the majority of participants found the AR application to be as intuitive and user-friendly as the Teach Pendant, enabling them to perform tasks efficiently. These findings, illustrated in Fig. 5-9, suggest that the AR app, even as an incomplete product, could serve as a viable alternative to the Teach Pendant for controlling collaborative robot systems.

A. Limitations and challenges

Apart from the input latency between the gesture and the robot's response, as well as the limited field of view of the Hololens 2 that restricts users' peripheral view, a limitation of using AR devices to control robots is the difficulty in environments with poor lightings or excessive obstacles around the area making it difficult to use gestures to control the robotic arm. The main challenge of these AR devices would be implementing safety measures, error correction protocols and fail-safes for new users in applications like surgery which could lead to serious consequences.

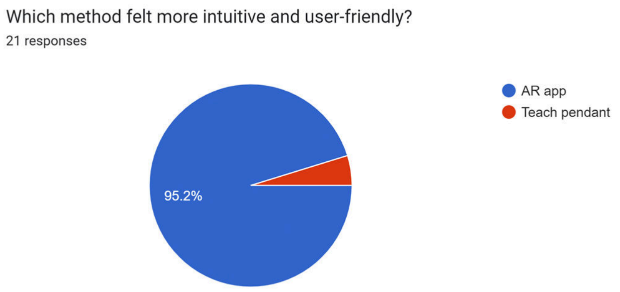


Fig. 5. Intuitiveness and user-friendliness of interfaces

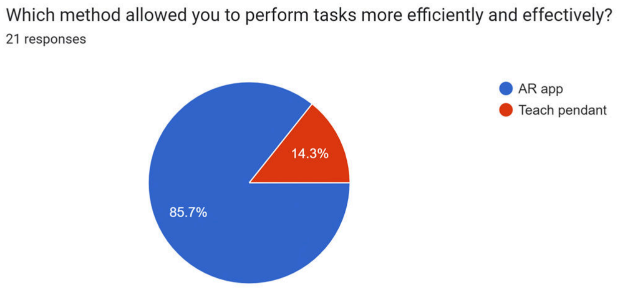


Fig. 6. Efficiency and effectiveness of interfaces

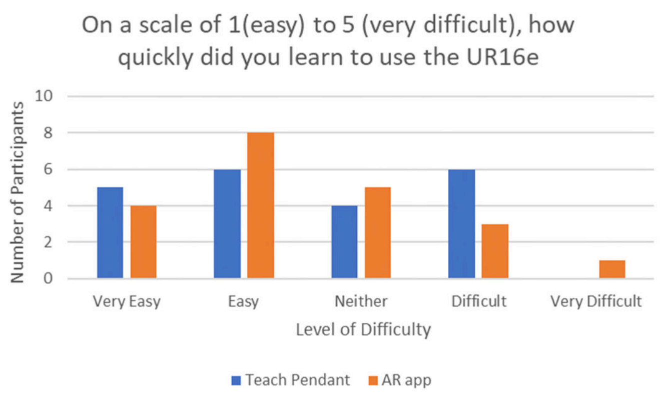


Fig. 7. Ease of operation

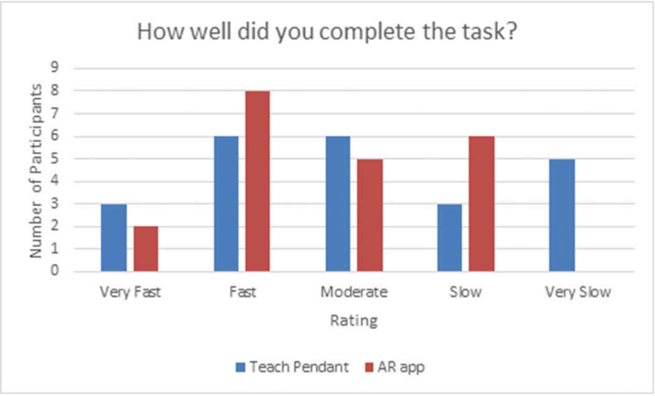


Fig. 8. Task completion rate

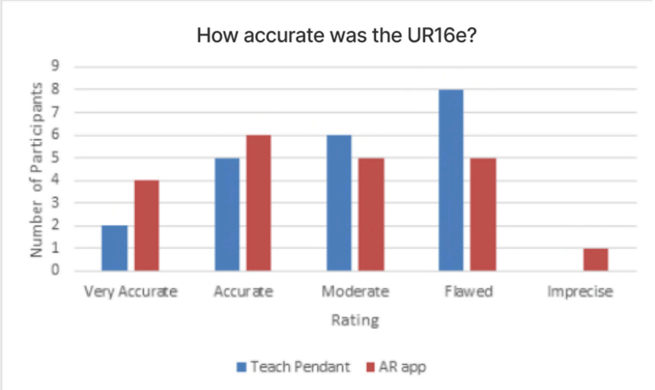


Fig. 9. Manuel control accuracy

VI. CONCLUSION AND FUTURE WORK

In conclusion, this study examined gesture-based robot control techniques and proposed a system architecture for an AR and gesture interface for collaborative robot control, which has been shown to enhance user experience. Although the Teach Pendant is the established industry standard, the AR application, despite being a work-in-progress, achieved similar results in terms of task completion, ease of learning and accuracy. The AR app's potential to match or even surpass the traditional Teach Pendant in these areas suggests it could be a viable alternative for robot control.

Future work should focus on further enhancing the AR application's user interface, exploring multi-robot control, automating tasks, and integrating IoT and cloud services. These improvements would expand the application's capabilities, making it a more intuitive and adaptable tool for robot control. In addition to improving the application's functionality, the AR experience must be developed to mitigate users from experiencing cybersickness by improving the lighting and minimising the visual clutter in the AR environment.

The study also explored various control methods for the UR16e robot, including Cartesian movement, slider control, and inverse kinematic algorithms. The findings highlighted challenges with using moveL commands in Cartesian control, making UR16e manipulation difficult. Conversely, the moveJ algorithm, used in both slider control and inverse kinematics, proved to be more accurate and stable, especially in the context of the AR app. While slider control posed some challenges, the inverse kinematic algorithm consistently optimised UR16e robot performance, solidifying its status as the preferred choice for future application development.

In summary, the AR app demonstrates comparable performance to the traditional Teach Pendant, showing significant promise as an effective tool for UR16e control. As development progresses, it is poised to become a strong competitor in the industry, offering a modern, user-friendly alternative to traditional methods.

REFERENCES

[1] Rudorfer, M., Guhl, J., Hoffmann, P., & Krüger, J. (2018, September). Holo Pick'n'Place. In 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA) (Vol. 1, pp. 1219-1222). IEEE.

[2] Kaczmarek, W., Panasiuk, J., Borys, S., & Banach, P. (2020). Industrial robot control by means of gestures and voice commands in off-line and on-line mode. Sensors, 20(21), 6358.

[3] Li, C., Fahmy, A. and Sienz, J. (2019) 'An augmented reality-based human-robot interaction interface using Kalman filter sensor fusion', Sensors (Switzerland), 19(20). Available at: https://doi.org/10.3390/s19204586.

[4] Neto, P., Simão, M., Mendes, N., & Safeea, M. (2019). Gesture-based human-robot interaction for human assistance in manufacturing. The International Journal of Advanced Manufacturing Technology, 101, 119-135.

[5] Park, K. B., Choi, S. H., Lee, J. Y., Ghasemi, Y., Mohammed, M., & Jeong, H. (2021). Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality. IEEE Access, 9, 55448-55464.

[6] Rosen, E., Whitney, D., Phillips, E., Chien, G., Tompkin, J., Konidaris, G., & Tellex, S. (2019). Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. The International Journal of Robotics Research, 38(12-13), 1513-1526.

[7] Soares, I., Petry, M. and Moreira, A.P. (2021) 'Programming robots by demonstration using augmented reality', Sensors, 21(17). Available at: https://doi.org/10.3390/s21175976.

[8] Zimmerman, K.B. (2019) Conference Proceedings of the Society for Experimental Mechanics Series Series Editor. Available at: http://www.springer.com/series/8922.

[9] De Pace, F., Manuri, F., Sanna, A., & Fornaro, C. (2020). A systematic review of Augmented Reality interfaces for collaborative industrial robots. Computers & Industrial Engineering, 149, 106806.

[10] Salman, A. E., & Roman, M. R. (2023). Augmented reality-assisted gesture-based teleoperated system for robot motion planning. Industrial Robot: the international journal of robotics research and application.

The development of an augmented reality gesture control human-robot interface

Hariharasudhan, Akhil

2024-11-11

Hariharasudhan A, Tang G, Webb P. (2024) The development of an augmented reality gesture control human-robot interface. In: 2024 12th International Conference on Control, Mechatronics and Automation (ICCMA), 11-13 November 2024, London, UK

https://doi.org/10.1109/iccma63715.2024.10843879