

Synchronized Control of Robotic Arm based on Virtual Reality and IMU Sensing

CHIH-JER LIN, TING-YI SIE, YU-SHENG CHANG

Graduate Institute of Automation Technology,
National Taipei University of Technology,
1, Sec. 3, Zhongxiao E. Rd., Taipei 10608,
TAIWAN

Abstract: - This study introduces a robotic control system that combines virtual reality integration and inertial measurement units (IMU) using mixed reality (MR) devices. The system integrates three main modules: (1) virtual reality (VR), which simulates remote reality by the Hololens2, (2) the wearable IMUs device, which captures the operator's hand movements; and (3) the robotic arm UR5, which is controlled by the user and the VR environment. The virtual and physical systems communicate via a Message Queuing Telemetry Transport (MQTT) communication architecture to establish communication between modules. To introduce a closed-loop control system for the robotic arm, model predictive control (MPC) was achieved with precise path planning to provide a flexible, intuitive, and reliable method to operate the remote-controlled manipulator. To validate the system integration and functions, two demonstrations were conducted: (a) the offline mode, where the VR module of the robotic arm was controlled by the IMUs device to check correctness, and (b) the online mode, where the control command was transferred to UR5 to complete a target mission via artificial potential field (APF) adjustment. The primary outcome of this study was the development of virtual and real industrial robotic arms to test the developed model in VR and shop floor labs.

Key-Words: - model predictive control (MPC), MQTT, Hololens2, artificial potential field (APF), virtual reality, path planning, visual localization, inertial measurement units (IMU).

Received: February 9, 2023. Revised: October 13, 2023. Accepted: November 19, 2023. Published: December 13, 2023.

1 Introduction

With the advancement of technology and artificial intelligence, the application of remote manipulators has been gradually growing, not only limited to industrial automation but also extended to the fields of healthcare, education, and environmental monitoring. However, this remote detection type still needs improvement, such as inaccurate operation, slow response, and the inability to sense the environment directly. Researchers have been working on solving problems related to remote-controlled robotic arms by integrating various technologies in recent years. Head-mounted display helmets and wearables are widely used in virtual reality simulations and operational interfaces, [1], [2].

The head-mounted display helmet can provide the operator with an immersive virtual environment, which allows the operator to experience the operation of the robotic arm in an immersive manner. This improves the intuitiveness and realism of the operation. The wearable device can accurately capture the movement information of the operator's arm to provide accurate input for remote control, [3].

In addition, a closed-loop control system for model predictive control is usually introduced to improve the control accuracy and stability of the remote control. The control system predicts the movement trajectory of the robotic arm based on the sensor data and optimizes the control instructions for smoother and more accurate path planning, [4]. This method can reduce the influence of external interference on operation and improve the system's control performance and reaction speed.

Remote-control technology is used in various fields, including manipulator operation and control of robotic systems. However, traditional remote controls suffer from communication delays, which affect real-time performance and accuracy. Researchers have recently addressed these challenges by combining various techniques, such as head-mounted displays and wearable devices, for realistic simulations and model predictive control, [5], [6], [7], [8]. Model prediction control utilizes a mathematical model of a manipulator that predicts the motion trajectory to generate a smooth and accurate trajectory, as shown in Figure 1. The model predictive control is a path-planning method based

on a system model that uses a mathematical model to predict future motion trajectories. By predicting and optimizing the model, smooth and precise trajectories can be generated to achieve a higher motion control performance. The model predictive control method exhibited good adaptability and response speed. It can cope with uncertainty and external interference such that the robotic arm can be controlled accurately in a complex environment. This study aims to investigate the feasibility of a remote-controlled manipulator for performing tasks using head-mounted display helmets, wearable devices, model prediction control, and intelligent path planning. To realize accurate and efficient remote-controlled manipulator operations, it promotes applications and development to enhance efficiency and safety and increase convenience and possibilities, [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

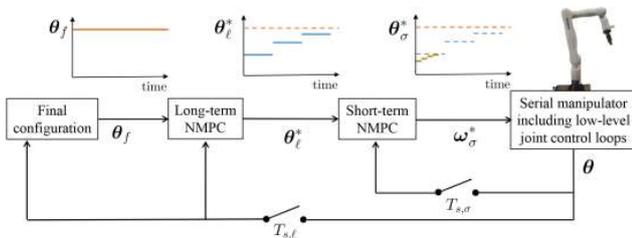


Fig. 1: Model Predictive Control illustration, [2]

2 System Description

The proposed system architecture comprises four parts: a self-made wearable device to capture the operator's gestures, a virtual reality integration platform to project the operation preview in real-time, a synchronized manipulator for remote control, and a transmission and computation terminal program. As shown in Figure 2, the first part is the arm pose capture device, which comprises three inertial measurement units (IMU) to measure the arm posture of the operator. Through the application of a Kalman filter, the IMU data can be calibrated and estimated to obtain accurate attitude information, which is analyzed by the program and transmitted to the robotic arm UR5 for manipulator synchronization control. The second part is a virtual reality integration platform, which receives the data from the first part and displays them on HoloLens 2. A self-developed C# program receives the operator's posture to display on the HoloLens 2. The third part is the terminal program, which integrates the commands to the actual hardware and transfers the program to the manipulator to perform the closed-loop control algorithm. The fourth part describes the real-time

hardware used in this experiment, including a UR5 collaborative manipulator, Robotiq 2F-85 two-finger servo gripper, and Intel RealSense D435 depth camera. Table 1 presents the software and corresponding applications used in this study.

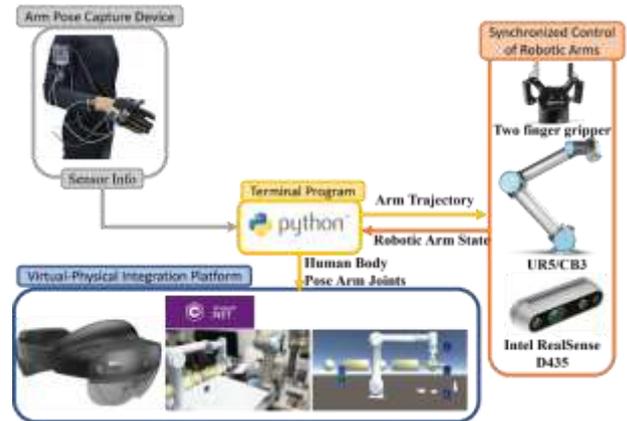


Fig. 2: System Architecture Diagram

Table 1. Software and applications used in the study

Development environment	<i>Integrated content and subparts</i>
Python	Analyze manipulator posture Object recognition(D435) Path planning(MPC&APF) Inverse Kinematics(IK) calculation, Data transmission
Arduino	Capturing human arm information
Visual Studio C# (Unity)	Manipulator model simulation Human posture visualization Hololens real-time demonstration Provide operation UI
SolidWorks	Draw and export 3D object models.

2.1 Wearable Manipulator Instructor and Gesture Gripper Control

Self-developed wearable devices include battery-charging, boost-voltage, microcontroller, and multiplexed I2C circuits. These circuits were integrated to capture the operator's gestures. Three BNO055 integrated 9-axis inertial measurement units (IMUs) are used for the operator's hand motion tracking. Each IMU chip contains an accelerometer and gyroscope sensors and is referred to as a 6-axis IMU or 6 D.O.F. IMU. The IMU includes a magnetometer besides the accelerometer and gyroscope sensors. The IMU data can be calibrated and estimated by applying a Kalman filter to obtain accurate attitude information from the operator. In terms of communication integration, all the three IMUs were equipped with I2C communication functions. The problem of conflicting I2C addresses is solved using multiplexers that connect multiple I2C devices with the same address simultaneously. In addition, a pair

of data gloves were fabricated in our laboratory, as shown in Figure 3(a). It uses a microcontroller and bending sensors to recognize the operator's hand-paw operation. The hand position and gesture information can be transmitted to the processing unit of the wearable device, and the information can be uploaded to the host computer through radio frequency (RF) wireless transmission to realize manipulator arm control based on human gestures. The overall hardware device usage is shown in Figure 3(b).

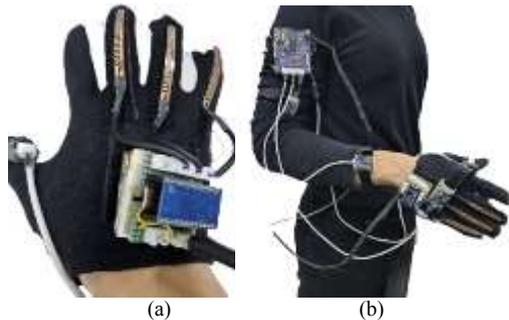


Fig. 3: (a) Self-made data glove (b) Overall hardware setup

2.2 Virtual-Reality Integration Platform

The real-time visualization environment of HoloLen2 is compiled and programmed using Unity software. In the experiment, we created a human-machine interface (HMI), as shown in Figure 4, and the most important aspect was the model. To match the hardware model used in the experiment, the UR5 model and gripper were imported into Unity. To comply with the virtual environment, it is necessary to disassemble UR5's movable joints one by one and reset the home coordinates of each component. The gripper model is also disassembled based on all movable joints. This disassembly method allows the gripper components to be operated independently, which better mimics the real-world behavior of the gripper. After the assembly is completed, the rotational direction must be confirmed. Unity has the possibility of a Gimbal Lock. This is mainly because during the rotation process, if a specific axis is rotated by 90°, the subsequent rotations will lose the rotation dimension, resulting in an unexpected rotation result. Finally, all the models were assembled in the Unity development environment according to the relative position of the assembly, as shown in Figure 5(a) and Figure 5(b).



Fig. 4: Unity user interface

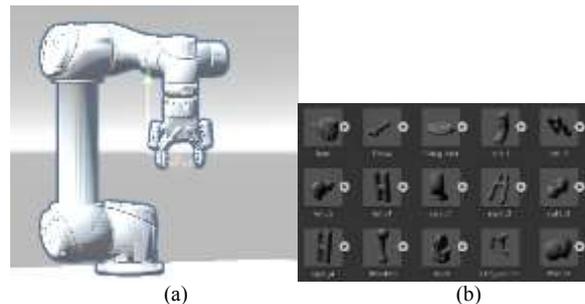


Fig. 5: (a) UR5 and Gripper Assembly (b) Total Components of the Model

2.3 MQTT-based Communication Architecture

The communication of the proposed system is based on a message-queuing telemetry transport (MQTT) communication architecture to integrate each device. The communication architecture is shown in Figure 6. The local computer is responsible for processing the hand posture and position returned by the wearable IMUs device. The IMU data are calculated and transferred to the joint angle using inverse kinematics. The values were corrected using model predictive control (MPC) and artificial potential field (APF) algorithms. The data are then uploaded to the MQTT, and HoloLens2 subscribes to the UR5 joint angles and gripper information through the MQTT. It then immediately displays the operation to local personnel. The remote computer communicates between the workstation equipment and the MQTT. The remote computer subscribes to the UR5 joint information and gripper status and sends the real UR5 information from the remote computer to the MQTT. The actual UR5 joint angle was provided to the local MPC algorithm for the calculation. The object coordinates of D435 were provided to the APF algorithm to perform object attraction for path planning.

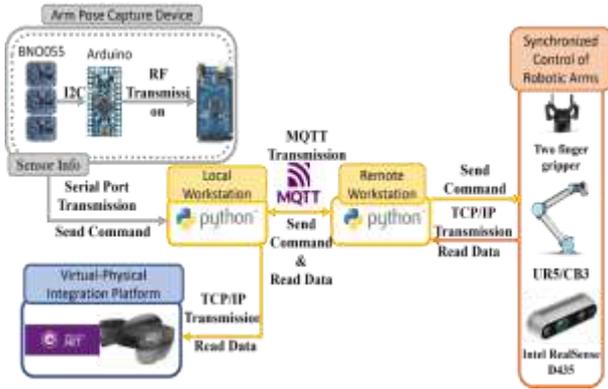


Fig. 6: Communication Architecture Diagram

3 Methodology

3.1 IMU-Based Human Hand Position and Orientation Measurement

The derivation of the angle of the quaternion estimation using the quaternion output from sensor fusion is as follows.

(1) Quaternion Pose Estimation

A quaternion consists of one real and three imaginary parts, represented by Eq. (1).

$$q \equiv \omega + xi + yj + zk = \begin{pmatrix} x \\ y \\ z \\ \omega \end{pmatrix} \quad (1)$$

Based on the Euler angle definition for the three rotation angles, the quaternion can be expressed using Eq. (2).

$$q = \begin{pmatrix} \cos\left(\frac{\psi}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \end{pmatrix} \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ 0 \\ \sin\left(\frac{\theta}{2}\right) \\ 0 \end{pmatrix} \begin{pmatrix} \cos\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \end{pmatrix} \quad (2)$$

Mathematically, owing to the symmetry between the rotation matrices and quaternions, the Euler angles can be expressed in terms of the quaternions, as shown in Eq. (3).

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \tan^{-1}\left(\frac{2yz+2wx}{w^2-x^2-y^2+z^2}\right) \\ \sin^{-1}\left(\frac{-2xz+2wy}{w^2+x^2-y^2-z^2}\right) \\ \tan^{-1}\left(\frac{2xy+2wz}{w^2+x^2-y^2-z^2}\right) \end{pmatrix} \quad (3)$$

(2) Human Arm Posture

The IMU modules were attached to the upper arm, forearm, and back of the hand, and the IMU data were used to compute arm-related information. The X, Y, and Z values for the hand were derived as follows.

$$R_{zyx} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = R_z(\psi)R_y(\theta)R_x(\phi) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\theta\cos\psi \\ \cos\theta\sin\psi \\ -\sin\theta \end{pmatrix} \quad (4)$$

In Eq. (4), the notation R_{zyx} represents rotation in the order of the z-, y-, and x-axes. The symbols ψ , θ , and ϕ represent the rotation angles around the z-axis, y-axis and x-axis, respectively. In Eq. (4), the IMU X-axis is computed by multiplying the rotation matrix by $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ to represent the IMU X-axis vector.

Finally, based on Figure 7 and Eqs. (4), the X, Y, and Z values of the arm can be obtained by multiplying with the arm length, as shown in Eq. (5).

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \overline{V_{ES}} \cdot R_{zyx} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \overline{V_{WE}} \cdot R_{zyx} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (5)$$

The X-, Y-, and Z-coordinates of the hand can be converted from the two IMUs of the upper arm and forearm. The 3-axis rotation angle of the IMU at the back of the hand is directly expressed as the roll, pitch, and yaw values of the wrist in space, which can be used to express the six degrees of freedom of the hand in space.

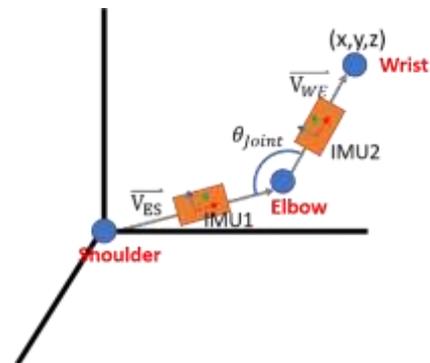


Fig. 7: IMU Vector Relationship Diagram

3.2 Path Planning and Visual Localization

(1) Visual Localization

In visual localization, we must perform several coordinate system conversions to convert 2D planar

coordinates into 3D world coordinates. For object recognition and localization, we used a RealSense depth camera to capture the object and obtain its depth information. The object's position in the image can be converted through three-dimensional coordinate conversion into the position in the coordinate system of the robotic arm base. The object position is recognized by removing noise and calculating the center point, and coordinate conversion is performed to achieve object recognition and positioning. This process consists of several steps: the first step is from camera modelling to depth computation; the second step is object recognition and coordinate conversion; and the final step is to achieve stable and accurate object positioning within 3 mm, as shown in Figure 8.

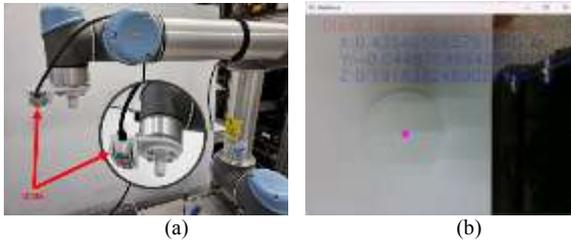


Fig. 8: (a)D435 Mounting Schematic (b) Object Coordinate Calculation

(2) Artificial Potential Field

In the remote control of a robotic arm, an artificial potential field (APF) can be used to control the movement of the robotic arm and quickly guide it to the target position. An attractive potential energy field (APF) is a common type of potential energy field. Attractive APFs allow the robotic arm to be subjected to a force toward the target position, which assists the remote operator in quickly acquiring an object. The attractive potential field is given by Eq. (6).

$$U_{att}(q) = \frac{1}{2}\zeta\|O(q) - O(q_f)\|^2 \quad (6)$$

where $O(q)$ denotes the current position of the robotic arm, $O(q_f)$ represents the target position, q represents the joint angles of the robotic arm, and ζ is a parameter that adjusts the influence of the attractive potential field.

The gradient of the attractive potential field can be expressed by Eq. (7).

$$F_{att}(q) = -\nabla U_{att}(q) = -\zeta(O(q) - O(q_f)) \quad (7)$$

where $F_{att}(q)$ represents the force exerted on the manipulator and can be used to control the direction and velocity of the motion of the robotic arm. When

the manipulator enters the potential field, $F_{att}(q)$ points toward the target position. Adjusting the parameter ζ that influences the attractive potential field can control the magnitude and range of attraction. Thus, it influences the trajectory of the manipulator movement, as shown in Figure 9.

(3) Model Predictive Control

The model predictive control (MPC) is a method used to implement system trajectory planning and control. MPC uses mathematical models to predict the system behavior, which makes optimal control decisions based on future predictions. The MPC derivation process can be described as follows. There are four elements in the MPC model: prediction, rolling optimization, error compensation, and derivation of the MPC formula, where P denotes the prediction step, and M denotes the control step.

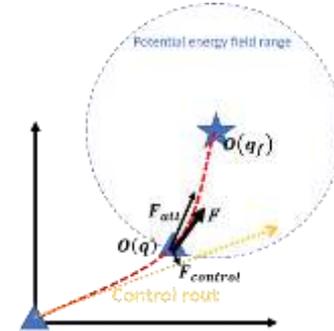


Fig. 9: Schematic diagram of APF

3.3 Model Establishment

When modeling predictive control, the system under control is subjected to step inputs, and step-response outputs are observed. Assuming that the corresponding outputs of the model at moments T , $2T$, ..., PT are a_1, a_2, \dots, a_p , according to the superposition principle of linear systems, the augmented representation can be expressed as shown in Eq. (8).

$$y(k) = \sum_{i=1}^{P-1} a_i \cdot \Delta u(k-i) + a_i \cdot \Delta u(k-P) \quad (8)$$

(1) Prediction Model

From Eq. (8) for $y(k)$, an expression can be obtained to predict the output for n steps, as expressed in Eq. (9).

$$y_0(k+j) = \sum_{i=j+1}^{P-1} a_i \Delta u(k+j-i) + a_p \Delta u(k+j-P), j = 1, 2, \dots, n \quad (9)$$

If y_0 is the predicted output from past inputs, it can be obtained using Eq. (10).

$$\hat{y}(k+j) = \sum_{i=1}^j a_i \Delta u(k+j-i) + y_0(k+j) \quad (10)$$

In matrix form, this can be represented by Eq. (11).

$$\hat{Y} = A\Delta U + Y_0 \quad (11)$$

(2) Rolling Optimization

First, the desired trajectory must be determined to achieve a smooth and stable output. Therefore, a first-order filter is often used to generate the desired trajectory, denoted by w . The core of the rolling optimization is to find the optimal solution of the objective function to determine the control increment Δu . The objective function is essentially a quadratic polynomial in terms of Δu , which can be given by Eq. (12).

$$J = \sum_{i=1}^P [y(k+i) - w(k+i)]^2 \cdot q + \sum_{j=1}^M [\Delta u(k+j-1)]^2 \cdot r \quad (12)$$

The objective function is primarily separated into two parts: the first is to minimize the difference between the actual and desired outputs, and the second is to approach the target with as little energy as possible. In Eq. (12), q and r are the weighting coefficients of the two objectives. The relative settings of q and r are inversely proportional. For example, if the ratio of q to r is 10, this indicates a greater emphasis on minimizing the control effort (energy consumption). In contrast, if the ratio of q to r is 0.1, this signifies a greater emphasis on quickly approaching the target. Subsequently, transforming it into a matrix form leads to the expression of ΔU , as shown in Eq. (13).

$$\frac{\delta J}{\delta \Delta U} = 0 \rightarrow \Delta U = (A^T Q A + R)^{-1} \cdot A^T \cdot Q (W - Y_0) \quad (13)$$

(3) Feedback Correction and Error Compensation

Suppose that, at the current time k , we predict the output of P and at time $k+1$, the current value of y is $y(k+1)$. A prediction error exists at this point, as expressed in Eq. (14).

$$e(k+1) = y(k+1) - \hat{Y}_0(k+1) \quad (14)$$

Using this error to correct future predictions, Eq. (15), where H denotes a weighted matrix.

$$\tilde{Y}(k+1) = \hat{Y}_0(k+1) + H \cdot e(k+1) \quad (15)$$

The prediction process can be simplified as follows: At time k , the prediction of the forward P steps is as described in Eq. (16).

$$\tilde{Y}(k+1), \tilde{Y}(k+2), \dots, \tilde{Y}(k+P) \quad (16)$$

Therefore, by moving to time $k+1$ and continuing the prediction forward, Eq. (17) can be obtained as.

$$\tilde{Y}(k+2), \tilde{Y}(k+3), \dots, \tilde{Y}(k+P+1) \quad (17)$$

If we use shift matrix S , the initial prediction $\hat{Y}_0(k+1)$ can be expressed as shown in Eq. (18).

$$\hat{Y}_0 = S \cdot \tilde{Y} \quad (18)$$

In summary, MPC is characterized by allowing some error in the predicted model, which allows optimized control to be achieved within a limited period of time. At each step of the control execution, the next step is recalculated and reoptimized (rolling optimization). It has strong tuning capability and can be used for calculations when the control system is not precisely modeled. Figure 10 shows the test results of the experimental control for robot UR5, where the movements are prerecorded trajectories.

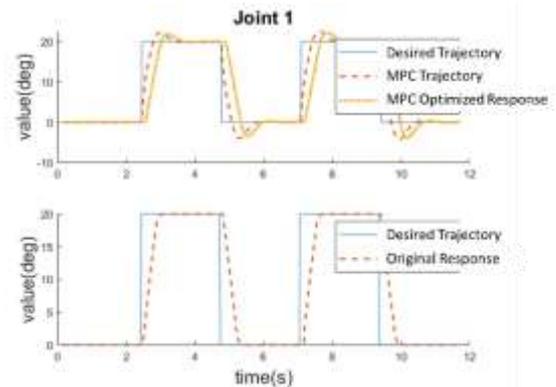


Fig. 10: Joint Responses with and without Controller

4 Result and Discussion

In this study, two experimental modes were implemented: offline and online. In offline mode, trajectory planning and smooth control are achieved through curve fitting to enable the execution of predetermined arm movements. In online mode, a feedback-controlled system was used to compensate for disturbances and ensure accurate movements.

Figure 11 illustrates (a) a flowchart of the object update, (b) offline teaching, and (c) the online mode.

(1) Offline mode control

In assembly lines, it is necessary to perform the same action repeatedly. We introduced the "offline mode" based on this architecture to cope with this situation. In this mode, operators interact with virtual objects through their actions. Hololens2 provides a local, fully simulated, and visually interactive environment in these operations. After simulating the system, the script for the entire action process was provided to the remote workstation for execution. The overall flow is shown in Figure 12. The real-time visualization environment of HoloLen2 is compiled and programmed using Unity software. In the experiment, users controlled the virtual model in a unity development environment according to the relative position of the assembly, as shown in Figure 4.

(2) Online mode control

In addition to the applications mentioned above, in which repeated playback is required for offline mode control, real-time remote control is required when dealing with more complex problems. Compared to storing an entire batch of scripts and transmitting them all at once, the requirements for the online mode are more stringent for controllers.

The online mode is equivalent to a closed-loop controller across the network in terms of architecture, as shown in Figure 13, where the local operation of the user is sent directly to the remote manipulator via the MQTT. The visualization part of the remote environment provides positional updates of the interactive model in the Hololens2 through a D435 camera. Local and remote network interference is inevitable, and packet loss can cause problems such as unsynchronized operation and discontinuous tracking angles of the joints.

(3) Anti-Interference Test

Because network interference is unavoidable, we approximated the synchronization effect of remote UR5 when the connection was poor by adding two degrees of simulated noise to the joint synchronization control. The original UR5 tracking results are presented in Figure 14. It can be seen that the original UR5 controller tries to track these bouncing disturbances, which cause abnormal vibrations during the UR5 synchronization operation and increase the difficulty of remote control. Figure 15 shows the tracking response of the MPC control strategy used, and it can be seen that after the MPC adaptation, the response of the

joints has been improved to a great extent, and the interference resistance is more favorable compared to Figure 14.

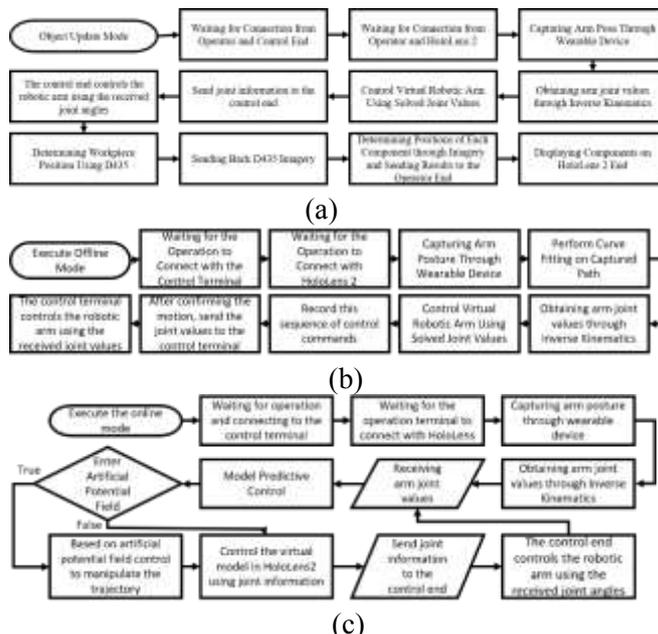


Fig. 11: (a)Object update, (b) Offline Mode Control, (c) Online Mode Control

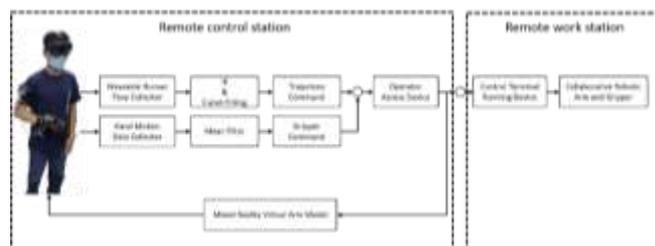


Fig. 12: Offline-mode diagram

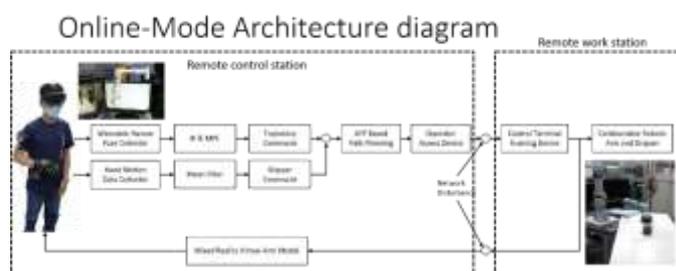


Fig. 13: Online-mode diagram

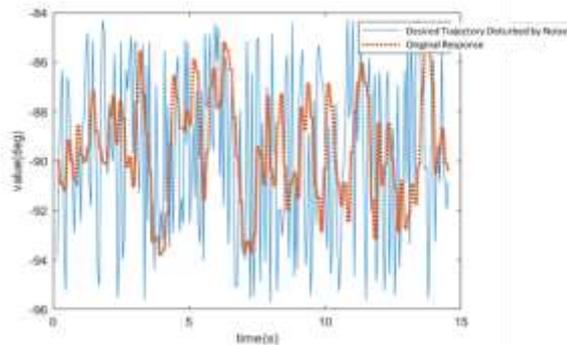


Fig. 14: Joint control response with interference (original)

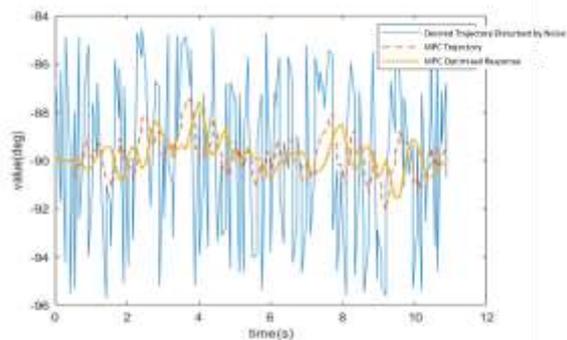


Fig. 15: Joint control response with interference (MPC optimized)

5 Conclusions

A VR (Virtual Reality) integrated controller was implemented to test the system integration and functions. Two control architectures were investigated in this study: offline and online. The offline mode uses a curve-fitting method for trajectory planning and implements the VR model using the HoloLen2 model to check the control curve to achieve a predefined arm motion. The online mode considers the need for remote control and introduces an APF controller with feedback for trajectory planning and control. This allows the real-time controller to compensate for the robotic trajectory, and even if disturbances are encountered during remote control, it ensures the accuracy and stability of the robotic motion. In summary, the proposed controllers provide a reliable control strategy for remote-controlled systems and offer an effective solution for achieving accurate motion.

Acknowledgement:

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financial support of this research under Contract No. NSTC 112-2218-E-027-007.

References:

- [1] J. Guhl, S. Tung, and J. Kruger, "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)*, Limassol, Cyprus, 2017, pp. 1-4, doi: 10.1109/ETFA.2017.8247749.
- [2] Oleinikov, S. KUSDAVLETOV, A. SHINTEMIROV and M. RUBAGOTTI, "Safety-Aware Nonlinear Model Predictive Control for Physical Human-Robot Interaction," in *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5665-5672, July 2021, doi: 10.1109/LRA.2021.3083581.
- [3] M. Rubagotti, T. Taunyazov, B. Omarali and A. Shintemirov, "Semi-Autonomous Robot Teleoperation With Obstacle Avoidance via Model Predictive Control," in *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2746-2753, July 2019, doi: 10.1109/LRA.2019.2917707.
- [4] N. Zhang, Y. Zhang, C. Ma, and B. Wang, "Path planning of six-DOF serial robots based on improved artificial potential field method," *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macao, Macao, 2017, pp. 617-621, doi: 10.1109/ROBIO.2017.8324485.
- [5] K. -H. Lee, V. Pruks and J. -H. Ryu, "Development of shared autonomy and virtual guidance generation system for human interactive teleoperation," *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Jeju, Korea (South), 2017, pp. 457-461, doi: 10.1109/URAI.2017.7992775.
- [6] A. Vick, J. Guhl, and J. Krüger, "Model predictive control as a service — Concept and architecture for use in cloud-based robot control," *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 2016, pp. 607-612, doi: 10.1109/MMAR.2016.7575205.
- [7] S. Kleff, A. Meduri, R. Budhiraja, N. Mansard and L. Righetti, "High-Frequency Nonlinear Model Predictive Control of a Manipulator," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 7330-7336, doi: 10.1109/ICRA48506.2021.9560990.
- [8] S. V. Ghouskhanehee and A. Alfi, "Model Predictive Control of transparent bilateral

- teleoperation systems under uncertain communication time-delay," *2013 9th Asian Control Conference (ASCC)*, Istanbul, Turkey, 2013, pp. 1-6, doi: 10.1109/ASCC.2013.6606130.
- [9] Nicola Piccinelli, Riccardo Muradore,"A bilateral teleoperation with interaction force constraint in unknown environment using nonlinear model predictive control," *European Journal of Control*, Vol. 62, 2021, pp.185-191, ISSN: 0947-3580.
- [10] Bukeikhan Omarali, Tasbolat Taunyazov, Askhat Bukeyev, and Almas Shintemirov. 2017. Real-Time Predictive Control of an UR5 Robotic Arm Through Human Upper Limb Motion Tracking. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI '17)*. Association for Computing Machinery, New York, NY, USA, 237–238.
- [11] H. Lim, Y. Kang, C. Kim, J. Kim and B. -J. You, "Nonlinear Model Predictive Controller Design with Obstacle Avoidance for a Mobile Robot," *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Beijing, China, 2008, pp. 494-499, doi: 10.1109/MESA.2008.4735699.
- [12] M. Noroozi and L. Kies, "Performance Analysis of Universal Robot Control System Using Networked Predictive Control," *2022 7th International Conference on Robotics and Automation Engineering (ICRAE)*, Singapore, 2022, pp. 227-232, doi: 10.1109/ICRAE56463.2022.10056165.
- [13] Barnali Das, Gordon Dobie," Delay compensated state estimation for Telepresence robot navigation,"*Robotics and Autonomous Systems*, Vol. 146, 2021, 103890, ISSN 0921-8890.
- [14] F. Cursi, V. Modugno and P. Kormushev, "Model Predictive Control for a Tendon-Driven Surgical Robot with Safety Constraints in Kinematics and Dynamics," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 7653-7660, doi: 10.1109/IROS45743.2020.9341334.
- [15] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Koley and E. Todorov, "An integrated system for real-time model predictive control of humanoid robots," *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Atlanta, GA, USA, 2013, pp. 292-299, doi: 10.1109/HUMANOIDS.2013.7029990.
- [16] J. Lee, M. Seo, A. Bylard, R. Sun and L. Sentis, "Real-Time Model Predictive Control for Industrial Manipulators with Singularity-Tolerant Hierarchical Task Control," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, 2023, pp. 12282-12288, doi: 10.1109/ICRA48891.2023.10161138.
- [17] Cho J, Park JH. Model Predictive Control of Running Biped Robot. *Applied Sciences*. 2022; 12(21):11183. <https://doi.org/10.3390/app122111183>.
- [18] Künhe, F., Gomes, J., & Fetter, W. (2005, September). Mobile robot trajectory tracking using model predictive control. In *II IEEE Latin-American robotics symposium*, Vol. 51.
- [19] Taïx Michel, Flavigné David, Ferré Etienne. Human interaction with motion planning algorithm. *Journal of Intelligent & Robotic Systems*, 2012, 67: 285-306.
- [20] M. S. Ahn, H. Chae, D. Noh, H. Nam, and D. Hong, "Analysis and Noise Modeling of the Intel RealSense D435 for Mobile Robots," *2019 16th International Conference on Ubiquitous Robots (UR)*, Jeju, Korea (South), 2019, pp. 707-711, doi: 10.1109/URAI.2019.8768489.

Contribution of Individual Authors to the Creation of a Scientific Article

- Chih-Jer Lin, Yu-Sheng Chang carried out the simulation and the optimization.
- Yu-Sheng Chang has implemented the Algorithms in Visual Studio C# (Unity).
- Ting-Yi Sie and Yu-Sheng Chang have organized and executed the experiments of Section 4.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financial support of this research under Contract No. NSTC 112-2218-E-027-007.

Conflict of Interest

The authors have no conflicts of interest to declare.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US