Title page

Inverse Kinematic Algorithm for 8-DOF Redundant Manipulators Based on Weighted Least-Norm Method and Parameterization Method

Zi-Ming Chen, born in 1984, is currently an associate professor at *Yanshan University, China*. He received his Ph.D. degree from *Yanshan University, China*. His research interests include the design and analysis theory of parallel mechanism, robot technology.
Tel: +86-13780373347, E-mail: chenzm@ysu.edu.cn

Yu Guo, born in 1995, is currently a Ph.D. candidate at the *School of Mechanical Engineering, Yanshan University, China*. He received his master's degree in mechanical engineering from *Yanshan University, China*.
E-mail: gajade@qq.com

Zong-Guo Xu, born in 1997, is currently a master candidate at the *School of Mechanical Engineering, Yanshan University, China*.
E-mail: 2550838063@qq.com

Chao-Yang Ji, born in 1998, is currently a master candidate at the *School of Mechanical Engineering, Yanshan University, China*.
E-mail: 2322835711@qq.com

Yan-Wen Li, born in 1969, is currently a professor at *Yanshan University, China*. She received her Ph.D. degree from *Yanshan University, China*. Her research interests include machine vision, robotics technology, and application research.
Tel: +86-13703350152, E-mail: ywl@ysu.edu.cn

Corresponding author: Yan-Wen Li E-mail: ywl@ysu.edu.cn

ORIGINAL ARTICLE

Inverse Kinematic Algorithm for 8-DOF Redundant Manipulators Based on Weighted Least-Norm Method and Parameterization Method

Zi-Ming Chen^{1,2} • Yu Guo¹ • Zong-Guo Xu,¹ • Chao-Yang Ji,¹ • Yan-Wen Li^{1,2}

© Chinese Mechanical Engineering Society and Springer-Verlag Berlin Heidelberg 2017

Abstract: Eight-degree-of-freedom redundant manipulators are considered more suitable for performing special tasks in unstructured environments due to their high degree of freedom and flexibility. The 8-DOF tunnel shotcrete manipulator is a kind of equipment that is applied to the concrete spraying support in the process of tunnel excavation. Since the mechanism itself is a redundant manipulator with link offset, its inverse kinematics is extremely difficult to solve. To address the problems that the conventional redundant manipulator inverse kinematics solution is difficult to solve analytically and the numerical algorithm requires several iterations with accumulated errors, a synthesis inverse kinematics algorithm based on the weighted minimum norm method and the joint angle parameterization method is proposed to overcome these defects. Firstly, a set of optimized joint angle approximation solutions is obtained based on the weighted least norm method of joint angle limit avoidance optimization function, and secondly, two joint variables are parameterized as known parameters in the optimized approximation solutions based on the joint angle parameterization method, and further, the analytical expressions of the remaining joint angles are derived by algebraic methods. Finally, the superiority of the inverse kinematic synthesis algorithm over the joint angle parameterization method in terms of joint motion is verified by comparing numerical examples of linear motion trajectories. Numerical simulation results show that the algorithm has the advantages of avoiding joint angle limits, high computational accuracy, no cumulative error, and the algorithm can meet the general real-time motion control requirements as demonstrated by the experimental analysis of algorithm time

consumption.

Keywords: 8-DOF • Inverse kinematics • Weighted Least-Norm method • Parameterization method

1 Introduction

Tunnel shotcrete robot is a kind of construction machinery used for concrete shotcrete support of tunnel wall after blasting, which plays an irreplaceable role in tunneling engineering. The inverse kinematics of the shotcrete robot arm is one of the key problems that restrict the realization of intelligent shotcrete spraying in tunnels, and its inverse kinematics is extremely difficult to solve due to the existence of redundant degrees of freedom, complex joint structure, and link offset of the robot arm mechanism. The degrees of freedom (DOF) of the current large tunnel shotcrete manipulator are mostly 8-DOF [1], and the inverse kinematics problem of the robot arm is a prerequisite for analyzing the motion characteristics of the mechanism and realizing its trajectory planning and motion control.

The methods for solving the inverse kinematics of redundant robots are mainly divided into closed-form solution method, numerical solution method, and intelligent algorithm. The closed-form solution method mainly includes geometric and algebraic solution methods [2], in addition to two methods that combine algebraic and geometric methods, namely the arm angle parameterization method [3] and the joint angle parameterization method [4], Xu et al. [5] applied the above two methods to the inverse kinematics solution of a 7-DOF space manipulator, where the arm angle parameterization method has obvious geometric significance and is convenient for arm configuration control, but the method is limited by the robot arm geometric configuration and is not universal.

✉ Yan-Wen Li
ywl@ysu.edu.cn

¹ School of Mechanical Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China.

² Parallel Robot and Mechatronic System Laboratory of Hebei Province, Yanshan University, Qinhuangdao, Hebei 066004, China.

The joint angle parameterization method, also called the fixed joint method [6], can be used for the inverse kinematic solution of general redundant robotic arms, because after parameterizing one or more joints, it is equivalent to the reduction of the degrees of freedom of the robotic arm, and then the difficulty of the inverse kinematic solution will be reduced, but it will lose the motion flexibility of the original redundant mechanism itself [7]. Rong et al. [8] performed a kinematic analysis of a large shotcrete robot by converting the joint-coupled motion of an 8-DOF robot arm to kinematic constraints for an inverse kinematic solution, but this inverse kinematic approach was only applicable to the mechanism itself, not to other types of manipulators. Wu et al. [9] used the interval search genetic algorithm based on trajectory to solve inverse kinematics of an 8-DOF shotcrete manipulator, however, this method cannot guarantee the joint angle limitation effectively. Shimizu et al. [10] parametrized the self-motion variable of the SRS (i.e., 3R-1R-3R) 7-DOF anthropomorphic manipulator and obtained the analytical inverse solutions through geometric method. The self-motion of the manipulator is represented by full circumferential rotation around the intersection of the shoulder and wrist joint, but this solution method is only suitable for the SRS manipulator.

The numerical solution method for solving the inverse kinematics is generally based on the inverse Jacobi matrix at the velocity domain, and the main methods are the Jacobi pseudo-inverse method [11,12], the gradient projection method [13-15], and the weighted minimum norm method [16-18], which approximates the exact solution by giving initial values and iterating. However, the numerical solution method and intelligent algorithms lead to the emergence of cumulative errors, and the computational complexity of the algorithm makes it difficult to ensure real-time performance. To address the shortcomings of the numerical solution method, Yang et al. [19] proposed a method to avoid the inverse of the Jacobi matrix based on the weighted minimum norm method, and the simulation results proved that the method can effectively reduce the solution complexity. Zu et al. [20] proposed a quadratic computational method combining the gradient projection method and the fixed joint method for effectively solving the inverse solution of redundant robots for ensuring the solution accuracy and the real-time performance of the system, which has some universality. Yan et al. [21] solved the inverse kinematics of a 7-DOF redundant manipulator based on a quadratic algorithm. The simulation results show that the algorithm has the advantages of avoiding joint limits and accurate solutions.

In addition, there are some intelligent algorithms based on particle swarm optimization algorithm [22], genetic algorithm [23,24], neural network algorithm [25-27], and hybrid biogeography-based optimization algorithm [28] for inverse kinematics solution, but these algorithms generally have problems of poor stability and low applicability, such as slow convergence and prematurity.

Aiming at the inverse kinematics problem of 8-DOF shotcrete manipulator, this paper proposes an inverse kinematics solution method for redundant manipulator based on weighted minimum norm method and joint angle parameterization method, which solves the problems that the shotcrete manipulator cannot obtain closed-form solution, and that the numerical inverse solution algorithm or intelligent algorithm cannot solve the problem of low efficiency and poor stability. At the same time, it retains the redundant characteristics of the manipulator, and has the advantages of avoiding joint limit and no error of closed solution. It is an effective method to solve the inverse kinematics of redundant manipulators.

2 Forward Kinematics

The robotics research model is an 8-DOF tunnel shotcrete manipulator as shown in Figure1, whose configuration is R-R-P-R-R-P-R-R series mechanism, where R represents the revolute pair and P represents the prismatic pair. The overall structure of the manipulator has divided into three parts: the big arm (joint 1, 2, 3), the forearm (joint 4, 5, 6), and the gunjet (joint 7, 8, and nozzle). Joint 3 and joint 6 are prismatic and the rest revolute. The offset between joint 3 and joint 4 is a_4 . Joint 5 is a revolute joint of height d_5 for yaw motion of the forearm and the offset between joint 4 and joint 5 is a_5 .

Figure 1 8-DOF tunnel shotcrete manipulator

The Denavit-Hartenberg (D-H) link frame assignments are given in accordance with the convention described in

Craig (1986) [29]. This assignment results in the interlink homogeneous transformation matrix

$${}^{i-1}_iT = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where θ_i denotes the i -th joint angle, $s\theta_i = \sin(\theta_i)$, $c\theta_i = \cos(\theta_i)$, $s\alpha_{i-1} = \sin(\alpha_{i-1})$, $c\alpha_{i-1} = \cos(\alpha_{i-1})$. The corresponding D-H parameters are given in Table 1.

Figure 2 Shotcrete manipulator link frame assignment

The link frame assignments for shotcrete manipulator are given in Figure 2, The associated interlink homogeneous transformation matrices, ${}^{i-1}_iT$, $i = 1, \dots, 8$, are easily calculated from the above Eq. (1) based on the D-H parameters values listed in Table 1. The forward kinematics function with respect to joint angle $\theta = [\theta_1, \theta_2, d_3, \theta_4, \theta_5, d_6, \theta_7, \theta_8]^T$ is obtained by multiplying the homogeneous transformation matrices of associated interlink in order

$${}^0_8T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T {}^6_7T {}^7_8T \quad (2)$$

Table 1 Denavit-Hartenberg parameters for manipulator

Link i	Link twist $\alpha_{i-1}/(\text{rad})$	Link length $a_{i-1}/(\text{mm})$	Link offset $d_i/(\text{mm})$	Joint angle $\theta_i/(\text{rad})$
1	0	0	0	θ_1
2	$\pi/2$	0	0	θ_2
3	$\pi/2$	0	d_3	0
4	$-\pi/2$	a_4	0	θ_4
5	$-\pi/2$	a_5	d_5	θ_5
6	$-\pi/2$	0	d_6	0
7	0	0	0	θ_7
8	$-\pi/2$	0	0	θ_8

3 Inverse kinematics optimization algorithm based on weighted least norm method

3.1 An optimization method

For a manipulator with redundant degrees of freedom, since the operating space dimension is greater than the joint space dimension, for a given generalized velocity of the end-effector, there are infinitely many groups of solutions to the corresponding joint velocity, and the inverse kinematics formula of the velocity is

$$\dot{\theta} = J^+ \dot{X} + (I - J^+ J) \eta \quad (3)$$

where operating space $X \in \mathbb{R}^m$, joint space $\theta \in \mathbb{R}^n$, $n > m$. $J \in \mathbb{R}^{m \times n}$ is the Jacobian, $J^+ = J^T (JJ^T)^{-1}$ is the pseudo-inverse of the Jacobian. $I \in \mathbb{R}^{n \times n}$ is the identity matrix. $\eta \in \mathbb{R}^{n \times 1}$ is any vector. $J^+ \dot{X}$ is the minimum norm solution of joint velocity, $(I - J^+ J) \eta$ is the homogeneous solution of joint velocity, Define the optimization objective function $H(\theta)$ and the optimization coefficient K , and let $\eta = K \cdot \nabla H$ to get the gradient projection method:

$$\dot{\theta} = J^+ \dot{X} + K(I - J^+ J) \nabla H \quad (4)$$

It can be seen from Eq. (4) that the gradient projection algorithm only optimizes the secondary task index function $H(\theta)$ in the Jacobian null space without affecting the main task. However, the gradient projection method has certain drawbacks. For example, the K value is not well chosen [30]. If it is too large, it will cause oscillation and cause unnecessary self-motion. If it is too small, the optimization effect will not be achieved and the algorithm will be unstable.

The weighted least norm method was first proposed by Whitney [31] to solve the redundancy problem. Chan [32] and Dubey used the weighted least norm method to achieve joint limits avoidance for the redundant manipulator. Compared with the gradient projection method, it will not cause unreasonable large self-motion, and will only affect when necessary. The core idea of the algorithm is that when the joint limit is approached, the weighting factor of the corresponding joint increases, thus penalizing the corresponding joint velocity to avoid joint overrun; when there is a tendency to move away from the joint limit or when there is no motion of the joint, no weighting is done to the joint velocity and the joint is free.

3.2 Weighted least norm method based on optimization function of avoiding joint limits

The weighted second norm that defines the joint velocity is as follows:

$$\|\dot{\theta}\| = \sqrt{\dot{\theta}^T \mathbf{W} \dot{\theta}} \quad (5)$$

where $\mathbf{W} \in \mathbb{R}^{8 \times 8}$ is a symmetric positive definite matrix. For simplicity, $\mathbf{W}$ is generally taken as a diagonal matrix:

$$\mathbf{W} = \text{diag}(w_1, w_2, \dots, w_8) \quad (6)$$

where $w_i, i=1, 2, \dots, 8$ is the weight coefficient. Introduce the following transformations:

$$\mathbf{J}_w = \mathbf{J} \mathbf{W}^{-\frac{1}{2}}, \quad \dot{\theta}_w = \mathbf{W}^{\frac{1}{2}} \dot{\theta} \quad (7)$$

where $\mathbf{J}_w$ and $\dot{\theta}_w$ are the weighted Jacobian matrix and the weighted joint velocity vector, respectively. Therefore, the forward velocity kinematics can be expressed as:

$$\dot{\mathbf{X}} = \mathbf{J}_w \dot{\theta}_w \quad (8)$$

Then, the minimum norm solution of Eq. (8) can be expressed as:

$$\dot{\theta}_w = \mathbf{J}_w^+ \dot{\mathbf{X}} \quad (9)$$

where $\mathbf{J}_w^+ = \mathbf{J}_w^T (\mathbf{J}_w \mathbf{J}_w^T)^{-1}$ is the pseudo-inverse of the weighted Jacobian matrix. Combined with Eq. (7), the joint velocity solving formula of the weighted minimum norm method can be obtained:

$$\dot{\theta} = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W} \mathbf{J}^T)^{-1} \dot{\mathbf{X}} \quad (10)$$

The optimization index function $H(\theta)$ [33] to achieve the limits avoidance of the joint angle is as follows:

$$H(\theta) = \sum_{i=1}^n \frac{1}{4} \frac{(\theta_{\max}^i - \theta_{\min}^i)^2}{(\theta_{\max}^i - \theta_i)(\theta_i - \theta_{\min}^i)} \quad (11)$$

where $\theta_{\max}^i$ and $\theta_{\min}^i$ are the upper and lower limits of the joint angle of the i -th joint, respectively. The partial derivative of $H(\theta)$ with respect to θ_i can be expressed as:

$$\frac{\partial H(\theta)}{\partial \theta_i} = \frac{(\theta_{\max}^i - \theta_{\min}^i)^2 (2\theta_i - \theta_{\max}^i - \theta_{\min}^i)}{4(\theta_{\max}^i - \theta_i)^2 (\theta_i - \theta_{\min}^i)^2} \quad (12)$$

When the joint angle limit is approached, $\partial H / \partial \theta_i$ will tend to infinity and the weight coefficient w_i will increase to infinity to penalize further motion of the joint; when it is in the middle of the joint angle range, the value of $\partial H / \partial \theta_i$ will be zero, and the joint will be free to move

and the associated weight coefficient w_i should be equal to 1, given by the following expression:

$$w_i = \begin{cases} 1 + \left| \frac{\partial H(\theta)}{\partial \theta_i} \right|, & \Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| \geq 0 \\ 1, & \Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| < 0 \end{cases} \quad (13)$$

The value of $\Delta |\partial H / \partial \theta_i|$ is the difference between $\partial H / \partial \theta_i$ at the current moment and $\partial H / \partial \theta_i$ at the previous moment, in other words, $\Delta |\partial H / \partial \theta_i|$ is the rate of change $\partial H / \partial \theta_i$. When the joint angle at the current moment is moving towards the joint limit compared with the previous moment, the value of $\Delta |\partial H / \partial \theta_i|$ is positive; when there is no change in the joint angle, the value of $\Delta |\partial H / \partial \theta_i|$ is zero; when there is a tendency to move away from the joint limit, the value of $\Delta |\partial H / \partial \theta_i|$ is negative.

Through the joint angle $\theta(t_1)$ at the previous time and the known sampling period Δt , the joint angle at the current time can be easily obtained:

$$\theta(t_2) = \theta(t_1) + \dot{\theta} \Delta t \quad (14)$$

The joint angle $\theta(t_2)$ obtained from Eq. (14) is equivalent to that obtained by integrating the velocity over a certain time. The solution has the advantages of avoiding the limit of joint angle and the smallest weighted norm of joint speed.

4 Inverse kinematics analytical algorithm based on joint angle parameterization method

For the discrete-time real-time motion planning of the redundant manipulator, the joint angles obtained from Eq. (14) are based on the previous moment and obtained by integrating the velocity over a period of time, so the joint angles obtained are a set of optimized approximate solutions, and if a long trajectory is given, the manipulator tip will produce cumulative errors. To improve the accuracy of the inverse solution results, the joint angle parameterization method is used to parameterize the $n-m=2$ joints of the robotic arm, so that robotic arm becomes a non-redundant mechanism to reduce the difficulty of solving the analytical solution.

4.1 Analytical expression of each joint angle variable

Assume that the end pose of the manipulator represented by the homogeneous transformation matrix is given as:

$${}^0_8T = \begin{bmatrix} {}^0_8R & {}^0_8P \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where 0_8R represents the orientation of the end coordinate system in the base coordinate system, and 0_8P represents the position of the end coordinate system in the base coordinate system.

4.1.1 Analytical expression for joint angle 1

The forward kinematics equation is left multiplied by the matrix to obtain the following formula:

$$L = {}^0T^{-1} {}^0_8T = {}^1T {}^2T {}^3T {}^4T {}^5T {}^6T {}^7T {}^8T = R \quad (16)$$

where L and R are the matrices on both sides of this equation. All elements in the matrix are represented in the appendix.

To simplify the calculation, joint angle 5 and joint angle 6 are parameterized. In the first step, according to Eq. (16), the equation relation between joint 1, joint 5, and joint 6 can be obtained as follows:

$$\begin{cases} L(2,4) = c_1 p_y - s_1 p_x \\ R(2,4) = c_5 d_6 \\ L(2,4) = R(2,4) \end{cases} \quad (17)$$

where $c_1 = \cos(\theta_1)$, $c_5 = \cos(\theta_5)$, $s_1 = \sin(\theta_1)$. $L(2,4)$ and $R(2,4)$ represent the second row and fourth column of the left matrix and the right matrix on both sides of the Eq. (16). Therefore, the analytical expression of joint angle 1 can be obtained as follows:

$$\begin{aligned} \theta_1 &= A \tan 2(C_1, \pm \rho_1) - A \tan 2(p_y, -p_x) \\ C_1 &= d_6 \cos(\theta_5) \\ \rho_1 &= \sqrt{p_y^2 + p_x^2 - C_1^2} \end{aligned} \quad (18)$$

where, the function of $A \tan 2(Y, X)$ represents the four-quadrant inverse tangent, which returns the four-quadrant inverse tangent ($\tan^{-1}$) of Y and X .

There are two values of θ_1 obtained, which can be selected according to the actual available range of joint motion. Here, the available range of joint angle 1 is $(-\pi, \pi)$, and there are two cases in which the solution

results need to be converted to the available range:

$$\theta_{li} = \begin{cases} \theta_{li} - 2\pi, & \theta_{li} > \pi \\ \theta_{li} + 2\pi, & \theta_{li} < -\pi \end{cases}, i = 1, 2 \quad (19)$$

The interval conversion of the subsequent joint variables is also carried out according to the above method, which will not be described in detail later.

4.1.2 Analytical expression for the sum of joint angle 2 and joint 4

According to Eq. (16), the following equation can be obtained:

$$L(1,3) = r_{13}c_1 + r_{23}s_1 = c_7s_{24} - c_5s_7c_{24} = R(1,3) \quad (20)$$

$$L(2,3) = r_{23}c_1 - r_{13}s_1 = -s_5s_7 = R(2,3) \quad (21)$$

$$L(3,3) = r_{33} = -c_7c_{24} - c_5s_7s_{24} = R(3,3) \quad (22)$$

where $c_{24} = \cos(\theta_2 + \theta_4)$, $s_{24} = \sin(\theta_2 + \theta_4)$.

Eq. (20) $\times c_{24}$ + Eq. (22), this simplifies to the following expression:

$$(r_{13}c_1 + r_{23}s_1)c_{24} + r_{33}s_{24} = -c_5s_7 \quad (23)$$

To eliminate joint variable θ_7 to solve the sum of joint angle 2 and joint angle 4, Eq. (21) and Eq. (23) can be obtained simultaneously:

$$s_5[(r_{13}c_1 + r_{23}s_1)c_{24} + r_{33}s_{24}] = (r_{23}c_1 - r_{13}s_1)c_5 \quad (24)$$

The angle sum θ_{24} of joint angle 2 and joint angle 4 can be obtained by solving Eq. (24):

$$\theta_{24} = A \tan 2(C_5, \pm \rho_5) - A \tan 2(A_5, B_5) \quad (25)$$

where, In Eq. (26), A_5 , B_5 , C_5 , and ρ_5 are respectively:

$$\begin{aligned} A_5 &= s_5(r_{13}c_1 + r_{23}s_1) \\ B_5 &= s_5r_{33}s_{24} \\ C_5 &= c_5(r_{23}c_1 - r_{13}s_1) \\ \rho_5 &= \sqrt{A_5^2 + B_5^2 - C_5^2} \end{aligned} \quad (26)$$

4.1.3 Analytical expression for joint angle 2

According to Eq. (16), the following equation can be obtained:

$$\begin{aligned} L(1,4) &= p_x c_1 + p_y s_1 \\ &= a_4 c_2 + d_3 s_2 + a_5 c_{24} - d_5 s_{24} - d_6 s_5 c_{24} = \mathbf{R}(1,4) \end{aligned} \quad (27)$$

$$\begin{aligned} L(3,4) &= p_z \\ &= a_4 s_2 - d_3 c_2 + a_5 s_{24} + d_5 c_{24} - d_6 s_5 s_{24} = \mathbf{R}(3,4) \end{aligned} \quad (28)$$

Eq. (27) $\times c_2$ + Eq. (28) $\times s_2$, this simplifies to the following expression:

$$n_1 c_2 + n_2 s_2 = a_4 \quad (29)$$

where n_1 and n_2 are respectively:

$$\begin{aligned} n_1 &= p_x c_1 + p_y s_1 - a_5 c_{24} + d_5 s_{24} + d_6 s_5 c_{24} \\ n_2 &= p_z - a_5 s_{24} - d_5 c_{24} + d_6 s_5 s_{24} \end{aligned} \quad (30)$$

Further, the expression of joint angle 2 can be obtained according to Eq. (27)

$$\theta_2 = A \tan 2(a_4, \pm \rho_2) - A \tan 2(n_1, n_2) \quad (31)$$

where, $\rho_2 = \sqrt{n_1^2 + n_2^2 - a_4^2}$.

4.1.4 Analytical expression for joint angle 4 and joint variable 3

Eq. (27) $\times s_2$ - Eq. (28) $\times c_2$, this simplifies to the following expression:

$$d_3 = n_1 s_2 - n_2 c_2 \quad (32)$$

Since the sum θ_{24} of the angles of joint angle 2 and joint angle 4 has been obtained, and θ_2 has been obtained above, the value of joint angle 4 can be easily obtained:

$$\theta_4 = \theta_{24} - \theta_2 \quad (33)$$

4.1.5 Analytical expression for joint angle 8

The analytical expressions for the four joint angles have been derived from the above formulae, leaving only the joint variables θ_7 and θ_8 . Since joint angle 7 and joint angle 8 are only related to the orientation of the manipulator tip, to facilitate the solution of these two joint angles, the following equation is obtained by rearranging the forward kinematic formulae:

$$\mathbf{L} = {}^0_8 \mathbf{T} {}^7_8 \mathbf{T}^{-1} = {}^0_1 \mathbf{T} {}^1_2 \mathbf{T} {}^2_3 \mathbf{T} {}^3_4 \mathbf{T} {}^4_5 \mathbf{T} {}^5_6 \mathbf{T} {}^6_7 \mathbf{T} = \mathbf{R} \quad (34)$$

According to the equality of the corresponding elements on both sides of the matrix, the following equations are obtained:

$$\begin{cases} L(2,3) = -r_{22} c_8 - r_{21} s_8 \\ \mathbf{R}(2,3) = c_1 c_5 - s_5 s_1 c_{24} \\ L(2,3) = \mathbf{R}(2,3) \end{cases} \quad (35)$$

The analytical expression of joint variable θ_8 can be obtained by solving the above Eq. (35):

$$\theta_8 = A \tan 2(C_8, \pm \rho_8) - A \tan 2(r_{22}, r_{21}) \quad (36)$$

where C_8 and ρ_8 are respectively:

$$\begin{aligned} C_8 &= s_5 s_1 c_{24} - c_1 c_5 \\ \rho_8 &= \sqrt{r_{22}^2 + r_{21}^2 - C_8^2} \end{aligned} \quad (37)$$

4.1.6 Analytical expression for joint angle 7

According to Eq. (34), the following equation can be obtained:

$$\begin{cases} L(3,2) = r_{33} \\ \mathbf{R}(3,2) = -c_7 c_{24} - c_5 s_7 s_{24} \\ L(3,2) = \mathbf{R}(3,2) \end{cases} \quad (38)$$

The analytical expression of joint variable θ_7 can be obtained by solving the above Eq. (38):

$$\theta_7 = A \tan 2(-r_{33}, \pm \rho_7) - A \tan 2(A_7, B_7) \quad (39)$$

where, A_7 , B_7 , and ρ_7 are respectively:

$$\begin{aligned} A_7 &= c_{24} \\ B_7 &= c_5 s_{24} \\ \rho_7 &= \sqrt{A_7^2 + B_7^2 - r_{33}^2} \end{aligned} \quad (40)$$

Except for joint variables d_3 and θ_4 , the calculated solutions for each joint angle expression are two solutions. Considering the joint angle range of the robotic arm and the continuity of the actual motion, a solution that is closest to the current joint angle should be selected. Figure 3 shows the multiple solutions of the joint angles and the solution sequence steps. The multiple solutions are screened one by one according to the order in this figure. Finally, a set of analytical solutions with the smallest

change in joint angles and continuous motion characteristics can be obtained.

Figure 3 Multiple solutions and solving steps of joint variables

4.2 A synthesis algorithm based on weighted least-norm method and parameterization method

The flow chart of the inverse kinematics synthesis algorithm is shown in Figure 4. Firstly, a set of optimized joint variables θ^* was obtained based on the weighted least norm method. Further, θ_5 and d_6 are extracted from the optimized joint variables as the parameterized joint variables. In this case, the manipulator was equivalent to a non-redundant manipulator. Then, based on the joint angle parameterization method, the formulas of the parameterized non-redundant manipulator were deduced to obtain the analytical expressions of the other joint angles. Finally, suitable solutions are selected according to the principle of continuity of joint motion.

Figure 4 Flow chart of inverse kinematics synthesis algorithm

5 Numerical results

The known link constants are $a_4=597\text{mm}$, $a_5=330\text{mm}$, $d_5=380\text{mm}$, and the range of joint angles is shown in Table 2. In Table 2, the unit of revolute pair is radians, and the

unit of prismatic pair is meter.

Table 1 Range of motion of all joint angles

θ_i	θ_1	θ_2	d_3	θ_4	θ_5	d_6	θ_7	θ_8
$\theta_{\min}^i$	$-\pi$	$\frac{\pi}{4}$	2	$-\pi$	$-\frac{5\pi}{6}$	2	$-\pi$	$-\pi$
$\theta_{\max}^i$	π	$\frac{5\pi}{6}$	4	0	$\frac{\pi}{6}$	5	π	π

5.1 Verify inverse kinematics algorithm

Suppose the initial joint angle is θ_0 , the corresponding initial pose is X_0 , and the expected end pose is X_n .

$$\theta_0 = [0^\circ, 100^\circ, 2000\text{mm}, -100^\circ, -120^\circ, 3000\text{mm}, 0^\circ, 0^\circ] \quad (41)$$

$$X_0 = [4794.02\text{mm}, -1500\text{mm}, 1315.22\text{mm}, 3.1416\text{rad}, 0\text{rad}, -2.0944\text{rad}] \quad (42)$$

$$X_n = [7191.04\text{mm}, 1500\text{mm}, 2630.45\text{mm}, 3.3416\text{rad}, 0.2\text{rad}, -1.8944\text{rad}] \quad (43)$$

where the vector $X = [p_x, p_y, p_z, \alpha, \beta, \gamma]^T$ is represented by 6 elements, in which the first three elements represent the position of the end-effector and the last three elements represent the position of the end-effector represented by the Euler angles.

The revolute pair and the prismatic pair in the 8-DOF shotcrete manipulator are equivalently represented by cylinders and prisms respectively. The simulation model of the redundant manipulator mechanism established in MATLAB is shown in Figure 5.

Figure 5 The simulation model of manipulator mechanism

The *tploy* quintic polynomial trajectory planning function in the MATLAB robot toolbox is used to interpolate the linear trajectory. The initial and final velocity is set to zero and the simulation time is 5 seconds. Figure 6 shows the comparison curve between the desired

trajectory and the actual trajectory obtained by the synthesis algorithm.

Figure 6 The desired trajectory and the actual trajectory

The position tracking error of the tip of the manipulator is defined:

$$\Delta E_p = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} \quad (44)$$

where Δx , Δy , and Δz are the errors of the actual trajectory and the desired trajectory in the three directions of x, y, and z respectively. The trajectory error curve obtained by the simulation is shown in Figure 7.

Figure 7 Position tracking error of the end of the manipulator during movement

From the simulation results of Figure 6 and Figure 7, it can be seen that the actual trajectory completely coincides with the desired trajectory, and the position tracking error is zero, which proves the correctness of the inverse kinematics synthesis algorithm.

5.2 Inverse kinematics algorithm analysis

The curve of each joint angle obtained by the above linear trajectory simulation is shown in Figure 8 and Figure 9. The configuration changes of the redundant manipulator during the simulation are shown in Figure 10. It can be seen from Figure 8 to Figure 10 that the joint angles change curve is smooth and continuous, and at the same time, the change of the forearm movement is greater than that of the big arm movement. Such motion characteristics can help reduce energy consumption and alleviate the vibration problem of the manipulator.

Figure 8 The angle curves of all revolute joints by the synthesis algorithm

Figure 9 The displacement curves of all prismatic joints by the synthesis algorithm

Figure 10 The configuration changes of the manipulator during the movement are simulated by the synthesis algorithm

To verify the feasibility of the algorithm, it is assumed that the inverse kinematic solution is carried out for the given same linear trajectory using only the joint angle parameterization method without optimization characteristics, and the values of θ_5 and d_6 are given by Eq. (41), where the values of both are set to be constant from the beginning to the end of the trajectory. The manipulator at this time is equivalent to a non-redundant 6-DOF manipulator. The joint angle curves obtained by the simulation of the linear trajectory are shown in Figure 11 and Figure 12, and the configuration changes of the manipulator during the movement are shown in Figure 13.

Figure 11 The displacement curves of all revolute joints by the parameterization algorithm

As can be seen from Figure 12, the inverse kinematic solution obtained using the parameterization method in this case, the range of motion of the robot arm joint 3 exceeds the joint angle limit and the corresponding motion cannot be achieved in reality. In addition, the parameterization

method does not have the joint velocity norm optimization, so it does not have the energy consumption optimization characteristics.

Figure 12 The displacement curves of all prismatic joints by the parameterization algorithm

Comparing the configuration changes of the manipulator in Figure 10 and Figure 13, it can be seen that the change in the joint variables of the big arm is greater than that of the forearm., so there is an unreasonable robotic arm movement phenomenon. Therefore, if this method is used alone, the joint angle range of the manipulator may be exceeded, and the situation of inability to move may occur, resulting in the target point becoming unreachable.

Figure 13 The configuration changes of the manipulator during the movement are simulated by the parameterization method

Compared with the joint angle parameterization method without optimization characteristics, the inverse kinematic synthesis algorithm can optimize the joint motion by using the redundant characteristics of the mechanism, and the joint variables θ_5 and d_6 that are parameterized during the parameterization are given by the optimized approximate

solutions based on the weighted least norm method, which can effectively avoid the arbitrariness of artificially given values. Comparing the results of these two algorithms, it can be demonstrated that the inverse kinematic synthesis algorithm based on the weighted least norm method and the joint angle parameterization method has certain superiority in the joint limit avoidance and motion optimization of the redundant manipulators.

For the inverse kinematics numerical solutions such as the gradient projection method and the weighted minimum norm method, to ensure the accuracy of the end-effector trajectory, it is necessary to add the pose feedback and iterate the pseudo-inverse of the Jacobi matrix in each control cycle to gradually converge the error to the tolerance interval. The algorithm in this paper only needs to calculate the Jacobi pseudo-inverse once in each sampling period, and there is no iterative process, so the computational complexity is lower, and the high accuracy of the solution can be guaranteed because the analytical expression of the inverse kinematics is obtained.

The processor used for the simulation experiments in this paper is Intel(R) Core(TM) i5-4200U @1.6GHZ, RAM is 8GB, The inverse kinematics algorithm is written in MATLAB, and the inverse kinematics algorithm time consumption experiment is carried out. The linear trajectory is divided into 500 points, and the calculation time of the divided 500 points is averaged, and the average time consumption of the whole inverse kinematics algorithm is 0.7ms per step, among which the average time consumption of the optimization algorithm based on the weighted minimum norm method is 0.45ms, accounting for about 64%, and the average time consumption of the parameterization method is 0.19ms, accounting for about 27%, which fully meets the computational efficiency requirements for general motion control situations.

6 Conclusions

- (1) This study proposes an inverse kinematic synthesis algorithm for the 8-DOF shotcrete manipulator based on the weighted least norm method and the joint angle parameterization method. The algorithm first obtains the optimized approximate solutions for the joint variables based on the weighted least norm method and then obtains the analytical expressions for the remaining joint variables based on the joint angle parameterization method. Finally, the correctness of the algorithm is verified by simulation results.
- (2) Compared with the traditional inverse kinematics numerical algorithm, the inverse kinematics synthesis

algorithm in this paper avoids multiple iterations, has higher computational efficiency, and has no errors in the inverse kinematics results. The weighted minimum norm algorithm is used to give the parameterized values required by the joint angle parameterization method. This algorithm has a certain universality for the inverse kinematics problems of general redundant manipulators.

- (3) The research results of this paper provide a reference for the realization of intelligent upgrading of tunnel shotcrete manipulator, which has high application value and lays the foundation for further realization of fully automatic spraying of tunnel shotcrete manipulator and modernization of tunnel engineering.

7 Declaration

Acknowledgements

The authors sincerely thanks to Professor YanWen Li of Yanshan University for his critical discussion and reading during manuscript preparation.

Funding

Supported by National Natural Science Foundation of China (Grant No. 51775474), Hebei Provincial Natural and Science Foundation of China (Grant Nos. E2020203197), and Science and Technology Support Program of Hebei Provincial Science and Technology Research and Development Program (Grant Nos. 19221909D)

Availability of data and materials

The datasets supporting the conclusions of this article are included within the article.

Authors' contributions

The author' contributions are as follows: ZiMing Chen was in charge of the whole trial; Yu Guo wrote the manuscript; the remaining authors assisted with sampling and laboratory analyses.

Competing interests

The authors declare no competing financial interests.

Consent for publication

Not applicable

Ethics approval and consent to participate

Not applicable

References

- [1] Lin X, Song D, Qin M, et al. An Automatic Tunnel Shotcrete Robot[C]. 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22-24 Nov, 2019: 3858-3863.
- [2] Paul R P, Hong Z. Computationally Efficient Kinematics for Manipulators with Spherical Wrists Based on the Homogeneous Transformation Representation[J]. The International Journal of Robotics Research, 1986, 5 (2): 32-44.
- [3] Kreutz-Delgado K, Long M, Seraji H. Kinematic Analysis of 7-DOF Manipulators[J]. The International Journal of Robotics Research, 1992, 11 (5): 469-481.
- [4] Lee S, Bejczy A K. Redundant arm kinematic control based on parameterization[C]. Proceedings. 1991 IEEE International Conference on Robotics and Automation, Sacramento, 1991: 458-465.
- [5] Wen Fu X, Jin Tang Z, Lei Y, et al. Parameterized Inverse Kinematics Resolution Method for a Redundant Space Manipulator with Link Offset[J]. Journal of Astronautics, 2015, 36 (1): 33-39.
- [6] Yu C, Jin M, Liu H. An analytical solution for inverse kinematic of 7-DOF redundant manipulators with offset-wrist[C]. 2012 IEEE International Conference on Mechatronics and Automation, Chengdu, China, 5-8 Aug, 2012: 92-97.
- [7] Huo X, Liu Y, Jiang L, et al. Inverse kinematic optimizations of 7R humanoid arms based on a joint parameterization[C]. 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3-6 Aug, 2014: 113-118.
- [8] Xuewen R. Analyzing the kinematics of a shotcrete robot[J]. MINING & PROCESSING EQUIPMENT, 1999 (11): 25-27+2-4.
- [9] Wu D, Zhang W, Qin M, et al. Interval Search Genetic Algorithm Based on Trajectory to Solve Inverse Kinematics of Redundant Manipulators and Its Application[C]. 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May-31 Aug, 2020: 7088-7094.
- [10] Shimizu M, Kakuya H, Yoon W, et al. Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution[J]. IEEE Transactions on Robotics, 2008, 24 (5): 1131-1142.
- [11] Buss S R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods[J]. IEEE Journal of Robotics and Automation, 2004, 17 (1-19): 16.
- [12] Karpińska J, Tchoń K, Janiak M. Approximation of Jacobian Inverse Kinematics Algorithms: Differential Geometric vs. Variational Approach[J]. Journal of Intelligent & Robotic Systems, 2012, 68 (3): 211-224.
- [13] Liegeois A. Automatic supervisory control of the configuration and behavior of multibody mechanisms[J]. IEEE transactions on systems, man, and cybernetics, 1977, 7 (12): 868-871.
- [14] Dubey R V, Euler J A, Babcock S M. An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist[C]. Proceedings. 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 24-29 April, 1988: 28-36 vol.1.
- [15] Liu J, Tong Y, Ju Z, et al. Novel Method of Obstacle Avoidance Planning for Redundant Sliding Manipulators[J]. IEEE Access, 2020, 8: 78608-78621.
- [16] Ji X, Congwei Z, Wei W. General-Weighted Least-Norm Control for Redundant Manipulators[J]. IEEE Transactions on Robotics, 2010, 26 (4): 660-669.
- [17] Kim I, Oh J-H. Inverse Kinematic Control of Humanoids under Joint Constraints[J]. International Journal of Advanced Robotic Systems, 2013, 10 (1): 74.
- [18] Huang S, Peng Y, Wei W, et al. Clamping weighted least-norm method for the manipulator kinematic control: Avoiding joint limits[C]. Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28-30 July, 2014: 8309-8314.
- [19] Yang F, Li H, Wang Y, et al. An optimization method for solving the inverse kinematics of redundant manipulator[J]. Jiqiren(Robot), 2012, 34 (1): 17-21.
- [20] Di Z, Zhenwei W, Dalong T. Efficient inverse kinematic solution for redundant manipulators[J]. Chinese Journal of Mechanical Engineering, 2005, 41 (6): 71.
- [21] Wenbin Y, Lei S. An optimization method for inverse kinematics of a 7-DOF redundant manipulator[C]. 2015 34th Chinese Control Conference (CCC), Nanjing, China, 28-30 July, 2015: 4472-4479.
- [22] Huang H-C, Chen C-P, Wang P-R. Particle swarm optimization for solving the inverse kinematics of 7-DOF robotic manipulators[C]. 2012 IEEE international conference on systems, man, and cybernetics (SMC), Seoul, Korea (South), 14-17 Oct, 2012: 3105-3110.
- [23] Lin Y, Zhao H, Ding H. Solution of inverse kinematics for general robot manipulators based on multiple population genetic algorithm[J]. Journal of Mechanical Engineering, 2017, 53 (3): 1-8.
- [24] Zhou Z, Guo H, Wang Y, et al. Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm[J]. International Journal of Advanced Robotic Systems, 2018, 15 (4): 1729881418792992.
- [25] Köker R, Çakar T, Sari Y. A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators[J]. Engineering with Computers, 2014, 30 (4): 641-649.
- [26] Duka A-V. Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm[J]. Procedia Technology, 2014, 12: 20-27.
- [27] Almusawi A R J, Dülger L C, Kapucu S. A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm (Denso VP6242)[J]. Computational Intelligence and Neuroscience, 2016, 2016: 5720163.
- [28] Ren Z-W, Wang Z-H, Sun L-N. A hybrid biogeography-based optimization method for the inverse kinematics problem of an 8-DOF redundant humanoid manipulator[J]. Frontiers of Information Technology & Electronic Engineering, 2015, 16 (7): 607-616.
- [29] Craig J J. Introduction to robotics: mechanics and control, [M]. Pearson Education India, 2009.
- [30] Sun K, Xie Z, Huang J, et al. Gradient projection method of kinematic redundant manipulator based on continuous scale factor[J]. Journal of Jilin University (Engineering and Technology Edition), 2009, 39 (5): 1257-1261.
- [31] Whitney D E. Resolved motion rate control of manipulators and human prostheses[J]. IEEE Transactions on man-machine systems, 1969, 10 (2): 47-53.
- [32] Chan T F. A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators[J]. IEEE Transactions on Robotics and Automation, 1995, 11: 286-292.
- [33] Zghal H, Dubey R V, Euler J A. Efficient gradient projection optimization for manipulators with multiple degrees of redundancy[C]. Proceedings., IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13-18 May, 1990: 1006-1011.

Biographical notes

Zi-Ming Chen, born in 1984, is currently an associate professor

at Yanshan University, China. He received his Ph.D. degree from Yanshan University, China. His research interests include the design and analysis theory of parallel mechanism, robot technology.

E-mail: chenzm@ysu.edu.cn

Yu Guo, born in 1995, is currently a Ph.D. candidate at the School of Mechanical Engineering, Yanshan University, China. He received his master's degree in mechanical engineering from Yanshan University, China.

E-mail: gajade@qq.com

Zong-Guo Xu, born in 1997, is currently a master candidate at the School of Mechanical Engineering, Yanshan University, China.

E-mail: 2550838063@qq.com

Chao-Yang Ji, born in 1998, is currently a master candidate at the School of Mechanical Engineering, Yanshan University, China.

E-mail: 2322835711@qq.com

Yan-Wen Li, born in 1969, is currently a professor at Yanshan University, China. She received her Ph.D. degree from Yanshan University, China. Her research interests include machine vision, robotics technology, and application research.

E-mail: ywl@ysu.edu.cn

Appendix

The left matrix of Eq. (16):

$$L = \begin{bmatrix} c_1 r_{11} + s_1 r_{21} & c_1 r_{12} + s_1 r_{22} & c_1 r_{13} + s_1 r_{23} & c_1 p_x + s_1 p_y \\ c_1 r_{21} - s_1 r_{11} & c_1 r_{22} - s_1 r_{12} & c_1 r_{23} - s_1 r_{13} & c_1 p_y - s_1 p_x \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The right matrix of Eq. (16):

$$R = \begin{bmatrix} R(1,1) & R(1,2) & R(1,3) & R(1,4) \\ c_7 c_8 s_5 - c_5 s_8 & -c_5 c_8 - c_7 s_5 s_8 & -s_5 s_7 & c_5 d_6 \\ R(3,1) & R(3,2) & R(3,3) & R(3,4) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The elements in the right matrix of Eq (16):

$$R(1,1) = c_8 (s_7 s_{24} + c_5 c_7 c_{24}) + s_5 s_8 c_{24}$$

$$R(1,2) = c_8 s_5 c_{24} - s_8 (s_7 s_{24} + c_5 c_7 c_{24})$$

$$R(1,3) = c_7 s_{24} - c_5 s_7 c_{24}$$

$$R(1,4) = a_4 c_2 + d_3 s_2 + a_5 c_{24} - d_5 s_{24} - d_6 s_5 c_{24}$$

$$R(3,1) = s_5 s_8 s_{24} - c_8 (s_7 c_{24} - c_5 c_7 s_{24})$$

$$R(3,2) = s_8 (s_7 c_{24} - c_5 c_7 s_{24}) + c_8 s_5 s_{24}$$

$$R(3,3) = -c_7 c_{24} - c_5 s_7 s_{24}$$

$$R(3,4) = a_4 s_2 - c_2 d_3 + (a_5 - d_6 s_5) s_{24} + d_5 c_{24}$$

The left matrix of Eq. (34):

$$L = \begin{bmatrix} c_8 r_{11} - r_{12} s_8 & r_{13} & -c_8 r_{12} - r_{11} s_8 & p_x \\ c_8 r_{21} - r_{22} s_8 & r_{23} & -c_8 r_{22} - r_{21} s_8 & p_y \\ c_8 r_{31} - r_{32} s_8 & r_{33} & -c_8 r_{32} - r_{31} s_8 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The right matrix of Eq. (34):

$$R = \begin{bmatrix} R(1,1) & R(1,2) & R(1,3) & R(1,4) \\ R(2,1) & R(2,2) & R(2,3) & R(2,4) \\ R(3,1) & R(3,2) & R(3,3) & R(3,4) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The elements in the right matrix of Eq (34):

$$R(1,1) = s_7 c_1 s_{24} - c_7 (s_1 s_5 - c_5 c_1 c_{24})$$

$$R(1,2) = c_7 c_1 s_{24} + s_7 (s_1 s_5 - c_5 c_1 c_{24})$$

$$R(1,3) = -c_5 s_1 - s_5 c_1 c_{24}$$

$$R(1,4) = a_5 c_1 c_{24} - d_5 c_1 s_{24} - d_6 (c_5 s_1 + s_5 c_1 c_{24}) + a_4 c_1 c_2 + c_1 d_3 s_2$$

$$R(2,1) = s_7 s_1 s_{24} + c_7 (c_1 s_5 + c_5 s_1 c_{24})$$

$$R(2,2) = c_7 s_1 s_{24} - s_7 (c_1 s_5 + c_5 s_1 c_{24})$$

$$R(2,3) = c_1 c_5 - s_5 s_1 c_{24}$$

$$R(2,4) = a_5 s_1 c_{24} - d_5 s_1 s_{24} + d_6 (c_1 c_5 - s_5 s_1 c_{24}) + a_4 c_2 s_1 + d_3 s_1 s_2$$

$$R(3,1) = c_5 c_7 s_{24} - s_7 c_{24}$$

$$R(3,2) = -c_7 c_{24} - c_5 s_7 s_{24}$$

$$R(3,3) = -s_5 s_{24}$$

$$R(3,4) = a_4 s_2 - c_2 d_3 + a_5 s_{24} + d_5 c_{24} - d_6 s_5 s_{24}$$