



Preserving Human Expertise: Exploring Mixed Reality Trajectory Planning for Robotic Belt Grinding

W.-X. WANG¹, H.-L. HE^{2*}, T.-Y. ZHOU³, L. WAN⁴

¹. College of Mechanical and Vehicle Engineering, Chongqing University, No. 174, Shazhengjie, Shapingba, Chongqing, 400044, P.R. China;

². State Key Laboratory of Mechanical Transmissions for Advanced Equipment, Chongqing University, No. 174, Shazhengjie, Shapingba, Chongqing, 400044, P.R. China.

³. School of Engineering, Cardiff University, Cardiff, UK

*Corresponding author. Tel : + 86 15210579790

E-mail address: ¹wx.wang@cqu.edu.cn, ²202307021055@stu.cqu.edu.cn, ³ZhouT13@cardiff.ac.uk

Abstract

This paper is dedicated to learning the experience of manual work and proposes a new method for robot grinding and polishing trajectory planning based on mixed reality and pose measurement. The method will capture the movements of a skilled operator when surface polishing a blade part, and generate the path and direction of robot polishing based on this manual polishing trajectory. This paper takes the Fanuc M-710iC/50 robot as the experimental object, conducts DH modeling and forward and inverse kinematics analysis, establishes a virtual robot model with real scale, adds motion control attributes through Unity 3D, and build it to the Hololens device. According to the technical requirements of Vuforia Dynamic Model Target, the experimental fixture is improved, and the recognition accuracy and stability are greatly improved compared with those before the improvement; The Vuforia Model Target technology is used to realize robot tracking registration and fixture pose measurement; The pose information generated after the recognition is used to record the trajectory of the fixture in the process, and then the robot grinding and polishing program is generated. This trajectory planning method not only enhances traditional methods but also incorporates the flexibility of manual work, offering promising applications in robot trajectory planning.

Keywords: Industrial robot, Mixed Reality, Trajectory planning, Pose measurement

1. Introduction

In the aerospace parts manufacturing industry, there is a growing trend towards the use of automated manufacturing technologies. These technologies have proven to be effective in enhancing production efficiency, reducing costs, and minimizing human error. However, certain surface finishing tasks still require manual grinding performed by skilled operators. It is important to note that manual polishing can pose safety and health risks to the operators, while also making it challenging to ensure consistent quality throughout the process. To address these concerns, the current common practice is to utilize industrial robots to replace manual grinding and polishing of parts. This not only ensures consistent quality and standardizes the process, but also reduces the need for rework, thereby saving time and cost in the manufacturing process.

Currently, research on belt grinding for industrial robots focuses primarily on tool path planning. However, for complex geometries, guiding the robot accurately through the grinding and polishing

operation without collisions remains a tough and time-consuming task. There are now two basic types of robot education methods: online teaching and offline programming [1]. Online teaching in robotics includes an experienced operator manually adjusting the robot end-effector to the desired position and attitude using a trainer. The robot controller then creates motion commands based on these locations and attitudes. This teaching method has several advantages, including intuitiveness, low programming requirements, and low setup costs. However, its accuracy heavily depends on the operator's expertise and talents, and additional tests are required to verify the reliability and safety of the finished pathways. On the other hand, the offline programming method uses computer graphics to simulate a 3D virtual robot and workpiece. Trajectories are planned on the workpiece's surface, which are subsequently discretized into point positions. A program is built to lead the robot end-effector to successively reach all trajectory locations. With the use of computer algorithms, this method enables the instruction of complex workpieces [2].

However, teaching 3D modeling and planning trajectories for specific robots and workpieces in offline settings is time-consuming and requires significant programming skills that operators who typically undertake training may not possess. Moreover, offline programming relies on precise robot and work unit models, necessitating the use of additional sensors for calibration and workpiece positioning. An alternative approach to robot trajectory planning is programming by demonstration (PbD). In PbD, an expert user demonstrates a trajectory to the robot, extracts key information from the demonstrated trajectory, and then executes a trajectory optimization procedure that the robot follows. Currently, PbD technology is predominantly used for non-contact tasks. A method for teaching robot force control and tool path through PbD has been developed for tasks. However, these methods are not suitable for contact finishing processing tasks like robotic grinding and polishing due to accuracy limitations.

This paper presents a robot motion trajectory planning method that combines PbD and mixed reality technology. The method utilizes pose measurement and mixed reality [7][8] to plan the robot toolpath, using the toolpath demonstrated by expert users as input information. The goal is to improve productivity and safety in robotic machining, address traditional demonstration methods' limitations, and reduce cognitive pressure for operators in the industrial robot field and trajectory planning. The process involves the expert user holding a workpiece with a fixture already clamped, demonstrating the grinding path. The Hololens, a mixed reality device worn by the expert user, measures the real-time coordinates of the clamped fixture on the workpiece. These coordinates are then transformed into the robot work program using a robot kinematics algorithm, enabling the robot to replicate the grinding path demonstrated by the expert user. The feasibility and effectiveness of the proposed method are validated through experiments conducted with the FANUC M710ic/50 robot.

2. Robot Kinematics

To accurately describe the motion of a virtual robot in a mixed reality space, modeling of the robot linkages and joints is required. The DH representation [9] is usually used to model the individual joints of the robot, and this modeling method is independent of the order and complexity of the robot structure and applies to robots of any configuration.

2.1 Robot Kinematics Modeling

Figure 1 shows the coordinate system of each linkage established by the FANUC M710ic/50 robot, and Table.1 shows the D-H parameter table of this robot, according to which the robot kinematic model is established by the D-H model. The linkage transformation matrix can be expressed as Eq (1).

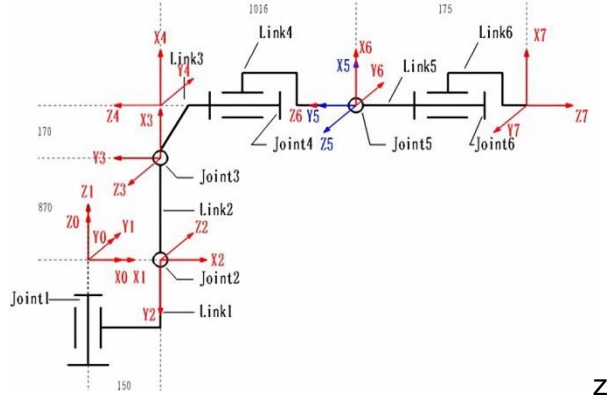


Figure 1 – Link coordinate system.

$$\begin{aligned}
 {}^i T_{i+1} &= Rot(z, \theta_{i+1}) \times Trans(0, 0, d_{i+1}) \times Trans(a_{i+1}, 0, 0) \times Rot(x, \alpha_{i+1}) \\
 &= \begin{bmatrix} C\theta_{i+1} & -S\theta_{i+1}C\alpha_{i+1} & S\theta_{i+1}S\alpha_{i+1} & a_{i+1}C\theta_{i+1} \\ S\theta_{i+1} & C\theta_{i+1}C\alpha_{i+1} & -C\theta_{i+1}S\alpha_{i+1} & a_{i+1}S\theta_{i+1} \\ 0 & S\alpha_{i+1} & C\alpha_{i+1} & d_{i+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)
 \end{aligned}$$

Where $C\theta_i = \cos\theta_i$, $S\theta_i = \sin\theta_i$, and so on. $\theta_i, d_i, a_i, \alpha_i$ are the joint angle, the link offset, the link length, and the link twist angle, respectively.

Table 1 - Nominal values of kinematic parameters

i	$\theta_i/^\circ$	d_i/mm	a_i/mm	$\alpha_i/^\circ$
1	θ_1	0	150	-90
2	θ_2-90	0	870	180
3	θ_3	0	170	-90
4	θ_4	-1016	0	90
5	θ_5	0	0	-90
6	θ_6	-175	0	180

Rod angle θ_i : the angle of rotation of the X_i and X_{i-1} axes about the Z_i axis.

Connecting rod distance d_i : the distance between two perpendiculars a_{i-1} and a_i along Z_i axis.

Connecting rod length a_i : The length of the common perpendicular of the joint axis at both ends of connecting rod i .

Rod torsion angle α_i : the rotation angle perpendicular to the axis of the joints of both ends of rod i in the plane of a_i .

The robot kinematic equation is Eq (2), which is the product of each linkage transformation matrix.

$${}^0_1 T = \begin{pmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & a_1 \cos\theta_1 \\ \sin\theta_1 & 0 & \cos\theta_1 & a_1 \sin\theta_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^3_4 T = \begin{pmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned}
{}^1_2T &= \begin{pmatrix} \cos \theta_2 & \sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & -\cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^4_5T &= \begin{pmatrix} \cos \theta_5 & 0 & -\sin \theta_5 & 0 \\ \sin \theta_5 & 0 & \cos \theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
{}^2_3T &= \begin{pmatrix} \cos \theta_3 & 0 & -\sin \theta_3 & a_3 \cos \theta_3 \\ \sin \theta_3 & 0 & \cos \theta_3 & a_3 \sin \theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^5_6T &= \begin{pmatrix} \cos \theta_6 & \sin \theta_6 & 0 & 0 \\ \sin \theta_6 & -\cos \theta_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned} \tag{2}$$

2.2 Robot Positive Kinematics

The transition from joint space to Cartesian space is a fundamental problem in positive kinematics for robots. It involves establishing a model that connects the parameters of the robot's joint space with the parameters of the Cartesian space. This model serves as a theoretical foundation for studying motion control of mixed reality robots, allowing for the manipulation of robot motion by using gestures in the mixed reality space. In this study, we focus on analyzing the forward kinematics of the FANUC M710ic/50 robot. Specifically, we aim to determine the position of the end-tool coordinate system relative to the base coordinate system of the robot, given the known joint variables and linkage parameters. We solve for this position by utilizing the robot's DH model and linkage parameters, obtaining the transformation matrix of the neighboring linkage coordinate system using Eq (3).

$$\begin{aligned}
{}^{i-1}_iT &= Rot(Z, \theta_i) \cdot Trans(0, 0, d_i) \cdot Trans(a_i, 0, 0) \cdot Rot(X, \alpha_i) \\
&= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3}$$

Where ${}^{i-1}_iT (i = 1, 2, 3, 4, 5, 6)$ describes the transformation matrix of the latter linkage with respect to the previous linkage position, respectively. The transformation matrix T of the robot end-tool coordinate system, i.e., the linkage coordinate system, with respect to the robot base coordinate system is the product of the transformation matrices of the neighboring connecting rod coordinate systems as in Eq (4):

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

2.3 Robot Inverse Kinematics

Inverse kinematics of industrial robots refers to the reverse solution of the angle values of each joint of the robot according to the pose of the robot end-effector. The method proposed in this paper requires viewing the motion of the virtual robot in real-time in the mixed reality space and needs to convert the tool path information demonstrated by the expert user into the robot joint angle information for storage. Therefore, an inverse kinematic analysis of the experimental robot is required.

2.4 Inverse Kinematics Analytic Solution

It is known that FANUC M710ic/50 is a six-axis robot that satisfies the Pieper principle, and it has 8

groups of closed inverse solutions. The joint variables $\theta_1, \theta_2, \dots, \theta_6$, can be solved by multiplying the matrices on both sides of Eq (4) by the inverse matrix of each link transformation matrix, and their values are shown in Eq (5).

$$\begin{aligned}
\theta_1 &= A \tan 2(d_6 a_y + p_y, d_6 a_x + p_x) \\
\theta_2 &= A \tan 2(w_1, w_2) - A \tan 2\left(\frac{w_3 + a_2^2 - d_4^2 - a_3^2}{2a_2 \sqrt{w_3}}, \pm \sqrt{w_3 - \left(\frac{w_3 + a_2^2 - d_4^2 - a_3^2}{2a_2 \sqrt{w_3}}\right)^2}\right) \\
\theta_3 &= A \tan 2(w_4, w_5 - a_2) - A \tan 2\left(d_4, \pm \sqrt{w_4^2 + (w_5 - a_2)^2 - d_4^2}\right) \\
\theta_4 &= A \tan 2(a_x S_1 - a_y C_1, -a_x C_1 C_{23} - a_y S_1 C_{23} + a_z S_{23}) \\
\theta_5 &= A \tan 2(a_x (C_1 C_{23} C_4 - S_1 S_4) + a_y (S_1 C_{23} C_4 + C_1 S_4) - a_z S_{23} C_4) \\
\theta_6 &= A \tan 2(-n_x k_1 - n_y k_2 + n_z S_{23} S_4, o_x k_1 + o_y k_2 - o_z S_{23} S_4)
\end{aligned} \tag{5}$$

In the above Eq (6):

$$\begin{aligned}
w_1 &= (d_6 a_x + p_x) C_1 + (d_6 a_y + p_y) S_1 - a_1 \\
w_2 &= d_6 a_z + p_z \\
w_3 &= w_1^2 + w_2^2 \\
w_4 &= w_1 S_2 + w_2 C_2 \\
w_5 &= w_1 C_2 - w_2 S_2 \\
k_1 &= C_1 C_{23} S_4 + S_1 C_4 \\
k_2 &= S_1 C_{23} S_4 - C_1 C_4 \\
S_i &= \sin \theta_i, C_i = \cos \theta_i, S_{ij} = \sin(\theta_i - \theta_j), C_{ij} = \cos(\theta_i - \theta_j)
\end{aligned} \tag{6}$$

Where $n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z$ are the robot end-tool coordinate system attitude information, and p_x, p_y, p_z are the robot end-tool coordinate system pose information. From Eq (6), if the position matrix (n, o, a, p) of the robot end-tool coordinate system with respect to the robot base coordinate system is known, the joint variables $\theta_1, \theta_2, \dots, \theta_6$ can be solved, which is the solution of the robot inverse kinematics.

2.5 Selection Algorithm for the Analytical Solutions

For a general six-axis industrial robot, the analytical solution's non-uniqueness leads to the possibility that the end may correspond to multiple sets of solutions in one pose. To satisfy the continuity of robot motion, this paper proposes a selection algorithm for the inverse kinematics analytical solutions based on the minimum stroke criterion. The selection process is shown in Figure 2.

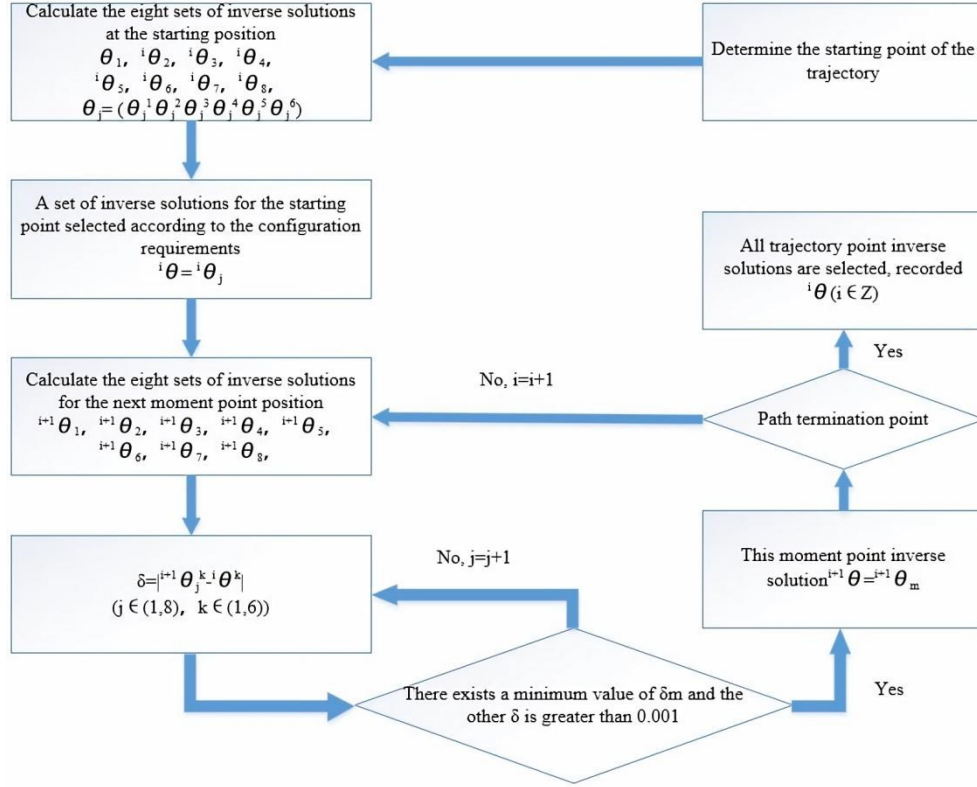


Figure 2 – Process of selecting an inverse kinematics solution

To verify the accuracy of the inverse kinematics algorithm, we selected five points in the manual machining area. The robot was then operated to sequentially move the end-tool coordinate system to these selected points. The position data of the end-tool coordinate system was recorded from the robot controller (see Table 2), while the joint variable data was also obtained from the robot controller (see Table 3). Additionally, we solved the joint variable data using algebraic analysis and recorded the results in Table 4. By comparing Table 3 and Table 4, we observed that the solutions obtained from the algebraic analysis method matched the solutions in the robot controller, with three decimal place accuracy. This indicates that the inverse kinematics solution proposed in this paper is consistent with the solution in the robot controller. Hence, we can conclude that the inverse kinematics solution and the optimal solution selection method presented in this paper are correct.

Table 2 The end pose read from the teach pendant

Number	X/mm	Y/mm	Z/mm	w/°	p/°	r/°
1	1286.822	92.290	681.122	179.774	-24.915	42.522
2	1396.741	192.753	774.803	35.648	-19.897	152.164
3	1543.229	-416.577	1269.307	23.074	-54.196	111.252
4	1076.385	-56.987	409.462	-169.863	-8.214	-92.661
5	920.477	905.013	1282.032	-35.140	7.685	-93.941

Table 3 The joint angle read from the teach pendant

Number	J1/°	J2/°	J3/°	J4/°	J5/°	J6/°
1	1.998	3.495	-11.087	161.892	60.583	-133.130
2	5.473	10.254	-21.547	20.157	76.584	28.367
3	-12.587	24.532	12.367	-47.569	32.954	74.593
4	-1.695	-6.115	-25.565	-169.874	54.756	84.610
5	48.583	6.535	6.034	-33.919	69.687	-31.899

Table 4 Inverse kinematics calculation based on algebraic analysis method

Number	J1/°	J2/°	J3/°	J4/°	J5/°	J6/°
1	1.998	3.495	-11.087	161.892	60.583	-133.130
2	5.473	10.254	-21.547	20.157	76.584	28.367
3	-12.587	24.532	12.367	-47.569	32.954	74.593
4	-1.695	-6.115	-25.565	-169.874	54.756	84.610
5	48.583	6.535	6.034	-33.919	69.687	-31.899

3. Object Recognition and Tracking Based on Vuforia Engine

Based on physical CAD data, Hololens utilizes Vuforia Model Target technology to effectively identify and track 3D objects in the real world. The positional information is extracted from the provided tracking data, enabling the registration of the robot model and identification of the fixture model for tracking. Subsequently, the corresponding virtual model is deployed for superimposition. To expedite the process of locating the physical object in the visual image and initiating object tracking, it is crucial to select an initial recognition pose that generates a model target reference line view, as depicted in Figure 3. Typically, a pose that reveals more geometric information of the object in front of the camera is chosen to enhance recognition accuracy and efficiency.

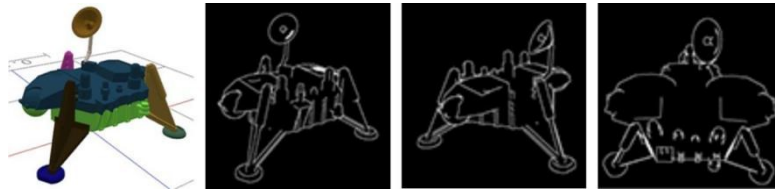


Figure 3 – Model target guide views.

3.1 Robot Tracking and Registration based on Vuforia Static Model Target

The robot grinding and throwing trajectory planning method based on mixed reality and pose measurement proposed in this paper needs to register the virtual robot to the real robot and overlap with the real robot so that it has the same working environment and the same base coordinate system as the real robot to complete the trajectory planning in the real scene instead of the real robot. To facilitate the representation of the relationship between each coordinate system, each coordinate

system is defined as shown in Figure 4. and the robot recognition process is shown in Figure 5.

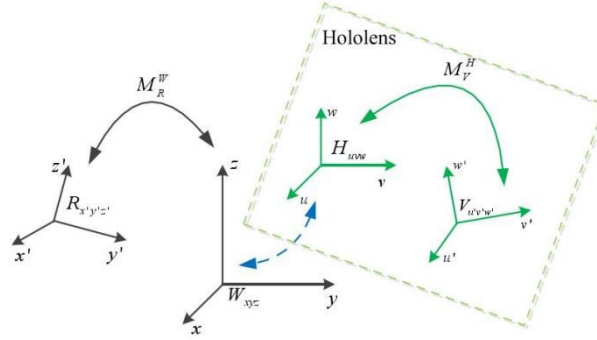


Figure 4 – Relationship of each coordinate system.

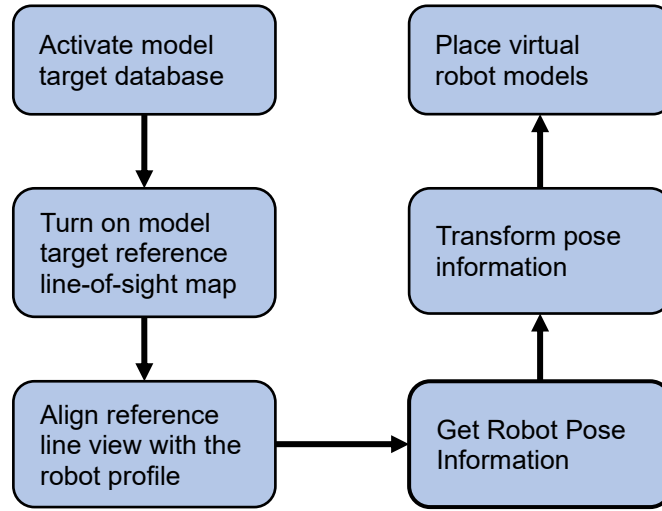


Figure 5 – Robot 3D registration process.

Where, R_{xyz} , W_{xyz} , H_{uvw} , V_{uvw} , M_R^W , M_V^H , are the real robot base coordinate system, world coordinate system, Hololens virtual scene coordinate system, virtual robot base coordinate system, transformation matrix between real robot base coordinate system and world coordinate system, transformation matrix between virtual robot base coordinate system and Hololens virtual scene coordinate system, respectively. Their coordinate relationships are $W = M_R^W \cdot R$, $H = M_V^H \cdot V$. Therefore, making the base coordinate system of the real robot coincide with the base coordinate system of the virtual robot, that is $V = R$, it is necessary to make $W = H$, $M_R^W = M_V^H$.

The Hololens device establishes the space coordinate system of the virtual scene at the initial point when the virtual scene is opened. The world coordinate system can be established arbitrarily. It is advisable to also establish the world coordinate system at the initial point so that $W = H$ is established. The Vuforia Static Model Target technique is used to identify the one-axis profile of the real robot, as shown in Figure 6. This helps obtain the pose matrix of the base coordinate system with respect to the world coordinate system. The virtual robot model is then placed under this pose matrix, ensuring $M_R^W = M_V^H$. This completes the robot tracking registration, as shown in Figure 7.



Figure 6 – One-axis profile.

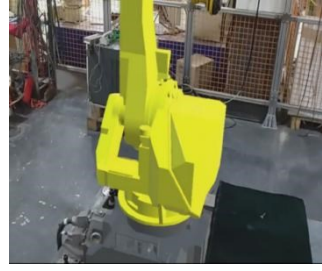
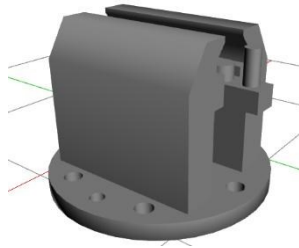


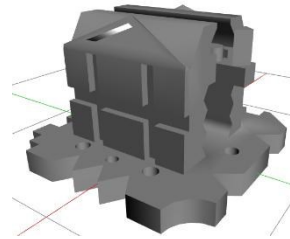
Figure 7 – Tracking and registration.

3.2 Fixture Model Optimization and Pose Recognition

In traditional fixture design, complex structures are typically avoided to minimize machining challenges and ensure precise machining. However, according to Vuforia Dynamic Model Target's technical requirements, rigid objects with vibrant surfaces and intricate geometric details tend to exhibit more stable tracking performance. Therefore, to ensure accurate and stable tracking recognition, it is necessary to enhance and optimize the traditional fixture model. The models before and after improvement are illustrated in Figure 8(a) and Figure 8(b), respectively, while the recognition effect is demonstrated in Figure 9(a) and Figure 9(b).



(a) Before improvement.

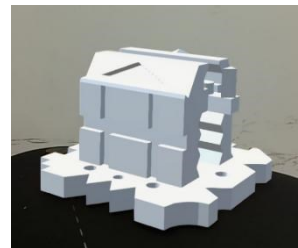


(b) After improvement.

Figure 8 – Fixture structure optimization



(a) Before recognition.



(b) After recognition.

Figure 9 – Fixture tracking results

In the field of robot inverse kinematics, it is essential to determine the robot's endpoint coordinate system and the positional matrix in relation to the robot's base coordinate system. This is done to calculate the necessary rotation angles for each axis. However, Vuforia Engine faces limitations as it cannot simultaneously track and recognize the poses of two objects. As a result, it is unable to directly calculate the positional matrix of the fixture with respect to the robot base coordinate system. To overcome this, a solution is implemented where the pose Matrix_{MB} of the robot base coordinate system with respect to the world coordinate system is recorded upon successful recognition of the robot pose. Subsequently, the recognition process is halted after placing a virtual robot under this pose matrix. Once the fixture is recognized, the pose Matrix_{MF} of the fixture coordinate system with

respect to the world coordinate system is obtained. By applying the principles of coordinate system transformation theory, it becomes possible to derive the pose Matrix_{BF} of the fixture coordinate system with respect to the robot base coordinate system. The relationship between the coordinate systems is illustrated in Figure 10.

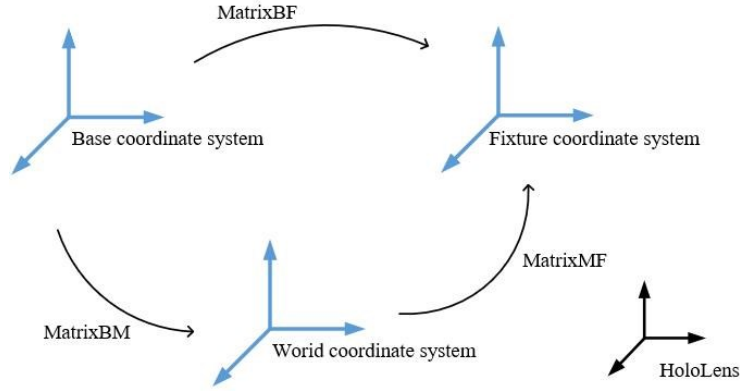


Figure 10 – Relationship between coordinate systems.

3.3 Fixture Pose Information Extraction

The fixture tracking has a chi-square transformation relationship, as illustrated in Figure 11. To describe the process of extracting and transforming fixture pose information, the following coordinate systems are defined: the robot's base coordinate system $\{B\}$, the HoloLens camera coordinate system $\{H\}$, the mixed reality spatial coordinate system $\{M\}$, and the fixture coordinate system $\{F\}$.

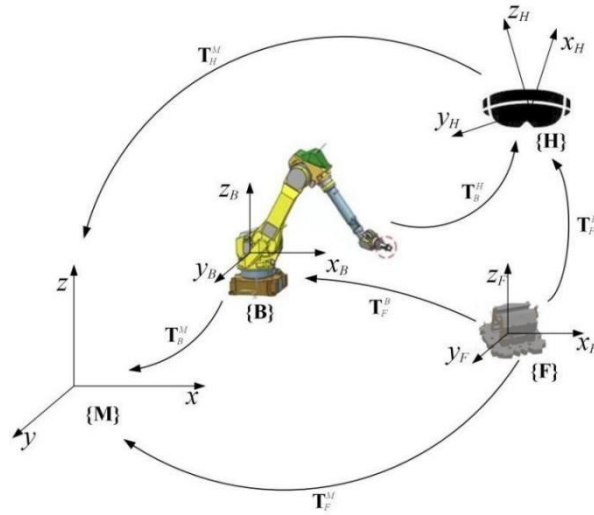


Figure 11 – Extracting robot pose information.

From Figure 11, it can be seen that there is a transformation relationship in the robot recognition process as in Eq (7), where T represents a 4×4 chi-square matrix, and T_F^H and T_B^H are the transformation relationship between the fixture coordinate system and the robot base coordinate system with respect to the HoloLens camera coordinate system. T_H^M , T_B^M , and T_F^M are the pose matrices of HoloLens camera coordinate system, robot base coordinate system, and fixture coordinate system in the mixed-reality space coordinate system, respectively.

In Figure 11, a transformation relationship is illustrated in the robot recognition process, described by

Eq (7). Here, T represents a 4×4 chi-square matrix. T_F^H and T_B^H denote the transformation relationship between the fixture coordinate system and the robot base coordinate system, relative to the HoloLens camera coordinate system. Similarly, T_H^M , T_B^M , and T_F^M represent the pose matrices of the HoloLens camera coordinate system, robot base coordinate system, and fixture coordinate system, respectively, in the mixed-reality space coordinate system.

$$\begin{cases} T_F^M = T_F^H \cdot T_H^M \\ T_B^M = T_B^H \cdot T_H^M \\ T_F^B = T_F^M \cdot (T_B^M)^{-1} \end{cases} \quad (7)$$

Based on eq (7), the T_F^M can be solved using the same method as T_B^M through robot recognition. Additionally, the T_F^B can be calculated using T_B^M and T_F^M . By following the afore mentioned calculation process, real-time position information of the fixture coordinate system in the base coordinate system of the robot can be obtained while the craftsman holding the blade is in motion. This position information can be utilized as the position state of the robot end relative to the base coordinate system of the robot for analyzing the robot kinematics and solving the robot joint space parameters. Consequently, the virtual robot end can achieve the robot base coordinate system parameters, allowing the virtual robot end to reach the pose state of the fixture.

4. Mixed Reality Platform Construction and Software Programming

This system uses the mixed reality display technology of the HoloLens device, as well as Vuforia Model Target physical recognition and tracking technology. It transforms artificial demonstration trajectory data into robot motion control parameters and generates robot processing programs. The system is developed specifically for HoloLens and follows the system architecture shown in Figure 12. To ensure cross-platform compatibility, the system uses the development environment of Open XR and its development standards. Vuforia is used for object recognition and tracking, while the Mixed Reality Toolkit (MRTK) provides cross-platform input systems and building blocks for spatial interaction and UI. The logical functions of the system are programmed in C#, including spatial coordinate data conversion, model pose data changes, model target database activation, robot kinematics algorithm writing, trajectory data storage, and robot motion program generation.

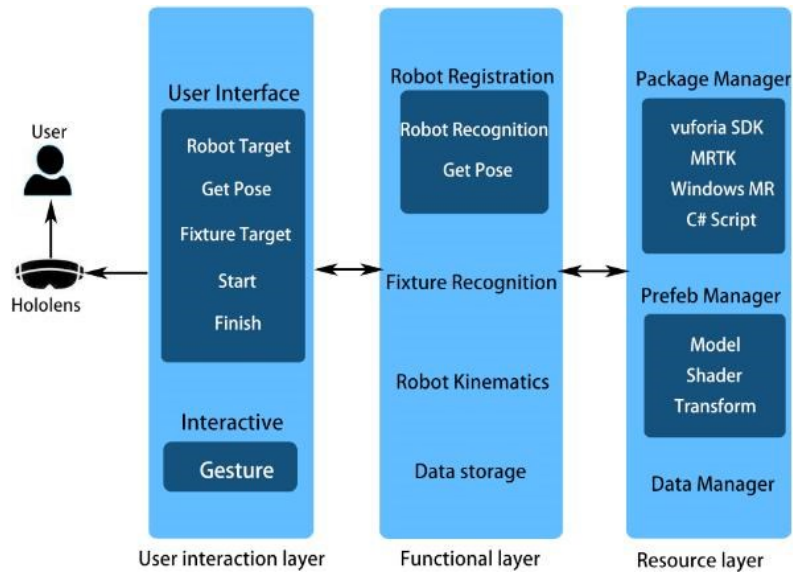


Figure 12 – System architecture.

The system interface is developed in Unity3D, with assistance from HTML, CSS, and the MRTK Mixed Reality Development Toolkit for client interface building blocks. The HoloLens mixed reality display technology is utilized to present the system screen, allowing users to interact with the interface through gestures and access the system's functions. The system is developed and debugged using the Unity 3D engine, as depicted in Figure 13. The process involves building the FANUC M710ic/50 robot axes model and fixture model using 3DsMAX, importing the models into MTG and Unity, training the model in MTG to generate the robot and fixture Model Target database imported into Unity, assembling the axes model in Unity, and assigning appropriate materials and physical attributes. Additionally, the MRTK is imported to enable gesture interaction and assist in designing the interaction interface. The Vuforia Engine is imported to generate the object recognition and tracking module. The logic programming of the entire system is implemented using C#, and the system is installed and deployed to HoloLens using Visual Studio.

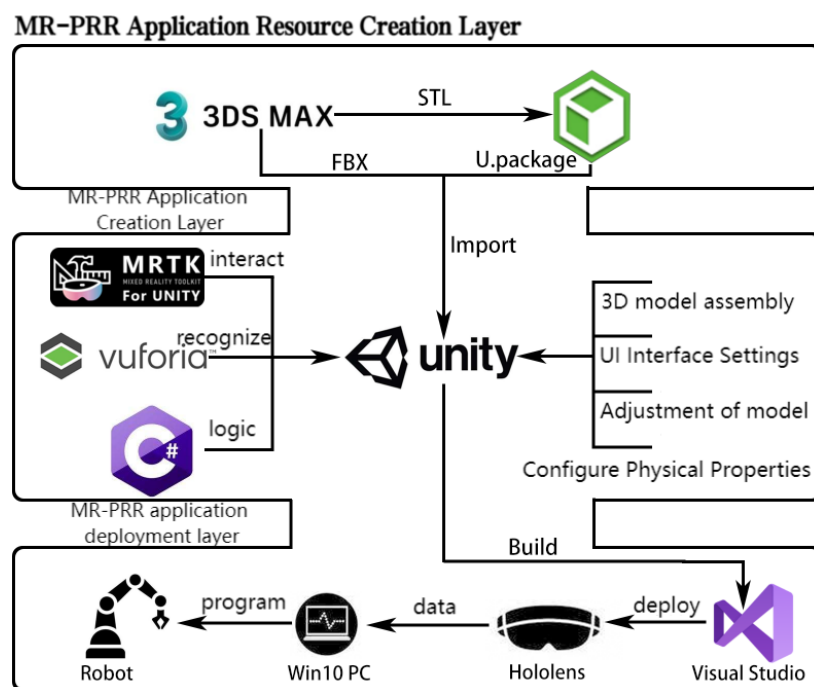


Figure 13 – Software development process.

4.1 Robot Modeling

Robot modeling FANUC M710ic/50 is a series robot. The model is assembled according to the motion relationship, as shown in Figure 14, and the parent-child motion relationship between each axis is established, as shown in Figure 15.

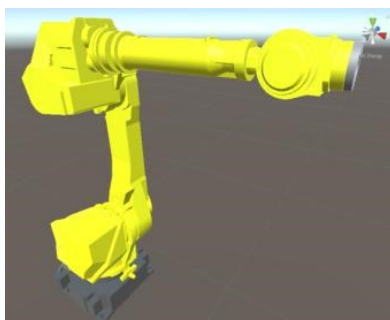


Figure 14 – Robot model.

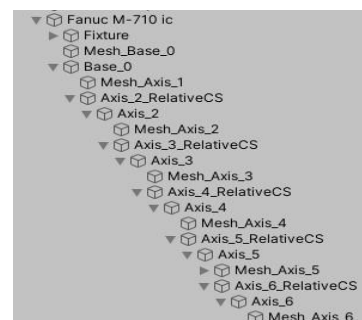


Figure 15 – Movement relationship.

Figure 15 illustrates the Mesh-Axis-I, which represents the mesh model of each axis. Axis-i refers to the coordinate system of each axis, while Axis-i-RelativeCS represents the relative coordinate system of each axis. To enable interaction with the robot model using hands, components such as Box Collider, Object Manipulator, and Near Interaction Grabbable should be added to each axis model in Unity.

4.2 User Interface Design

Figure 16 shows the software user interface, with a total of five mixed reality interactable buttons, each with When the user clicks the button, the Interactable script fires the bound event. The PressableButton script binds the button object (Cube) to set the visual feedback after pressing. The NearInteractionTouchable script enables buttons to interact with hands to enter control commands.



Figure 16 – User interface.

Trigger events for buttons:

Robot Target: To start identifying the robot pose, activate the Robot Model Target dataset.

Get Pose: Get the robot pose matrix, under the pose information, place the robot pref and disable the Robot Model Target dataset.

Fixture Target: Activate the Fixture Model Target dataset and start identifying the fixture pose

Start Demonstrate: Create a robot motion program file, calculate the inverse kinematics solution, write the inverse solution data to the file, and start trajectory planning.

Finish Demonstrate: Update the file data and end the trajectory planning.

4.3 Robot Motion Control Script

In a real working environment, the robot achieves desired positioning by coordinating its end effector through 6-axis motion. To ensure that the virtual robot and the real robot have the same motion state in mixed reality, motion control scripts need to be added in Unity3D, along with the robot kinematics in 2. The Unity script inherits the MonoBehaviour class and utilizes two functions, FixedUpdate() and transform.Rotate(), to move the virtual robot's end effector and the axes accurately. The FixedUpdate() function is invoked at regular intervals after the script is called and is typically used to control object movement. transform.localRotate() enables rotation of each axis around the joint. The specific implementation process is depicted in Figure 17, and the experimental demonstration is illustrated in Figure 18.

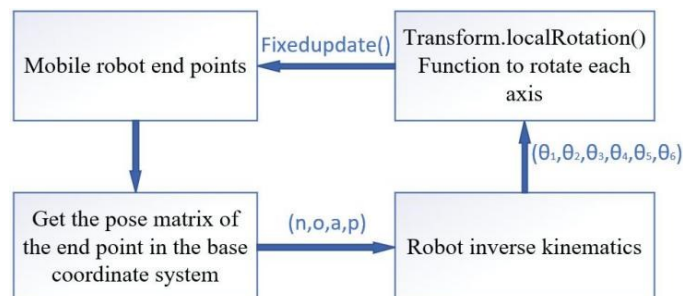


Figure 17 – Motion control.

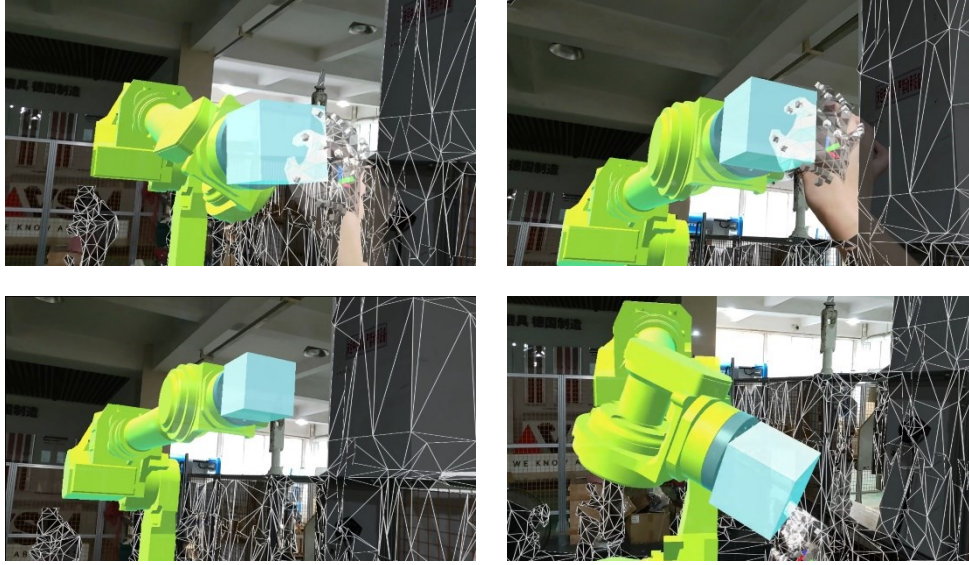


Figure 18 – Demonstration experiment.

5. Conclusion and Prospect

This paper proposed a new robot trajectory planning method, which plans the robot's machining trace by measuring the skilled person's grinding and throwing trace information through Hololens. Improve the productivity of robot processing, circumvent the disadvantages of traditional planning methods, combine the advantages of manual grinding and polishing, and reduce the operator's cognitive pressure on robot processing and trajectory planning. At present, only the robot reproduces the manual grinding and polishing path for simple experimental verification, and the reproduction accuracy needs to be measured later. The manual grinding and polishing path is complex and irregular, and it is impossible to avoid degradation of machining quality due to handshakes, etc. Therefore, path optimization algorithms need to be incorporated at a later stage to simplify the complex manual paths into raster-like paths similar to those generated by robot trajectory planning software, as well as to eliminate the effects caused by hand jitter. In addition, the follow-up work needs to be improved in the following aspects:

1. The mixed reality trajectory planning system described in this paper does not include a collision detection mechanism. To address this issue, the system can be enhanced by introducing the virtual robot to detect collisions with objects in the real environment. This will help avoid any interference caused by collisions when the robot replicates the artificial machining trajectory.
2. The robot is able to reproduce the artificial machining path, but lacks the ability to learn the manual grinding strength. To address this, force sensors are introduced to measure the manual processing strength data.

Copyright Statement

We confirm that we hold copyright on all of the original material included in this paper and we have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. And we confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

Acknowledgements

This study was supported by the China Postdoctoral Science Foundation(NO.2020M673126), the Fundamental Research Funds for the Central Universities(No.2023CDJXY-024). the Natural Science Foundation of Chongqing(CSTB2024NSCQ-MSX0784).

References

- [1] Pan Z, Polden J, Larkin N, et al. Recent Progress on Programming Methods for Industrial Robots. *Robotics and Computer-Integrated Manufacturing*, 28(2),87-94,2012.
- [2] Krüger J, Lien T K, Verl A. *Cooperation of human and machines in assembly lines*. CIRP annals, 58(2),628–646,2012.
- [3] Calinon S, Billard A. Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM. *Proc the 22nd international conference on Machine learning*. Bonn, Germany,105–112,2005.
- [4] Kormushev P, Nenchev D N, Calinon S, Caldwell DG (2011). Upper-body kinesthetic teaching of a free-standing humanoid robot. *Proc IEEE Int. Conf Robotics Automation (ICRA2011)*. Piscataway, NJ, USA , 3970–3975.
- [5] Kormushev P, Calinon S, Caldwell DG (2011). *Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input*. *Adv Robot* 25(5),581–603
- [6] Irish A, Mantegh I, Janabi-Sharifi F (2010). A PbD approach for learning pseudo-periodic robot trajectories over curved surfaces. *In Advanced Intelligent Mechatronics (AIM), IEEE/ASME International Conference*. 1425–1432.
- [7] Ohta Y, Tamura H. *Mixed reality: merging real and virtual*. Springer-VerlagBerlin, Heidelberg,1999.
- [8] Marin G, Dominio F, Zanuttigh P. Hand gesture recognition with leap motion and kinect devices. *In: 2014 IEEE International conference on image processing (ICIP)[C]*, 1565–1569,2014.
- [9] Denavit J, Hartenberg R S. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22(6),215-221,1995.