

Research Article

A Method for Obtaining Optimal Path in Angle and Avoiding Collision for Robotic Belt Grinding

Tie Zhang , Xiaohong Liang , Ye Yu , and Bin Zhang 

School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China

Correspondence should be addressed to Tie Zhang; merobot@scut.edu.cn

Received 4 June 2019; Revised 21 August 2019; Accepted 3 September 2019; Published 11 November 2019

Academic Editor: Francisco J. Montáns

Copyright © 2019 Tie Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The angular variation of the joints may be large, and collision between workpieces and tools may occur in robotic grinding. Therefore, this paper proposes an optimal robotic grinding path search algorithm based on the recursive method. The algorithm is optimized by changing the position of the tool coordinate system on the belt wheel; thus, the pose of the robot during grinding is adjusted. First, the position adjustment formula of the tool coordinate system is proposed, and a coordinate plane is established to describe the grinding path of the robot based on the position adjustment formula. Second, the ordinate value of this coordinate plane is dispersed to obtain the search field of the optimal robotic grinding path search algorithm. Third, an optimal robotic grinding path search algorithm is proposed based on the recursive method and single-step search process. Finally, the algorithm is implemented on the V-REP platform. Robotic grinding paths for V-shaped workpieces and S-shaped workpieces are generated using this algorithm, and a grinding experiment is performed. The experimental results show that the robotic grinding paths generated by this algorithm can smoothly complete grinding operations and feature a smaller angular variation of the joint than other methods and no collision.

1. Introduction

Currently, industrial robots are widely used for curved belt grinding. Generally, the robotic grinding path is generated by offline programming. When the generated robotic grinding path is executed to grind a curved surface, the angular variation of the joint may sometimes be large, and therefore, the robot joint angle may exceed the limit. Collision may occur in robotic grinding, such as collision between workpieces and tools and collision between robots and tools. To avoid exceeding the limiting joint angle and collision in robotic grinding, the robotic grinding path generation algorithm should be optimized to make the generated robotic curved surface grinding path feature smaller angular variation of the joint and no collision.

Many scholars have made in-depth research on the path planning of robot belts grinding. For example, Tsai et al. studied the robot path planning problem of the automatic mold polishing system and concentrated the solution into a core system to generate a trajectory that meets the

requirements [1]. Basañez and Rosell have proposed a graphical task-level robot programming tool that allows users to set parameters and select polishing curves according to their need [2]. Zhao et al. proposed a robot blade grinding system controlled by a PC and a robot controller [3]. Wei Wang proposed a new algorithm based on the optimization curve interval by considering the curvature constraint of grinding the workpiece surface, integrating the precision and efficiency of the robot belt grinding [4]. Kharidege et al. proposed an automatic planning and programming system that can accommodate workpiece surfaces of different shapes [5]. Liu et al. divided the surface of complex parts into several simple surfaces when planning the path, which improved the efficiency of path planning [6]. Considering the collision problem of grinding the surface of complex parts, many scholars have studied the collision-free path planning of robots and proposed many classical algorithms, such as the C-space method [7, 8] and artificial potential field method [9–12]. Scholars such as Kuffner and Lavelle proposed the RRT

algorithm, which can solve the robotic collision-free path planning issue in high-dimensional C space [13]. Scholars such as Shibata and Murakami proposed a quick robotic collision-free path planning algorithm, which can transform C space to deformed C space by using the artificial attraction potential energy method. Deformed C space does not include a restricted area, thus greatly simplifying robotic collision-free path planning in C space. This algorithm approximates restricted areas and potential energy functions to reduce calculation effort and realize quick planning [14]. Scholars such as Li et al. improved the artificial potential field method based on the regression search method. The algorithm plans a collision-free track by using the new potential field function to make the algorithm avoid local optimal solutions and unreachable areas [15].

The artificial potential field method is simple and highly efficient; thus, the approach is widely applied in the track planning of manipulators. Tsuji et al. noted that track planning based on the artificial potential field method can be divided into force control methods and motion speed control methods [16]. Ge and Cui proposed a new repulsive potential field function based on the artificial potential field and further studied the goals nonreachable with obstacles nearby (GNRON) issue associated with the artificial potential field method [17]. Masoud et al. added a component tangent to the repulsion field, which can assist robots in bypassing obstacles [18, 19]. Bosscher and Hedman combined the potential field method with machine vision and applied their repulsion speed approach to a real-time collision avoidance algorithm for robotic manipulators [20]. Bayro-Corrochano et al. transformed the potential field to the position correction of a robot to drive a 5-degree-of-freedom robot to avoid obstacles [21]. Wu et al. obtained the minimal distance between a robot and obstacles using a camera and moved the robotic manipulator to avoid obstacles by altering the driving speed [22].

In recent years, intelligent methods, such as artificial neural network algorithms, genetic algorithms, and fuzzy algorithms, have been applied in robotic collision-free path planning. Scholars such as Gómez-Bravo combined the genetic algorithm with the RRT algorithm to avoid collision in the hybrid environment of CaPaMan robots and SCARA robots [23]. Scholars such as Pires proposed a robotic collision-free avoidance method based on a multitarget genetic algorithm [24]. Lee et al. used recurrent neural network and reinforcement learning combined with V-REP simulation platform to design the shortest path planning of mobile robots considering the shortest time [25].

Research on collision-free optimization and joint angle optimization in robotic belt grinding is very limited. Therefore, this paper studies joint angle optimization and collision-free optimization methods for the development of robotic belt grinding path generation algorithms. First, the position adjustment formula of the tool coordinate system is proposed, and a coordinate plane is established to describe the robotic grinding path based on the position adjustment formula. Second, the ordinate value of this coordinate plane

is discretized to obtain the search field of the optimal robotic grinding path search algorithm. Finally, the optimal robotic grinding path search algorithm is proposed based on the recursive method and single-step search process. The practicability of this algorithm is verified by simulation and grinding experiments. Meanwhile, the author's previously published article [26] failed to consider the problem of excessive joint angle increment in the collision avoidance and planning process, which is unfavorable to robot motion control. Thus, based on the modeling method of [26], this paper improves the trajectory retrieval algorithm. Not only collision avoidance but also the total change of joint angle in the process of robot motion is considered, so that the algorithm can obtain a robot trajectory with a small total joint change in the situation of ensuring that the motion trajectory does not collide. Moreover, this paper validates this conclusion by using V-shaped parts experiments.

2. Generation of Robotic Belt Grinding Path

2.1. Principle of Robotic Belt Grinding Path Generation. In order to generate the robot trajectory, it is necessary to generate cutter points on the grinding surface of the workpiece and establish the coordinate system $\{M_i\}$ on each cutter point. The method of generating cutter points is as follows. First, the machined surface of the workpiece is extracted and converted into NURBS surface [27]. Then, the processing path of NURBS curve format is generated by using the equal section method. Then, according to the set bow height error, the curvature estimation method is used to generate cutter points. Due to the limitation of space, this paper simply introduces the method of generating cutter points, which can be referred to the article [26] published by the author before. The cutter point coordinate system is shown in Figure 1, where $\{T\}$ represents the inherent coordinate system of the workpiece and i is the sequence number of the cutter point. The direction of the Z axis of $\{M_i\}$ is normal to the curved surface pointing to the exterior. The direction of the X -axis of $\{M_i\}$ is the tangent of the grinding cutter path at the cutter point. The direction of the Y -axis of $\{M_i\}$ can be determined by the right-hand rule.

The coordinate position of the cutter points, the normal to the grinded surface, and the tangent of the grinding cutter path are described by the workpiece inherent coordinate system $\{T\}$. According to the definition of the pose matrix, the pose matrix ${}^T_{M_i}T$ of the coordinate system $\{M_i\}$ to the workpiece inherent coordinate system $\{T\}$ can be obtained.

Figure 2 shows the robotic grinding platform. The robot body is the MOTOMAN-MH24 robot produced by YASKAWA, and its model is YR-MH00024-A00. $\{B\}$ is the base coordinate system of the robot. $\{6\}$ is the coordinate system of the robot end joint. $\{G\}$ is the tool coordinate system of the belt wheel. $\{M_i\}$ is the coordinate system of the i^{th} cutter point on the ground curved surface. $\{T\}$ is the inherent coordinate system of the workpiece. The pose matrix 6_7T of the coordinate system $\{T\}$ to the coordinate system $\{6\}$ depends on the structure of the workpiece fixture. The pose matrix B_GT of the tool coordinate system $\{G\}$ to the base coordinate system $\{B\}$ can be obtained by calibration.

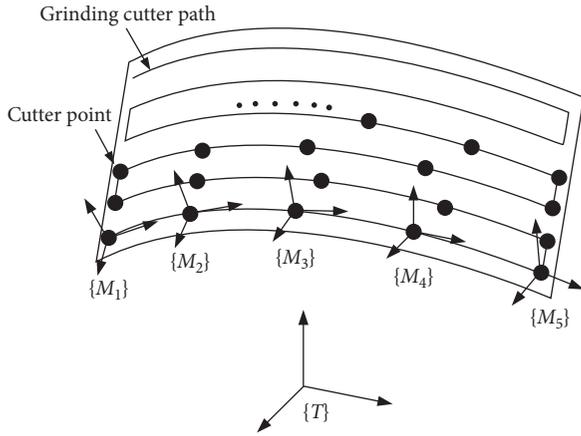


FIGURE 1: Establishment of cutter point coordinate system.

The grinding process essentially allows the cutter points to contact the belt wheel in sequence. When the i^{th} cutter point is ground, the coordinate system $\{M_i\}$ coincides with the tool coordinate system $\{G\}$, and we can obtain the following:

$${}^B_{M_i}T = {}^B_GT, \quad (1)$$

where ${}^B_{M_i}T$ can be expanded as follows:

$${}^B_{M_i}T = {}^T_{M_i}T {}^6_T T {}^B_6T. \quad (2)$$

By combining equation (1) with equation (2), we can obtain the following:

$${}^B_6T = ({}^6_T T)^{-1} ({}^T_{M_i}T)^{-1} {}^B_GT. \quad (3)$$

By inputting the pose matrix B_6T obtained from formula (3) into the reverse kinematics solution algorithm, we can obtain the joint space pose $\theta = [\theta_1 \sim \theta_6]$ of the robot when the i^{th} cutter point is ground.

By substituting the pose matrix B_6T , ${}^6_T T$, and ${}^T_{M_i}T$ into formula (3) in sequence, we can obtain the corresponding B_6T of the i^{th} cutter point. According to the reverse kinematics solution algorithm and the corresponding B_6T of each cutter point, we can obtain the robot path for grinding this curved surface.

2.2. Large Angular Variation and Collision Occur in Robotic Grinding. The tool coordinate system $\{G\}$ is fixed in the generation of the robotic grinding path, namely, B_GT does not vary. For example, the tool coordinate system $\{G\}$ is fixed at the middle position of the front end of the belt, as shown in Figure 3.

For some workpieces, the joint angle of the generated robotic grinding path will vary greatly. The joint angle may exceed the limit in robotic grinding. For example, Figure 4 shows a V-shaped workpiece. The row pitch of the cutter path is set to 5 mm on the V-shaped curve, and the bow height error between the adjacent cutter points is set to 0.05 mm. The coordinate system $\{M\}$ is established at each cutter point, as shown in Figure 5.

The pose matrix B_GT of the tool coordinate system $\{G\}$ to the robot base coordinate system $\{B\}$ is set as follows:

$${}^B_GT = \begin{bmatrix} 0.007 & 0 & 1 & 890.3631 \\ 1 & -0.0012 & -0.007 & 235.6854 \\ 0.0012 & 1 & 0 & 323.1888 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

The pose matrix ${}^6_T T$ of the workpiece coordinate system $\{T\}$ to the coordinate system $\{6\}$ is set as follows:

$${}^6_T T = \begin{bmatrix} 0.9237 & 0.3827 & 0.0158 & 9.6032 \\ -0.3826 & 0.9239 & -0.0067 & -3.9836 \\ -0.0171 & 0.0001 & 1 & 85.6760 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

By substituting the pose matrix B_GT , ${}^6_T T$, and every ${}^T_{M_i}T$ into formula (3), we can obtain the robot path for grinding the V-shaped workpiece. The robot path for grinding the V-shaped workpiece is shown in Figure 6. As indicated, the angular variation of the 4th joint is large.

Collision may occur when the generated robotic grinding path is executed. As shown in Figure 7, when the robot is machining part A of the S-shaped workpiece, part B of the S-shaped workpiece collides with the belt wheel and the surface shape of the workpiece is damaged.

Given the disadvantages of the robotic grinding path generation algorithm outlined above, it is necessary to optimize the robotic grinding path generation process to make the generated robotic grinding path feature smaller angular variation of the joint and no collision.

3. Optimization Method in Robotic Grinding

3.1. Optimization Principle. The pose matrix B_GT in formula (3) can be adjusted to optimize the robotic grinding path. As shown in Figure 8(a), the $\{G\}$ coordinate system can be moved by a certain distance along the negative direction of the X-axis to prevent collision between the belt wheel and part B area of the S-shaped workpiece. As shown in Figure 8(b), the $\{G\}$ coordinate system can be rotated by a certain angle at the edge of the belt wheel to adjust the robot pose during robotic grinding.

A proper position of $\{G\}$ is searched for each cutter point in the generation of the robotic grinding path. The contact position between the workpiece and the belt wheel during grinding can be varied over different positions of $\{G\}$. Then, the robot pose during robotic grinding can be adjusted by changing the position of $\{G\}$ to obtain a robotic grinding path with a smaller joint change and no collision.

The initial position of the tool coordinate system $\{G\}$ is set at the midpoint of the belt wheel width (as shown in Figure 3). The initial tool coordinate system is marked $\{G_0\}$. The Z-axis of $\{G_0\}$ is vertical to the axis of the belt wheel and parallel to the horizontal plane. The X-axis of the $\{G_0\}$ is parallel to the axis of the belt wheel and points to the left. The pose matrix ${}^B_{G_0}T$ of $\{G_0\}$ to the robot base coordinate system $\{B\}$ can be obtained by calibration. $\{G_0\}$ enables translational

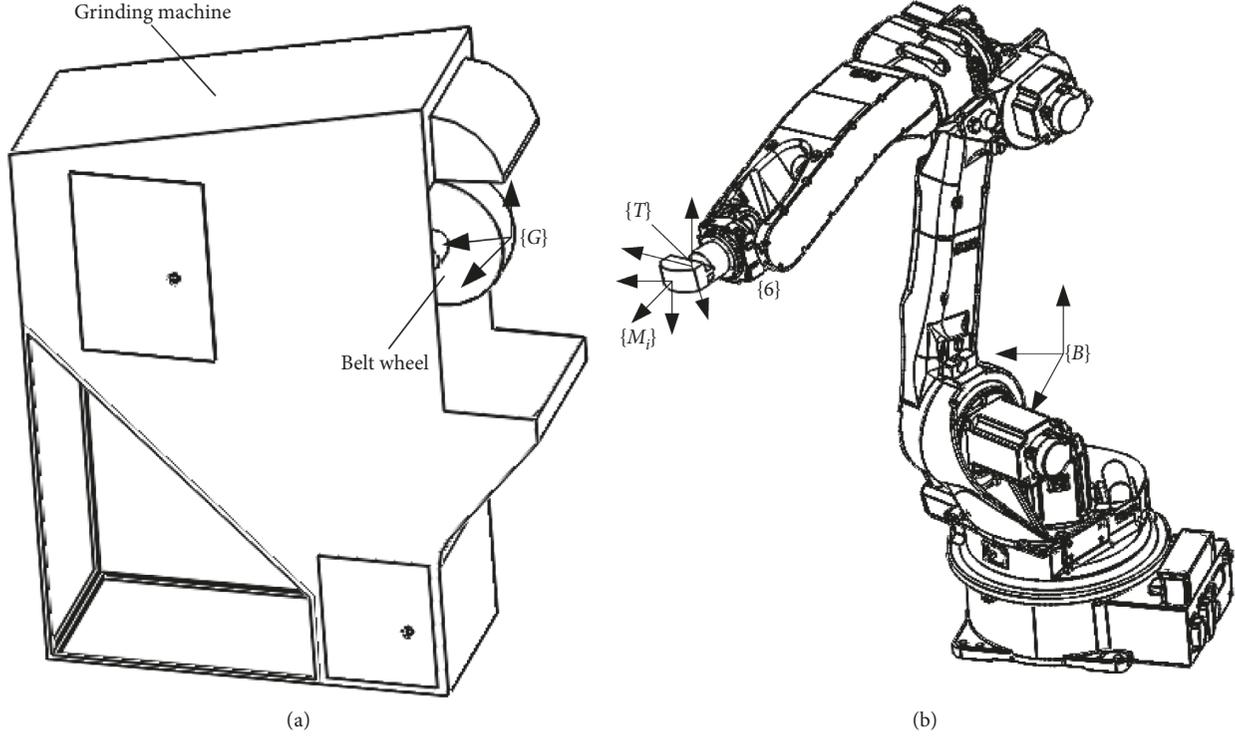


FIGURE 2: Robotic grinding platform.

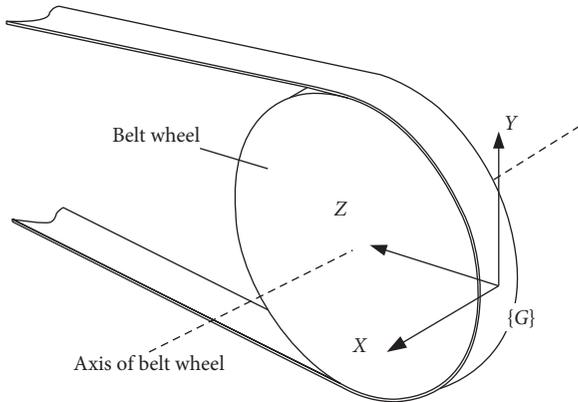


FIGURE 3: Fixed {G} coordinate system.

adjustment within the width range of the belt wheel, as shown in Figure 9.

Given that the width of the belt wheel is w , $\{G_0\}$ is the origin of translational adjustment. Rightward movement of the $\{G_0\}$ coordinate system is the forward direction of translational adjustment. Leftward movement is the negative direction of translational adjustment. Variable a is the distance of translational adjustment, and the range of a is $[-w/2, w/2]$. $\{G\}$ is the coordinate system after translational adjustment for $\{G_0\}$. The pose matrix of $\{G\}$ to the robot base coordinate system $\{B\}$ is ${}^B_G T$; thus, we obtain the following:

$${}^B_G T = {}^B_{G_0} T \cdot {}^{G_0}_G T. \quad (6)$$

In this formula, ${}^{G_0}_G T$ is the pose matrix of $\{G\}$ to $\{G_0\}$. The corresponding calculation formula is expressed as follows:

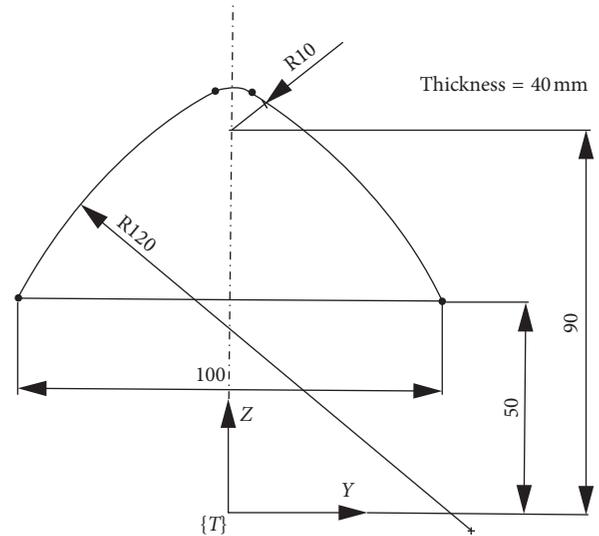


FIGURE 4: Size of V-shaped workpiece.

$${}^{G_0}_G T = \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

The tool coordinate system is indicated as $\{L\}$ or $\{R\}$ when $\{G_0\}$ lies on the left or right edge of the belt wheel, respectively. $\{L\}$ can rotate clockwise around the Y-axis, and $\{R\}$ can rotate anticlockwise around the Y-axis, as shown in Figure 10.

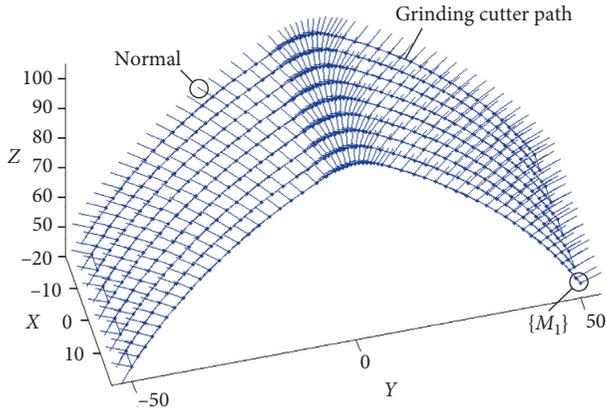


FIGURE 5: Cutter point of V-shaped workpiece.

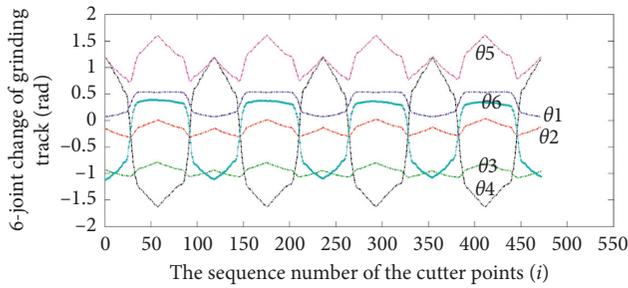


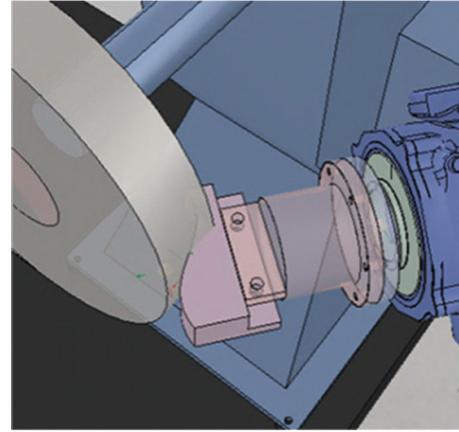
FIGURE 6: Angular variation of the robot track for grinding V-shaped workpiece.

Given that $\{L\}$ rotates around the Y-axis clockwise by d degrees, the tool coordinate system is indicated as $\{G\}$ after rotational adjustment. The pose matrix ${}^B_G T$ of $\{G\}$ to the robot base coordinate system $\{B\}$ is as follows:

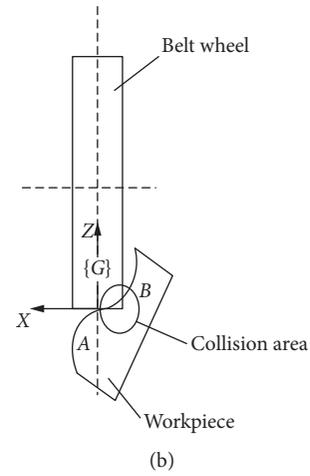
$${}^B_G T = {}^B_{G_0} T \cdot {}^{G_0}_L T \cdot {}^L_G T = {}^B_{G_0} T \cdot \begin{bmatrix} 1 & 0 & 0 & \frac{w}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \cos d & 0 & -\sin d & 0 \\ 0 & 1 & 0 & 0 \\ \sin d & 0 & \cos d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To simplify formula (8), we can obtain the following:



(a)



(b)

FIGURE 7: Collision situation. (a) Collision in machining. (b) Top view of grinding part.

$${}^B_G T = {}^B_{G_0} T \cdot \begin{bmatrix} \cos d & 0 & -\sin d & \frac{w}{2} \\ 0 & 1 & 0 & 0 \\ \sin d & 0 & \cos d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Similarly, after $\{R\}$ rotates about the Y-axis anticlockwise by d degrees, the tool coordinate system $\{G\}$ is obtained. The pose matrix ${}^B_G T$ of $\{G\}$ to the robot base coordinate system $\{B\}$ is

$${}^B_G T = {}^B_{G_0} T \cdot \begin{bmatrix} \cos d & 0 & \sin d & -\frac{w}{2} \\ 0 & 1 & 0 & 0 \\ -\sin d & 0 & \cos d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

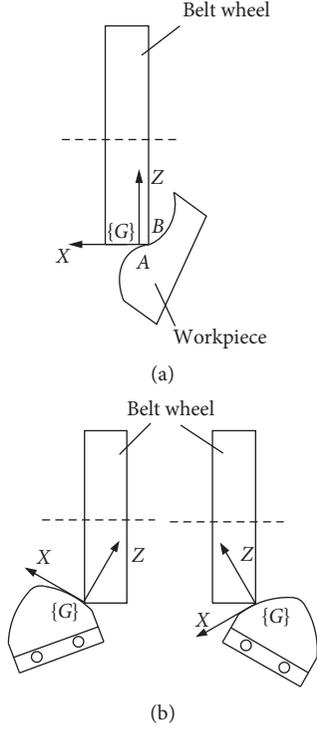


FIGURE 8: Adjustment method of the tool coordinate system $\{G\}$. (a) Translational adjustment for $\{G\}$. (b) Rotational adjustment for $\{G\}$.

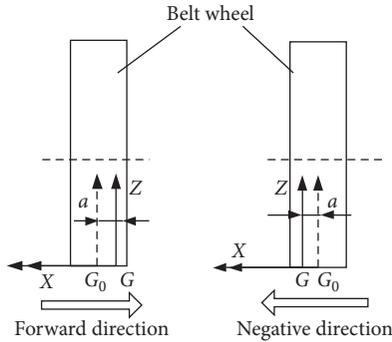


FIGURE 9: Translational adjustment for $\{G\}$.

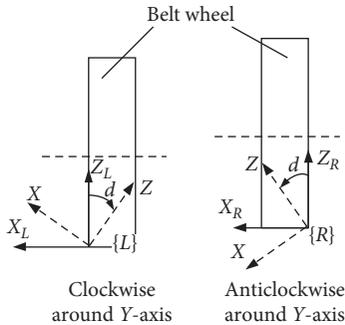


FIGURE 10: Rotational adjustment for the tool coordinate system.

3.2. Consolidation of Tool Coordinate System Adjustment Formula. The grinding area of the belt wheel can be divided into three areas (shown as Figure 11): a left rotational

adjustment area, a right rotational adjustment area, and a translational adjustment area. The tool coordinate system can make corresponding adjustments in these areas. Because the left rotational adjustment area, the right rotational adjustment area, and the translational adjustment area do not overlap, the calculation formula for the tool coordinate system after adjustment can be merged and described by one variable t .

Given that the variable t is 0 at $\{G_0\}$, the width of the belt wheel is w and D is a constant representing the range of angles at which the workpiece can rotate at the edge of the abrasive pulley; thus, we can obtain the following:

- (1) For $t \in [(-w/2) - D, -w/2]$, the tool coordinate system $\{G\}$ is in the left rotational adjustment area. The clockwise rotation angle around the Y -axis is $d = -(w/2) - t$. After d is substituted into formula (9), the corresponding ${}^B_G T(t)$ of the variable t can be obtained:

$${}^B_G T(t) = {}^B_{G_0} T \cdot \begin{bmatrix} \cos\left(\frac{w}{2} + t\right) & 0 & \sin\left(\frac{w}{2} + t\right) & \frac{w}{2} \\ 0 & 1 & 0 & 0 \\ -\sin\left(\frac{w}{2} + t\right) & 0 & \cos\left(\frac{w}{2} + t\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

- (2) For $t \in [-w/2, w/2]$, the tool coordinate system $\{G\}$ is in the translational adjustment area. After t is substituted into formulas (6) and (7), the corresponding ${}^B_G T(t)$ of the variable t can be obtained:

$${}^B_G T(t) = {}^B_{G_0} T \cdot \begin{bmatrix} 1 & 0 & 0 & -t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

- (3) For $t \in [w/2, (w/2) + D]$, the tool coordinate system $\{G\}$ is in the right rotational adjustment area. The anticlockwise rotation angle around the Y -axis is $d = t - w/2$. After d is substituted into formula (10), the corresponding ${}^B_G T(t)$ of the variable t can be obtained:

$${}^B_G T(t) = {}^B_{G_0} T \cdot \begin{bmatrix} \cos\left(t - \frac{w}{2}\right) & 0 & \sin\left(t - \frac{w}{2}\right) & -\frac{w}{2} \\ 0 & 1 & 0 & 0 \\ -\sin\left(t - \frac{w}{2}\right) & 0 & \cos\left(t - \frac{w}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (13)$$

As long as the specific value of t is given, the pose matrix ${}^B_G T(t)$ can be obtained by equations (11) to (13); that is, each value of t corresponds to a specific position of $\{G\}$. A proper t value shall be specified for each cutter point. The ${}^B_G T(t)$ of each cutter point and ${}^T_{M_i} T$ are substituted into formula (3) to obtain a robotic grinding path with smaller joint angular variation and no collision.

3.3. Coordinate Plane Description of Robotic Grinding Path.

To describe the process of generating the optimal robotic grinding path, the coordinate plane in Figure 12 is established. The ordinate value of the coordinate plane is the sequence number of cutter points. The abscissa value is the variable t . Every point (i, t) in the coordinate plane is a node marked Node (i, t) . The notation Node (i, t) indicates that the robotic grinding pose of the i^{th} cutter point is generated by using the pose matrix ${}^B_G T(t)$.

Each cutter point must be ground only once; therefore, each cutter point corresponds to a single node. The nodes can be sorted according to the sequence number of the corresponding cutter point. The sorted nodes compose the robotic grinding path.

The shaded region is the collision area. The nodes of the robotic grinding path shall not fall into the shaded region.

4. Optimal Search Algorithm for Robot Grinding Path

4.1. Generation of Search Field. The coordinate plane in Figure 12 indicates that the robotic grinding path is a list composed of a series of nodes. The optimal search algorithm searches the optimal robotic grinding path in the collection that contains all nodes.

t is a continuous variable such that each column of the coordinate plane includes limitless nodes. The positive integer M can be set, and t can be uniformly dispersed over M values, where M is the input of the method and a constant value; then, we can obtain a node matrix with M rows and N columns, where N is the total number of cutter points, as shown in Figure 13.

A structure can be established to save the information of a node. The members of the node structure are described as follows:

- (1) The variable t is used to save the corresponding ordinate value of the node
- (2) The variable i is used to save the corresponding cutter point sequence number of the node
- (3) The array RobotPose[6] with a length of 6 is used to save the robot posture of the node
- (4) The array Origin[3] with a length of 3 is used to save the origin position of $\{M_i\}$
- (5) The array AxisZ[3] with a length of 3 is used to save the direction vector of the Z-axis of $\{M_i\}$
- (6) The array AxisX[3] with a length of 3 is used to save the direction vector of the X axis of $\{M_i\}$

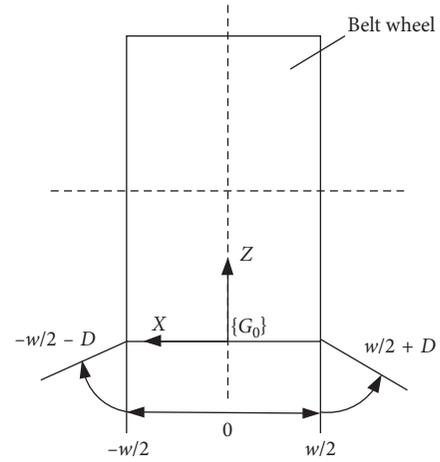


FIGURE 11: Division of the belt wheel grinding area.

The node structure is established for each node of the matrix in Figure 13. Then, we can obtain the node structure matrix NodeMatrix with M rows and N columns, as shown in Figure 14. The NodeMatrix is used as the search field of the optimal search algorithm. The optimal search algorithm searches the elements in the NodeMatrix; if the search is successful, we can obtain a robotic grinding path with smaller joint angular variation and no collision.

4.2. Single-Step Search. After NodeMatrix is obtained, the optimal search algorithm is designed. Based on the above description and actual grinding conditions, the optimal robotic grinding path should satisfy the following three requirements: the variation of the robot joint angles is small; the parts are continuously cut; and there is no collision between the workpieces and the belt wheel.

To better describe the algorithm, this section first describes the single-step search process of the algorithm. The optimal search algorithm is designed based on the single-step search process and the recursive method. The robotic grinding path generated by the algorithm should satisfy the three abovementioned requirements.

In each single-step search, the algorithm selects only one node from a column of NodeMatrix and stores the selected nodes into the track list RobotTrackList. The nodes in the i^{th} column of NodeMatrix are the searchable nodes of the i^{th} cutter point.

Provided that the algorithm searches the i^{th} column of NodeMatrix, the last node of the track list RobotTrackList corresponds to the $(i-1)^{\text{th}}$ cutter point, which is marked Node $_{i-1}$. The nodes of the i^{th} column are searched as follows:

Step 1: traverse all nodes of the i^{th} column of NodeMatrix and determine the max joint angle $\text{Max } \theta_i$ and the minimal joint angle $\text{Min } \theta_i$ ($i=1\sim 6$) of the entire track when these nodes act as Node $_i$ of RobotTrackList. Iteratively, calculate the total joint change $\Delta\theta$ using

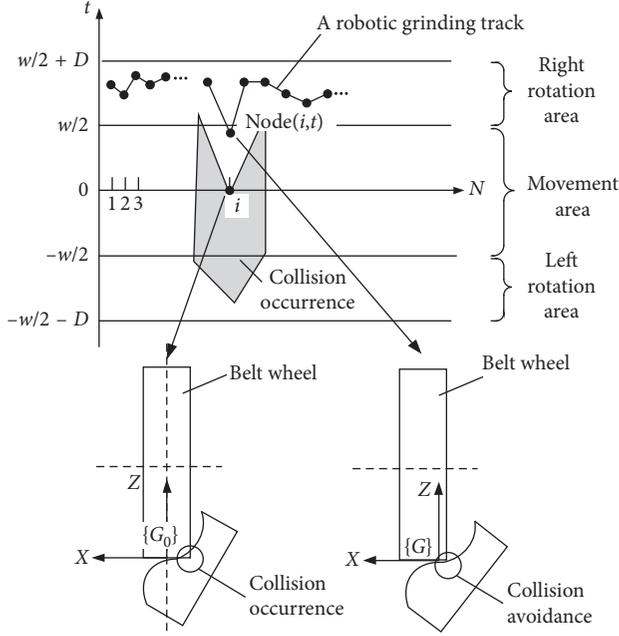


FIGURE 12: Coordinate plane description of robotic grinding path.

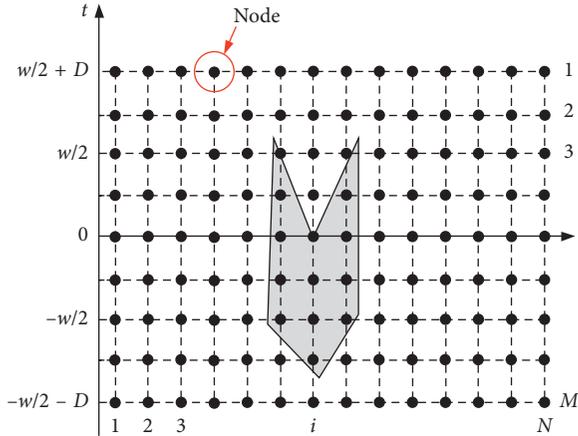


FIGURE 13: Node matrix.

$$\Delta\theta = \sum_{i=1}^6 |\text{Max } \theta_i - \text{Min } \theta_i|. \quad (14)$$

Step 2: sort the nodes of the i^{th} column of NodeMatrix in the ascending order of $\Delta\theta$. After sorting, the first node of the i^{th} column of NodeMatrix indicates the node that guarantees minimal joint angular variation.

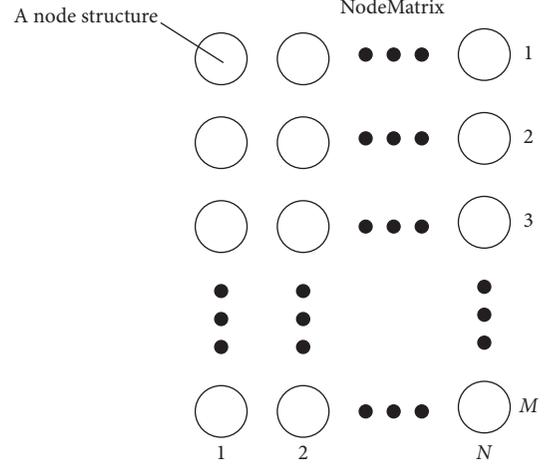


FIGURE 14: Node structure matrix NodeMatrix.

Then, the first node is inserted into the end of RobotTrackList and acts as Node_i , which corresponds to the i^{th} cutter point, as shown in Figure 15.

Step 3: calculate the difference Δt between Node_{i-1} and Node_i of RobotTrackList. If the value of Δt is large, the distance on the belt wheel between Node_{i-1} and Node_i is wide, as shown in Figure 16.

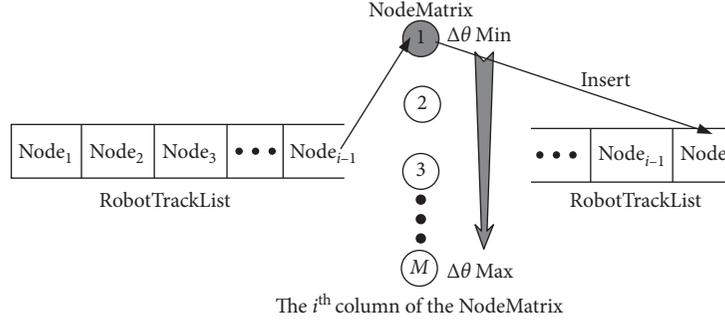
To ensure that the workpiece is continuously cut during grinding, the intermediate nodes should be inserted between Node_{i-1} and Node_i .

For example, when the track is executed to Node_{i-1} , the part is located on the left edge of the belt wheel; when the track is executed to Node_i , the part is located on the right edge of the belt wheel, as shown in Figure 17. To ensure continuous grinding of the part, the additional intermediate nodes should be inserted between two adjacent nodes. The inserted intermediate nodes are equivalent to additional postures inserted between two machining postures.

Provided that the difference Δt between two adjacent nodes does not exceed ΔT , define middleNum as the number of intermediate nodes between Node_{i-1} and Node_i . The value of middleNum can be calculated by the following equation:

$$\text{middleNum} = \left\lceil \frac{t_i - t_{i-1}}{\Delta T} \right\rceil, \quad (15)$$

where $\lceil * \rceil$ is an integer function and ΔT is a constant, which is the maximum limit to the change of adjacent t values.


 FIGURE 15: Search of i^{th} node.

Through linear interpolation of $Node_{i-1}$ and $Node_i$, the members in the structures of the intermediate nodes can be calculated as follows:

$$\left\{ \begin{array}{l} t_j = t_{i-1} + \Delta T \cdot j, \\ i_j = i - 1 + \frac{\Delta T \cdot j}{t_i - t_{i-1}}, \\ \text{Origin}_j = \text{Origin}_{i-1} + \frac{(\text{Origin}_i - \text{Origin}_{i-1})\Delta T \cdot j}{t_i - t_{i-1}}, \\ \text{Axis}Z_j = \text{Axis}Z_{i-1} + \frac{(\text{Axis}Z_i - \text{Axis}Z_{i-1})\Delta T \cdot j}{t_i - t_{i-1}}, \\ \text{Axis}X_j = \text{Axis}X_{i-1} + \frac{(\text{Axis}X_i - \text{Axis}X_{i-1})\Delta T \cdot j}{t_i - t_{i-1}}, \end{array} \right. \quad (16)$$

$j = (1 \sim \text{middleNum})$. After t_j , Origin_j , $\text{Axis}Z_j$, and $\text{Axis}X_j$ are calculated, the coordinate system $\{M_j\}$ of the intermediate nodes can be established and ${}^T_{M_j}T$ can be obtained. ${}^B_G T(t_j)$ can be obtained by equations (11) and (13) according to t_j . Then, ${}^T_{M_j}T$ and ${}^B_G T(t_j)$ are substituted into equation (3), and the robot grinding posture of $Node_j$ can be obtained by the inverse kinematics algorithm.

After all intermediate nodes are obtained, these nodes are inserted between $Node_{i-1}$ and $Node_i$, as shown in Figure 18.

Step 4: Because $Node_i$ and the intermediate nodes may fall into the collision area, collision detection shall be performed for $Node_i$ and the intermediate nodes.

- (1) If $Node_i$ and the intermediate nodes do not fall into the collision area, let $Node_i$ be the starting point, and conduct the next search round of search.
- (2) If $Node_i$ or one of the intermediate nodes is within the collision area, collision will occur during grinding; thus, the current $Node_i$ and the intermediate nodes shall be abandoned; then, the

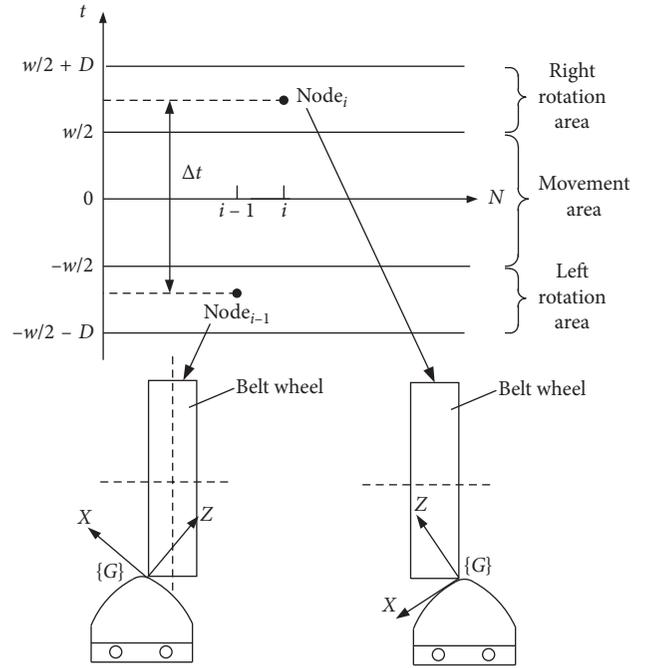


FIGURE 16: Wide distance between positions of two adjacent nodes.

search starting point shall return to $Node_{i-1}$, and the next node of the i^{th} column of NodeMatrix is searched.

As shown in Figure 19, the first node of the i^{th} column of NodeMatrix is searched and Track1 is obtained. Because collision is detected in the partial nodes of Track1, Track1 shall be abandoned and the search starting point returns to $Node_{i-1}$. The second node of the i^{th} column of NodeMatrix is searched, and Track2 is obtained. The algorithm detects whether collision occurs in the $Node_i$ and the intermediate nodes of Track2. If collision still occurs, the Track2 shall be abandoned and the third node of the i^{th} column of NodeMatrix shall be searched; if no collision occurs, the nodes on Track2 shall be placed into the RobotTrackList, and the next round of search is conducted.

4.3. Optimal Robotic Grinding Path Search Algorithm Based on Recursive Method. The optimal robotic grinding path search algorithm is proposed based on the single-step search

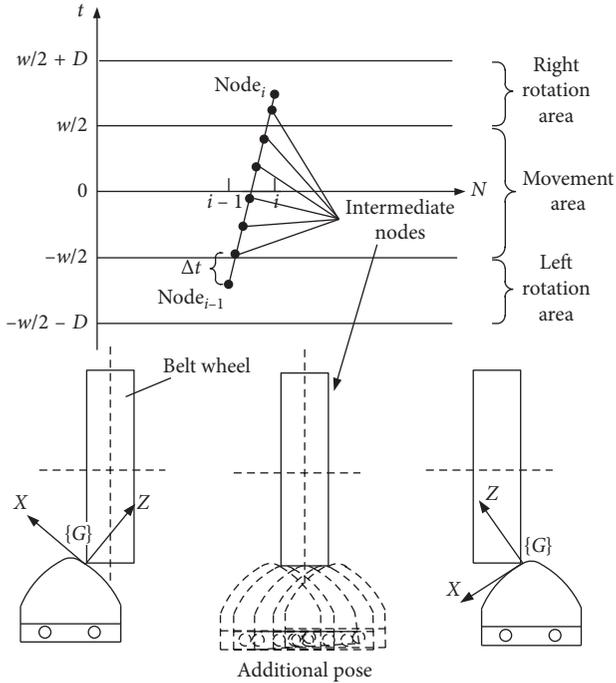


FIGURE 17: Insert the intermediate nodes.

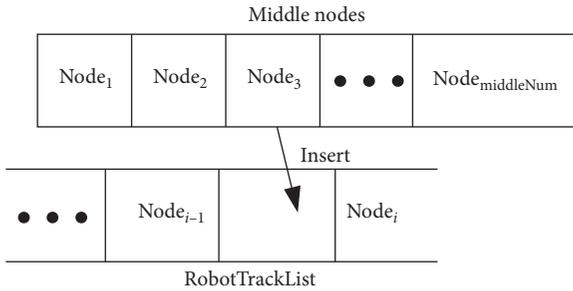


FIGURE 18: Insertion track list of intermediate nodes.

method. Provided that the i^{th} cutter point has been searched, Node_i of RobotTrackList is the k_i^{th} node of the i^{th} column of NodeMatrix . The nodes of the $(i+1)^{\text{th}}$ column of NodeMatrix are searched. All situations in searching can be summarized by the following three situations:

Situation 1: when the k_{i+1}^{th} node of the $(i+1)^{\text{th}}$ column of NodeMatrix is searched, this node is used as Node_{i+1} and the intermediate nodes are generated. Then, collision detection is conducted for Node_{i+1} and the intermediate nodes. If no collision is detected, the search function is recursively called to search the nodes of the $(i+2)^{\text{th}}$ column of NodeMatrix . Because the i^{th} layer of the search function is called when Node_{i+1} is searched, this is the $(i+1)^{\text{th}}$ layer recursive call of the search function, as shown in Figure 20.

Situation 2: when the k_{i+1}^{th} node of the $(i+1)^{\text{th}}$ column of NodeMatrix is searched, this node is used as Node_{i+1} , and the intermediate nodes are generated. Then, collision detection is conducted for Node_{i+1} and the intermediate nodes. If collision is detected and k_{i+1} is not

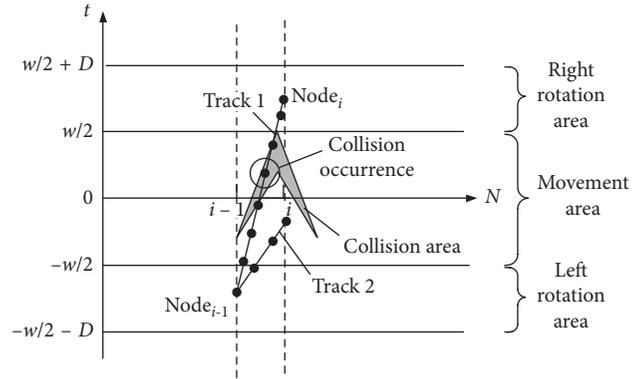


FIGURE 19: Collision conditions.

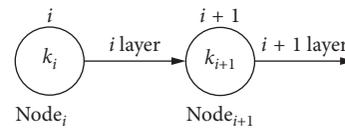


FIGURE 20: Situation 1.

equal to M , Node_{i+1} and the intermediate nodes are abandoned. Then, the i^{th} layer of the search function continues, and the $(k_{i+1}+1)^{\text{th}}$ node of the $(i+1)^{\text{th}}$ column of NodeMatrix is searched, as shown in Figure 21.

Situation 3: when the k_{i+1}^{th} node of the $(i+1)^{\text{th}}$ column of NodeMatrix is searched, this node is used as Node_{i+1} , and the intermediate nodes are generated. Then, collision detection is conducted for Node_{i+1} and the intermediate nodes. If collision is detected and k_{i+1} is equal to M , the algorithm cannot search the node without collision from the $(i+1)^{\text{th}}$ column of NodeMatrix . Node_i , Node_{i+1} and the corresponding intermediate nodes of two nodes in the RobotTrackList would be abandoned. Then, the i^{th} layer of the search function is terminated and returns to the $(i-1)^{\text{th}}$ layer of the search function. The $(i-1)^{\text{th}}$ layer of the search function continues, and the $(k_{i+1})^{\text{th}}$ node of the i^{th} column of NodeMatrix is searched, as shown in Figure 22.

By combining the three situations and based on the recursive method, the implementation steps of the optimal robotic grinding path search algorithm can be established. Provided that the NodeMatrix has been obtained, the numbers of rows and columns of NodeMatrix are M and N . The first-column node of NodeMatrix closest to the initial robot posture is denoted Node_1 . Node_1 is inserted into the RobotTrackList .

Algorithm 1 shows the pseudocode of the recurse search method.

5. Simulation and Experiment

Through the analysis of the above sections, in theory, the proposed algorithm can optimize the grinding path of the robot and can generate a path with small angle change and

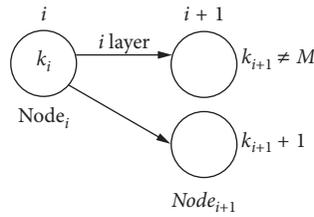


FIGURE 21: Situation 2.

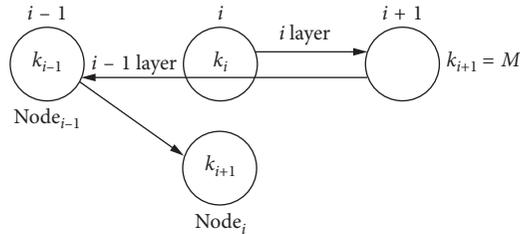


FIGURE 22: Situation 3.

no collision. However, in order to better verify the feasibility of the algorithm, simulation and grinding experiments of V-shaped workpiece are needed to verify the advantage that the algorithm can generate processing path with small angle change. Furthermore, the collision avoidance performance of grinding path can be verified by simulation and experiment of S-shaped workpiece.

5.1. Algorithm Implementation

5.1.1. Algorithm Implementation Platform. The optimal robotic grinding path search algorithm is implemented on the V-REP platform. V-REP is an open-source robot simulation and integration development platform. The platform can accurately simulate the entire robot system. In addition, V-REP provides a rich and powerful application program interface (API), so users can quickly validate the algorithm. The scenario simulation, collision detection, and motion simulation functions in V-REP are mainly used in implementing the algorithm.

Visual Studio2013 and Qt are used to program a secondary development program for V-REP. The secondary development program includes the optimal robotic grinding path search algorithm, the robot forward kinematics algorithm, and the robot inverse kinematics algorithm. The V-REP software runs the scenario simulation, collision detection, and motion simulation functions. The secondary development program can transfer the robot posture to the V-REP for simulation. V-REP can transfer the simulation results and collision detection results to the secondary development program to provide the necessary information for the optimal robotic grinding path search algorithm.

5.1.2. General Flow of the Optimal Robotic Grinding Path Search Algorithm. Table 1 shows the input parameters of the optimal robotic grinding path search algorithm.

The general flow of the algorithm is described as follows:

Input: the workpiece cutter points, the parameters in Table 1, the empty track list RobotTrackList, and the empty matrix NodeMatrix

- (1) According to the cutter points of the workpiece, the parameters in Table 1, and the tool coordinate system position adjustment formula, the coordinate plane in Figure 12 can be established. Then, the positive integer M can be set, and the ordinate value of the coordinate plane is dispersed to obtain the NodeMatrix. The NodeMatrix has M rows and N columns, where N is the total number of cutter points.
- (2) Make the node in the first column of NodeMatrix closest to the initial robot posture Node_1 . Then, insert Node_1 into RobotTrackList.
- (3) The recursive search function $\text{RecurseSearch}()$ is called to search the nodes in NodeMatrix. When the function $\text{RecurseSearch}()$ is terminated and the algorithm returns to this position, step (4) is performed.
- (4) Extract the last node structure in RobotTrackList. Determine whether the variable i of the structure is equal to N .

If $i = N$, the “optimization success” and the track list RobotTrackList are outputted.

If $i \neq N$, $\text{RecurseSearch}()$ does not successfully complete a search. “Optimization failure” is outputted, and the track list RobotTrackList is cleared.

5.2. Simulation Analysis and Grinding Experiment of V-Shaped Workpiece

5.2.1. Simulation Analysis of V-Shaped Workpiece. The optimal robotic grinding path search algorithm is used to generate the robotic grinding path of the V-shaped workpiece shown in Figure 4. The cutter points in Figure 5 are imported into the algorithm. w is set to 50 mm, D is set to 45

```

Extract the last node structure in the RobotTrackList, and get the variable  $i$  of this structure.
IF ( $i = N$ )
    The track search is complete. The search function is terminated and returns to the function call point.
ELSE
    Traverse all nodes of the  $(i + 1)^{th}$  column of the NodeMatrix. The total joint variation  $\Delta\theta$  of the robot grinding path is calculated
    when these nodes act as  $Node_{i+1}$ . The nodes of the  $(i + 1)^{th}$  column of NodeMatrix are sorted in ascending order of  $\Delta\theta$ .
    A variable  $k$  is created, and  $k = 1$ .
    For  $k = 1$  to  $M$ 
        The  $k^{th}$  node of the  $(i + 1)^{th}$  column of NodeMatrix is searched. Make this node  $Node_{i+1}$ , and the intermediate nodes are
        generated. Then, collision is detected for  $Node_{i+1}$  and the intermediate nodes.
        IF (no collision occurs)
            The search function for this round is complete. Insert the  $Node_{i+1}$  into the RobotTrackList at the end. Exit this function
            and call for the next round of search.
        ELSE
            IF ( $k < M$ )
                 $Node_{i+1}$  and the intermediate nodes in the RobotTrackList are abandoned.
            ELSE
                The algorithm cannot search the proper node from the  $(i + 1)^{th}$  column of NodeMatrix.  $Node_i$ ,  $Node_{i+1}$ , and the
                corresponding intermediate nodes of two nodes in the RobotTrackList are abandoned. The  $i^{th}$  layer of the search function is
                terminated and returns to the  $(i - 1)^{th}$  layer of the search function.
        End
    End
End
End

```

ALGORITHM 1: Recurse search function (RobotTrackList).

degrees, M is set to 140, and ΔT is set to 1. ${}^B_{G_0}T$ is set as the matrix in equation (4). ${}^6_T T$ is set as the matrix in equation (5). The robot track for grinding this V-shaped workpiece is obtained using the proposed algorithm and is shown in Figure 23.

The joint variation of the track in Figure 6 and that of the track in Figure 23 are indicated in Table 2. Comparing these two tracks shows that the maximal joint variation of the track obtained by the optimal algorithm decreases from 2.8134 rad to 1.067 rad.

The coordinate plane of this optimized track is illustrated in Figure 24. Figure 24 shows that no collision occurs in the track. The collision area can be obtained by traversing every structure in NodeMatrix and using the collision detection function of the V-REP. Because collision must be detected for every structure in NodeMatrix, the process is time-consuming in the generation of the collision area. The algorithm spends 3528 s generating the collision area in Figure 24. When the optimal robotic grinding path search algorithm is used to generate the robotic grinding path, there is no need for the collision area. Collision can only be detected for the searched node and the intermediate nodes in Step (3) of the RecurseSearch() function, as detailed in Section 4.3. The algorithm takes only 38.91 s to generate the robotic grinding path of the V-shaped workpiece.

5.2.2. Grinding Experiment of V-Shaped Workpiece. A V-shaped workpiece is ground by the optimal grinding path generated in Section 5.2.1. The effect after grinding is shown in Figure 25.

A three-coordinate measurement instrument is used to sample the surface profile of Figure 25(b), as shown in

TABLE 1: List of input parameters.

Parameter	The meaning of the parameter
w	Width of belt wheel (unit: mm)
D	The range of rotation area on the belt wheel (unit: degree)
M	Dispersion quantity of variable t (unit: 1)
ΔT	Maximal difference between two adjacent nodes (unit: 1)
${}^B_{G_0}T$	Pose matrix of initial tool coordinate system $\{G_0\}$ to the base coordinate system $\{B\}$
${}^6_T T$	Pose matrix of workpiece coordinate system $\{T\}$ to the coordinate system $\{6\}$

Figure 26. The precision of the instrument is $2\ \mu\text{m}$, and the instrument uses its probe for sampling. The probe of the instrument carries out 5 parallel scans in the thickness direction of the workpiece. The sampling points are plotted in the 3D coordinate system, as shown in Figure 27.

Given that the i^{th} sampling point is P_i . Then, the distance between P_i and the ideal profile curved surface is δ_i . δ_i is defined as the shape error at each sampling point. Taking the ordinal number of the sampling points as the X-axis and the shape error of each sampling point as the Y-axis, the error distribution map of the sampling points can be obtained as shown in Figure 28. The maximum, mean, and standard deviation of the errors are shown in Table 3.

The bow error is set to 0.05 mm, but the mean error and the maximum error actual measured are 0.094 mm and 0.236 mm. The measured value and the set value vary for the following reasons: First, the belt wheel is nonrigid, as it is made of rubber. When the belt wheel contacts the workpiece, slight deformation occurs. Second, the open chain

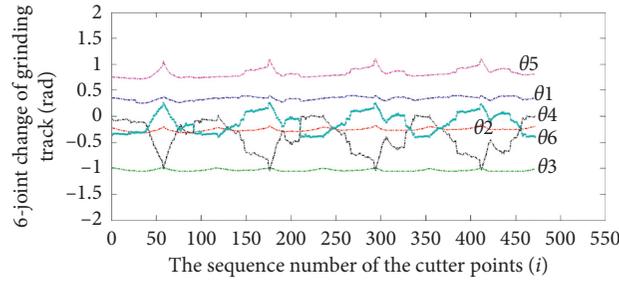


FIGURE 23: Optimal robot grinding path of V-shaped workpiece.

structure of the robot is not very rigid and deforms during grinding. Third, the tool coordinate system features a calibration error.

The curved surface is divided into several areas, as shown in Figure 29. The roughness instrument measures each area and records the measured results. The measuring range of the instrument is Ra0.025 to Ra16.

The measured results are shown in Table 4. The mean roughness of the surface is $1.151 \mu\text{m}$, and the standard deviation is $0.110 \mu\text{m}$. The roughness mean values in the X and Y directions are approximate, and the maximum standard deviation is only $0.141 \mu\text{m}$. This finding indicates that the processing quality of the surface is consistent.

The experimental results show that the robot grinding path generated by the algorithm can ensure surface quality.

5.3. Simulation Analysis and Grinding Experiment of S-Shaped Workpiece

5.3.1. Simulation Analysis of S-Shaped Workpiece. The collision avoidance capability of the algorithm can be validated using an S-shaped workpiece. The dimensions of the S-shaped workpiece are shown in Figure 30. The spacing of the cutter path is set to 5 mm. The bow height error between the adjacent two cutter points is set to 0.05 mm. The cutter points of this S-shaped workpiece are generated as shown in Figure 31, and the coordinate system $\{M\}$ is established on every cutter point.

The cutter points in Figure 31 are imported into the optimal robotic grinding path search algorithm. w is set to 50 mm, D is set to 45 degrees, M is set to 140, and ΔT is set to 1. ${}^B_{G_0}T$ is set as the matrix in equation (4). ${}^6_T T$ is set as the following matrix:

$${}^6_T T = \begin{bmatrix} 0.9239 & 0.3827 & 0 & 18.5084 \\ -0.3827 & 0.9239 & 0 & -7.7414 \\ 0 & 0.0001 & 1 & 106.9650 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (17)$$

The algorithm takes 36.07 s to generate the robotic grinding path of the S-shaped workpiece. The robotic grinding path of the S-shaped workpiece is shown in Figure 32. The joint changes of the grinding path are shown in Table 5.

The coordinate plane of the track is illustrated in Figure 33. The algorithm spends 3036 s generating the collision area. Figure 33 shows no collision in the track.

5.3.2. Grinding Experiment of S-Shaped Workpiece. An S-shaped workpiece is ground by the optimal grinding path generated in Section 5.3.1. The effect after grinding is shown in Figure 34.

A three-coordinate measurement instrument is used to sample the surface profile of Figure 34(b). The sampling points are plotted in the 3D coordinate system, as shown in Figure 35.

The error distribution map of the sampling points is shown in Figure 36. The maximum, mean, and standard deviation of the errors are shown in Table 6.

The mean error and the maximum error actual measured are 0.081 mm and 0.202 mm, which are equivalent to the shape error of the V-shaped workpiece.

The curved surface is divided into several areas, as shown in Figure 37. The roughness instrument measures each area and records the measured results. The measured results are shown in Table 7.

The mean roughness of the surface is $1.215 \mu\text{m}$, and the standard deviation is $0.152 \mu\text{m}$. The roughness mean values in the X and Y directions are approximate, and the maximum standard deviation is $0.221 \mu\text{m}$. This finding indicates that the processing quality of the surface is consistent.

The experimental results show that the robot grinding path generated by the algorithm can ensure surface quality.

6. Discussion and Conclusions

To obtain a robotic grinding path with a small joint angular variation and no collision, the robot optimal grinding path search algorithm is proposed. The algorithm optimizes the grinding pose of the robot by adjusting the position of the tool coordinate system on the grinding wheel. First, the coordinate plane is established to describe the grinding path. Second, the ordinate value of the coordinate plane is dispersed to obtain the search field of the algorithm. Third, the single-step search process of the algorithm is proposed. Finally, based on the single-step search process and the recursive method, the optimal robotic grinding path search algorithm is obtained.

TABLE 2: Joint variation of the robotic grinding path before and after optimization (rad).

	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_4$	$\Delta\theta_5$	$\Delta\theta_6$	Max $\Delta\theta$
Before optimization	0.4711	0.3512	0.2951	2.8134	0.8810	1.5024	2.8134
After optimization	0.1508	0.1678	0.0881	1.0657	0.3917	0.6646	1.0657

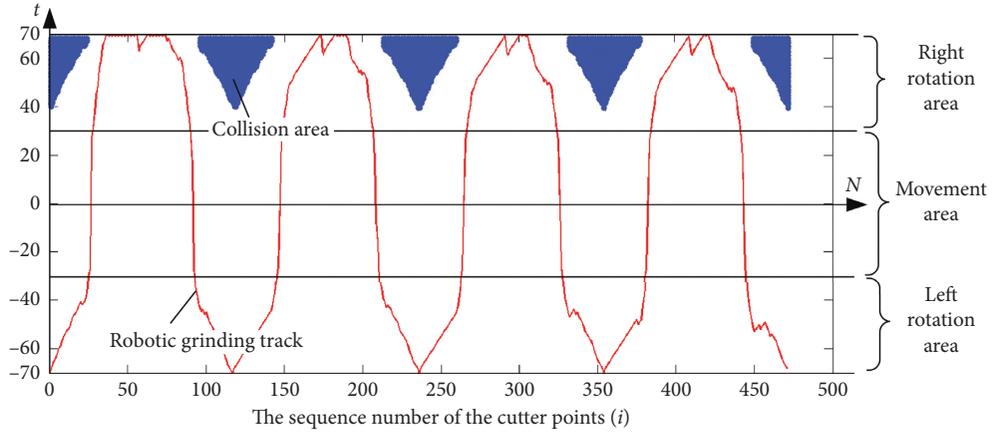


FIGURE 24: Coordinate plane description of the optimal grinding track of V-shaped workpiece.

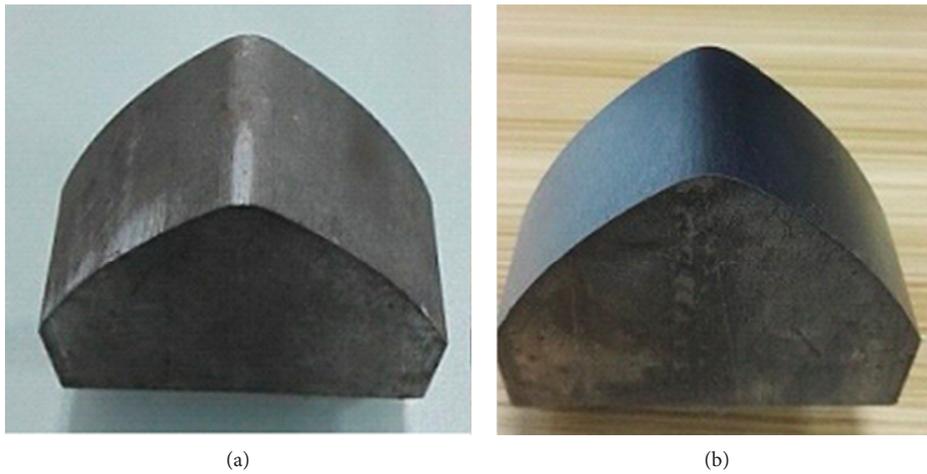


FIGURE 25: V-shaped workpiece. (a) Before grinding. (b) After grinding.

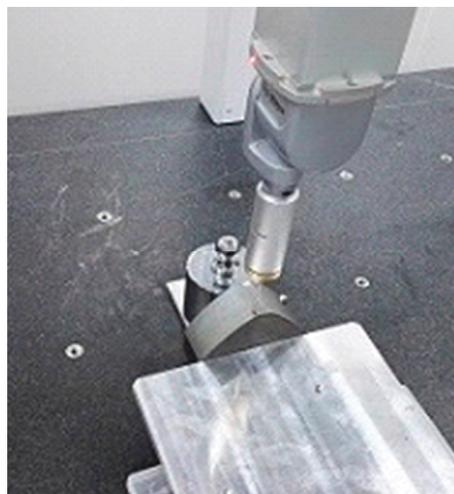


FIGURE 26: Surface sampling of workpiece.

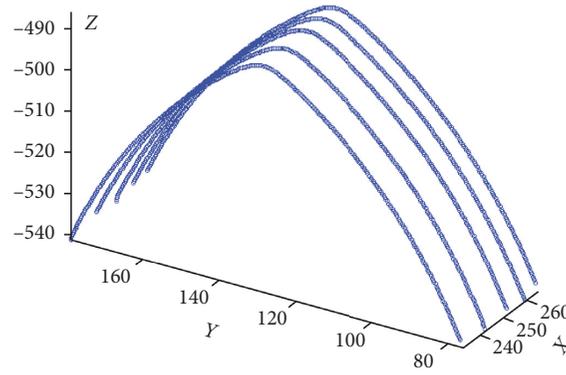


FIGURE 27: Sampling points.

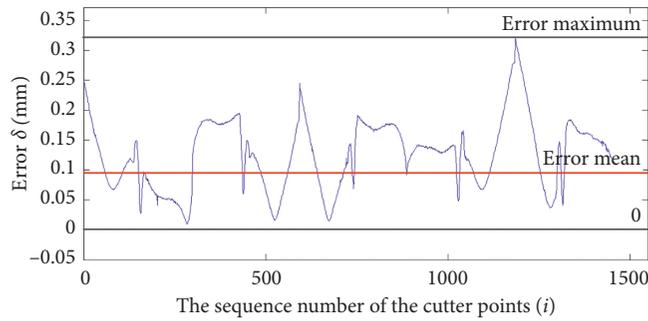


FIGURE 28: Error distribution map of the sampling points.

TABLE 3: Maximum, mean, and standard deviation of shape error of sampling points (mm).

Error mean	Error standard deviation	Error maximum
0.094	0.043	0.236

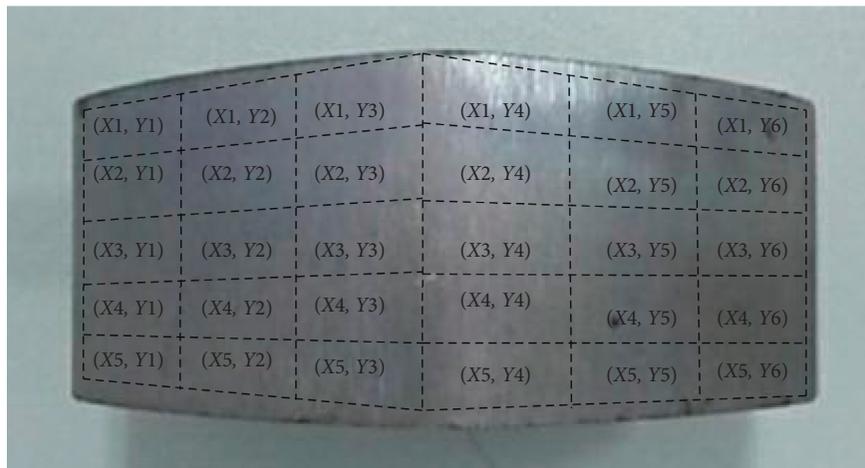


FIGURE 29: Curved surface division.

The optimal robotic grinding path search algorithm is implemented on the V-REP platform. The algorithm is used to generate the grinding path of a V-shaped workpiece and an S-shaped workpiece. Then, the generated

grinding path is used to carry out simulations and grinding experiments of the two parts. The simulation and experimental results demonstrate the feasibility of the algorithm.

TABLE 5: Joint variation of the grinding path of S-shaped workpiece (rad).

	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_4$	$\Delta\theta_5$	$\Delta\theta_6$	Max $\Delta\theta$
Before optimization	0.1099	0.2345	0.2446	0.9062	0.7344	0.2375	0.9062

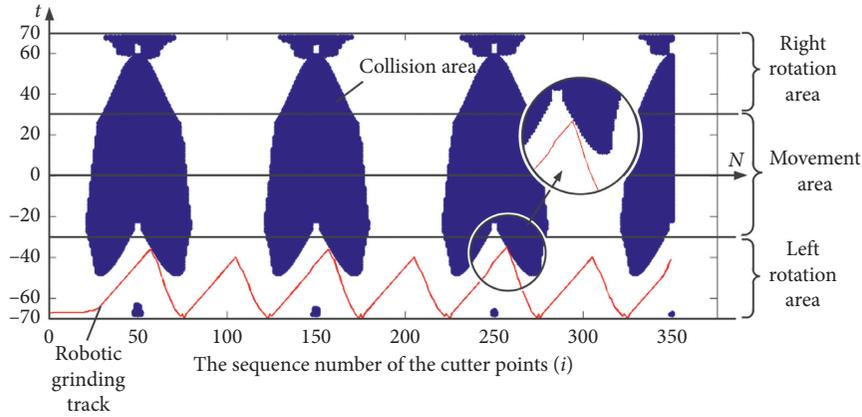


FIGURE 33: Coordinate plane description of the grinding track of S-shaped workpiece.

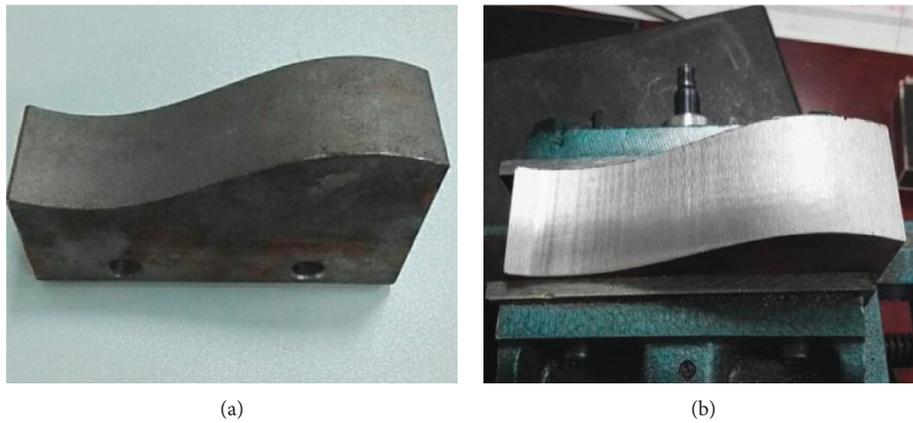


FIGURE 34: S-shaped workpiece.(a) Before grinding. (b) After grinding.

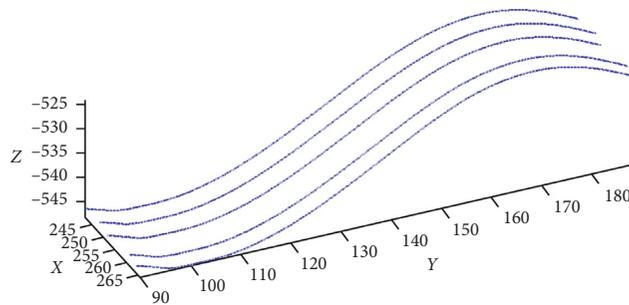


FIGURE 35: Sampling points.

Referring to the modeling method in the author’s previous paper [26], this paper improves the algorithm of robot grinding trajectory calculation in [26], which not only meets the requirements of collision but also achieves

the grinding trajectory with small total variation of joint angle, thereby improving the smoothness of the robot movement process, improving the grinding efficiency, and reducing the cost.

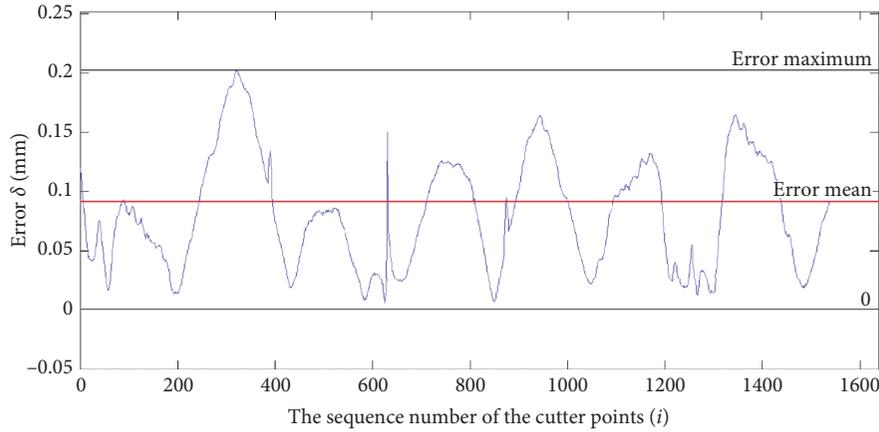


FIGURE 36: Error distribution map of the sampling points.

TABLE 6: Maximum, mean, and standard deviation of shape error of sampling points (mm).

Error mean	Error standard deviation	Error maximum
0.081	0.047	0.202

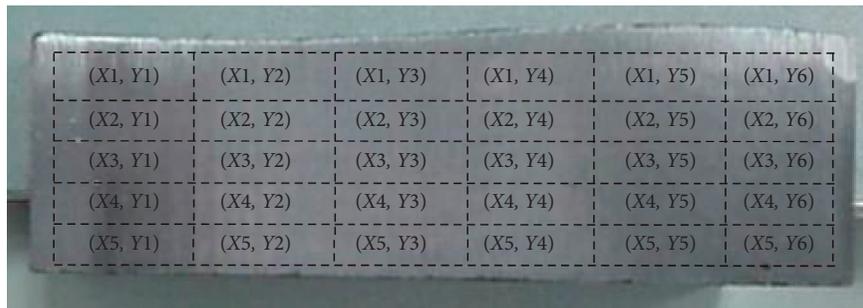


FIGURE 37: Curved surface division.

TABLE 7: Surface roughness of S-shaped workpiece.

	Y1	Y2	Y3	Y4	Y5	Y6	Mean	Std.
X1	1.053	1.434	1.215	1.068	1.426	1.038	1.206	0.185
X2	1.480	1.042	1.455	1.434	1.311	1.120	1.307	0.186
X3	1.002	1.199	1.090	1.289	1.175	1.067	1.137	0.104
X4	1.387	1.129	1.131	1.274	1.256	1.092	1.212	0.113
X5	1.408	1.400	1.072	1.072	1.200	1.120	1.212	0.156
Mean	1.266	1.241	1.193	1.227	1.274	1.087	1.215	
Std.	0.221	0.171	0.157	0.157	0.100	0.035		0.152

However, the proposed algorithm still has some limitations. Because the algorithm is conditioned by the minimum angle change when selecting the value of the variable “ t ,” the final robot path is the local optimal solution with the minimum angle change. As for the search method of global optimal solution, further research is needed. In addition, the surface roughness of the workpiece is related to the force in the process of robot grinding. In this paper, the measurement of surface quality is mainly to show that the robot path obtained by this method can ensure that there is no collision, the angle change is small, and the

quality will not deteriorate. As for how to improve the grinding quality, we will do further research about force control method in the future.

Data Availability

No data were used to support this study.

Conflicts of Interest

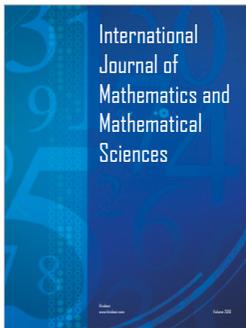
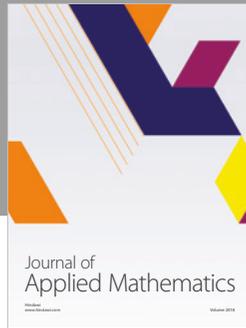
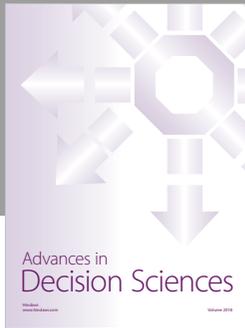
The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Science and Technology Major Project of China (grant number 20152X04005006); Science and Technology Planning Project of Guangdong Province, China (grant numbers 2014B090921004, 2014B090920002, and 2015B010918002); and Science and Technology Major Project of Zhongshan, China (grant number 2016F2FC0006).

References

- [1] M. J. Tsai, J. J. Fang, and J. L. Chang, "Robotic path planning for an automatic mold polishing system," *International Journal of Robotics and Automation*, vol. 19, no. 2, pp. 81–90, 2004.
- [2] L. Basanez and J. Rosell, "Robotic polishing systems," *IEEE Robotics and Automation Magazine*, vol. 12, no. 3, pp. 35–43, 2005.
- [3] Y. Zhao, J. Zhao, L. Zhang, L. Qi, and Q. Tang, "Path planning for automatic robotic blade grinding," in *Proceedings of the 2009 International Conference on Mechatronics and Automation (ICMA)*, pp. 1556–1560, IEEE, Changchun, China, August 2009.
- [4] W. Wang and C. Yun, "A path planning method for robotic belt surface grinding," *Chinese Journal of Aeronautics*, vol. 24, no. 4, pp. 520–526, 2011.
- [5] A. Kharidege, D. T. Ting, and Z. Yajun, "A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot," *International Journal of Advanced Manufacturing Technology*, vol. 19, no. 9–12, pp. 3921–3934, 2017.
- [6] J. Liu, X. Huang, S. Fang, H. Chen, and N. Xi, "Industrial robot path planning for polishing applications," in *Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1764–1769, IEEE, Qingdao, China, December 2016.
- [7] T. Lozano-Perez, "Automatic planning of manipulator transfer movements," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 10, pp. 681–698, 1981.
- [8] T. Lozano-Perez, "Spatial planning: a configuration space approach," *IEEE Transactions on Computers*, vol. C-32, no. 2, pp. 108–120, 1983.
- [9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [10] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: theory and experiments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1423–1436, 1990.
- [11] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.
- [12] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 23–32, 1992.
- [13] J. J. Kuffner and S. M. Lavelle, "RRT-connect: An efficient approach to single-query path planning," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 995–1001, IEEE, San Francisco, CA, USA, 2000.
- [14] M. Shibata, T. Murakami, and K. Ohnishi, "Fast collision-free track planning for robot manipulator based on distortional configuration space," *IEEE Transactions on Sensors and Micromachines*, vol. 115, no. 12, pp. 1476–1483, 2008.
- [15] G. Li, A. Yamashita, H. Asama et al., "An efficient improved artificial potential field based regression search method for robot path planning," in *Proceedings of 2012 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1227–1232, IEEE, Chengdu, China, August 2012.
- [16] T. Tsuji, P. G. Morasso, and M. Kaneko, "Track generation for manipulators based on artificial potential field approach with adjustable temporal behavior," in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 438–443, IEEE, Osaka, Japan, November 1996.
- [17] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [18] S. A. Masoud and A. A. Masoud, "Constrained motion control using vector potential fields," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 3, pp. 251–272, 2000.
- [19] A. A. Masoud, "Decentralized self-organizing potential field-based control for individually motivated mobile agents in a cluttered environment: a vector-harmonic potential field approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 3, pp. 372–390, 2007.
- [20] P. Bosscher and D. Hedman, "Real-time collision avoidance algorithm for robotic manipulators," in *Proceedings of the 2009 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 113–122, IEEE, Woburn, MA, USA, November 2009.
- [21] E. Bayro-Corrochano, O. Carbajal-Espinosa, and A. Loukianov, "Applications of potential fields and conformal geometric algebra for humanoid manipulation maneuvering," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1378–1383, IEEE, Taipei, China, October 2010.
- [22] P. W. Wu, Y. C. Lin, C. M. Wang et al., "Grasping the object with collision avoidance of wheeled mobile manipulator in dynamic environments," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, pp. 5300–5305, Tokyo, Japan, November 2013.
- [23] F. Gómez-Bravo, G. Carbone, and J. C. Fortes, "Collision free trajectory planning for hybrid manipulators," *Mechatronics*, vol. 22, no. 6, pp. 836–851, 2012.
- [24] E. J. S. Pires, P. B. de Moura Oliveira, and J. A. T. Machado, "Manipulator trajectory planning using a MOEA," *Applied Soft Computing*, vol. 7, no. 3, pp. 659–667, 2007.
- [25] M. H. Woo, S.-H. Lee, and H. M. Cha, "A study on the optimal route design considering time of mobile robot using recurrent neural network and reinforcement learning," *Journal of Mechanical Science and Technology*, vol. 32, no. 10, pp. 4933–4939, 2018.
- [26] T. Zhang and J. Su, "Collision-free planning algorithm of motion path for the robot belt grinding system," *International Journal of Advanced Robotic Systems*, vol. 15, no. 4, pp. 1–13, 2018.
- [27] L. Piegl, "On NURBS: a survey," *IEEE Computer Graphics and Applications*, vol. 11, no. 1, pp. 55–71, 1991.



Hindawi

Submit your manuscripts at
www.hindawi.com

