A postprocessing and path optimization based on nonlinear error for multijoint industrial robot-based 3D printing

Guoqiang Fu¹,²,³, Tengda Gu¹,³, Hongli Gao¹,³ and Caijiang Lu¹,³

Abstract
Multijoint industrial robots can be used for 3D printing to manufacture the complex freeform surfaces. The post-processing is the basis of the precise printing. Due to the nonlinear motion of the rotational joint, nonlinear error is inevitable in multijoint industrial robots. In this article, the postprocessing and the path optimization based on the nonlinear errors are proposed to improve the accuracy of the multijoint industrial robots-based 3D printing. Firstly, the kinematics of the multijoint industrial robot for 3D printing is analyzed briefly based on product of exponential (POE) theory by considering the structure parameters. All possible groups of joint angles for one tool pose in the joint range are obtained in the inverse kinematics. Secondly, the nonlinear error evaluation based on the interpolation is derived according to the kinematics. The nonlinear error of one numerical control (NC) code or one tool pose is obtained. The principle of minimum nonlinear error of joint angle is proposed to select the appropriate solution of joint angle for the postprocessing. Thirdly, a path smoothing method by inserting new tool poses adaptively is proposed to reduce the nonlinear error of the whole printing path. The smooth level in the smoothing is proposed to avoid the endless insertion near the singular area. Finally, simulation and experiments are carried out to testify the effectiveness of the proposed postprocessing and path optimization method.

Keywords
3D Printing, multijoint industrial robot, nonlinear error, postprocessing, path optimization

Introduction
The 3D printing is quite popular and growing rapidly with unique advantages in the field of prototype and mold manufacturing.¹ Industrial robots realize motion control by the change of joint space, which is more flexible than in Cartesian coordinate system. The combination of 3D printing and industrial robots can overcome the space limitations of traditional printing, as shown in Figure 1, and realize the high-precision printing of complex freeform surfaces.

At present, there are many kinds of research on multijoint industrial robots for 3D printing in the field of architecture. Zhu used ABB manipulator to build a flexible...
additive remanufacturing platform, which can implement reverse engineering for defective parts. Keating and Oxman built a multifunctional and multimaterial processing platform using KUKA robot, which combines additive manufacturing and subtractive manufacturing. Gosselin et al. deposited materials layer-by-layer on ultrahigh performance concrete through an extrusion printing head mounted on a six-axis robot. Jean-Pierre et al. developed a novel five-axis robot for industrial large printing applications, which can print on fixed surfaces like walls or buildings. Multijoint industrial robots for 3D printing are also used in the traditional manufacturing field. Zhang et al. used industrial robots to design manufacturing paths along curved surfaces, simulate, and print parts with curved surfaces. Tam et al. adapted stress line additive manufacturing for structurally performative fabrication using a six-axis industrial robot.

Complex paths are the basis of 3D printing of complex surfaces of the robot. Postprocessing is the link between complex paths and printing robots. Therefore, it is a very important step in 3D printing technology. Denavit–Hartenberg parametric method was used in most postprocessing studies to establish the kinematics of robots. In this method, the cumulative error can increase with the increase of degrees of freedom (DOF). POE theory was also used for kinematics modeling. An et al. gave a generalized solution of kinematics problem based on POE, which can be used to analyze the kinematics problem of serial robots. Ayiz and Kucuk used POE theory to solve kinematics problems of industrial robots. POE theory allows global description of rigid body motion and greatly simplifies the analysis of mechanism. At present, the most common methods for solving inverse kinematics of robots are algebraic method, geometric method, and numerical method. Yahya et al. proposed a geometric method to find the optimal solution of inverse kinematics of redundant or hyper-redundant robots from infinite solutions. Ananthanarayanan and Ordóñez took the analytical solution as the initial value and iteratively solved the objective function to obtain the solution of inverse kinematics. Modern intelligent algorithms are widely used to solve inverse kinematics of robots with complex structures. Beheshti et al. adopted adaptive fuzzy logic method to determine inverse kinematics solution of a 3-DOF planar robot. Yang et al. made the multisolution a mono-one with the application of genetic algorithm to search for all global optimum solutions. Karla and Prakash combined the neural-genetic method of artificial neural network with genetic algorithm to solve the inverse kinematics problem of planar robots. These methods can obtain accurate inverse kinematics solutions but require a lot of training.

The inverse kinematics solution of robots is a multisolution problem. At present, the criterion of shortest travel is generally adopted to select the appropriate solution, but the differences between joints and the actual requirements of each joint of the robot are ignored. Guang-Dao et al. proposed a criterion of weighted shortest travel and selected the appropriate solution by finding the minimum value of weighted Euclidean distance. Xiao-Gang and Feng-Rong divided the result of inverse solution with the geometric attitude of the robot to facilitate selection. A precise path is a prerequisite to ensure the accuracy and performance of the motion system. Macfarlane and Croft used a concatenation of fifth-order polynomials to provide a smooth trajectory between two way points. Chen and Xi used freeform surfaces and CAD information to optimize deviation and generate trajectories through the trajectory integration algorithm. Due to the nonlinear motion of the rotational joint, nonlinear error is inevitable in multijoint industrial robots. Considering the similarity between industrial robots for 3D printing and five-axis machine tools, nonlinear error can also be applied to the path optimization of industrial robots for 3D printing. Lin et al. used nonlinear error evaluation method with rotational tool center point (RTCP) interpolation and particle swarm optimization to solve the
optimal setting problem of workpieces. Li et al. used cubic spline to process tool paths, densified the data to reduce nonlinear error, and smoothed the paths. Path optimization can be considered in postprocessing but can only be used to process printing code.

In this article, a postprocessing method based on nonlinear error is proposed to optimize the 3D printing path. Firstly, the kinematics of multijoint industrial robots is analyzed by POE theory, and all possible joint angle groups of a tool pose are solved in the range of joint angle according to kinematics. Secondly, the nonlinear error evaluation method is derived, and the principle of minimum nonlinear error of joint angle is used to select the appropriate solution of joint angle for the postprocessing. Thirdly, the path optimization of adaptive insertion of new tool poses is proposed to reduce the nonlinear error of the whole printing path. Finally, simulation and experiment are carried out on 3D printing system of a multijoint industrial robot.

The rest of the article is organized as follows: The second section establishes the kinematics model of multijoint industrial robots based on POE theory. In the third section, a multisolusion selection method for inverse kinematics of the robot based on nonlinear error is presented. In the fourth section, an optimization method of 3D printing paths based on nonlinear error is proposed. Simulation and experiment will be carried out in the fifth section.

Kinematics of multijoint industrial robot for 3D printing

Forward kinematics of the robot is the basis of the postprocessing. The universal forward kinematics of multijoint industrial robots is established by POE theory. The structure parameters of the robot are introduced, and the positions of each axis in the tool coordinate system (TCS) are obtained.

Kinematics configuration analysis

Industrial robots need to set the origin. If the origin position is not accurate, the robot may have a high repeatability accuracy but not positional accuracy. The absolute origin of the robot is located at the intersection of the first joint axis and the common perpendicular of the second joint axis. Generally speaking, the printing origin is located on the printing platform. For most slicing software, the workspace of the cuboid will set the printing origin at the suitable vertex, while the workspace of the cylinder will set the printing origin at the center of the bottom circle. For this article, the printing origin is set at the vertex of the workspace by default. The relative position between the absolute origin and the printing origin is approximately measured. The position is substituted into the inverse kinematics of the robot, and the motion angle of each joint is solved to make the robot move to this position. At this time, there must be a deviation between the position of the printing nozzle and the printing origin. By adjusting the movement of the printing platform, the printing nozzle can be infinitely close to the printing origin so as to determine the printing origin of the multijoint industrial robot for 3D printing.

There are one base coordinate system (BCS), one TCS, and one workpiece coordinate system (WCS) of the multijoint industrial robot for 3D printing. The origin of TCS is set at the printing nozzle and the origin of WCS is set at the printing origin, as shown in Figure 2. The displacements of three axes calculated by the postprocessing should be expressed in TCS. Vector \( V_U \) represents the position vector of the workpiece relative to the base. The position of axial line of \( A \)-axis is expressed as vector \( V_A \), where \( A = -(a_1 + a_3 + a_4) \), \( A_2 = 0 \). Vector \( V_B \) represents the position of \( B \)-axis relative to TCS, where \( B = -(a_3 + a_4) \). The position of axial line of \( C \)-axis is expressed as vector \( V_C \), where \( C = -(a_3 + a_4) \), \( C_2 = a_5 \). Vector \( V_D \) represents the position of \( D \)-axis relative to TCS, where \( D = -a_4, D_2 = a_5 \). Four vectors are structural parameters of the robot. Vector \( V_W \) represents the position vector of the workpiece relative to the base, where \( W = -(a_1 + a_3 + a_4) \), \( W_2 = a_5 - a_2 - d_1 \). The vector \( V_U \) represents the position vector of the workpiece relative to the base. As a result, vectors \( V_A, V_B, V_C, V_D, V_W, \) and \( V_U \) can be obtained.

Forward kinematics based on POE

The clear geometric meaning of POE theory makes it an effective tool for kinematic modeling. One twist can represent the motion of one rigid body. The twist coordinates can be expressed as in equation (1)

\[
\xi = [\omega^T, \nu^T]^T
\]
where \(\omega\) is the unit vector of the rotational axis; \(\nu\) is the linear velocity of the rotational motion; vector \(\nu\) can be obtained as \(\nu = p \times \omega\); \(p\) is the position vector in the base for a point fixed on the axis; and the exponential matrix of the twist is represented as in equation (2):

\[
g = e^{\mathbf{\hat{g}} \theta} = \begin{cases} 
\begin{bmatrix} e^{\omega \theta} & (I - e^{\omega \theta})(\omega \times \nu) + \omega \nu T \theta \end{bmatrix} & \text{if } \omega \neq 0 \\
[I \ \nu \theta] & \text{if } \omega = 0 
\end{cases}
\]

(2)

where \(e^{\omega \theta} = I + \omega \sin \theta + \omega^2 (1 - \cos \theta)\) (3)

Using the POE formula, the forward kinematics of multijoint industrial robots with \(n\) joint can be obtained as in equation (4):

\[
g_{\mathbf{a}}(\theta) = e^{\mathbf{\hat{g}} \theta_1} e^{\mathbf{\hat{g}} \theta_2} \ldots e^{\mathbf{\hat{g}} \theta_n}
\]

(4)

To obtain the position of the printing nozzle relative to the workpiece, the WCS was originally set as the global coordinate system. However, to facilitate the inverse kinematics solution, the BCS is set as the global coordinate system. Finally, the kinematic POE equation is coordinately translated, and the position of the printing nozzle relative to the workpiece can still be obtained. As shown in Figure 3, the order of the open chain is base→A-joint→B-joint→C-joint→D-joint→printing nozzle. At first, motion twists and exponential matrices of all axes are established according to POE theory.

For rotary axes, the motion is rotational. The motion twist is the product of one unit twist and angular. In addition, twists can represent arbitrary straight line in space from the view of geometry. The motion twist of the rotary axis is expressed as in equation (5):

\[
\xi = \left[\begin{array}{c}
\omega^T, (p \times \omega)^T
\end{array}\right]^T = \begin{bmatrix}
\omega_x, \omega_y, \omega_z, p_x \omega_z - p_z \omega_x, p_z \omega_y - p_y \omega_z
\end{bmatrix}^T
\]

(5)

where \(p = [p_x, p_y, p_z]^T\) represents the coordinates of one point in the global coordinate system. This point is in the rotation axis of the rotary axis.

In the kinematic chain of the multijoint industrial robot for 3D printing in “Kinematics configuration analysis” section, vectors \(V_4, V_B, V_C, V_D\), and \(V_W\) can calculate positions of A-joint, B-joint, C-joint, and D-joint in global coordinate system (BCS). As shown in Figure 3, point \(R_A\) is the intersection point of rotation axis of A-joint. The coordinates of \(R_A\) in BCS are \(R_A = -V_W + V_A\). Point \(R_B\) is the intersection point of rotation axis of B-joint. The coordinates relative to BCS are \(R_B = -V_W + V_B\). Point \(R_C\) is the intersection point of rotation axis of C-joint. The coordinates relative to BCS are \(R_C = -V_W + V_C\). Point \(R_D\) is the intersection point of rotation axis of D-joint. The coordinates relative to BCS are \(R_D = -V_W + V_D\). Then, unit twists of A-axis, B-axis, and C-axis are expressed as in equation (6):

\[
\begin{align*}
\xi_1 &= \begin{bmatrix} \omega_1, (p_1 \times \omega_1) \end{bmatrix}^T = [0, 0, 1, -W_y, -(A_y - W_y), 0]^T \\
\xi_2 &= \begin{bmatrix} \omega_2, (p_2 \times \omega_2) \end{bmatrix}^T = [0, 1, 0, -(B_x - W_x), 0, B_y - W_y]^T \\
\xi_3 &= \begin{bmatrix} \omega_3, (p_3 \times \omega_3) \end{bmatrix}^T = [0, 1, 0, -(C_x - W_x), 0, C_y - W_y]^T \\
\xi_4 &= \begin{bmatrix} \omega_4, (p_4 \times \omega_4) \end{bmatrix}^T = [0, 1, 0, -(D_x - W_x), 0, D_y - W_y]^T
\end{align*}
\]

(6)

According to the order of the open chain, the transformation matrix of multijoint industrial robot for 3D printing can be obtained with the product of exponential matrix of each axis. The movements of all axes relative to their local zero positions are positive relative to the global coordinate system (BCS). Then, the POE formula of the tool relative to the base of the multijoint industrial robot for 3D printing is expressed as:

\[
T_i^b = \xi_0 \cdot \xi_1^{\alpha_0} \cdot \xi_2^{\alpha_1} \cdot \xi_3^{\alpha_2} \cdot \xi_4^{\alpha_3} \cdot \xi_5^{\alpha_4} = \begin{bmatrix}
x_x & o_x & a_x & P_x' \\
y_y & o_y & a_y & P_y' \\
z_z & o_z & a_z & P_z'
\end{bmatrix}
\]

(7)

where \(\xi_i = [0, 0, 0, 0, 0, 1]^T\) represents the twist of the base and the printing nozzle, respectively. The attitude matrix is \(R = [n_x, n_y, n_z]\), and the position matrix is \(P = [P_x', P_y', P_z']^T\). However, it can be seen from Figure 2 that the fourth joint of the multijoint industrial robot in this article is a false joint. Due to the limitation of the mechanical structure, the end-effector is always parallel to the
printing platform, so that \(\delta = -\beta - \gamma\) can be obtained from the geometry, as expressed in equation (8)

\[
\begin{bmatrix}
P'_{x} \\
P'_{y} \\
P'_{z}
\end{bmatrix} = \begin{bmatrix}
\cos \alpha [a_1 \cos(\beta + \gamma) + a_2 \cos(\beta + a_1 + a_4)] \\
\sin \alpha [a_1 \cos(\beta + \gamma) + a_2 \cos(\beta + a_1 + a_4)] \\
a_3 \sin(\beta + \gamma) + a_2 \sin(\beta + d_1 - a_5)
\end{bmatrix} 
\]

(8)

The 3D printing code is the position parameter of the printing nozzle in the WCS. So, the forward kinematics equations of the printing nozzle relative to the workpiece are expressed as

\[
\begin{bmatrix}
P_{x} \\
P_{y} \\
P_{z}
\end{bmatrix} = \begin{bmatrix}
\cos \alpha[a_1 \cos(\beta + \gamma) + a_2 \cos(\beta + a_1 + a_4)] - U_x \\
\sin \alpha [a_1 \cos(\beta + \gamma) + a_2 \cos(\beta + a_1 + a_4)] - U_y \\
a_3 \sin(\beta + \gamma) + a_2 \sin(\beta + d_1 - a_5) - U_z 
\end{bmatrix} 
\]

(9)

All possible solutions of joint angle in inverse kinematics

For the kinematics model of robots established by POE theory, the Paden–Kahan subproblem is needed to solve the inverse kinematics. It requires the TCS relative to the BCS. As shown in Figure 3, given that \([P_x, P_y, P_z]^T\) is the TCS relative to the WCS, \([P_x, P_y, P_z]^T\) is converted to the BCS and expressed as \([P'_x, P'_y, P'_z]^T\) in the BCS, where \(P'_x = P_x + U_x, \ P'_y = P_y + U_y\), and \(P'_z = P_z + U_z\).

Firstly, ignoring the periodicity of trigonometric functions and the stroke range of the rotation axis, one rotation angle \(a, b, c\) of A-joint, B-joint, and C-joint is expressed as in equation (10)

\[
\begin{bmatrix}
a \\
b \\
c
\end{bmatrix} = \begin{bmatrix}
\arctan \frac{P'_y}{P'_x} \\
\arctan \frac{P'_x - d_1 + a_5}{r \pm a_1 - a_4} \pm \arccos \frac{a_3^2 + (r \pm a_1 - a_4)^2 + (P'_z - d_1 + a_5)^2 - a_2^2}{2a_2 \sqrt{(r \pm a_1 - a_4)^2 + (P'_z - d_1 + a_5)^2}} \\
\pm \arccos \frac{a_2^2 + a_3^2 - (r \pm a_1 - a_4)^2 - (P'_z - d_1 + a_5)^2}{2a_2 a_3}
\end{bmatrix} 
\]

(10)

where \(r = \sqrt{P'_x^2 + P'_y^2}\).

Secondly, considering the periodicity of trigonometric function and the travel range of rotation axis, according to the structure of multi-joint industrial robot, the stroke range of \(\alpha\) is \([-\pi, \pi]\), the stroke range of \(\beta\) is \([-\pi/2, 3\pi/2]\), and the stroke range of \(\gamma\) is \([-\pi, \pi]\). The range of the inverse trigonometric function and the stroke range of the A-joint, B-joint, and C-joint are given in Table 1. For A-joint, when the range of \(\alpha\) is \([-\pi, \pi]\), the tangent function is shown in Figure 4. Thus, the stroke range of equations can be divided into four regions, and the dual solution of equations is given in Table 2.

Combining the solutions of joint angles of A-joint, B-joint, and C-joint, the four solutions of \((\alpha, \beta, \gamma)\) are represented as follows. When \(P_x = 0\), \(r = \sqrt{(P_x + U_x)^2 + (P_y + U_y)^2}\), the solutions are represented as in equation (11)

\[
\begin{bmatrix}
a \\
b \\
c
\end{bmatrix} = \begin{bmatrix}
\pm \frac{\pi}{2} \\
\arctan \frac{P_x + U_x - d_1 + a_5}{r \pm a_1 - a_4} \pm \arccos \frac{a_3^2 + (r \pm a_1 - a_4)^2 + (P_x + U_x - d_1 + a_5)^2 - a_2^2}{2a_2 \sqrt{(r \pm a_1 - a_4)^2 + (P_x + U_x - d_1 + a_5)^2}} \\
\pm \arccos \frac{a_2^2 + a_3^2 - (r \pm a_1 - a_4)^2 - (P_x + U_x - d_1 + a_5)^2}{2a_2 a_3}
\end{bmatrix} 
\]

(11)

When \(P_x \neq 0\), \(a = \arctan [(P_y + U_y)/(P_x + U_x)]\) and \(r = \sqrt{(P_x + U_x)^2 + (P_y + U_y)^2}\). \(\alpha, \beta,\) and \(\gamma\) are all within the feasible domain, and there will be four groups of solutions. Take an example, when \(a > 0\), the four groups of solutions are as follows
\[
\begin{align*}
\alpha_1 &= a \\
\beta_1 &= \arctan \frac{P_z + U_z - d_1 + a_5}{r - a_1 - a_4} + \arccos \frac{a_2^2 + (r - a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2 - a_3^2}{2a_2 \sqrt{(r - a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2}} \\
\gamma_1 &= \arccos \frac{a_2^2 + a_3^2 - (r - a_1 - a_4)^2 - (P_z + U_z - d_1 + a_5)^2}{2a_2 a_3}
\end{align*}
\]

(12)

\[
\begin{align*}
\alpha_1 &= a \\
\beta_2 &= \arctan \frac{P_z + U_z - d_1 + a_5}{r - a_1 - a_4} - \arccos \frac{a_2^2 + (r - a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2 - a_3^2}{2a_2 \sqrt{(r - a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2}} \\
\gamma_2 &= -\arccos \frac{a_2^2 + a_3^2 - (r - a_1 - a_4)^2 - (P_z + U_z - d_1 + a_5)^2}{2a_2 a_3}
\end{align*}
\]

(13)

\[
\begin{align*}
\alpha_2 &= a - \pi \\
\beta_1 &= \arctan \frac{P_z + U_z - d_1 + a_5}{r - a_1 - a_4} + \arccos \frac{a_2^2 + (r + a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2 - a_3^2}{2a_2 \sqrt{(r + a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2}} \\
\gamma_1 &= \arccos \frac{a_2^2 + a_3^2 - (r + a_1 - a_4)^2 - (P_z + U_z - d_1 + a_5)^2}{2a_2 a_3}
\end{align*}
\]

(14)

\[
\begin{align*}
\alpha_2 &= a - \pi \\
\beta_2 &= \arctan \frac{P_z + U_z - d_1 + a_5}{r + a_1 - a_4} - \arccos \frac{a_2^2 + (r + a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2 - a_3^2}{2a_2 \sqrt{(r + a_1 - a_4)^2 + (P_z + U_z - d_1 + a_5)^2}} \\
\gamma_2 &= -\arccos \frac{a_2^2 + a_3^2 - (r + a_1 - a_4)^2 - (P_z + U_z - d_1 + a_5)^2}{2a_2 a_3}
\end{align*}
\]

(15)

**Multisolution selection method based on nonlinear error**

**Nonlinear error evaluation based on multijoint industrial robot for 3D printing**

The general software preprocessing generates a tool position file, which is the required printing path file. A continuous printing path is usually represented approximately by a set of consecutive line segments. Since the motion of the multijoint industrial robot in 3D printing process is converted into the motion of each motion axis, the actual motion trajectory is the composite trajectory of interpolation motion of each axis. As shown in Figure 5, there is a certain difference between the actual path and the ideal path, resulting in nonlinear error in 3D printing process of multijoint industrial robots.

When the interpolation calculation of multijoint industrial robot is carried out, the deviation between the actual

| Solution of rotation angles | Range of inverse trigonometric functions | Stroke of A-joint, B-joint, or C-joint |
|-----------------------------|----------------------------------------|--------------------------------------|
| \(a = \arctan(\frac{P_y'}{P_x'})\) | \([-\pi/2, \pi/2]\) | \([-\pi, \pi]\) |
| \(b = \arctan(\frac{P_y'+d_1+a_5}{r+a_1-a_4}) + \arccos \frac{a_2^2 + (r+a_1-a_4)^2 + (P_y'-d_1+a_5)^2 - a_3^2}{2a_2 \sqrt{(r+a_1-a_4)^2 + (P_y'-d_1+a_5)^2}}\) | \([-3\pi/2, 3\pi/2]\) | \([-\pi/2, 3\pi/2]\) |
| \(c = \pm \arccos \frac{a_2^2 + a_3^2 - (r+a_1-a_4)^2 - (P_y'-d_1+a_5)^2}{2a_2 a_3}\) | \([-\pi, \pi]\) | \([-\pi, \pi]\) |
the composite trajectory of each axis and the ideal trajectory of path planning is defined as nonlinear error. Suppose that there are two adjacent path points $P_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1})$ and $P_i = (x_i, y_i, z_i)$ on the trajectory, whose distance is $D = |P_{i-1} - P_i|$. Assuming that the current feed velocity is $F$ and the sampling interval is $T$, then

$$n = \frac{D}{F \cdot T} \quad (16)$$

where $n$ is the number of intervals required to cover the distance $D$. Note that $n$ obtained from equation (16) might be a noninteger, which needs to be adjusted to the nearest integer by slightly adjusting the feed velocity $F$ (still provided by $n$). For $P_{i-1}$ and $P_i$, the joint angles are $Q_{i-1} = (\alpha_{i-1}, \beta_{i-1}, \gamma_{i-1})$ and $Q_i = (\alpha_i, \beta_i, \gamma_i)$, and $n$ intervals are inserted evenly between $Q_{i-1}$ and $Q_i$, as shown in Figure 6.

For the $j$'th interval ($j\in[0, n]$), the joint angle $Q_{i,j}$ is

$$Q_{i,j} = \left(1 - \frac{j}{n}\right)Q_{i-1} + \frac{j}{n}Q_i \quad (17)$$

Each joint angle $Q_{i,j} = (\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j})$ can be solved by inverse kinematics equations (10) and (11) to get $P_{i,j}$. The distance $\varepsilon_{i,j}$ from $P_{i,j}$ to line segment $P_{i-1}P_i$ can be calculated. In fact, it is to solve the problem of the distance from a point to a line in space. It can be solved by vector cross multiplication, which is obtained by dividing the area of the parallelogram by the length of the base. So, the nonlinear error of the actual path point $P_{i,j}$ is

$$\varepsilon_{i,j} = \frac{|P_{i-1}P_i \times P_{i-1}P_{i,j}|}{|P_{i-1}P_i|} \quad (18)$$

Then, the maximum nonlinear error $\varepsilon_i$ between $P_{i-1}$ and $P_i$ is

$$\varepsilon_i = \max_{j=1}^{n} \varepsilon_{i,j} \quad (19)$$

**Optimization of joint angle selection based on nonlinear error**

The existence of a solution depends entirely on the workspace of the robot. In this workspace, there may be multiple inverse kinematic solutions of the kinematic equation. The more nonzero parameters of the connecting rod, the more ways to realize a certain target. However, the control of the robot needs a unique and exact solution. Therefore, the multisolution phenomenon of the robot will cause some problems, which need to be optimized for multiple solutions. If the obstacle avoidance requirement is not considered, the unique optimal solution can be obtained through the following steps.

The inverse solution of point A on the space path is solved to obtain $m$ groups of solutions, which is $Q_{i,k}$, where $k \in [1, m]$. Given that the previous point is $P_{i-1}$, the appropriate inverse solution is $Q_{i-1}$. According to “Nonlinear error evaluation based on multi-joint industrial robot for 3D printing” section, $n$ intervals are inserted between $Q_{i-1}$ and $Q_i$, and the forward kinematics of the $n+1$ angle value after interpolation is solved to obtain $P_{i,j,k}$, where $j\in[0, n]$. $\varepsilon_{i,j,k}$ can be calculated by equation (15), and the nonlinear error $\varepsilon_{i,k} = \max\{\varepsilon_{i,j,k}, 0 < j < n, 1 < k < m\}$ between $P_{i-1}$ and $P_i$ is found therefrom.

Finally, by comparing the nonlinear error $\varepsilon_i$ value of $m$ groups of solutions, the inverse solution corresponding to the minimum $\varepsilon_i$ value is an appropriate solution of $P_i$. 

**Figure 4.** The curve of tangent function when $ \alpha $ is in the range of $[-\pi, \pi]$. 

**Figure 5.** Nonlinear error theory model.

**Table 2.** The solutions of joint angle of A-joint.

| $a = \arctan(P'_j/P'_k)$ | Double solutions of $a$ | Region belonged to |
|--------------------------|------------------------|-------------------|
| $a > 0$                  | $a_1 = a$              | $\mathbb{1}$      |
|                          | $a_2 = a - \pi$        | $\mathbb{1}$      |
| $a < 0$                  | $a_1 = a$              | $\mathbb{2}$      |
|                          | $a_2 = a + \pi$        | $\mathbb{2}$      |
| $a = 0$                  | $a_1 = 0$              | $\mathbb{**}$     |
|                          | $a_2 = \pi$            | $\mathbb{**}$     |


Path optimization based on new tool pose insertion

Path optimization method

In the multijoint industrial robot for 3D printing, the printing path is generated by the path point file obtained by slicing software. The traditional path generation algorithm has defects, and the nonlinear error can make the actual print nozzle deviate from its ideal path, especially when the nonlinear error is too large. To avoid sacrificing the print quality, the path should be optimized in the postprocessing to reduce the nonlinear error in the printing path. It is necessary to select an appropriate error estimation method to determine whether to reduce the nonlinear error between two adjacent path points. If the error exceeds the allowable range, the optimization algorithm should be used; otherwise, the original generated path will be used for 3D printing.

Taking $e = \max \{e_i, 0 \leq i \leq n\}$ as the evaluation standard of nonlinear error, the nonlinear error between adjacent path points is calculated, respectively. If $e = \max \{e_i, 0 \leq i \leq n\}$ is greater than the allowable error value, a path point is inserted between adjacent path points, and then, the error between the adjacent path points after inserting the new path points is calculated to meet the requirements. By inserting a new path point, the size of the printed step can be controlled, and the data of path points can be made dense so as to control the nonlinear error effectively.

Therefore, firstly, the adjacent points $P_i$ and $P_{i+1}$ in the original print path file are denoted as $P_a$ and $P_b$, respectively, and the nonlinear error $e$ between the two points is calculated. If $e$ is greater than the allowable error value $[e]$, a new path point between $P_a$ and $P_b$ is inserted, and this path point is marked as $P_h$. If $e$ is still greater than $[e]$, a new path point continues to be inserted between $P_a$ and $P_b$, and this path point is denoted as $P_h$ until the nonlinear error $e$ between $P_a$ and $P_h$ is less than the allowable value $[e]$. At this time, $P_h$ is replaced by $P_a$, and the next path point is denoted as $P_h$. Then, the above process is repeated, and the nonlinear error $e$ between two points is compared with the allowable error value $[e]$, until the error requirements are met between adjacent points $P_a$ and $P_h$ in the original path. Finally, $P_{i+1}$ and $P_{i+2}$ are denoted as $P_a$ and $P_h$, respectively, and the above process is repeated until all path points in the original path point file are interpolated and optimized. It should be noted that acceleration and deceleration performance should also be considered to make the path points dense, otherwise, excessive insertion of path points will cause the jitter of multijoint industrial robots. The specific control flowchart is shown in Figure 7.

Smoothing in singular position

Singular configuration is an inherent characteristic of the industrial robot mechanism. There are two types of it. The first type is the boundary singular configuration, which appears when the robot actuator is located at the boundary of the working area. This singular configuration is not very serious, as long as the actuator is controlled away from the boundary of the working area. The second type is the internal singular configuration. When two or more joint axes are collinear, this kind of problem will appear, and this kind of singular configuration is quite difficult to deal with.

Therefore, the singular configuration must be avoided or processed near the singular point so that the robot can pass through the singular configuration on the premise of guaranteeing the working performance and motion accuracy. When the robot reaches the singular position, the 3D
printing path optimization algorithm proposed in “Path optimization method” section is easy to fall into the situation of infinite interpolation.

To avoid endless insertions near or in the singular region, the maximum number of inserted points between two points is defined. The smooth level $S$ is proposed to represent the smooth order of the inserted path points, and $S_{\text{max}}$ represents the maximum smooth level allowed. If $S_{\text{max}}$ is exceeded, the interpolation between the current adjacent points is stopped, and it is judged whether the next set of adjacent points is smooth.

Taking two points $M$ and $N$ on the path near or in the singular region as an example, where $S_{\text{max}} = 2$ is set. The first step is to evaluate the smoothness of the path between the two points of $MN$. According to the optimization algorithm in “Path optimization method” section, a new path point $R_1$ and its smooth level $S_1$ are obtained and inserted into the midpoint of the line $MN$, as shown in Figure 8(a). The second step is to smooth the path between $MR_1$. The new path point $R_2$ and its smooth level $S_2$ are still obtained first, and then optimized and inserted into the midpoint of the line $MR_1$, as shown in Figure 8(b). The smooth level is $S_2 = 2$. The third step is to smooth $MR_2$. Since $S_{\text{max}} = 2$, the smooth level exceeds $S_{\text{max}}$ at this time, and then, the current interpolation is aborted. Similarly, when smoothing $R_2R_1$, the smooth level also exceeds $S_{\text{max}}$, and the same is not interpolated. Then, the path between the next set of adjacent points is smoothed, that is, $R_1N$, as shown in Figure 8(c). Due to the smooth level limitations, $R_1R_3$ and $R_3N$ do not need to be smoothed. At this point, the path between $R_1N$ has been smoothed. Figure 8(d) shows the final smooth of the path $MN$, and the maximum number of inserted path points is 3.

**Simulation and experiment**

**Verification of multisolution selection method based on nonlinear error**

Figure 2 is a structure diagram of 3D printing system with multijoint industrial robots. This position is its initial position, and at this point, $V_A = [-285, 0, 0]^T$ mm, $V_B = [-220, 0, -56]^T$ mm, $V_C = [-220, 0, 79]^T$ mm, and $V_W = [-285, 0, -179]^T$ mm. In this experiment, $V_U = [225, -70, 12]^T$
To verify the correctness of the multisolution selection method based on nonlinear error, two path points $P_1$ and $P_2$ are obtained from the 3D printing path, where $P_1 = [165, 42, 120]^T$ and $P_2 = [165, 84, 120]^T$. And $Q_1 = [14.2811, 92.7825, -142.7475]^T$ is known. The verification of the inverse kinematics algorithm of the robot is performed according to the following steps.

1. According to the inverse kinematics formula in "All possible solutions of joint angle in inverse kinematics" section, all possible four groups of joint angle groups at the path point $P_2$ are solved, as given in Table 3.

2. According to “Optimization of joint angle selection based on nonlinear error” section, the interpolation interval is 10, and each joint angle group of the path point $P_2$ obtained by inverse kinematics can be used to calculate the corresponding joint angle interpolation. Then, the actual path points are obtained according to the kinematics, and nonlinear errors are calculated by equation (18), as given in Table 4. Similarly, the nonlinear error can be obtained by solving the remaining three sets of inverse solutions, as given in Table 5. From Tables 4 and 5, the nonlinear error of the entire path segment $P_1P_2$ under all possible joint angle groups can be obtained, as given in Table 6.

3. The nonlinear error of the entire path segment calculated under all possible joint angle groups is

![Figure 8. (a–d) The smooth steps for adjacent path points in singular area.](image)

**Table 3.** Four groups of solutions for inverse kinematics (unit: °).

| No. | $\alpha$ | $\beta$ | $\gamma$ |
|-----|----------|----------|----------|
| 1   | 26.9802  | 87.2732  | -137.2418|
| 2   | 26.9802  | -71.5542 | 137.2418 |
| 3   | -153.0198| 129.7027 | 84.9129  |
| 4   | -153.0198| 222.4277 | -84.9129 |

**Table 4.** Nonlinear error calculated using the first group of inverse solutions.

| No. | Joint angle interpolation (unit: °) | Actual path point (unit: mm) | Nonlinear error/mm |
|-----|------------------------------------|-----------------------------|--------------------|
| 1   | $\alpha$ 92.2316 $\beta$ -141.1969 $\gamma$ | 165.4623 46.0454 120.0644 | 0.4667 |
| 2   | $\alpha$ 91.6806 $\beta$ -141.6464 | 165.8261 50.1319 120.1146 | 0.8340 |
| 3   | $\alpha$ 91.1297 $\beta$ -141.0958 | 166.0898 54.2571 120.1504 | 1.1002 |
| 4   | $\alpha$ 90.5788 $\beta$ -140.5452 | 166.2517 58.4185 120.1719 | 1.2635 |
| 5   | $\alpha$ 90.0279 $\beta$ -139.9947 | 166.3102 62.6135 120.1790 | 1.3224 |
| 6   | $\alpha$ 89.4769 $\beta$ -139.4441 | 166.2637 66.8395 120.1719 | 1.2754 |
| 7   | $\alpha$ 88.9260 $\beta$ -138.8935 | 166.1108 71.0939 120.1504 | 1.1210 |
| 8   | $\alpha$ 88.3751 $\beta$ -138.3429 | 165.8501 75.3719 120.1146 | 0.8578 |
| 9   | $\alpha$ 87.8241 $\beta$ -137.7924 | 165.4803 79.6769 120.0644 | 0.4846 |
compared. According to the principle of minimum nonlinear error, the nonlinear error calculated using the first group of inverse solutions is selected to be the smallest. So, the appropriate solution is the first group, that is, $\mathbf{Q}_2 = [26.9802, 87.2732, -137.2418]^T$. In the actual printing process, according to the motion stability of the robot and the restriction of the actual working range, a group of suitable solutions is selected using the optimum condition of minimum nonlinear error. The result is shown in Figure 9.

### Verification of path optimization

The deer type specimen is selected as the experimental model, as shown in Figure 10, to verify the path optimization algorithm in “Path optimization based on new tool pose insertion” section. The original paths are obtained from slicing software. To verify the effectiveness of the

| No. | Group 2 | Group 3 | Group 4 |
|-----|---------|---------|---------|
| 1   | 69.3923 | 78.8918 | 39.8563 |
| 2   | 127.6094| 149.6099| 80.7996 |
| 3   | 171.2009| 222.5926| 129.9146|
| 4   | 197.6827| 295.2905| 182.2190|
| 5   | 205.6827| 342.6508| 222.7342|
| 6   | 194.9998| 339.1675| 235.9161|
| 7   | 166.6082| 279.8986| 213.4346|
| 8   | 122.5514| 184.4697| 157.8652|
| 9   | 65.7922 | 84.2131 | 81.4331 |

Table 6. Nonlinear error of the entire path segment $P_1P_2$ (unit: mm).

| No. | $\varepsilon_{1,1}$ | $\varepsilon_{1,2}$ | $\varepsilon_{1,3}$ | $\varepsilon_{1,4}$ |
|-----|---------------------|---------------------|---------------------|---------------------|
| 1   | 1.3224              | 205.6827            | 342.6508            | 235.9161            |

Figure 9. The simulated position and pose of the robot: (a) The first solution, (b) the second solution, (c) the third solution, and (d) the fourth solution.
algorithm, a section of contour code from the original path program is selected for simulation. The contour is shown in Figure 11. There are 20 path points along this path. The original path code needs to be postprocessed to generate the angle code that can be used for 3D printing. The relevant code is given in Table 7. The starting and ending points of the program segment are \([30.830, 41.791, 1.300]^T\) and \([64.912, 70.010, 1.300]^T\), respectively, and the corresponding angle codes are \([-6.29, 24.14, -107.66]^T\) and \([0, 21.30, -94.50]^T\).

The inverse kinematics of these 20 points is solved to obtain their respective joint angles. According to robot motor interpolation and motion control algorithm, the actual running path of multijoint industrial robot is simulated, as shown in Figure 12, which is the comparison of ideal path and actual nonlinear error path. It can be seen that the actual path of some path points deviates greatly from the ideal path, which results in the actual contour of the 3D printing deviating from the ideal contour of the 3D printing part. After the path is superimposed layer-by-layer, the surface of the actual 3D printing part is rough and the printing quality is poor. The nonlinear error model established in this article is used to analyze the nonlinear error of this path and obtain the histogram of nonlinear error distribution, as shown in Figure 13. It is not hard to see that the nonlinear error of some path points is relatively large, and the average value of the nonlinear error is 11.47 μm.

The allowable error is set as 5 μm, and the nonlinear error optimization method proposed in this article is adopted. When nonlinear error of a program is detected to be greater than the allowable error, the path point is inserted. It can be seen from Figure 13 that nonlinear error

| No. | G-code          | Angle code          |
|-----|-----------------|---------------------|
| 1   | G1 X30.830 Y41.791 E492.12910 | G1 A-6.29 B24.14 C-107.66 E492.12910 |
| 2   | G1 X31.008 Y41.860 E492.13548 | G1 A-6.27 B24.13 C-107.60 E492.13548 |
| 3   | G1 X31.528 Y42.343 E492.15090 | G1 A-6.15 B24.11 C-107.42 E492.15090 |
| 4   | G1 X32.517 Y43.680 E492.21428 | G1 A-5.84 B24.06 C-107.11 E492.21428 |
| 5   | G1 X33.191 Y44.899 E492.26409 | G1 A-5.55 B24.03 C-106.90 E492.26409 |
| 6   | G1 X36.021 Y52.400 E492.52649 | G1 A-3.86 B23.91 C-106.06 E492.52649 |
| 7   | G1 X38.225 Y58.609 E492.74508 | G1 A-2.48 B23.79 C-105.35 E492.74508 |
| 8   | G1 X39.428 Y61.119 E492.83755 | G1 A-1.92 B23.72 C-104.93 E492.83755 |
| 9   | G1 X40.729 Y62.618 E492.90360 | G1 A-1.59 B23.63 C-104.44 E492.90360 |
| 10  | G1 X41.837 Y63.344 E492.94760 | G1 A-1.43 B23.56 C-104.02 E492.94760 |
| 11  | G1 X44.458 Y61.301 E493.05822 | G1 A-1.85 B23.35 C-102.96 E493.05822 |
| 12  | G1 X47.443 Y60.264 E493.16340 | G1 A-2.05 B23.11 C-101.76 E493.16340 |
| 13  | G1 X50.600 Y60.232 E493.26849 | G1 A-2.03 B22.83 C-100.48 E493.26849 |
| 14  | G1 X53.606 Y61.206 E493.37367 | G1 A-1.81 B22.54 C-99.26 E493.37367 |
| 15  | G1 X57.415 Y63.286 E493.51811 | G1 A-1.36 B22.16 C-97.69 E493.51811 |
| 16  | G1 X60.670 Y64.608 E493.63480 | G1 A-1.08 B21.80 C-96.32 E493.63480 |
| 17  | G1 X63.097 Y66.295 E493.73307 | G1 A-0.74 B21.52 C-95.28 E493.73307 |
| 18  | G1 X64.452 Y68.044 E493.80676 | G1 A-0.39 B21.36 C-94.70 E493.80676 |
| 19  | G1 X64.748 Y68.774 E493.83290 | G1 A-0.24 B21.32 C-94.57 E493.83290 |
| 20  | G1 X64.912 Y70.010 E493.87443 | G1 A0.00 B21.30 C-94.50 E493.87443 |
in sampling sections 6–7 and 11–17 is greater than the allowable error. Therefore, new path points should be inserted in these nine sampling points. The nonlinear error distribution after the nonlinear error optimized is shown in Figure 14. It can be seen that the nonlinear error is less than 5 μm, the nonlinear error is effectively controlled, and the sampling points are increased from 20 to 31, which is the reasonable densification of path points in the sampling segments, where the error does not meet the conditions.

As shown in Figure 15, after inserting 11 path points between the path points with large nonlinear error, the actual running path of multijoint industrial robot tends to the ideal path. Figure 16 shows 11 path points at the
specific insertion positions of this contour path. Therefore, it can be shown in Figure 17 that the 3D printing path optimization method based on nonlinear error can well solve the nonlinear error distribution problem in the actual printing process of multijoint industrial robots and play a role in optimizing the path. The contour code of the optimized path points is given in Table 8.

Experiment

Finally, on the 3D printing system of multijoint industrial robot, as shown in Figure 17, the optimized path generated by the 3D printing path optimization method is verified by actual 3D printing. In the actual 3D printing system, the “deer” type specimen shown in Figure 18 and the “plum blossom” type specimen in Figure 19 are also selected. For these specimens, the method proposed in “Path optimization based on new tool pose insertion” section is used to generate the printing path, and the layer height of the printing parameter is set as 0.2 mm. The printing material is poly lactic acid (PLA).

The actual process of 3D printing is shown in Figure 17. The original printing code and the optimized printing code are, respectively, 3D printed with the 3D printing system. Figure 18(a) is the 3D printing part without path optimization. In the enlarged image of Figure 18(b), the nonsmooth path of the printed part is shown. Figure 18(c) is the 3D printing part after path optimization. In the enlarged image of Figure 18(d), the smooth path of the printed part is shown. The results show that the proposed path optimization method is effective.

Another printing experiment was carried out on the 3D printing system of multijoint industrial robot to verify the feasibility of path optimization method. The experimental conditions and parameters are consistent with the last one. Figure 19(a) is the part printed without path optimization...
Table 8. The contour code with optimization.

| No. | G-code               | Angle code         |
|-----|---------------------|--------------------|
| 1   | G1 X30.830 Y41.791 E492.12910 | G1 A-6.29 B24.14 C-107.66 E492.12910 |
| 2   | G1 X31.008 Y41.860 E492.13548 | G1 A-6.27 B24.13 C-107.60 E492.13548 |
| 3   | G1 X31.328 Y42.343 E492.15909 | G1 A-6.15 B24.11 C-107.42 E492.15909 |
| 4   | G1 X32.517 Y43.680 E492.21428 | G1 A-5.84 B24.06 C-107.11 E492.21428 |
| 5   | G1 X33.191 Y44.899 E492.26049 | G1 A-5.55 B24.03 C-106.90 E492.26049 |
| I   | G1 X33.899 Y46.774 E492.32593 | G1 A-5.13 B24.01 C-106.70 E492.32593 |
| II  | G1 X34.960 Y49.587 E492.42400 | G1 A-4.49 B23.96 C-106.39 E492.42400 |
| 6   | G1 X36.021 Y52.400 E492.52649 | G1 A-3.86 B23.91 C-106.06 E492.52649 |
| III | G1 X36.572 Y53.952 E492.58114 | G1 A-3.51 B23.88 C-105.89 E492.58114 |
| IV  | G1 X37.399 Y56.281 E492.66316 | G1 A-2.99 B23.84 C-105.63 E492.66316 |
| 7   | G1 X38.225 Y58.609 E492.74508 | G1 A-2.48 B23.79 C-105.35 E492.74508 |
| 8   | G1 X39.428 Y61.119 E492.83755 | G1 A-1.92 B23.72 C-104.93 E492.83755 |
| 9   | G1 X40.729 Y62.618 E492.90360 | G1 A-1.59 B23.63 C-104.44 E492.90360 |
| 10  | G1 X41.837 Y63.344 E492.94760 | G1 A-1.43 B23.56 C-104.02 E492.94760 |
| V   | G1 X43.148 Y62.323 E493.00293 | G1 A-1.64 B23.46 C-103.49 E493.00293 |
| 11  | G1 X44.458 Y61.301 E493.05822 | G1 A-1.85 B23.35 C-102.96 E493.11083 |
| VI  | G1 X45.951 Y60.783 E493.11083 | G1 A-1.95 B23.23 C-102.36 E493.05822 |
| 12  | G1 X47.443 Y60.264 E493.16340 | G1 A-2.05 B23.11 C-101.76 E493.16340 |
| VII | G1 X49.022 Y60.248 E493.21596 | G1 A-2.04 B22.97 C-101.12 E493.21596 |
| 13  | G1 X50.600 Y60.232 E493.26849 | G1 A-2.03 B22.83 C-100.48 E493.26849 |
| VIII | G1 X52.103 Y60.719 E493.32108 | G1 A-1.92 B22.69 C-99.87 E493.32108 |
| 14  | G1 X53.606 Y61.206 E493.37367 | G1 A-1.81 B22.54 C-99.26 E493.37367 |
| IX  | G1 X55.511 Y62.246 E493.44591 | G1 A-1.58 B22.35 C-98.48 E493.44591 |
| 15  | G1 X57.415 Y63.286 E493.51811 | G1 A-1.36 B22.16 C-97.69 E493.51811 |
| X   | G1 X59.043 Y63.947 E493.57647 | G1 A-1.22 B21.98 C-97.01 E493.57647 |
| 16  | G1 X60.670 Y64.608 E493.63480 | G1 A-1.08 B21.80 C-96.32 E493.63480 |
| XI  | G1 X61.884 Y65.452 E493.68936 | G1 A-0.91 B21.66 C-95.80 E493.68936 |
| 17  | G1 X63.097 Y66.295 E493.73307 | G1 A-0.74 B21.52 C-95.28 E493.73307 |
| 18  | G1 X64.452 Y68.044 E493.80676 | G1 A-0.39 B21.36 C-94.70 E493.80676 |
| 19  | G1 X64.748 Y68.774 E493.83290 | G1 A-0.24 B21.32 C-94.57 E493.83290 |
| 20  | G1 X64.912 Y70.010 E493.87443 | G1 A0.00 B21.30 C-94.50 E493.87443 |

Figure 18. Printing experiment for “deer” type specimen: (a) the original one (b) the enlarged one (c) the optimized one (d) the enlarged one.
method. It can be seen from the enlarged figure of Figure 19(b) that the printing path is not smooth. Using the proposed path optimization method, the printing path code of parts is optimized. The result of 3D printing is shown in Figure 19(c). In the enlarged image in Figure 19(d), it can be seen that the path becomes smooth. As can be clearly observed in Figure 19, the actual print track on the surface of the printed part is closer to the ideal print track, which confirms the feasibility of the 3D printing path optimization method.

Conclusions

Multijoint industrial robots-based 3D printing can be used for the high-precision printing of the complex freeform surfaces. The postprocessing is the basis of multijoint industrial robots for the 3D printing. This article proposes the postprocessing and the path optimization according to the nonlinear errors for the multijoint industrial robots-based 3D printing. The main contributions are as follows:

1. All possible solutions of joint angles for one tool pose are presented in detail based on the precise forward kinematics of the multijoint industrial robot for 3D printing. The positions of all axes relative to the base are introduced into the kinematic chain of the robot. The periodicity of trigonometric function and the stroke range of joint angle are considered.

2. The nonlinear errors of tool poses are calculated based on the interpolation and the forward kinematics for the evaluation of all possible solutions of joint angles. The solution with the minimum nonlinear errors is chosen as the appropriate solution of joint angle.

3. One inserting new tool poses-based path smoothing method is proposed to reduce the nonlinear error of the whole path. One new path point is inserted at the midpoint of the adjacent points if the smoothing is insufficient. The smooth level of each point is proposed to avoid the endless insertion near the singular area.

In this article, the industrial robot only with three joints is used and the solutions of joint angles for the tool orientations are not proposed, which is important for the printing of the freeform surface. In addition, the maximum available value of the smooth level is not presented. It is related to the smoothness of the whole path. In the future, further research will be carried out in these aspects. The postprocessing of the industrial robot with six joints will be
proposed. In addition, the error calibration and error compensation of the robot will be proposed to improve the accuracy of 3D printing.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was financially supported by the National Natural Science Foundation of China [Nos 51805457, 51775452, 51605253], Sichuan Science and Technology Program [2019YJ0249], China Postdoctoral Science Foundation [2020M673211], Open Foundation of the State Key Laboratory of Fluid Power and Mechatronic Systems [No. GZKF-201709], and the Fundamental Research Funds for the Central Universities [2682019CX30].

ORCID iD
Guoqiang Fu https://orcid.org/0000-0003-2887-284X

References
1. Conner BP, Manogharan GP, Martof AN, et al. Making sense of 3-D printing: creating a map of additive manufacturing products and services. Addit Manuf 2014; 1–4: 64–76.
2. Zhu S. Mobile additive remanufacturing. J Mech Eng 2013; 49(23): 1.
3. Keating S and Oxman N. Compound fabrication: a multifunctional robotic platform for digital design and fabrication. Robot Comp Int Manuf 2013; 29(6): 439–448.
4. Gosselin C, Duballet R, Roux P, et al. Large-scale 3D printing of ultra-high performance concrete – a new processing route for architects and builders. Mater Des 2016; 100: 102–109.
5. Jean-Pierre G, Said Z, and Gabriel R-T. A novel 5-axis robot for printing high resolution pictures from media on 3D wide surfaces. In: IEEE international conference on industrial technology, Gippsland, VIC, Australia, 10–13 February 2009. Gippsland: IEEE.
6. Zhang GQ, Mondesir W, Martinez C, et al. Robotic additive manufacturing along curved surface - a step towards freeform fabrication. In: Proceedings of the 2015 IEEE conference on robotics and biomimetics, Zhuhai, China, 6–9 December 2015, pp. 721–726. Zhuhai: IEEE.
7. Tam KMM, Mueller KT, Coleman JR, et al. Stress line additive manufacturing (SLAM) for 2.5-D shells. J Int Assoc Shell Spat Struct 2016; 57(4): 249–259.
8. Xu D, Calderon CAA, Gan QJ, et al. An analysis of the inverse kinematics for a 5-DOF manipulator. Int J Autom Comput 2005; 2(2): 114–124.
9. Iliukhin VN, Mitkovskii KB, Bizyanova DA, et al. The modeling of inverse kinematics for 5 DOF manipulator. Procedia Eng 2017; 176: 498–505.
10. Csg Lee and Ziegler M. Geometric approach in solving inverse kinematics of PUMA robots. IEEE Trans Aerosp Electron Syst 1984; 20(6): 695–706.
11. Fu G, Fu J, Shen H, et al. Product-of-exponential formulas for precision enhancement of five-axis machine tools via geometric error modeling and compensation. Int J Adv Manuf Technol 2015; 81(1–4): 289–305.
12. Chen Q, Zhu S, and Zhang X. Improved inverse kinematics algorithm using screw theory for a six-DOF robot manipulator. Int J Adv Robot Syst 2015; 12(10): 140–148.
13. Fu G, Gao H, and Gu T. A universal postprocessor of general table-tilting type of five-axis machine tools without rotational tool center point function for actual NC code generation. In: Proceedings of the ASME 2018 13th international manufacturing science and engineering conference, College Station, Texas, USA, 18–22 June 2018, p. 9. College Station: ASME.
14. An HS, Seo TW, and Lee JW. Generalized solution for a subproblem of inverse kinematics based on product of exponential formula. J Mech Sci Technol 2018; 32(5): 2299–2307.
15. Ayíz C and Kucuk S. The kinematics of industrial robot manipulators based on the exponential rotational matrices. In: IEEE International Symposium on Industrial Electronics, Seoul, South Korea, 5–8 July 2009. Seoul: IEEE.
16. Yahya S, Mohamed HAF, Moghavvemi M, et al. A geometrical inverse kinematics method for hyper-redundant manipulators. In: 10th international conference on control, automation, robotics and vision, Hanoi, Vietnam, 17–20 December 2008, pp. 1954–1958. Hanoi: IEEE.
17. Ananthanaryanan H and Ordóñez R. Real-time inverse kinematics of (2n+1) DOF hyper-redundant manipulator arm via a combined numerical and analytical approach. Mech Mach Theory 2015; 91: 209–226.
18. Köker R, Öz C, Çakar T, et al. A study of neural network based inverse kinematics solution for a three-joint robot. Robot Auton Syst 2004; 49(3–4): 227–234.
19. Aggarwal L, Aggarwal K, and Urbanic RJ. Use of artificial neural networks for the development of an inverse kinematic solution and visual identification of singularity zone(s). Procedia CIRP 2014; 17: 812–817.
20. Chiddarwar SS and Ramesh Babu N. Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6 R serial robot by a fusion approach. Eng Appl Artif Intell 2010; 23(7): 1083–1092.
21. Alavandar S and Nigam MJ. New hybrid adaptive neuro-fuzzy algorithms for manipulator control with uncertainties-comparative study. ISA Trans 2009; 48(4): 497–502.
22. Wei L-X, Wang H, and Li Y. A new solution for inverse kinematics of manipulator based on neural network. In: Proceedings of the second international conference on machine learning and cybernetics, Xi’an, China, 5–5 November 2003, pp. 1201–1203. Xi’an: IEEE.
23. Beheshti MTH, Tehrani AK, and Ghanbari B. An optimized adaptive fuzzy inverse kinematics solution for redundant manipulators. In: International symposium on intelligent
control, Houston, TX, USA, 8–8 October 2003, pp. 924–929. Houston: IEEE.

24. Yang Y, Peng G, Wang Y, et al. A new solution for inverse kinematics of 7-DOF manipulator based on genetic algorithm. In: *International conference on automation and logistics*, Jinan, China, 18–21 August 2007, pp. 1947–1951. Jinan: IEEE.

25. Karlra P and Prakash NR. A neuro-genetic algorithm approach for solving the inverse kinematics of robotic manipulators. In: *IEEE international conference on systems, man and cybernetics, Volume 2*, Washington, DC, USA, 8 October 2003, pp. 1979–1984. Washington: IEEE.

26. Wang Y and Wang L. The inverse kinematics algorithm of redundant DOF robot. *Mater Sci Forum* 2004; 471–472: 592–595.

27. Park HA, Ali MA, and Lee CSG. Closed-form inverse kinematic position solution for humanoid robots. *Int J Hum Robot* 2012; 9(3): 1250022.

28. Guang-Dao W, Yin-Zhong L, and Wei-Tang S. Inverse kinematics of robot based on weighted optimization. *Modular Mach Tool Autom Manuf Tech* 2016; 5: 1–8.

29. Xiao-Gang L and Feng-Rong T. Algorithm for the inverse kinematics calculation of 6-DOF serial welding robots based on geometry. *Mach Des Manuf* 2015; (2): 29–35.

30. Macfarlane S and Croft AE. Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Trans Robot Autom* 2003; 19(1): 42–52.

31. Chen H and Xi N. Automated tool trajectory planning of industrial robots for painting composite surfaces. *Int J Adv Manuf Technol* 2006; 35(7–8): 680–696.

32. Lin Z, Fu J, Shen H, et al. On the workpiece setup optimization for five-axis machining with RTCP function. *Int J Adv Manuf Technol* 2014; 74(1–4): 187–197.

33. Li J, Zhouyang H, and Lou Y. Tool path optimization in postprocessor of five-axis machine tools. *Int J Adv Manuf Technol* 2013; 68(9–12): 2683–2691.