

Research Article

Solving Inverse Kinematics of the Shotcrete Manipulator Based on the Plane Two-Link Model and Trajectory Planning

Chunlei Luo,¹ Yi He ,¹ and Shitu Abubakar^{1,2}

¹College of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China

²Department of Mechanical Engineering, Ahmadu Bello University, Zaria 810107, Nigeria

Correspondence should be addressed to Yi He; hycmee@163.com

Received 28 May 2020; Revised 4 August 2020; Accepted 14 August 2020; Published 1 September 2020

Academic Editor: Keigo Watanabe

Copyright © 2020 Chunlei Luo 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.

Traditionally, the construction of tunnel support is to manually move the manipulator through the remote control. During the manual construction process, there is a large rebound rate and a reduction in the compactness of the concrete. Skilled operators can obtain better shot blasting quality, but training of a skilled worker is costly. In this study, an automatic shotcrete manipulator is proposed to complete spraying and improve the spraying quality. The model of the manipulator is simplified into a planar two-link manipulator, solved the inverse kinematics of the shotcrete manipulator, and implemented the automatic trajectory planning process. By simplifying the complex mechanical arm structure model into a simple planar two-link manipulator model, the inverse kinematics of the shotcrete manipulator is solved, and automatic trajectory planning software is developed, which can better help construction and reduce injury to personnel's health and can also help the construction party save materials and improve economic efficiency and construction efficiency.

1. Introduction

Due to the large number of tunnel excavations and constructions in the construction of subways, high-speed rail, trains, and highways, the use of concrete wet spray machines is required to complete the tunnel support. The State Council of China had issued and implemented the “13th Five-Year Plan for the Development of Modern Comprehensive Transportation System” on February 3, 2017, see Table 1 for related indicators. The manual hand-held shotcrete or remote-controlled robotic arm shotcrete is commonly used in the current construction industry, and the quality (rebound rate and thickness) and efficiency of the construction depend on the technical ability of the workers. Often, manual construction will cause a large rebound rate. Thus, there is a lot of waste in the tunnel construction, which increases the construction cost [1]. The intensive labor resulted in workers' fatigue, which cannot guarantee high construction efficiency. In addition, workers who work in dusty environments for a long time (see Figure 1) are prone to occupational hazards [2].

Many algorithms have been proposed for the automatic control of industrial robotic arms, involving inverse

kinematics, trajectory planning, motion planning, force control, and so on. Some general 6-degree-of-freedom inverse kinematics can solve the inverse solutions of some robotic arms that meet the Pieper principle [3]. Because inverse kinematics requires tedious matrix calculation processes and clever operation skills, a symbolic inverse kinematics solver is proposed [4]. For situations where formula methods cannot be used to solve inverse kinematics, numerical methods can be used [5]. Different trajectory planning methods are proposed to enable the robotic arm to produce continuous motion with excellent kinematic performance [6–8]. However, for the automatic control of the shotcrete manipulator, other aspects need to be considered.

Many researchers have studied the remote operation mode of the shotcrete manipulator, data acquisition during shotcrete, simulators, etc. These studies can help workers perform shotcrete [9–11]. Different methods have been proposed to automate the shotcrete process [12–14]. Girmscheid and Moser demonstrated fully automatic shotcrete robot (Meyco). The manipulator has 8 degrees of freedom. Due to the complex mechanical structure of the shotcrete manipulator, a closed solution cannot be obtained.

TABLE 1: Main indicators for the development of comprehensive transportation during the “13th Five-Year Plan.”

Operating mileage	2015 (year)	2020 (year)	Increase
Railway operating mileage (10,000 km)	12.1	15	2.9
High-speed railway operating mileage (10,000 km)	1.9	3	1.1
Highway operating mileage (10,000 km)	458	500	42
Expressway operating mileage (10,000 km)	12.4	15	2.6
Operating mileage of urban rail transit (km)	3300	6000	2700



FIGURE 1: Construction site.

Therefore, it can only be solved iteratively. In this study, they set up three different modes for the shotcrete manipulator to adapt to different tunnel surface environments after blasting, including the manual mode, semiautomatic mode, and fully automatic mode. The blasting area is divided differently, and mobile spraying is performed at different speeds to ensure the designed thickness. Nabulsi et al. [13] also studied Putzmeister PM-407. This manipulator has 5 degrees of freedom. It uses incoupling and iterative techniques to solve inverse kinematics. When normal vectors cannot be extracted correctly, the surface of the design tunnel is used directly to calculate the normal vector. In such studies, a large number of matrix operations are used to solve inverse kinematics, and only numerical solutions can be provided.

The objective of this research is to model and develop an analytical solution of the shotcrete trolley of Henan Gengli Engineering Equipment Company. The manipulator has 7 degrees of freedom. By reducing the complex structure model of the robot arm to a simple planar two-link manipulator model, the inverse kinematics can be solved based on the analytical solution. Based on this, the motion trajectory of shotcrete was planned, and the control platform was developed. The structure of this paper is as follows. Section 2 studies the inverse kinematics of shotcrete manipulators in combination with shotcrete motion and presents theoretical relationships. Section 3 explains how to perform effective trajectory planning and analyzes the sudden changes in joint angles in motion. Section 4 develops a robot control platform based on the derived inverse kinematics formula and trajectory planning and verifies the correctness of the analytical solution and the rationality of trajectory planning. Section 5 presents the conclusions.

2. Inverse Kinematics

The shotcrete manipulator depicted in Figure 2 has 8 degrees of freedom, and the end nozzle has one degree of freedom designed for the craft, and the movement mode is to rotate

around the axis to ensure the uniformity of shotcrete. This degree of freedom is not in the control system of the manipulator. Also, there are two moving joints in the same direction. Since these two moving joints control the displacement in the same direction, they are regarded as one moving joint in the inverse kinematics model. In order to reduce the rebound of the concrete as much as possible and improve the uniformity and compactness of the shotcrete, it is necessary to keep the nozzle and the sprayed surface vertical and, at the same time, keep the nozzle and the sprayed surface at a distance of 1.5 to 2 meters [15]. The principle of the structural design of the shotcrete manipulator is to facilitate the operation of the worker, and the automatic control of the manipulator is not considered. Thus, the inverse kinematics of the shotcrete manipulator may not meet the design principles of the industrial manipulator. For example, the axes of the last three joints intersect at a point so that the inverse kinematics of the shotcrete robotic arm has a closed solution. Although the axis of the shotcrete robot that we studied meets the intersection of the axes of the last three joints, it introduces redundant degrees of freedom when controlling the position of the nozzle.

2.1. Kinematics Model of the Shotcrete Manipulator. The Denavit and Hartenberg (D-H) modeling method was used to construct the coordinate system of the shotcrete manipulator as shown in Figure 3, the D-H parameters are given in Table 2, and the added nozzle coordinate system describes the target position and the sprayed surface angular relationship and distance. The two moving joints are merged into one joint and described by coordinate system 4.

2.2. Inverse Kinematics Solution. In the inverse kinematics solution, the coordinate system e of the nozzle is not considered, which simplifies the solution process. Forward kinematics can be described as

$${}^0T = {}^0T(\theta_1) {}^1T(\theta_2) {}^2T(\theta_3) {}^3T(d_4) {}^4T(\theta_5) {}^5T(\theta_6)$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}, \quad (1)$$

where

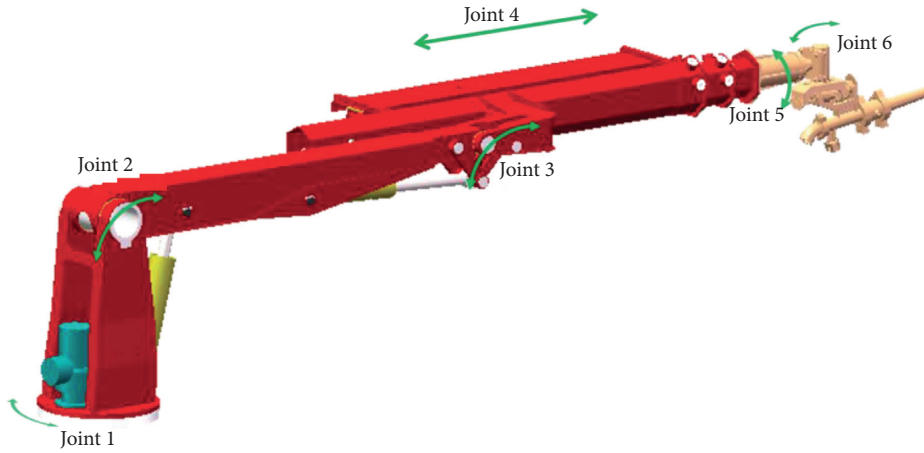


FIGURE 2: Structural model of the shotcrete manipulator.

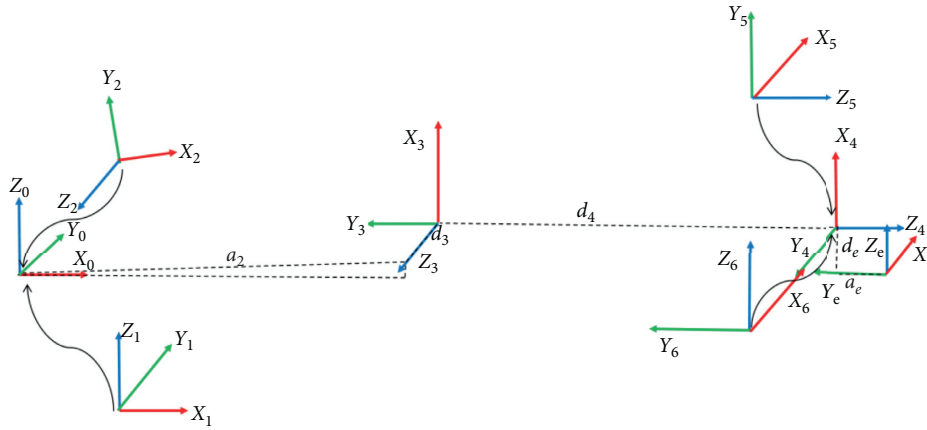


FIGURE 3: Kinematic model of the shotcrete manipulator.

TABLE 2: D-H parameters for the shotcrete manipulator.

Joint no.	α_{i-1} (°)	a_{i-1} (mm)	d_i (mm)	θ_i (°)	Angle limit (°/mm)
1	0	0	0	θ_1	-180~180
2	90	0	0	θ_2	0~70
3	0	a_2	$-d_3$	θ_3	-20~110
4	-90	0	d_4	0	0~4600
5	0	0	0	θ_5	-180~180
6	90	0	0	θ_6	-120~120
7	0	a_e	d_e	0	-

The initial values are $\theta_1 = 0$, $\theta_2 = 1.5500$, $\theta_3 = 88.4500$, $d_4 = 3406.0$, $\theta_5 = -90$, and $\theta_6 = 0$. The constant values are $a_2 = 3272.0$, $d_3 = 452.5$, $a_e = 442.0$, and $d_e = 398.1$.

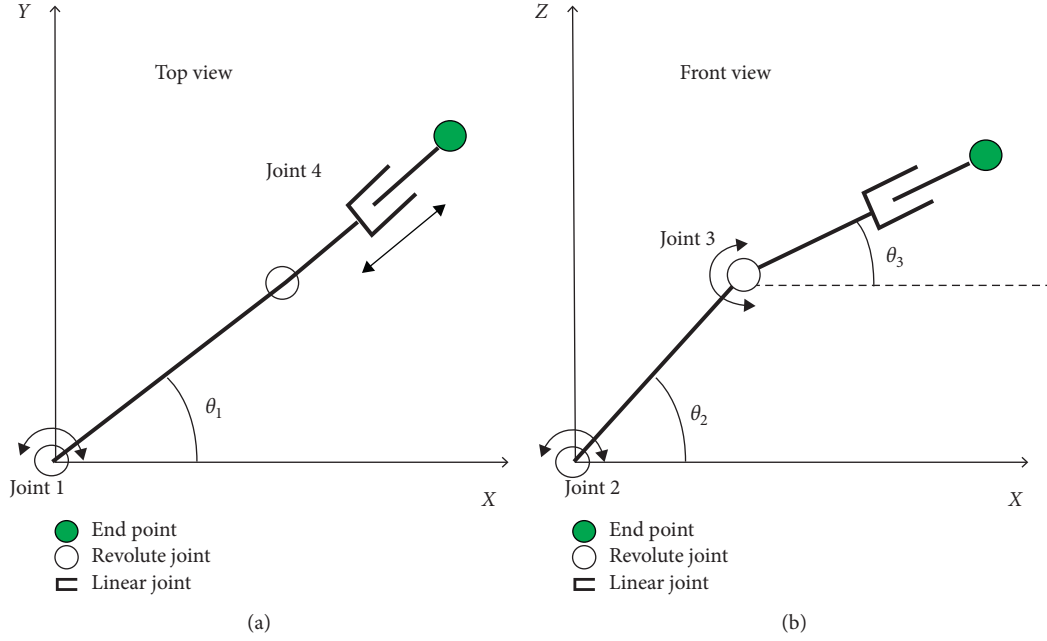


FIGURE 4: Simplified manipulator model 1: (a) top view; (b) front view.

$$\begin{aligned}
 {}^0_1T(\theta_1) &= \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^1_2T(\theta_2) &= \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^2_3T(\theta_3) &= \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^3_4T(d_4) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^4_5T(\theta_5) &= \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^5_6T(\theta_6) &= \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{2}$$

The vector P is a 3×1 position vector, R is a 3×3 rotation matrix, and the first, second, and third columns are orthogonal vectors to each other. According to the structure of

the manipulator, it is known that the front 4 joints determine the end position, but due to the redundant degrees of freedom, there are 3 known parameters here, and 3 equations can be established, but there are four unknowns, which are not directly solved according to the formula. The 1 to 4 numbers of joint will be solved by shotcrete trajectory analysis combined with geometric methods, and joint 5 and joint 6 will be solved by separating the variables using equation (1).

The 1 to 4 numbers of the joint of the manipulator (joint 1, joint 2, joint 3, and joint 4) are described as seen in Figure 4. The offset value d_3 of joint 3 is ignored here so that joint 2 and joint 3 can be described as a two-link model. The purpose is to be able to analyze the problem more conveniently. By solving the joint angle value using (P'_x, P'_y, P'_z) under this model, the joint angle value can also be solved in the original coordinate system model. Joint 1 shown in Figure 4 also includes joint 2 at the same time. In order to clearly describe the effect of joints on angle, joint 2 is not expressed, and the same is true in b in Figure 4. The rotation of joint angle 1 can determine the direction of the robot arm, so joint angle 1 can be obtained by the value of the nozzle in the X-Y plane:

$$\theta_1 = A \tan 2(P'_y, P'_x). \tag{3}$$

Separating variables θ_1 to obtain equation (4)

$${}^0_1T^{-1} {}^6_6T = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & P'_x \\ r_{21} & r_{22} & r_{23} & P'_y \\ r_{31} & r_{32} & r_{33} & P'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where $c_1 = \cos \theta_1$ and $s_1 = \sin \theta_1$.

Let (2, 1) and (2, 2) elements on both sides of equation (4) be equal.

$$-c_6s_5 = -s_1r_{11} + c_1r_{21}, \quad (5)$$

$$-s_6s_5 = -s_1r_{12} + c_1r_{22}. \quad (6)$$

Equations (5) and (6) are squared and added, respectively, and θ_5 can be solved:

$$\theta_5 = \arcsin\left(\sqrt{(-s_1r_{11} + c_1r_{21})^2 + (-s_1r_{12} + c_1r_{22})^2}\right). \quad (7)$$

By dividing equations (5) and (6), θ_6 can be solved:

$$\theta_6 = A \tan 2(s_1r_{12} - c_1r_{22}, -s_1r_{11} + c_1r_{21}). \quad (8)$$

When computing θ_6 , $\sin \theta_5 \neq 0$.

The four joint variables determine the position of the tip nozzle and their functional relationship is:

$$P'(P'_x, P'_y, P'_z) = f(\theta_1, \theta_2, \theta_3, d_4). \quad (9)$$

Detailed description of their specific relationship is given through the following formulas:

$$\begin{aligned} P'_x &= \cos \theta_1 [a_2 \cos \theta_2 + d_4 \cos(\theta_2 + \theta_3)], \\ P'_y &= \sin \theta_1 [a_2 \cos \theta_2 + d_4 \cos(\theta_2 + \theta_3)], \\ P'_z &= a_2 \sin \theta_2 + d_4 \sin(\theta_2 + \theta_3). \end{aligned} \quad (10)$$

As explained earlier, only three joint angles are required to determine the position of the end nozzle. There are four joint angles, and redundant degrees of freedom are introduced. The following is an analysis of the motion trajectory of the shotcrete manipulator to add constraints so that the equation can be solved. There are formulas for calculating joint angle 2 and joint angle 3 without adding constraints; however, they cannot really calculate the joint value:

$$\theta_3 = a \cos \left[\frac{(m+n)}{2a_2d_4} \right], \quad (11)$$

where $m = (P'_x)^2 / \cos^2 \theta_1$ and $n = (P'_z)^2 - (a_2^2 + d_4^2)$. θ_1 cannot be equal to -90 degree and 90 degree. Solving the joint 2 angle value is based on the geometric method (see Figure 5):

$$\begin{aligned} \beta &= A \tan 2\left(P'_z, \sqrt{(P'_x)^2 + (P'_y)^2}\right), \\ \alpha &= a \cos \left[\frac{(l + a_2^2 - d_4^2)}{2a_2 \sqrt{l}} \right], \end{aligned} \quad (12)$$

$$\theta_2 = \beta \pm \alpha,$$

where $l = P'_x \times P'_x + P'_y \times P'_y + P'_z \times P'_z$. $\theta_2 = \beta \pm \alpha$.

The moving area of the spray nozzle at the end of the shotcrete manipulator is divided into four pieces (see Figure 6). Due to the limitation of the mechanical structure of the manipulator and the range of motion of the joint, area3 and area4 are areas that the robotic arm cannot reach.

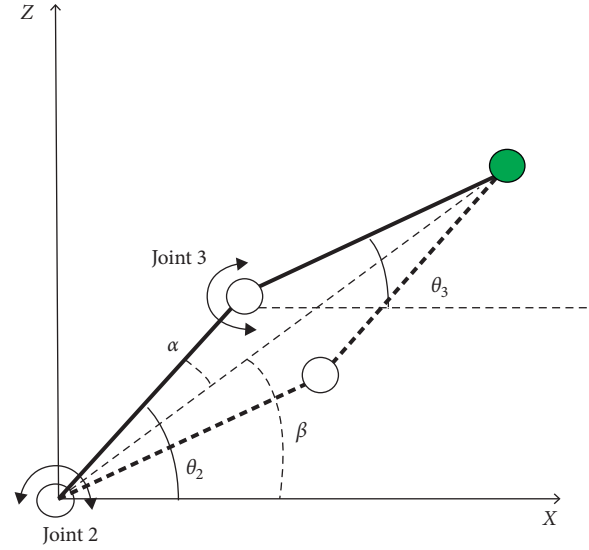


FIGURE 5: Two-link model. α is the angle between link 2 and connection of joint 2 and end point. β is the angle between axis X and connection of joint 2 and end point.

When moving joint 4 is completely retracted, d_4 is 0. Area2 represents the maximum range that the two links can move when d_4 equals to zero. When the nozzle moves in area2, this mode is called mode 1; if we change d_4 , making it more than the zero distance, the manipulator can reach farther area, and when joint 4 moves to another limit position, d_4 equals to the maximum distance. Therefore, area1 implies the movement range of two links when d_4 is greater than zero and less than the maximum distance. When the nozzle moves in area1, this mode is called mode 2. There are five movement statuses; statuses 1, 2, and 5 lie in area1, and all of them mean that d_4 is greater than zero; statuses 3 and 4 lie in area2; both of them also mean that d_4 is equal to zero. If the nozzle moves to area2, then θ_2 and θ_3 can be solved. If the nozzle moves to area1, the nozzle moves from one position to another. The two appropriate ways to solve the problem are changing the value of joint 2 and joint 3 simultaneously and changing the value of joint 3 and joint 4 simultaneously like CCD algorithm ideas [16]. When the shotcrete manipulator sprays the concrete in the tunnel, the manipulator will move to the back joints as fast as possible to complete spraying because the manipulator has a large amount of inertia. When the mode is changed from mode 1 to mode 2, then d_4 is set to zero mile. However, when the mode is changed from mode 2 to mode 1, d_4 will be computed according to the position of joint 3 and the desire position:

$$d_4 = \sqrt{(P_{x(d)} - P_{x(3)})^2 + (P_{y(d)} - P_{y(3)})^2 + (P_{z(d)} - P_{z(3)})^2}, \quad (13)$$

where $P_{x(d)}$, $P_{y(d)}$, and $P_{z(d)}$ are the position components (x, y, z) of the target point and $P_{x(3)}$, $P_{y(3)}$, and $P_{z(3)}$ are the position components (x, y, z) of coordinate 3 ($d_3 = 0$).

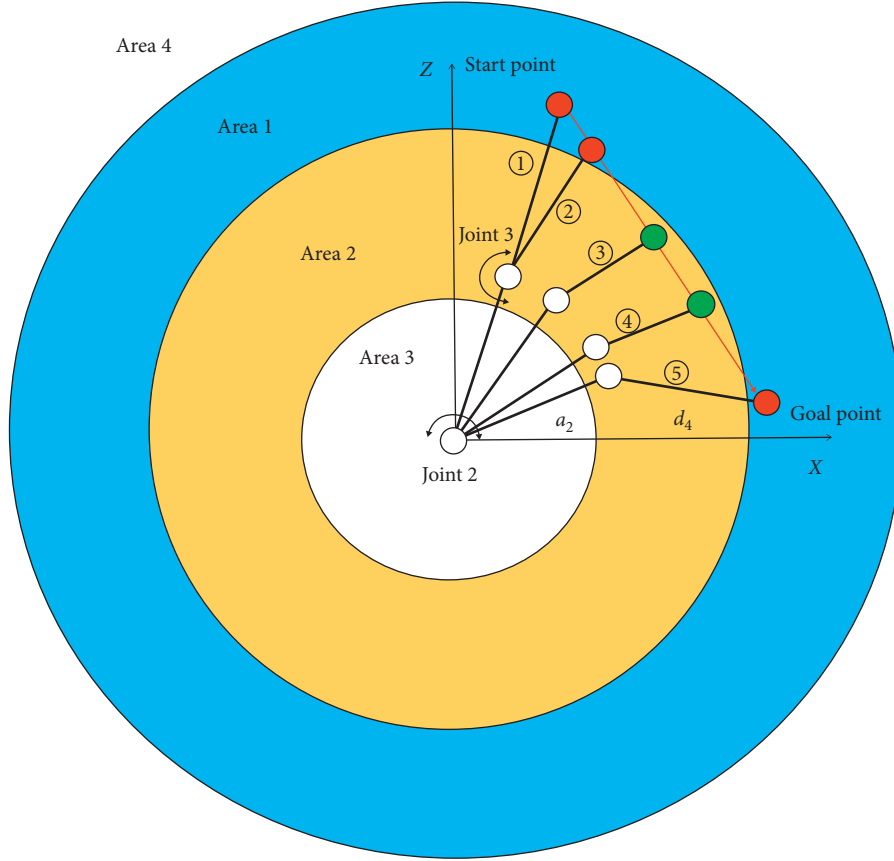


FIGURE 6: Changes in the joint angle during nozzle movement. a_2 is the constant distance, and d_4 is the moving distance of joint 4.

2.3. Kinematic Model considering Offset. In the aforementioned simplified kinematics model, we ignored the offset value d_3 between joint 3 and joint 4. The position value $P'(x, y, z)$ of the end point is different from the actual end point position $P(x, y, z)$. Now, considering the offset value d_3 (see Figure 7), the formula can be used to calculate the virtual position P' from the actual expected position P . So, the inverse kinematics formula derived above can describe the actual joint angle value:

$$\begin{aligned} \phi &= A \tan 2(d_3, \sqrt{P_x \times P_x + P_y \times P_y - d_3 \times d_3}), \\ \phi &= A \tan 2(P_y, P_x) + \phi. \end{aligned} \quad (14)$$

Then, we can compute the virtual desire position ($d_3 = 0$) using the real desire position:

$$\begin{aligned} P'_x &= \sqrt{P_x \times P_x + P_y \times P_y - d_3 \times d_3} \times \cos(\phi), \\ P'_y &= \sqrt{P_x \times P_x + P_y \times P_y - d_3 \times d_3} \times \sin(\phi), \\ P'_z &= P_z, \end{aligned} \quad (15)$$

where P_x, P_y, P_z is the real desire position and P'_x, P'_y, P'_z is the virtual desire position.

3. Trajectory Planning

To improve the performance of the manual operation of the shotcrete manipulator, reasonable trajectory planning of the spraying robotic arm is required. The nozzle needs to maintain a distance of 1.5–2 m from the sprayed surface, and the nozzle needs to be vertical to the sprayed surface. Therefore, the end nozzle needs to control position and attitude parameters to represent its spatial pose. For the position parameters, it needs to control 3 parameters (x, y, z), and the attitude parameters need to control 2 parameters (roll and yaw) (see Figure 8). These parameters are specified relative to the base coordinate system. After blasting, the tunnel surface would have been overexcavated and underexcavated as depicted in Figure 9. The overexcavation part requires more concrete construction than the blasted surface, and the underexcavation part needs further processing to enable shotcrete. In normal blasting, the area is sprayed once during the spraying process (from the tunnel side to the other side). During the spraying process, a thick construction can be completed, and the overdigged part is filled. Also, several shots of this area need to be shot back and forth.

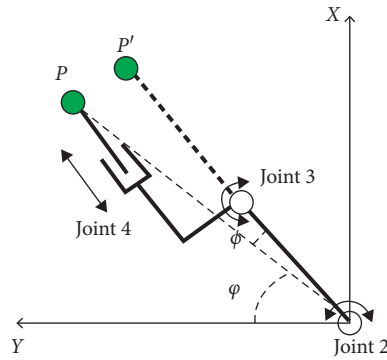


FIGURE 7: Two-link model (d_3 is not equal to zero) in tunnel coordinates. ϕ is the angle between axis Y and connection of joint 2 and end point, and ϕ' is the angle between link 2 and connection of joint 2 and end point.

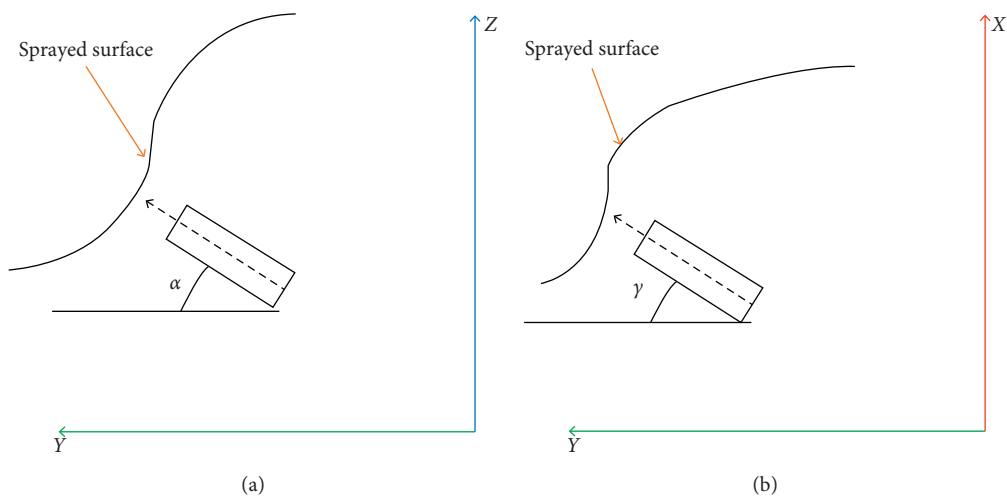


FIGURE 8: Roll (α) and yaw (γ) of the shotcrete manipulator for the nozzle during spraying. X_0, Y_0, Z_0 is the coordinate component of the tunnel. (a) Front view. (b) Top view.

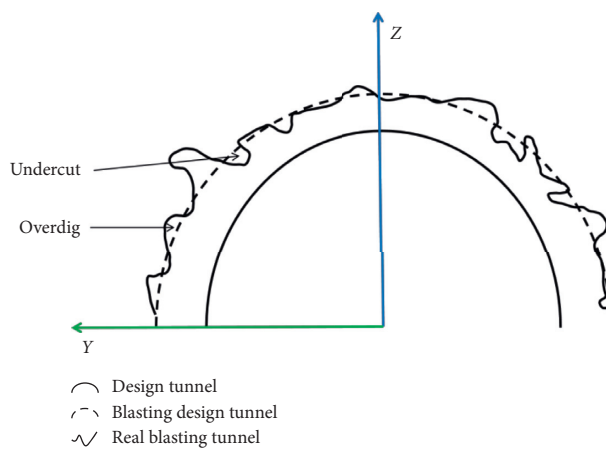


FIGURE 9: Illustration tunnel geometry.

3.1. Trajectory Generation. Tunnel models of various sizes were not considered in this research because environmental sensing such as Lidar is not incorporated. It is assumed that the geometric model of the tunnel is a

semicircle, and the real blasted surface is not considered when trajectory planning is performed. The attitude of the nozzle is calculated by a mathematical model. In general, the trajectory of the nozzle is rectangular, and this

geometry is used in the present study. By manually adjusting the shotcrete manipulator to specify the initial position and the distance between the tunnel trajectories, path length, and tunnel geometry, automatic trajectory planning program can generate a complete shot trajectory (see Figure 10(b)). Figure 10(a) specifically shows the detail in trajectory planning. This trajectory is only a schematic diagram, where d is along the arc direction described in the actual trajectory planning.

3.2. Problems in Shotcrete Manipulator Movement. During the inverse kinematics solution, the movement of some joints is restricted, and artificially added constraints may cause joint values to jump during the movement (see Figure 11). The shotcrete manipulator in Figure 11(a) is working in mode 1. State 0 indicates that joint angle 3 has reached the limit of positive values. When the shotcrete manipulator continues to move upward, the positive value of joint angle 3 is exceeded. If the shotcrete manipulator cannot meet state 1, it will jump to state 2 and cause the joint values of 2 and 3 to change suddenly. The shotcrete manipulator in Figure 11(b) works in mode 2. State 3 indicates joint 3 has reached the limit of positive values. If the shotcrete manipulator continues to move, the values of joints 2 and 3 will also change suddenly. Because in the shot trajectory planning, linear motion is used, this mode is also one of the reasons that causes the joint value to jump. We propose a solution. When joint angle 3 of the shotcrete manipulator approaches the extreme position, it does not move in the area's motion mode or a straight line but changes the motion mode or moves in an arc path or combines the two (see Figure 12); this can prevent the angular value of joint 2 and joint 3 from jumping.

4. Trajectory Planning Simulation

4.1. Verification of Inverse Kinematics. The inverse kinematics solution calculates the current joint angle value based on the previous joint angle vector value and the current motion mode. As it can be seen from Table 3, when the shotcrete robot moves in mode 1, the value of joint 4 is constant, and the movement of joints 2 and 3 and joint 1 enables the tip nozzle to reach the desired position. From Table 4, it can be seen that when the shotcrete manipulator moves in mode 2, the value of joint 2 is constant. The movement of joints 3 and 4 and joint 1 enables the tip nozzle to reach the desired position. From Table 5, we can see that the changes of the joint angle are continuous when switching between mode 1 and mode 2. Therefore, it is verified that the inverse kinematics solution combined with trajectory planning is correct.

A linear motion path containing two motion modes and conversion between modes is selected to verify the proposed inverse kinematics algorithm. Tables 3–5 only provide part of the data on the path to illustrate the correctness of the inverse kinematics algorithm. Solution no. means the

calculated joint angle, and pose no. represents the corresponding end point pose. Where $\text{step} = 0.005 \text{ m}$, the initial position of the movement is

$$\begin{aligned}\theta_1 &= \frac{\pi \times 6}{180}, \\ \theta_2 &= \frac{\pi \times 31}{180}, \\ \theta_3 &= \frac{-\pi \times 12.16}{180}, \\ \theta_4 &= 3.5059, \\ \theta_5 &= \frac{\pi \times 38.38}{180}, \\ \theta_6 &= \frac{\pi \times 57.52}{180}.\end{aligned}\tag{16}$$

The end position is

$$\begin{aligned}\theta_1 &= \frac{\pi \times 0}{180}, \\ \theta_2 &= \frac{\pi \times 3.64}{180}, \\ \theta_3 &= \frac{-\pi \times 6.24}{180}, \\ \theta_4 &= 3.4875, \\ \theta_5 &= \frac{\pi \times 60}{180}, \\ \theta_6 &= \frac{\pi \times 70}{180}.\end{aligned}\tag{17}$$

4.2. Results of Automatic Trajectory Generation. Based on the correct inverse kinematics algorithm and trajectory planning algorithm, we use OpenGL and Qt to develop host computer control software for controlling the shotcrete manipulator (see Figure 13). Simulation is included in control software to verify the correctness of the trajectory. If the parameters set are not reasonable, the parameters can be adjusted to regenerate the trajectory. Control software can monitor the motion status of the shotcrete manipulator in real time displaying the real shotcrete manipulator movement in the simulation graphical interface. Control software provides a friendly human-machine interaction interface that is simple and convenient to operate. The design principle of the control system does not change the original-spraying robot arm as much. Additional motion controller communicates with the original motion controller to achieve the motion control of the shotcrete manipulator.

By setting reasonable set of parameters, control software generates a set of executable trajectories, and simulations in software show this process (see Figure 14).

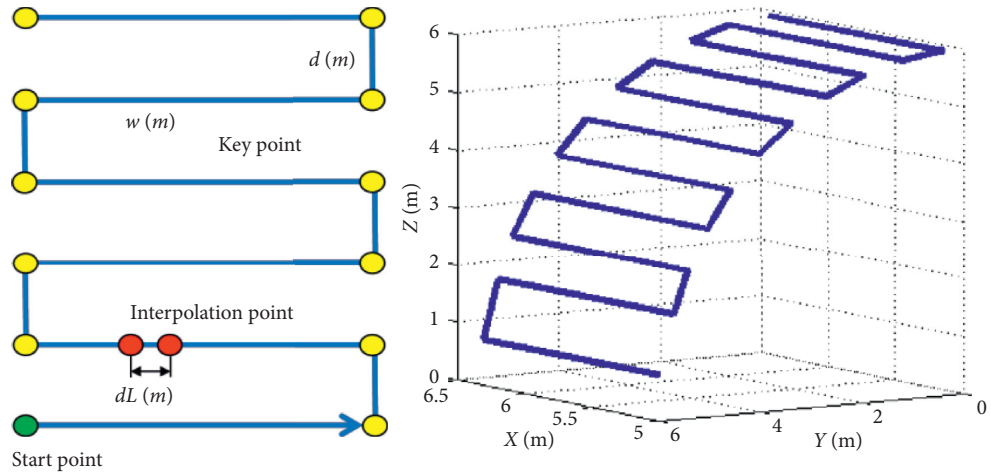


FIGURE 10: Desired shotcrete trajectory.

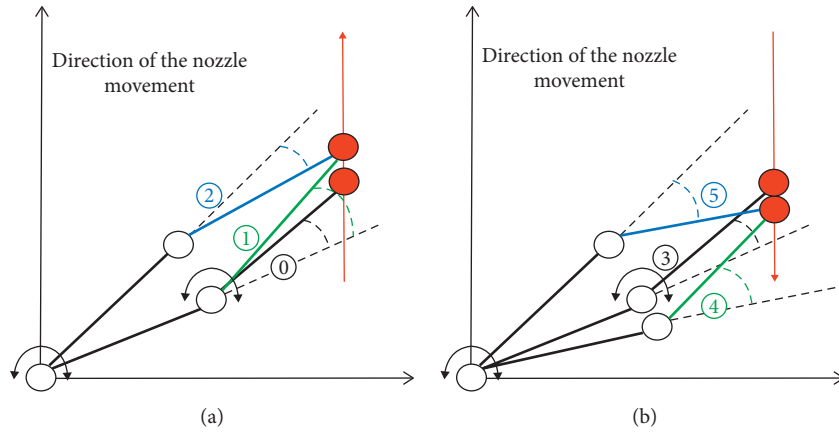


FIGURE 11: The problem of movement of the nozzle: (a) nozzle work in mode 1 and (b) nozzle work in model 2.

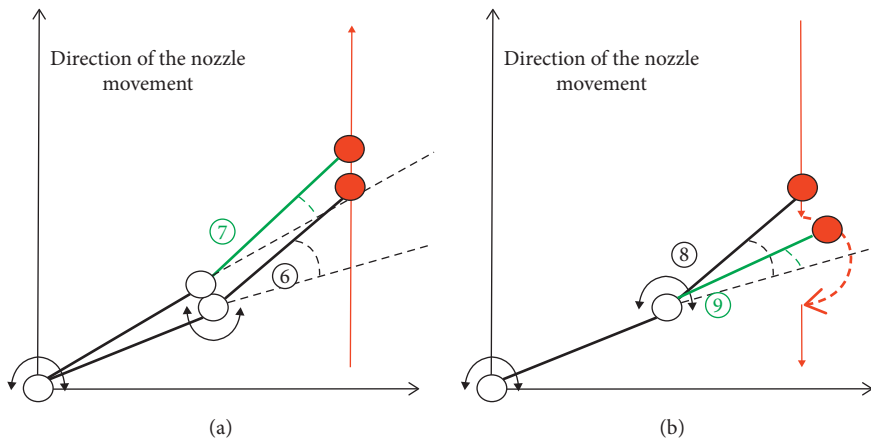


FIGURE 12: Solutions to problems in movement of the nozzle: (a) change mode 1 to mode 2 and (b) change model 2 to mode 1 and move in a circular path.

TABLE 3: Solutions of inverse kinematics in mode 1.

Solution no.	θ_1 (rad)	θ_2 (rad)	θ_3 (rad)	d_4 (m)	θ_5 (rad)	θ_6 (rad)
1	0.080018	0.456693	-0.227912	3.406000	0.757625	1.029767
2	0.079828	0.456685	-0.229316	3.406000	0.758306	1.030003
3	0.079638	0.456667	-0.230703	3.406000	0.758987	1.030239
4	0.079448	0.456642	-0.232074	3.406000	0.759668	1.030475
5	0.079258	0.456607	-0.233430	3.406000	0.760349	1.030712
6	0.079068	0.456566	-0.234769	3.406000	0.761029	1.030949
7	0.078879	0.456516	-0.236094	3.406000	0.761710	1.031188
8	0.078689	0.456458	-0.237403	3.406000	0.762390	1.031426
Pose no.	x (m)	y (m)	z (m)	Roll (rad)	Pitch (rad)	Yaw (rad)
1	6.157767	0.947736	2.294252	-0.773213	-0.150869	2.833495
2	6.159044	0.946654	2.289542	-0.773320	-0.152263	2.833642
3	6.160322	0.945572	2.284829	-0.773430	-0.153649	2.833789
4	6.161599	0.944491	2.280118	-0.773545	-0.155028	2.833936
5	6.162878	0.943409	2.275403	-0.773662	-0.156400	2.834083
6	6.164155	0.942327	2.270694	-0.773784	-0.157764	2.834230
7	6.165432	0.941244	2.265984	-0.773910	-0.159120	2.834378
8	6.166711	0.940163	2.261270	-0.774039	-0.160470	2.834526

TABLE 4: Solutions of inverse kinematics in mode 2.

Solution no.	θ_1 (rad)	θ_2 (rad)	θ_3 (rad)	d_4 (m)	θ_5 (rad)	θ_6 (rad)
1	0.005281	0.046679	-0.031888	3.440453	1.028433	1.200126
2	0.005119	0.046679	-0.033265	3.441585	1.029013	1.200762
3	0.004957	0.046679	-0.034638	3.442723	1.029592	1.201399
4	0.004794	0.046679	-0.036011	3.443869	1.030171	1.202038
5	0.004632	0.046680	-0.037382	3.445021	1.030749	1.202678
6	0.004470	0.046682	-0.038755	3.446179	1.031327	1.203319
7	0.004308	0.046683	-0.040128	3.447345	1.031905	1.203963
8	0.004146	0.046682	-0.041498	3.448517	1.032482	1.204607
Pose no.	x (m)	y (m)	z (m)	Roll (rad)	Pitch (rad)	Yaw (rad)
1	6.700704	0.487896	0.291913	-0.999891	-0.300959	2.949828
2	6.701981	0.486814	0.287199	-1.000311	-0.301886	2.950126
3	6.703259	0.485732	0.282486	-1.000733	-0.302810	2.950425
4	6.704536	0.484650	0.277768	-1.001155	-0.303733	2.950724
5	6.705813	0.483568	0.273064	-1.001579	-0.304650	2.951024
6	6.707091	0.482486	0.268356	-1.002004	-0.305566	2.951324
7	6.708368	0.481404	0.263644	-1.002430	-0.306481	2.951625
8	6.709646	0.480322	0.258920	-1.002857	-0.307394	2.951926

TABLE 5: Solutions of inverse kinematics between mode 1 and mode 2.

Solution no.	θ_1 (rad)	θ_2 (rad)	θ_3 (rad)	d_4 (m)	θ_5 (rad)	θ_6 (rad)
1	0.094466	0.541054	-0.283932	3.488624	0.706087	1.013583
2	0.094271	0.541054	-0.285317	3.488475	0.706780	1.013781
3	0.094076	0.392774	0.000274	3.406000	0.707473	1.013979
4	0.093880	0.399768	-0.014836	3.406000	0.708166	1.014177
5	0.011003	0.058919	-0.007261	3.406000	1.007931	1.178817
6	0.010838	0.046679	0.015329	3.406000	1.008522	1.179401
7	0.010674	0.046678	0.013928	3.406900	1.009113	1.179987
8	0.010509	0.046681	0.012523	3.407808	1.009704	1.180574
Pose no.	x (m)	y (m)	z (m)	Roll (rad)	Pitch (rad)	Yaw (rad)
1	6.061955	1.028885	2.647606	-0.736393	-0.116284	2.818948
2	6.063232	1.027803	2.642895	-0.736463	-0.117689	2.819082
3	6.064507	1.026721	2.638190	-0.792489	0.007869	2.822333
4	6.065786	1.025639	2.633476	-0.789817	0.000319	2.822551
5	6.655990	0.525765	0.456821	-0.983323	-0.278522	2.938954
6	6.657269	0.524683	0.452097	-0.986286	-0.268043	2.939934
7	6.658546	0.523601	0.447382	-0.986667	-0.269050	2.940218
8	6.659823	0.522519	0.442673	-0.987049	-0.270055	2.940503

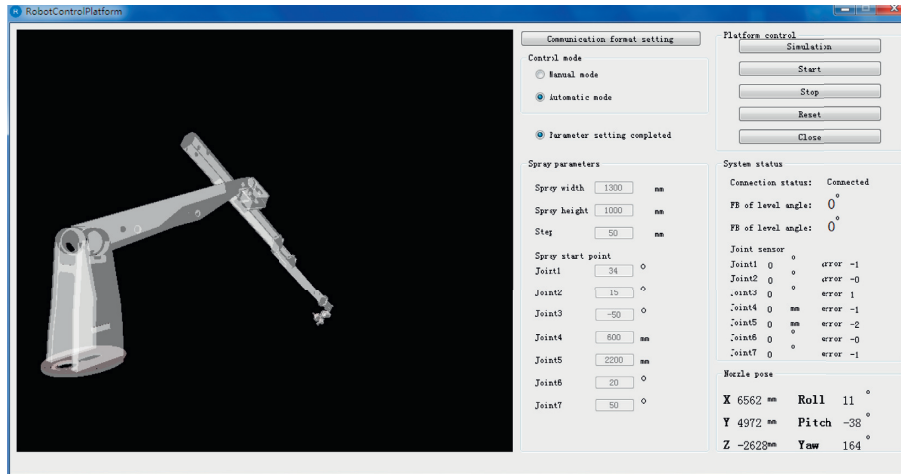


FIGURE 13: Robot control platform software interface.

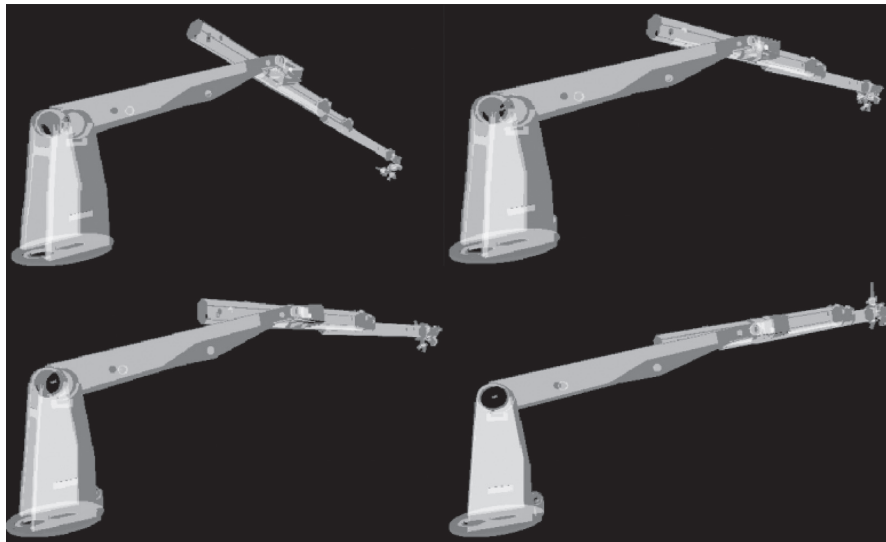


FIGURE 14: Motion simulation of the shotcrete manipulator.

5. Conclusions

A simple and effective inverse kinematics calculation method was proposed. Based on this inverse kinematics, trajectory planning of the shotcrete manipulator was realized. The two-link manipulator model is used to solve the inverse kinematics of the 6-degree-of-freedom shotcrete manipulator. The process is simple and efficient, does not require a large number of matrix operations, and can easily be obtained.

A combination of the motion trajectory shotcrete manipulator and its control principles is presented. Therefore, reasonable constraints enable the inverse kinematics of the robotic arm to be solved. In the simplified geometry model, it is described that, by giving some initial parameters, spraying trajectory can be automatically generated. Based on the inverse kinematics solution and automatic trajectory generation proposed, manipulator control software was developed using OpenGL and QT,

which provides a friendly human-computer interaction interface.

In this study, a simple geometry tunnel model is used. In the future, the real tunnel model will be obtained by sensing the environmental information with Lidar. At the same time, the normal vector of the tunnel section after blasting can be calculated to give the attitude of the tip nozzle, which can be more perpendicular to the sprayed surface to a large extent, ensuring a smaller rebound rate and better compaction.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank Sha Hao and Song FeiFei for their help in writing the article and colleague Wang Hu for his help in the work. This research was supported by the Changsha Important Science and Technology Specific Projects (grant no. kq1703022).

References

- [1] J. Höfler, J. Schlumpf, and M. Jahn, "Sika sprayed concrete handbook," in *Chapter 19: "Sprayed Concrete Applications"*, Zürich and Aichtal/Putzmeister, Aichtal, Germany, 2011.
- [2] R. M. Kessel, R. Mauermayer, and G. J. Praml, "Changes in lung function after working with the shotcrete lining method under compressed air conditions," *Occupational and Environmental Medicine*, vol. 46, no. 2, pp. 128–132, 1989.
- [3] D. Redl and J. F. Canny, "Efficient inverse kinematics for general 6R manipulators," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 648–657, 1994.
- [4] D. Zhang and H. Blake, "IKBT: solving closed-form inverse kinematics with behavior tree," *Journal of Artificial Intelligence Research*, vol. 65, pp. 457–486, 2019.
- [5] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, no. 16, pp. 1–19, 2004.
- [6] L. Biagiotti and C. Melchiorri, "Trajectory planning," in *Trajectory Planning for Automatic Machines and Robots* Springer Science & Business Media, Berlin, Germany, 2008.
- [7] R. L. Williams, "Simplified robotics joint-space trajectory generation with a via point using a single polynomial," *Journal of Robotics*, vol. 2013, Article ID 735958, 6 pages, 2013.
- [8] S. Macfarlane and A. C. Elizabeth, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 42–52, 2003.
- [9] C. E. Mower, W. Merkt, A. Davies, and S. Vijayakumar, "Comparing alternate modes of teleoperation for constrained tasks," in *Proceedings of the IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Vancouver, Canada, August 2019.
- [10] Á. Rodríguez and O. Río, *Analysis of Real Time Technical Data Obtained while Shotcreting: An Approach towards Automation*, Spanish National Research Council, Madrid, Spain, 2010.
- [11] J. Estévez, A. Suescun, G. Velez, and F. Ordás, "Development of a shotcrete training simulator for civil works," *International Journal of Robotic Engineering*, vol. 4, no. 1, 2019.
- [12] G. Girmscheid and S. Moser, "Fully automated shotcrete robot for rock support," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 3, pp. 200–215, 2001.
- [13] S. Nabulsi, A. Rodríguez, and O. Río, "Robotic machine for high-quality shotcreting process," in *Proceedings of the ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, Munich, Germany, June 2010.
- [14] M. Honegger, G. Schweitzer, O. Tschumi, and F. Amberg, "Vision supported operation of a concrete spraying robot for tunneling work," in *Proceedings of the Fourth Annual Conference on Mechatronics and Machine Vision in Practice*, Toowoomba, Australia, September 1997.
- [15] L. Uotinen and J. Suikkanen, "8th sprayed concrete web-proceedings," in *Proceedings of the Eight International Symposium on Sprayed Concrete-Modern Use of Wet Mix Sprayed Concrete for Underground Support*, Trondheim, Norway, June 2018.
- [16] A. Martín, A. Barrientos, and J. Del Cerro, "The natural-CCD algorithm, a novel method to solve the inverse kinematics of hyper-redundant and soft robots," *Soft Robotics*, vol. 5, no. 3, pp. 242–257, 2018.