Adaptive Trajectory Planning Method for Robotic Finishing of Concrete Shield Segments

Zian Xue[1,2], Yue Ma[1,2], Bin Li[1,2] and Qi Liu[1,2,*]

[1]*Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, School of Mechanical Engineering, Tianjin University of Technology, Tianjin 300384, China*

[2]*National Demonstration Center for Experimental Mechanical and Electrical Engineering Education, Tianjin University of Technology, Tianjin 300384, China*

Abstract: To address the challenges of low efficiency and inconsistent quality in the finishing of concrete shield segments, this paper proposes an adaptive trajectory planning method for a robotic system utilizing point cloud data. The system integrates an automated guided vehicle (AGV), a six-degree-of-freedom serial manipulator, and a 3D vision system to create an intelligent finishing robot. A "rectangular" offline programming trajectory is employed, coupled with multi-coordinate system transformations for precise path mapping. A 3D camera captures the segment's surface point cloud, which is subsequently registered, fused, and analyzed by a neural network to identify surface irregularities and compute a look-ahead tilt angle for adaptive trajectory compensation. Experimental results demonstrate that our method significantly enhances finishing uniformity and surface quality, offering a viable technical solution for the automated finishing of complex curved components.

Keywords: Serial robots, Trajectory planning, Offline programming, Adaptive control, Point cloud processing, Concrete pipe segments.

1. INTRODUCTION

In the prefabrication of concrete shield segments for tunnel engineering, the plastering operation is a critical step that directly impacts the surface quality and structural durability. Traditional manual finishing methods are not only inefficient and labor-intensive but also struggle to ensure consistent quality on complex curved surfaces, forming a significant bottleneck in the intelligent modernization of segment production.

In recent years, robotic technology has been progressively introduced into civil engineering to automate labor-intensive and repetitive manual tasks. In particular, serial-joint robots have garnered significant attention for finishing operations such as plastering and grinding, owing to their large working spaces and high motion flexibility [1-3]. Recent studies continue to explore trajectory generation for serial robots in complex operations [4] and their applications in sophisticated manufacturing processes [5]. However, most current applications remain dependent on pre-programmed fixed trajectories or simple teach-and-repeat methodologies, lacking the capability for real-time perception of actual workpiece geometry and consequent adaptive adjustment. When confronted with concrete components exhibiting manufacturing tolerances, installation pose deviations, or inherent deformation, such "blind-operation" robots fail to maintain consistent tool-to-surface contact, ultimately compromising finishing quality. Consequently, integrating high-precision vision systems for accurate trajectory planning has become crucial for high-quality automated finishing [6-7]. Notably, Vision-based calibration methods [8] and stereo vision localization techniques [9] have shown promising results in improving robotic accuracy

Recent advancements in robotics and automation continue to drive progress across intelligent manufacturing. Studies on automated guided vehicle (AGV) trajectory planning [10] and multi-equipment scheduling [11] offer valuable frameworks for deploying mobile robotic systems in industrial settings. Parallel developments in specialized offline programming [12] and point cloud processing—with applications extending from surface defect detection [13] to 3D reconstruction [14]—have enhanced robotic perceptual capabilities. Furthermore, adaptive control strategies for robotic trajectories are being actively refined [15]. Within the specific domain of segment finishing automation, contributions include the design and control algorithms for a plastering robot by Duan [16] and investigations into their site application by Li [17]. Complementary research in enabling technologies includes high-precision hand-eye calibration via 3D point cloud registration [18] and adaptive trajectory generation for complex surfaces [19]. Despite these valuable contributions, a significant research gap persists in the development of a fully integrated system that seamlessly combines reliable multi-station positioning, efficient offline programming for large curved surfaces, and real-time adaptive trajectory compensation driven by 3D perception, specifically for

*Address correspondence to this author at the Tianjin University of Technology, Tianjin 300384, China;
E-mail: rufei67@163.com

Received on 08-10-2025 Accepted on 14-11-2025 Published on 18-11-2025

E-ISSN: 2409-9694/25

concrete segment finishing. Most current solutions lack the model-based adjustment adaptability necessary to accommodate the geometric uncertainties inherent in freshly poured concrete surfaces within industrial production environments.

To address the identified challenges, this paper presents an intelligent robotic system tailored for finishing applications, specifically designed for automated surface smoothing of concrete segments. The integrated system comprises an automated guided vehicle (AGV) and a six-degree-of-freedom manipulator, enabling autonomous operations across the extensive curved surfaces of segments. The AGV utilizes QR code-based visual navigation to achieve precise positioning at multiple workstations, establishing a reliable global reference frame for all subsequent precision operations. Following successful positioning, a 3D vision sensor mounted on the robot's end-effector captures multi-view point cloud data of the segment surface. This data undergoes registration, stitching, and three-dimensional reconstruction to generate a faithful digital model of the actual surface topography. Based on this reconstructed 3D model, the system performs adaptive trajectory planning, generating optimized path points with integrated real-time adjustments. Finally, through precise conversions of each trajectory point across coordinate systems, the system drives the robotic arm to execute the complete automated finishing operation on the curved concrete segments.

This paper investigates an intelligent plastering robot for concrete shield segments. The structure of the paper is as follows: First, it outlines the overall architecture of the robotic system and its visual perception solution. It then details the methodology for offline programming of the plastering trajectories. Subsequent sections delve into point cloud processing, including stitching and surface reconstruction, and the adaptive optimization of the trajectories based on the reconstructed model. Finally, field experiments validating the system's comprehensive performance are presented. Collectively, this research provides an effective integrated solution for achieving automated, high-precision plastering on complex curved components.

2. INTRODUCTION TO THE PLASTERING ROBOT SYSTEM

Figure **1** depicts the operational scenario for the screeding robot. The system comprises an on-site platform support, the main robot unit, and concrete pipe segments situated within the product mold. The robot performs its screeding tasks at designated workstations on both sides of the platform. The product mold is centrally positioned and symmetrical within the platform frame. Owing to the substantial size of the concrete segments and the constraints of the workspace, the main robot body travels along tracks installed on both sides of the frame. Three distinct workstations are established on each side, each equipped with a QR code to facilitate precise robot positioning.

To facilitate the transformation of trajectory data from the workpiece to the robot coordinate system, three distinct coordinate frames are defined: the mold coordinate system, the QR code coordinate system, and the robot coordinate system.

The mold coordinate system $\{W\}$ is defined with point W as its origin. Its z-axis is oriented vertically

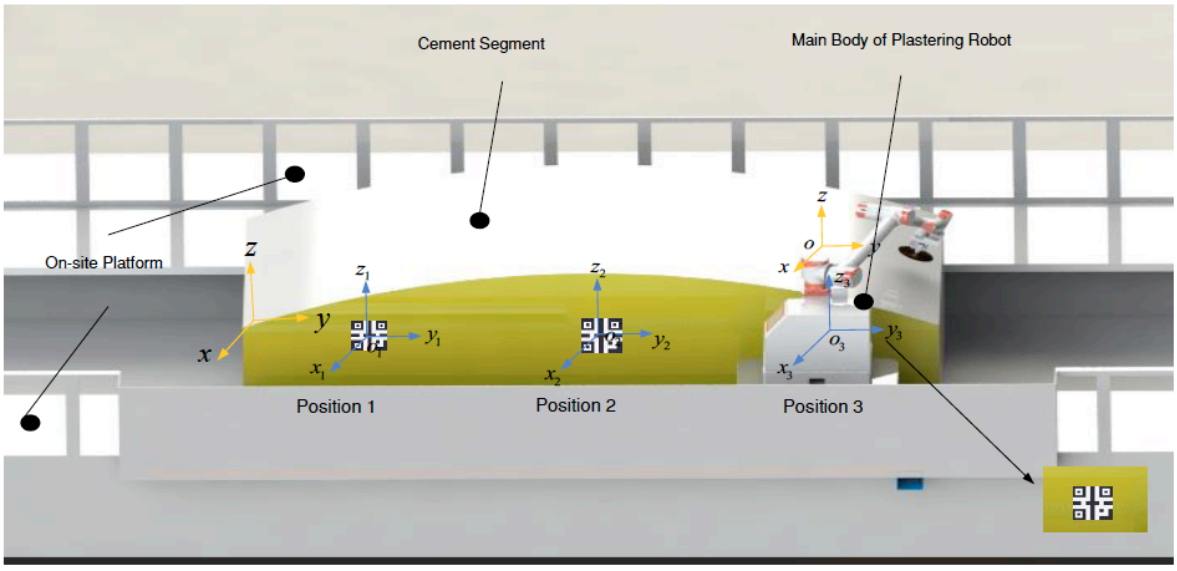


Figure 1: Schematic Diagram of the Finishing Scenario.

upward, the x-axis extends perpendicularly from the front operational platform toward the mold, and the y-axis is aligned parallel to the mold's longitudinal axis. This system provides the geometric reference for defining the position and dimensions of each section of the concrete pipe segment and serves as the basis for offline trajectory programming.

Three QR code coordinate systems $\{Q_1,Q_2,Q_3\}$ are established at workstations Q_1, Q_2 and Q_3 respectively, with their coordinate axes aligned with those of the $\{W\}$ system. Positioned on the product mold, these QR code coordinate systems enable precise calibration of the robot coordinate system and facilitate accurate multi-station robot positioning.

The robot reference coordinate system $\{O_1,O_2,O_3\}$ is established with point O on the robot base as its origin. Its z-is vertically upward, the x-axis extends perpendicularly from the platform support toward the mold, and the y-axis runs parallel to the mold's longitudinal direction. This system serves as the fundamental spatial reference for the robot, ensuring accurate execution of the finishing operations.

Figure **2** shows a schematic of the robotic finishing system. The primary robotic unit consists of a system control cabinet, an automated guided vehicle (AGV), a 6-degree-of-freedom serial manipulator, an end-effector, a 2D camera, and a 3D camera. The AGV is responsible for transporting the entire system between workstations. A 2D camera, mounted at the base of the robotic arm, scans QR codes on the segment facade to identify and locate the target workstation. Once stationed, the system initiates operation based on pre-programmed trajectories. The end-effector is a circular troweling disk that rotates to smooth the concrete surface. A 3D camera, installed on the robot's terminal joint, captures point cloud data of the segment surface. During operation, the robotic arm guides the troweling disk along the planned trajectories over the mold surface to complete the finishing task.

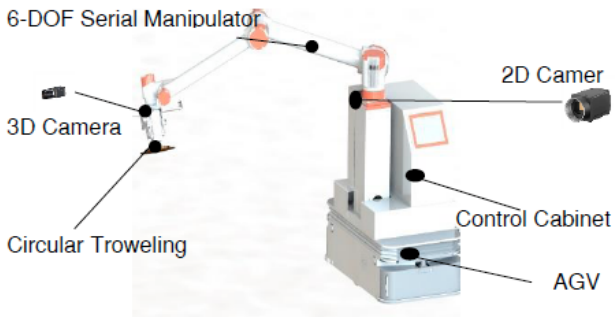


Figure 2: Schematic Diagram of the Robotic Finishing System.

3. ADAPTIVE TRAJECTORY PLANNING STRATEGY BASED ON POINT CLOUD DATA

To achieve precise robotic operations in complex working environments, point cloud data-based adaptive trajectory planning is essential. This methodology primarily encompasses two core processes: (1) offline programming of the initial smoothing trajectory for the segment surface, and (2) utilizing the robotic vision system to acquire point cloud data, model the working target, and subsequently perform adaptive trajectory adjustments, followed by experimental validation.

Considering the geometric characteristics of the concrete segment surface and the operational constraints of the robot's workspace, an optimized rectangular trajectory configuration consisting of three parallel paths was designed. The offline trajectory programming is conducted based on the surface dimensions and other relevant parameters. This process begins with the precise localization of key boundary and intermediate points for the rectangular trajectory. Trajectory equations are then formulated based on these defined points. All generated trajectory point coordinates undergo a systematic transformation: they are first converted into a local coordinate system referenced to individual QR codes, then into the respective QR code coordinate systems. Subsequently, the vision system identifies and measures the calibration QR codes at each workstation. Finally, the trajectory coordinates are transformed from the QR code coordinate systems to the robot base coordinate system for storage and execution.

Figure 3: Schematic Diagram of the Rectangular Trajectory.

During the surface point cloud data acquisition phase, operation commences with the AGV moving and its onboard 2D camera activating simultaneously. After the camera successfully scans and recognizes the QR codes, the AGV proceeds to each workstation in sequence. Point cloud data scanning then initiates. Following a pre-defined data acquisition path, both the robot arm and its end-effector-mounted 3D camera simultaneously begin collecting point cloud data. After data acquisition is complete at all three workstations,

the system processes the aggregated point clouds from the three scans.

The point cloud data processing follows a three-stage procedure.

(1) Coarse registration of point clouds is performed using coordinate transformation formulas.

(2) Fine registration is implemented through the Iterative Closest Point (ICP) algorithm.

(3) The precisely registered point clouds are stitched and fused into a complete model.

This process yields and stores the complete surface and boundary point cloud data of the concrete segment, establishing both the reference framework and operational target for adaptive trajectory planning.

The integrated point cloud is subsequently analyzed to identify regions with significant surface height variations. Based on this analysis, the trajectory is adaptively modified to ensure that the generated paths meet operational accuracy and safety standards when executed by the robotic system. This methodology significantly reduces on-site commissioning time and enables the execution of optimized troweling trajectories.

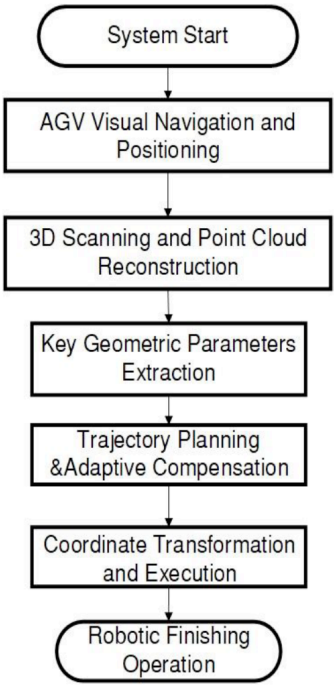


Figure 4: Workflow of the adaptive robotic finishing system

4. ADAPTIVE TRAJECTORY PLANNING METHOD FOR SURFACE FINISHING

4.1. Offline Programming of Trajectories

Offline programming is a fundamental methodology in robotic automation, enabling the preliminary planning and coding of robotic trajectories and motion sequences within a simulated environment, without requiring physical robot operation. For finishing robots, this approach leverages the segment's dimensional data to pre-design troweling paths, thereby eliminating the need for production-line debugging and significantly enhancing programming efficiency and operational flexibility. This foundation facilitates precise trajectory node positioning and ensures the effective execution of the finishing operation. To guarantee complete surface coverage while avoiding redundant or missed areas, a strategy employing three concentric rectangular paths is adopted. This configuration systematically covers the entire working surface from the segment boundaries towards the interior, enabling comprehensive surface treatment through structured path planning.

4.1.1. Trajectory Key Node Localization

The precise coordinates of key nodes within the troweling trajectory constitute the foundation for offline programming. The diameter of the robot's end-effector troweling disk is set to one-quarter of the segment's width. This ratio ensures adequate surface coverage per pass while maintaining mechanical flexibility and operational efficiency. Based on the predefined distribution of the three workstations along the segment, the QR code locations are determined. This enables the spatial calibration of the key nodes—including the start points, turning points, and end points of the rectangular trajectory—within the three-dimensional workspace.

Figure **5** illustrates the geometric configuration of the segment's side plane, highlighting key points along the plastering trajectory. Points A and D define the left and right boundaries of the segment's side plane. Points K and L represent the trisection points along the horizontal line AD. Vertical perpendiculars drawn from points K and L intersect the arc at points B and C, respectively, which serve as division points between the three trajectory segments. Points E, F, G, H, I, and J denote the nodal points of the three rectangular trajectories, functioning as start and end points for the three pairs of arc segments.

To prevent the troweling disk from exceeding the operational workspace and ensure uniform surface finishing, each trajectory segment maintains an offset of one troweling radius from the boundary points. The arc lengths of segments $\overset{\frown}{AE}$、$\overset{\frown}{FB}$、$\overset{\frown}{BG}$、$\overset{\frown}{HC}$、$\overset{\frown}{CI}$ $\overset{\frown}{JD}$ are all approximated to be equal to the troweling radius r. Given the arc radius R, the arc lengths L from each nodal point to point A are sequentially measured. Using the central angle formula $\theta = l/R$, the corresponding deflection angles θ_i. for each point are derived.

The trajectory arc AD, with center coordinates (x_c, y_c, z_c) and central angle θ_i, $\theta \in (0, \theta_D)$, can be expressed by the following parametric equations:

$$\begin{cases} x(\theta) = x_c + x_o \\ y(\theta) = y_c + R\cos(\theta); (\theta \in (0, \theta_D)) \\ z(\theta) = z_c + R\sin(\theta) \end{cases} \tag{1}$$

The variable x_0 represents the axial offset, which in this case corresponds to the diameter $2r$ of the troweling disk. Since the circular arc lies in the YOZ plane, this offset is applied along the X-axis direction. It is used to plan multiple parallel rectangular trajectories, achieving full coverage along the width of the pipe segment.

The coordinates of each point are thus given by：

$$P(\theta) = (x_c + x_o, y_c + R\cos(\theta), z_c + R\sin(\theta)) \tag{2}$$

The coordinates of these key nodes, calculated through the above procedure, provide the fundamental basis for generating the subsequent finishing trajectory.

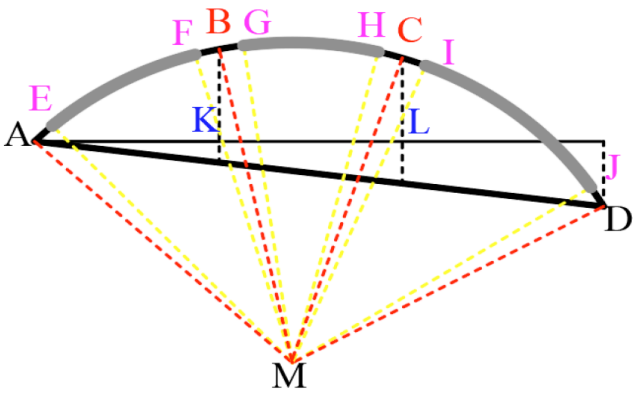


Figure 5: Schematic Diagram of the Segment Side-Plane Geometry.

4.1.2. Offline Generation of Finishing Trajectory

Following the calculation of the key node coordinates, the predefined rectangular path is formalized into a set of trajectory equations. The complete rectangular trajectory is divided into four distinct segments: the first and third segments constitute arc paths parallel to the y-axis, while the second and fourth segments form straight-line paths parallel to the x-axis. Using the first rectangular trajectory as an example, the positional and orientational expressions for each segment are defined as follows:

Position:

Segment 1:

$$\begin{cases} x(i) = x_c + x_o \\ y(i) = y_c + R\cos(\theta_0 + i\Delta\theta); \\ z(i) = z_c + R\sin(\theta_0 + i\Delta\theta) \end{cases} \quad (i \in (0, n)) \tag{3}$$

Segment 2:

$$\begin{cases} x(i) = x_c + x_o + (i-n)(L_x/n) \\ y(i) = y_c + R\cos\theta_F \\ z(i) = z_c + R\sin\theta_F \end{cases}; \quad (i \in (n+1, 2n)) \tag{4}$$

Segment 3:

$$\begin{cases} x(i) = x_c + x_o + L_x \\ y(i) = y_c + R\cos(\theta_F + (2n-i)\Delta\theta); \\ z(i) = z_c + R\sin(\theta_F + (2n-i)\Delta\theta) \end{cases} \quad (i \in (2n+1, 3n)) \tag{5}$$

Segment 4:

$$\begin{cases} x(i) = x_c + x_o + L_x - (i-3)L_x/n \\ y(i) = y_c + R\cos\theta_0 \\ z(i) = z_c + R\sin\theta_0 \end{cases}; \quad (i \in (3n+1, 4n)) \tag{6}$$

Orientation:

Segment 1:

$$\begin{cases} \alpha = 0 \\ \beta = 0 \\ \varphi = \theta_0 + i\Delta\theta + \pi/2 \end{cases} \quad (i \in (0, n)) \tag{7}$$

Segment 2:

$$\begin{cases} \alpha = 0 \\ \beta = 0 \\ \varphi = \theta_F + \pi/2 \end{cases} \quad (i \in (n+1, 2n)) \tag{8}$$

Segment 3:

$$\begin{cases} \alpha = 0 \\ \beta = 0 \\ \varphi = \theta_F + (2n-i)\Delta\theta - \pi/2 \end{cases} \quad (i \in (2n+1, 3n)) \tag{9}$$

Segment 4:

$$\begin{cases}\alpha = 0 \\ \beta = 0 \\ \varphi = \theta_0 - \pi/2 \end{cases} \quad (i \in (3n+1, 4n)) \tag{10}$$

In the above equations, x_o represents the axial offset, which in this case corresponds to the diameter of the troweling disk $2r$, θ_0 is the central angle corresponding to the arc length R of $\overset{\frown}{AE}$ and θ_F is the central angle corresponding to $\overset{\frown}{AF}$. $\Delta\theta$ is defined as $1/n$ of the trajectory arc, where $\Delta\theta = (\theta_F - \theta_0)/n$, $i \in N^*$, Each value of i presents a specific posture along the trajectory. L_x denotes the width of the rectangular trajectory, which is equal to half the width of the concrete pipe surface. Equations (3) to (6) define the positional expressions.

The three orientation angles of the robot are denoted as α, β and φ where α represents the rotation angle of the troweling disk around the y-axis, β denotes the rotation angle around the z-axis, and φ indicates the rotation angle around the x-axis. Equations (7) to (10) define the orientation expressions.

In summary, the initial trajectory for offline programming has been derived through the above formulations.

4.2. Adaptive Trajectory Planning for Finishing Operations

In practical applications, after the product mold is filled with concrete slurry, the surface typically exhibits significant irregularities, including extensive protrusions and depressions. Applying the pre-programmed offline trajectory directly under these conditions would lead to substantial inaccuracies and compromised finishing quality. To address this limitation, an adaptive trajectory planning methodology is employed. This approach utilizes a 3D camera mounted on the robot's end-effector to capture dense point cloud data of the actual surface. The collected data is then processed to identify and localize surface irregularities. Based on this analysis, the finishing trajectory is dynamically adjusted, thereby enhancing the overall process robustness and quality.

4.2.1. Point Cloud Data Acquisition and Processing

The robot is controlled to move a high-precision 3D camera along a predefined scanning path, enabling comprehensive scanning of the workpiece mold surface. During the scanning process, the camera capture interval is predetermined. Point cloud data from individual viewpoints is acquired at regular spatial intervals, with each viewpoint's data stored collectively as a distinct set.

The point cloud set before registration:

$$P = \{P_1, P_2, ..., P_n\} \tag{1}$$

The i-th Point Cloud Before Registration:

$$P_i = \{p_{i1}, p_{i2}, ..., p_{iN_i}\} \tag{2}$$

Here, the number of viewpoints n is determined by the preset camera capture interval time. Each $p_{ij} \in \boldsymbol{R}^3$ represents a point within the i-th point cloud, and N_i denotes the total number of points in the i-th point cloud. The number of points in each point cloud depends on the resolution of the 3D camera, ensuring it meets the requirements for defect identification. During the scanning process, the pose information $\boldsymbol{T_i}$ of the 3D vision sensor (linked to the robot's end-effector joint coordinates) is synchronously recorded, providing initial pose constraints for subsequent point cloud registration.

$$\boldsymbol{T_i} = \begin{bmatrix} \boldsymbol{R}_i & t_i \\ 0 & 1 \end{bmatrix} \tag{3}$$

where $\boldsymbol{R_i}$ is a 3×3 rotation matrix representing the orientation of frame $\{\boldsymbol{T}\}$ relative to frame $\{\boldsymbol{G}\}$, and t_i is a 3×1 translation vector specifying the coordinates of the origin O_t of frame $\{\boldsymbol{T}\}$ in frame $\{\boldsymbol{G}\}$.

4.2.2 Point Cloud Registration and Fusion

Multi-viewpoint point cloud processing is conducted through point cloud registration and fusion. Using the vision sensor pose information $\boldsymbol{T_i}$ recorded during acquisition, the local point clouds P are registered and merged into a unified global point cloud. The specific procedure is as follows:

Using the recorded pose $\boldsymbol{T_i}$, the coordinates p_{iN_i} of each point in a local cloud P_i are transformed from the end-effector coordinate system $\{\boldsymbol{T_i}\}$ to the global coordinate system $\{\boldsymbol{G}\}$, The transformed point is denoted as ${p_{iN_i}}^G$.

$${p_{iN_i}}^G = \boldsymbol{R_i} p_{iN_i} + t_i \tag{4}$$

The Iterative Closest Point (ICP) algorithm is then applied for fine registration and iterative refinement. Finally, all transformed point clouds are merged into a consolidated global point cloud, denoted as $p^{G'} = \bigcup_{i=1}^{n} p_i^{G'}$.

This global surface point cloud of the segment is subsequently processed by a pre-trained neural

network based on a PointNet++ architecture, which is designed for direct regression of defect geometric parameters from 3D point cloud data. The network was trained on a dataset of 342 point cloud samples, augmented through random rotation, scaling (±5%), and Gaussian noise injection to enhance robustness. The model achieves a mean absolute error of 1.8 mm in defect center localization and 0.9 mm in dimensional prediction on the test set, demonstrating accuracy adequate for this application. The network directly outputs key parameters of identified surface imperfections, including their spatial location (3D center coordinates), classification (protrusion/depression), and a representative dimension (effective radius).

These parameters directly enable the subsequent geometric computations, which comprise two main steps:

1): Using the end trowel radius as a baseline interval, the center coordinates of each significant protrusion or depression area are located.

2): Simultaneously, for each point in the path sequence, the inclination angle between the current point and a look-ahead point (offset by half a trowel radius along the motion direction) is calculated. This is defined as the look-ahead tilt angle α.

These parameters are critical for the adaptive adjustment of the trowel posture in the subsequent trajectory planning stage.

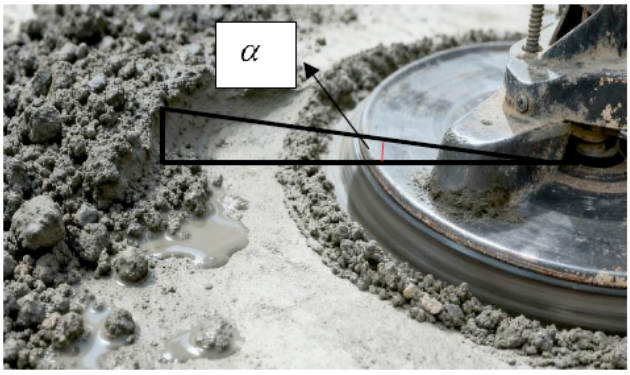


Figure 6: Schematic Diagram of the Look-ahead Tilt Angle.

4.2.3. Adaptive Finishing Trajectory Planning

Based on the predicted tilt angle identified from the surface point cloud by the neural network, adaptive modifications are applied to the original trajectory. This ensures the end-effector maintains optimal alignment with the local surface normal during movement. The revised trajectory equations for the four path segments are as follows:

Position:

$$\begin{cases} x_0(i) = x(i) \\ y_0(i) = y(i) + \delta_y(\alpha(i)); (t \in (0, 4n)) \\ z_0(i) = z(i) + \delta_z(\alpha(i)) \end{cases} \tag{15}$$

The orientation expressions remain unmodified.

In the equations above $\delta_y(\alpha(i)) = k \cdot r \sin(\alpha(i))$ represents the tilt angle compensation along the y-axis, and $\delta_z(\alpha(i)) = k \cdot r \cos(\alpha(i))$ denotes the tilt angle compensation along the z-axis. Here, k is the adaptive gain coefficient, calibrated based on practical conditions.

This compensation mechanism ultimately transforms the theoretical trajectory $P(i) = [x(i), y(i), z(i)]$ into the actual trajectory $P_0(i) = [x_0(i), y_0(i), z_0(i)]$ achieving the critical transition from idealized path planning to adaptive execution based on the actual surface.

4.2.4 Trajectory Point Coordinate Transformation

The three sets of rectangular trajectory equations in the $\{W\}$ coordinate system are obtained through the aforementioned calculations. Subsequently, the trajectory point coordinates in the $\{W\}$ system are derived via trajectory interpolation. However, during actual robotic operation, motion control relies on locally detected QR code benchmarks identified in real-time by the vision system, rather than a single global coordinate system. Therefore, the coordinates in the $\{W\}$ system cannot be directly used to control the robot.

To address this issue, the trajectory coordinates in the $\{W\}$ system must be transformed into the robot coordinate systems $\{O_i - x_{oi} y_{oi} z_{oi}\}, i \in (1,2,3)$ at each workstation. The specific procedure is as follows:

Step 1: Using the vector method, transform the trajectory point coordinates in the $\{W\}$ system into the corresponding QR code coordinate systems $\{Q_i - x_i y_i z_i\}, i \in (1,2,3)$:

$$Q_{ij} = \boldsymbol{\alpha}_i + (\boldsymbol{R}_i \cdot N_k)'; (i = 1,2,3) \tag{16}$$

The coordinate Q_{ij} represents the position of the j-th trajectory point in the i-th QR code coordinate system. Here, $\boldsymbol{\alpha}_i$ denotes the position vector of the origin of the $\{W\}$ system in the Q_i system, and $\boldsymbol{R}_i$ represents the rotation transformation matrix from the $\{W\}$ system to the i-th QR code coordinate system. N_k indicates the coordinates of the k-th trajectory point in the $\{W\}$ system.

Step 2: Using the vector method, transform the trajectory point coordinates in the QR code coordinate systems into the corresponding robot coordinate systems $\{O_i - x_{oi}y_{oi}z_{oi}\}, i \in (1,2,3)$ at the three workstations:

$$C_{ij} = \boldsymbol{\beta}_i + \left(\boldsymbol{R}_{oi} \cdot Q_{ij}\right)' \quad (17)$$

Here, C_{ij} represents the coordinates of the j-th trajectory point in the i-th robot coordinate system. $\boldsymbol{\beta}_i$ denotes the position vector of the origin of the Q_i system in the respective robot coordinate system, and $\boldsymbol{R}_{oi}$ represents the rotation transformation matrix from the i-th QR code coordinate system to the i-th robot coordinate system. Q_i refers to the trajectory coordinate sequence in the QR code coordinate system.

Ultimately, the trajectory point coordinates in the robot coordinate systems for the three workstations are obtained.

5. EXPERIMENTAL VALIDATION

This section presents the experimental validation of the proposed adaptive trajectory planning method. Field experiments were designed to compare its performance against a conventional fixed-trajectory approach under industrial production conditions. The evaluation encompasses both quantitative analysis based on 3D point cloud data and qualitative assessment of the finished surface quality.

5.1. Experimental Setup and Comparative Methodology

The experiments were conducted in a concrete segment production environment, as shown in Figure **7**. To ensure a fair comparison, segment samples from the same production batch were processed using two distinct methods:

(1): Fixed-Trajectory Method: This baseline method employed a pre-programmed offline trajectory without any online perception or adaptive capability, representing conventional robotic finishing.

(2): Proposed Adaptive Method: This method implemented the complete pipeline, including offline trajectory planning, 3D point cloud-based surface reconstruction, and real-time trajectory compensation using the look-ahead tilt angle.

All experiments were performed under identical conditions, utilizing the same robotic system, and maintaining strict consistency in concrete material, environmental parameters, and trowel motion settings. The 3D vision system captured high-density point clouds of the segment surface after processing with each method for subsequent quantitative analysis.

Figure 7: Concrete Segment Production Scenario.

5.2. Quantitative Results and Data Analysis

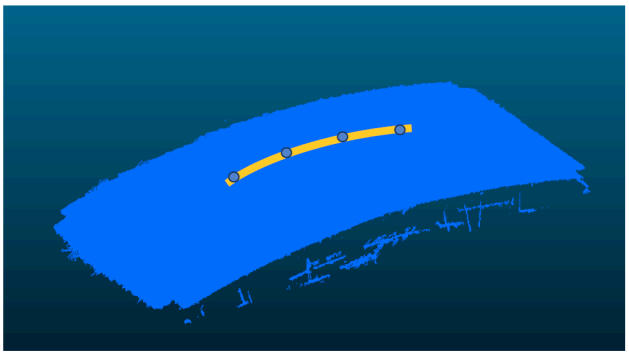
The quantitative evaluation focused on the flatness of the finished surface. A representative arc on the segment surface was selected, and surface height data at multiple points along this arc were extracted from the 3D point clouds. The comparative results are detailed in Table **1**.

The data unequivocally demonstrate the superiority of the adaptive method. While the fixed-trajectory method resulted in significant height variations (Range: 38.2 mm), the proposed method produced a consistently flat surface, reducing the variation range by 87.7% to merely 4.7 mm.

Table 1: Comparison of Surface Height Along a Representative arc Before and After Finishing

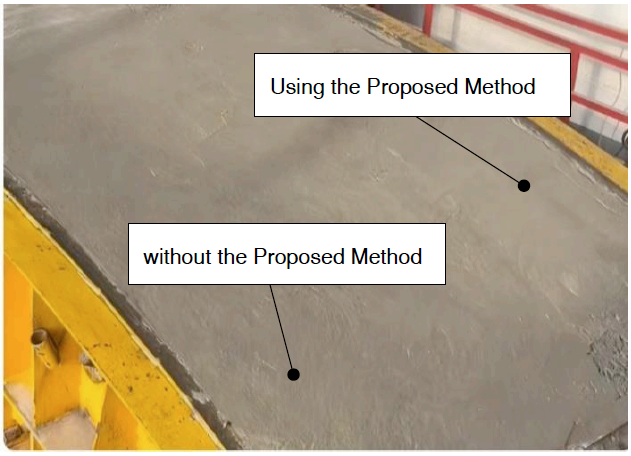
Location	Fixed-Trajectory Method (mm)	Adaptive Method (mm)
1	-15.8	2.3
2	-6.6	-2.4
3	22.4	1.6
4	8.9	-0.9
Range	**38.2**	**4.7**

The experimental results presented in this section, comprising the quantitative data in Table **1** and the visual evidence in Figure **8**, provide consistent and compelling validation of the proposed system's performance. The significant reduction in surface height variation and the clear visual improvement in finish quality unequivocally demonstrate the advantage of the adaptive method over the conventional fixed-trajectory approach under the tested conditions.



(**a**)

(**b**)



(**c**)

Figure 8: (**a**) Measurement setup; (**b**) Before finishing; (**c**) Conventional vs. Proposed method.

6. DISCUSSIONS AND CONCLUSION

6.1. Discussion

The experimental results demonstrate that the proposed adaptive trajectory planning method effectively addresses the core challenge of geometric uncertainty in concrete segment finishing. The significant improvement in surface quality, quantified by an 87.7% reduction in height variation, stems from the system's ability to transition from a static, pre-defined path to a dynamic, perception-driven process. This model-based adjustment capability, which integrates real-time 3D perception with trajectory compensation, is the key differentiator from prior fixed-trajectory or teach-and-repeat systems and directly tackles the research gap of lacking online adjustment capabilities in industrial finishing environments. The look-ahead tilt angle calculation is instrumental in ensuring the trowel maintains optimal contact with the surface, proactively compensating for undulations rather than reacting to them.

However, the study has certain limitations. The computational load for point cloud processing and neural network inference, while manageable in our experiments, presents a challenge for real-time application on very large surfaces or with faster cycle time requirements. Furthermore, the system's performance was validated under specific production conditions; its robustness across a wider range of concrete setting states and environmental factors warrants further investigation.

6.2. Conclusion

This paper has successfully tackled the issues of low automation and poor trajectory adaptability in concrete shield segment finishing by designing and implementing an integrated robotic system with an adaptive trajectory planning method. The principal achievements of this research are summarized as follows:

(1) A collaborative multi-coordinate system framework was established, enabling accurate and reliable trajectory mapping from offline programming to physical execution across multiple workstations, which is fundamental for deployable industrial automation.

(2) A structured offline programming methodology using rectangular trajectories was developed, providing complete surface coverage and establishing a high-quality initial path for subsequent adaptive refinements.

(3) A core technological contribution is the point cloud data-driven adaptive mechanism. This involves capturing the as-built surface geometry, fusing multi-view point clouds, employing a neural network for semantic analysis, and calculating a look-ahead tilt angle for real-time trajectory compensation, thereby ensuring constant tool-surface conformity.

(4) Comprehensive field validation confirmed the system's practical efficacy. The proposed method yielded superior finishing uniformity and surface quality compared to the fixed-trajectory baseline, proving its viability and reliability in a real-world production scenario.

In summary, this research provides a robust integrated solution that bridges the gap between offline planning and online adaptive execution for robotic finishing. The proposed method not only resolves the specific challenge of trajectory adaptation in concrete segment finishing but also offers a referable technical framework for the automated precision machining of other complex curved components. Future work will focus on enhancing computational efficiency for real-time performance, improving robustness under variable operational conditions, and advancing system intelligence through dynamic obstacle avoidance and multi-robot collaborative strategies.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China(NSFC) (grant 52375026), the Special Project for Basic Research Cooperation in Beijing-Tianjin-Hebei Region (24JCZXJC00270).

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] Xu Y, Liu Y, Liu X, *et al*. Trajectory Generation Method for Serial Robots in Hybrid Space Operations.Actuators, 2024, 13(3): 108. https://doi.org/10.3390/act13030108

[2] Feng Y, He J, Luo J, *et al*. A vision-based self-calibration method for industrial robots using variable pose constraints. Robotics and Computer-Integrated Manufacturing, 2026, 98103142-103142. https://doi.org/10.1016/j.rcim.2025.103142

[3] M Li, R Lu Target ball localization for industrial robots based on binocular stereo vision. Industrial Robot: the international journal of robotics research and application, 2025, 52(4): 600-609. https://doi.org/10.1108/IR-07-2024-0317

[4] Shinde V, Pawar P, Gadakh V. Research Progress on Industrial Robots: A Review. Recent Patents on Mechanical Engineering, 2026, 19(1): 27-61. https://doi.org/10.2174/0122127976326652241018114550

[5] Moorthy S, Sakthivel K S S, Joo H Y, *et al*. Distributed Observer-Based Adaptive Trajectory Tracking and Formation Control for the Swarm of Nonholonomic Mobile Robots with Unknown Wheel Slippage. Mathematics, 2025, 13(10): 1628-1628.

[6] Zheng Z Y, Gao J, Zheng Z Y, *et al*. High-precision hand-eye calibration method based on 3D vision point cloud registration. Journal of Machine Design, 2023, 40(S2): 51-56.

[7] Qiao D, Sike W, Xueqin C, *et al*. Pavement crack detection based on point cloud data and data fusion. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 2023, 381(2254): 20220165-20220165. https://doi.org/10.1098/rsta.2022.0165

[8] Liu R ,Jie B ,Tong Y , *et al*. Automatic virtual reduction of unilateral zygomatic fractures based on ICP algorithm: A preliminary study. Journal of stomatology, oral and maxillofacial surgery, 2025, 126(4S): 102220. https://doi.org/10.1016/j.jormas.2025.102220

[9] Kong F, Mei B, Fu Y, *et al*. Drilling task planning and offline programming of a robotic multi-spindle drilling system for aero-engine nacelle acoustic liners. Robotics and Computer-Integrated Manufacturing, 2025, 92102897-102897. https://doi.org/10.1016/j.rcim.2024.102897

[10] Müller F, Koch M, Hasse A. User Study to Validate the Performance of an Offline Robot Programming Method That Enables Robot-Independent Kinesthetic Instruction through the Use of Augmented Reality and Motion Capturing. Robotics, 2024, 13(3): 35. https://doi.org/10.3390/robotics13030035

[11] Zang Z, Zhang X, Gong X, *et al*. A spatio-temporal trajectory planning framework for AGVs based on motion primitive and dynamic programming in off-road environments. Advanced Engineering Informatics, 2026, 69(PB): 103934-103934. https://doi.org/10.1016/j.aei.2025.103934

[12] Yang Y, Sun S, Wu Y, *et al*. Integrating multi-equipment scheduling with accurate AGV path planning for U-shaped automated container terminals. Computers & Industrial Engineering, 2025, 209111427-111427. https://doi.org/10.1016/j.cie.2025.111427

[13] Duan Z Q. Design and control algorithm research of shield segment plastering robot. Southwest Jiaotong University, 2023.

[14] Li J F. Development and application research of segment plastering robot. China Machinery, 2023, (02): 48-51.

[15] Zhang L Y. Intelligent technology for metro shield tunnel segment production line. Modern Urban Rail Transit, 2022, (11): 47-52.

[16] Cai X Y. Research on the application of automated equipment in metro shield segment production. Stone, 2023, (09): 20-22.

[17] Yang H L. Adaptive trajectory generation and control strategy for robot welding of complex surfaces. Office Informatization, 2024, 29(17): 62-64. https://doi.org/10.1109/ISoIRS63136.2024.00011

[18] Yang S. Research on off-line programming technology for industrial robot laser cutting. Metal Working (Hot Working), 2025, (06): 50-53+59.

Yang X J, Guo Y W, Huang W. The application of off-line programming and simulation technology for industrial robots in intelligent manufacturing. Papermaking Equipment & Materials, 2024, 53(11): 103-105.

https://doi.org/10.31875/2409-9694.2025.12.07