



A real-time inverse kinematics solution based on joint perturbation for redundant manipulators

Qinhuan Xu and Qiang Zhan

Robotics Institute, Beihang University, Beijing, 100191, China

Correspondence: Qiang Zhan (qzhan@buaa.edu.cn)

Received: 18 June 2020 – Revised: 25 December 2020 – Accepted: 20 January 2021 – Published: 25 February 2021

Abstract. Aiming at the problem that the calculation of the inverse kinematics solution of redundant manipulators is very time-consuming, this paper presents a real-time method based on joint perturbation and joint motion priority. The method first seeks the pose nearest to the target pose in the manipulator's pose set through fine-tuning all the joints with different angle deviations at the same time and then regards this pose as the starting one to perform iterative calculations until the error between the current pose and the target pose is less than the pre-determined error, thus obtaining the inverse kinematics solution corresponding to the target pose. This method can avoid the pseudo-inverse calculations of the Jacobian matrix and significantly reduce the solving complexity. Two types of manipulators are taken as examples to validate the proposed method. Under the premise that the manipulator motion trajectory is satisfied, the Jacobian pseudo-inverse method and the proposed method are both adopted to solve the inverse kinematics. Simulations and comparisons show that the proposed method has better real-time performance, and the joint motions can be flexibly controlled by setting different joint motion priorities. This method can make the work cycle faster and improve the production efficiency of redundant manipulators in real applications.

1 Introduction

Until the present, in the manufacturing industry, traditional industrial manipulators with 6 or fewer degrees of freedom (DOFs) have been widely used in such tasks as transporting, assembling, spraying, and welding. However, with the diversification of production environments and task requirements, manipulators are required to have more flexible motions and obstacle avoidance capability. Therefore, manipulators with redundant DOFs have gradually attracted people's favour for higher flexibility and adaptability compared to traditional manipulators (Aouache et al., 2019; Akbaripour and Masehian, 2017; Omrcen and Zlajpah, 2007). The additional DOFs have led to redundant manipulators being able to complete tasks such as obstacle avoidance, joint limit avoidance, and load distribution optimization while performing specific production tasks (Liu et al., 2017), such as welding and polishing. To speed up the work cycle and improve production efficiency, higher requirements are imposed on the real-time performance of manipulators (Kalra et al., 2006; Kuhlemann et al., 2016). Compared to non-redundant ma-

nipulators, redundant manipulators have null space, and thus the inverse kinematics solution is not unique, and the process is more complicated and time-consuming. As a bottleneck problem of motion control and path planning, the high real-time inverse kinematics solution of redundant manipulators has caused much concern (Dasgupta et al., 2009; Kouabon et al., 2020).

Until the present, numerous research studies have focused on the inverse kinematics solution of redundant manipulators, and the methods can be divided into three types: geometric methods, numerical methods, and iterative methods (Jia et al., 2014). Transforming the inverse kinematics problem into a geometric one in a two-dimensional plane or a three-dimensional space, geometric methods (Tian et al., 2020; Faria et al., 2018; Zaplana et al., 2018) are suitable for redundant manipulators with a simple structure or special structure. Numerical methods include the Jacobian pseudo-inverse method (Hayashibe et al., 2006; Park et al., 2018), the gradient projection method (Zu et al., 2005), the generalized Jacobian method (Kumar et al., 2010), and the weighted

minimum norm method (Maciejewski and Klein, 1985; Mohamed et al., 2009). However, these methods usually make equivalent approximations to the kinematics models of redundant manipulators during the solving process, inevitably leading to error accumulation. The high non-linearity and complex coupling of redundant manipulators have also resulted in the use of iterative algorithms to obtain inverse kinematics solutions. Korein and Badler (1982) first proposed the use of the iterative method to calculate the inverse kinematics solution of redundant manipulators. The limitation of this kind of method, however, is that the convergence of the solving results depends on the setting of the initial pose. Damas (2012) proposed an algorithm that can learn the input–output relationship of kinematics online and is suitable for solving forward and inverse kinematics of manipulators. Ayusawa and Nakamura (2012) proposed a fast inverse kinematics method that calculates the gradient vector with the Newton–Euler iterative algorithm to obtain the inverse solution. In addition, some methods based on genetic algorithms (Zeng et al., 2018), neural networks (Xia and Wang, 2001), particle swarm optimization (PSO) (Umar et al., 2019), and differential evolution (DE) (Lopezfranco et al., 2018) can also solve the inverse kinematics problem of redundant manipulators in an iterative manner. However, these iterative algorithms require extensive calculations and are very time-consuming.

The inverse kinematics algorithms of redundant manipulators are mainly evaluated from indexes such as real-time, reliability, stability, and universality (Tolani et al., 2000). The geometric method is fast and accurate for solving inverse kinematics, but it depends on the specific structure and geometric characteristics of manipulators and lacks universality. The numerical method and iterative method have the advantages of strong universality as they can be applied to hyper-redundant manipulators as well as constraints and objective functions in optimization problems, but they are not suitable for online real-time path planning because of extensive calculations, poor real-time performance, and low precision.

This paper presents a real-time method based on joint perturbation to obtain the inverse kinematics solution of redundant manipulators, which has no constraints on the DOFs, sizes, and structures of redundant manipulators. The remainder of this paper is organized as follows. In Sect. 2, the inverse kinematics method based on joint perturbation is introduced. In Sect. 3, simulations and comparisons are provided to validate the proposed method. Finally, conclusions of the paper are given in Sect. 4.

2 Inverse kinematics method based on joint perturbation

2.1 Kinematics model of redundant manipulators

If the dimension of a manipulator’s work space is m , and the dimension of its joint space is n , when $m < n$, the manipula-

tor has $n - m$ redundant DOFs. The kinematics model of the redundant manipulator can be established with the modified D–H method (Denavit–Hartenberg; Lv and Liu, 2016). First, the D–H coordinate systems of all the links are established. The z axis of coordinate system $\{i\}$ is collinear with the axis of joint i , which is located at the beginning of link i . The x axis coincides with the common perpendicular of joint i and joint $i + 1$, pointing from joint i to joint $i + 1$, and the y axis can be determined by the right-hand rule. The D–H transformation parameters are determined according to the relations of neighbouring frames, and then the homogeneous transformation matrix of frame $\{i\}$ relative to frame $\{i - 1\}$ ${}^i_{i-1}T$ can be attained.

$${}^i_{i-1}T = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where a_{i-1} represents the length of the common perpendicular to the axes of joint $i - 1$ and joint i , α_{i-1} represents the angle between the axes of joint $i - 1$ and joint i , d_i represents the offset of link i relative to link $i - 1$, and θ_i represents the rotation angle of link i relative to link $i - 1$ around the axis of joint i . Multiplying all the homogeneous transformation matrices in order results in the forward kinematics model n_0T of the manipulator.

$${}^n_0T = {}^1_0T {}^2_1T \dots {}^{n-1}_{n-2}T {}^n_{n-1}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Define $q = (\theta_1 \theta_2 \dots \theta_n)^T$ as the joint variables of the manipulator, and the pose of the manipulator is $P = f(q)$. When the current joint angles of the manipulator are $q_0 = (\theta_{01} \theta_{02} \dots \theta_{0n})^T$, the current pose of the manipulator P_0 can be obtained by

$$P_0 = f(q_0). \quad (3)$$

The target pose of the manipulator P_t can be determined according to task requirements. The inverse kinematics solution of the manipulator is used to obtain the joint variables q_t according to the initial pose P_0 and the target pose P_t . A redundant manipulator with n DOFs has numerous inverse solutions, from which the optimal one should be selected in an actual motion control. This paper presents a real-time inverse kinematics method based on joint perturbation, in which two definitions including the joint motion priority and joint perturbation coefficient matrix are used.

2.2 Joint motion priority

First, define the motion priority of each joint as $k_1, k_2, k_3 \dots k_{n-1}, k_n, 0 \leq k_i \leq 1 (i = 1, 2, 3 \dots n)$. The larger the k_i is, the higher the motion priority joint i has. When $k_i = 0$, the motion of joint i will be limited to zero. The priority of

each joint can be determined according to task requirements. For example, when a manipulator’s motion is planned based on energy optimization, according to the rule of “moving big joints less and small joints more”, the joint motion priorities can be set as $k_1 \leq k_2 \leq k_3 \dots k_{n-1} \leq k_n$. When a manipulator’s motion is planned based on time minimization, the joint motion priorities can be set as $k_1 \geq k_2 \geq k_3 \dots k_{n-1} \geq k_n$ to follow the rule of “moving big joints more and small joints less” (Xu et al., 2010). In particular, when a manipulator’s motion is not constrained by tasks, the joint motion priorities can be set as $k_1 = k_2 = k_3 \dots k_{n-1} = k_n$ so that all the joints have the same motion priority.

2.3 Joint perturbation coefficient matrix

Suppose the initial joint variables of a manipulator are $q_0 = (\theta_{01}\theta_{02}\dots\theta_{0n})^T$, which correspond to the initial pose P_0 of the manipulator. A very small angle $\delta\theta$ ($\delta\theta > 0$) is set as the basic perturbation angle, and the perturbation angle of joint i is $\Delta\theta_i$. Thus, the perturbation angle of joint i is the product of the motion priority k_i and the basic perturbation angle $\delta\theta$.

$$\Delta\theta_i = k_i\delta\theta \tag{4}$$

A rotary joint has two types of possible motions: counter-clockwise or clockwise, so the coefficient of $\delta\theta$ can be 1 or -1 . When the perturbation direction of joint i is counter-clockwise, $\Delta\theta_i = k_i\delta\theta$; when the perturbation direction is clockwise, $\Delta\theta_i = -k_i\delta\theta$.

$$\Delta q = \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \vdots \\ \Delta\theta_n \end{pmatrix} = \begin{pmatrix} \pm k_1\delta\theta \\ \pm k_2\delta\theta \\ \vdots \\ \pm k_n\delta\theta \end{pmatrix} \tag{5}$$

Hence, the perturbation variable Δq of a manipulator with n DOFs has 2^n possible combinations. Each combination of the coefficients of the basic perturbation angle $\delta\theta$ of n joints constitutes a column vector, and all the column vectors constitute a matrix K , which is defined as the joint perturbation coefficient matrix. Figure 1 shows the joint perturbation combinations of three types of manipulators with 1 DOF, 2 DOFs, and 3 DOFs, and the construction of the joint perturbation coefficient matrix is introduced as follows.

When $n = 1$, the manipulator has only 1 DOF and two types of possible motions, as shown in Fig. 1a. Therefore, the joint perturbation coefficient matrix of the manipulator is

$$K_1 = \begin{bmatrix} 1 & -1 \end{bmatrix}. \tag{6}$$

When $n = 2$, the manipulator has 2 DOFs and four types of possible motions, as shown in Fig. 1b, and the joint perturbation coefficient matrix is

$$K_2 = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ K_1 & K_1 \end{bmatrix}. \tag{7}$$

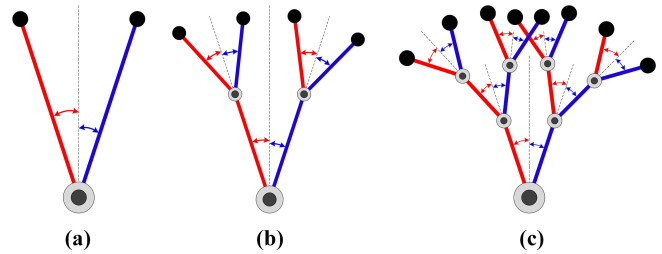


Figure 1. Joint perturbation combinations of manipulator with n DOFs. (a) $n = 1$. (b) $n = 2$. (c) $n = 3$.

When $n = 3$, the manipulator has 3 DOFs and eight types of possible motions, as shown in Fig. 1c, and the joint perturbation coefficient matrix is

$$K_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ K_2 & K_2 \end{bmatrix}. \tag{8}$$

By this analogy, the joint perturbation coefficient matrix of a redundant manipulator with n DOFs is

$$K_n = \begin{bmatrix} 1 & \dots & 1 & -1 & \dots & -1 \\ & & K_{n-1} & & & K_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ K_{n-1} & K_{n-1} \end{bmatrix}. \tag{9}$$

In Eq. (9), $K_n \in R^{n \times 2^n}$, the elements of the first 2^{n-1} columns and the first row are all 1, and the elements of the last 2^{n-1} columns and the first row are all -1 . When the joint motion priorities are determined, the joint perturbation coefficient matrix considering the joint motion priority can be obtained by multiplying row i of matrix K_n by k_i . Similar to rotary joints, the motion priority of prismatic joints is set as k_p , and the basic perturbation displacement as δd , and the coefficient of δd can be 1 or -1 . When $\Delta d = k_p\delta d$, the perturbation direction is positive, and when $\Delta d = -k_p\delta d$, the perturbation direction is negative. Therefore, this algorithm can be used for manipulators with prismatic joints.

2.4 Calculation of basic perturbation angle

The basic perturbation angle not only determines the amplitude of joint perturbation in an iterative calculation process but also plays a decisive role in the solving speed and accuracy of the algorithm. The smaller the basic perturbation angle, the higher the accuracy of the solution. As the number of iterations increases, the solving speed will decrease. Therefore, it is necessary to take a bigger basic perturbation angle while satisfying the solution accuracy. For a manipulator with n DOFs, the link lengths are $L = [l_1, l_2, \dots, l_n]$, and

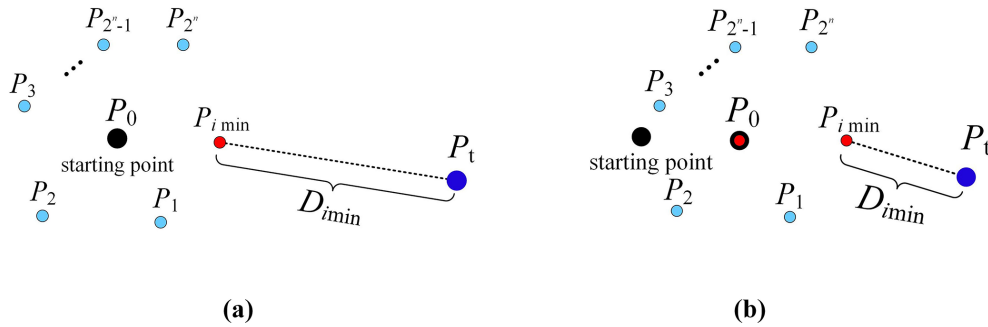


Figure 2. Iterative calculation process of the joint perturbation method. (a) The first iteration. (b) The second iteration.

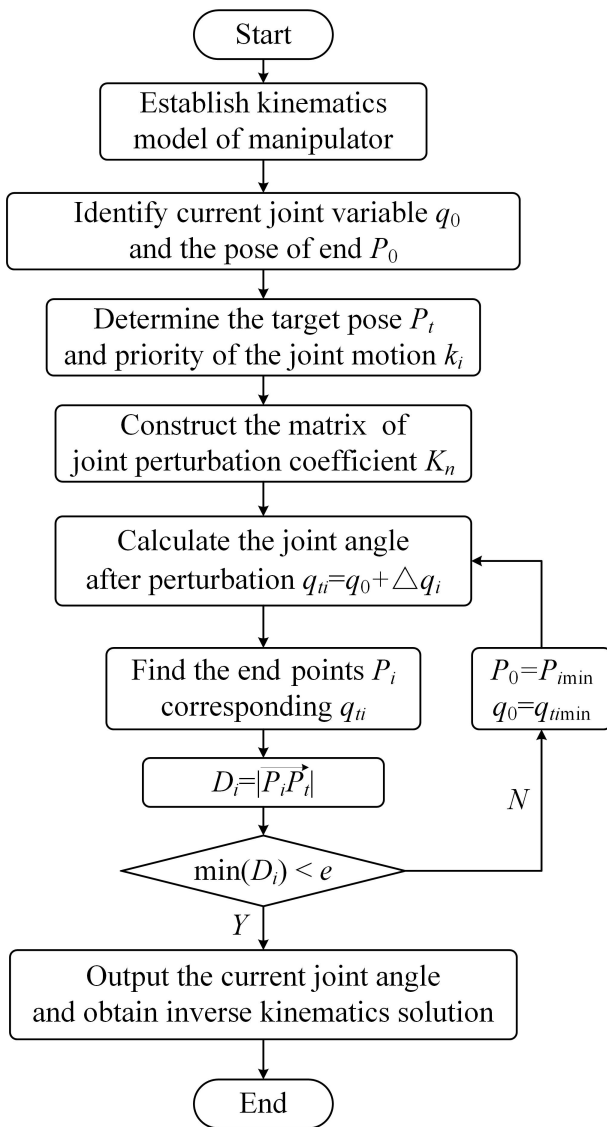


Figure 3. The flow chart of the algorithm based on joint perturbation.

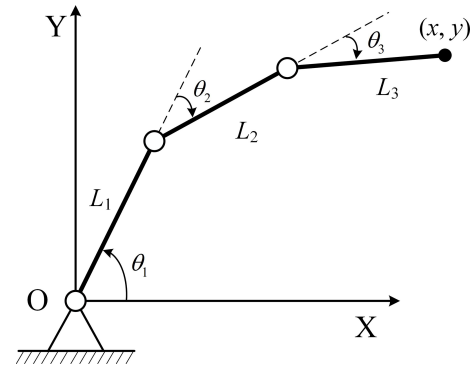


Figure 4. The schematic diagram of a planar manipulator.

then $l_{\Sigma} = l_1 + l_2 + \dots + l_n$, and the maximum displacement Δs of the manipulator position for one perturbation should satisfy

$$\Delta s = l_{\Sigma} \delta\theta + (l_{\Sigma} - l_1) \delta\theta + \dots + l_n \delta\theta \leq e, \tag{10}$$

where e is the maximum error allowed by the solution accuracy or predetermined. In this paper, the calculation formula of the basic perturbation angle is defined as

$$\delta\theta = \frac{e}{l_1 + 2l_2 + \dots + il_i + \dots + nl_n}. \tag{11}$$

2.5 Inverse kinematics solution method

The joint angles after perturbation q_{ti} ($i = 1, 2, 3 \dots 2^n$) can be calculated according to the joint perturbation coefficient matrix and the basic perturbation angle $\delta\theta$.

$$q_{ti} = q_0 + \Delta q_i = \begin{pmatrix} \theta_{01} \\ \theta_{02} \\ \vdots \\ \theta_{0n} \end{pmatrix} + \delta\theta \cdot K_n(i) \tag{12}$$

In Eq. (12), the vector $K_n(i)$ is the column i of the joint perturbation coefficient matrix K_n , so the pose of the manipulator P_i ($i = 1, 2, 3 \dots 2^n$) can be calculated by

$$P_i = f(q_{ti}) = f(q_0 + \Delta q_i). \tag{13}$$

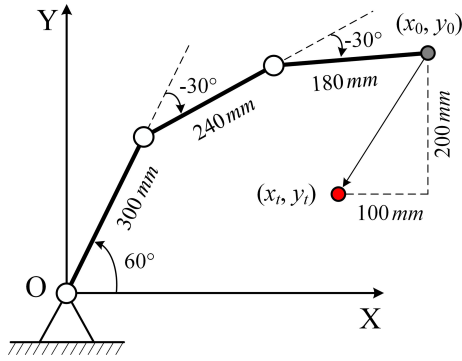


Figure 5. Parameters setting of a planar manipulator with 3 DOFs.

2^n different poses ($P_1, P_2 \dots P_{2^n}$) can be obtained through one single operation for a redundant manipulator with n DOFs, and then the distance $D_i = |\vec{P_i P_t}|$ between each pose P_i and the target pose P_t can be calculated. The minimum of D_i can be determined by comparisons, $D_{i\min} = \min |\vec{P_i P_t}|$. Set the pose nearest to the target pose P_t as $P_{i\min}$, so the joint variable corresponding to $P_{i\min}$ is $q_{ti\min}$. Further, set the pose $P_{i\min}$ as the starting pose and reassign P_0 and q_0 as $P_0 = P_{i\min}, q_0 = q_{ti\min}$. The iterative calculations will continue for the target pose P_t until $\min |\vec{P_i P_t}| \leq e$. When the iterative calculations end, the calculated joint variable q_t is the inverse solution of the target pose P_t . The process of the iterative calculations from the starting pose P_0 to the target pose P_t is shown in Fig. 2.

2.6 The main steps of the method

The main steps of the joint perturbation method are summarized as follows:

1. Establish the kinematics model of redundant manipulators with n DOFs.
2. Identify the current joint variables $q_0 = (\theta_{01}\theta_{02} \dots \theta_{0n})^T$ and calculate the corresponding manipulator pose P_0 .
3. Determine the target pose P_t and the priority of each joint motion k_i according to actual task requirements.
4. Construct the joint perturbation coefficient matrix K_n , determine the basic perturbation angle $\delta\theta$, and obtain the perturbation angle of each joint $\Delta q = k_i \cdot \delta\theta$.
5. Calculate the joint angles after perturbation $q_{ti} = q_0 + \Delta q_i (i = 1, 2, 3 \dots 2^n)$.
6. Find all the poses P_i for the 2^n kinds of possibilities by using the forward kinematics model of the manipulator.
7. Calculate the distance $D_i = |\vec{P_i P_t}|$ between the current manipulator pose P_i and the target pose P_t .

8. Obtain $D_{i\min}$ by comparison and verify whether $D_{i\min} \leq e$ is satisfied. If $D_{i\min} \leq e$, the current joint angles are the joint variables q_t corresponding to the target pose P_t . If $D_{i\min} > e$, set the pose corresponding to joint variables $q_{ti\min}$ as the starting pose and repeat steps 5–7.

A flow chart of the method is shown in Fig. 3.

3 Simulations and analyses

3.1 Simulation verification of a manipulator with 3 DOFs

A planar manipulator with 3 DOFs is used to verify the inverse kinematics method based on joint perturbation. The schematic diagram of the manipulator mechanism is shown in Fig. 4. If only the position of the manipulator is considered, the mechanism is a planar manipulator with 1 redundant DOF.

The manipulator’s position coordinates (x, y) can be obtained with the forward kinematics model.

$$\begin{cases} x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{cases} \quad (14)$$

The joint perturbation coefficient matrix of the manipulator is

$$K_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \quad (15)$$

The motion priorities of the three joints are k_1, k_2 , and k_3 . The coefficient matrix considering the joint motion priorities is

$$K_3 = \begin{bmatrix} k_1 & k_1 & k_1 & k_1 & -k_1 & -k_1 & -k_1 & -k_1 \\ k_2 & k_2 & -k_2 & -k_2 & k_2 & k_2 & -k_2 & -k_2 \\ k_3 & -k_3 & k_3 & -k_3 & k_3 & -k_3 & k_3 & -k_3 \end{bmatrix}. \quad (16)$$

Set the link lengths of the planar redundant manipulator as $L_1 = 300, L_2 = 240$, and $L_3 = 180$ mm, and the current joint angles are $\theta_{01} = 60, \theta_{02} = -30$, and $\theta_{03} = -30^\circ$. The current manipulator position coordinates $P_0(x_0, y_0)$ can be calculated with the current joint angles, with $x_0 = 537.8461$ and $y_0 = 379.8076$ mm. The target position coordinates are set to be $P_t(437.8461, 179.8076)$, as shown in Fig. 5.

Let the motion priorities of the three joints be $k_1 = 0.6, k_2 = 0.8$, and $k_3 = 1.0$, and set the allowable error $e = 0.01$ mm and the basic perturbation angle as $\delta\theta = 4.34 \times 10^{-4}$. When the joint motion priorities are considered, the coefficient matrix K_3 of the manipulator is

$$K_3 = \begin{bmatrix} 0.6 & 0.6 & 0.6 & 0.6 & -0.6 & -0.6 & -0.6 & -0.6 \\ 0.8 & 0.8 & -0.8 & -0.8 & 0.8 & 0.8 & -0.8 & -0.8 \\ 1.0 & -1.0 & 1.0 & -1.0 & 1.0 & -1.0 & 1.0 & -1.0 \end{bmatrix}. \quad (17)$$

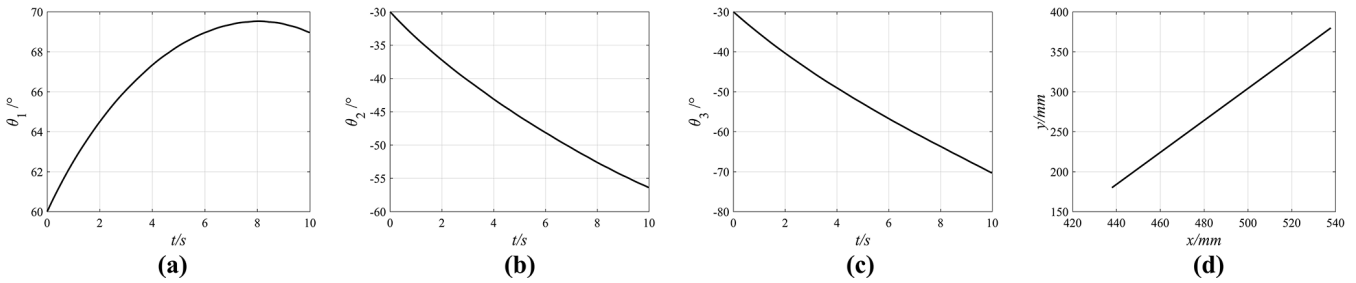


Figure 6. Motion curves of the joints and the manipulator end. (a) The first joint. (b) The second joint. (c) The third joint. (d) Trajectory of the manipulator end.

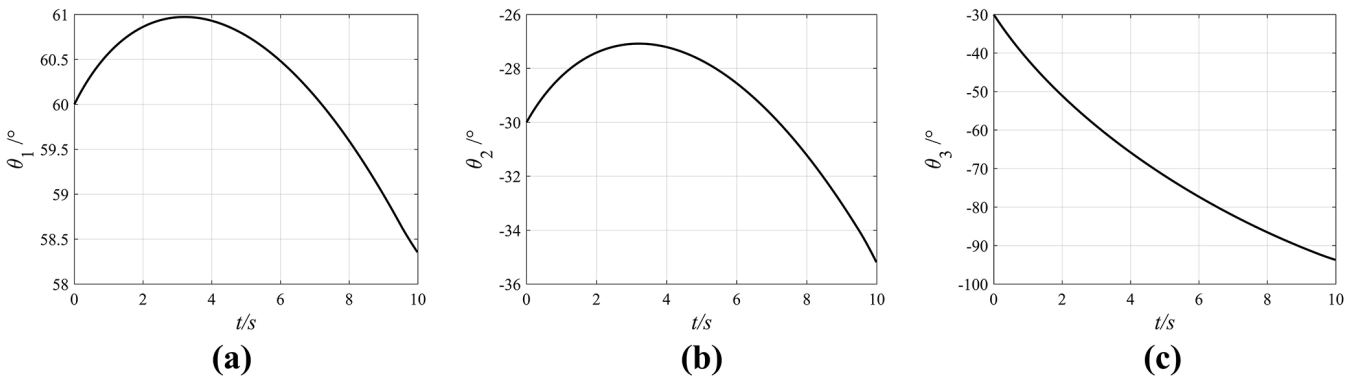


Figure 7. Motion curves of the three joints ($k_1 = 0.2, k_2 = 0.6, k_3 = 1$). (a) The first joint. (b) The second joint. (c) The third joint.

Therefore, the perturbation angle of each joint can be calculated as

$$\Delta q = \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{pmatrix} = \begin{pmatrix} \pm k_1 \delta\theta \\ \pm k_2 \delta\theta \\ \pm k_3 \delta\theta \end{pmatrix} = \begin{pmatrix} \pm 2.60 \times 10^{-4} \text{ }^\circ \\ \pm 3.47 \times 10^{-4} \text{ }^\circ \\ \pm 4.34 \times 10^{-4} \text{ }^\circ \end{pmatrix}. \tag{18}$$

Suppose that the manipulator takes $t = 10\text{s}$ to move from $P_0(x_0, y_0)$ to the target position $P_t(x_t, y_t)$ along a straight line at a constant speed. The position changes of the manipulator $\Delta x = x_t - x_0 = -100$ and $\Delta y = y_t - y_0 = -200$ mm are given, and thus, the distance between the initial position and the target position is

$$l = \sqrt{(x_t - x_0)^2 + (y_t - y_0)^2} = 223.6068 \text{ mm}. \tag{19}$$

Therefore, the velocity of the manipulator end is $v = l/t$. The linear interpolation is performed in the motion path, and the interpolation period is taken as $\Delta t = 10 \text{ ms}$. So, the number of interpolations h is

$$h = \frac{t}{\Delta t} = 1000. \tag{20}$$

Therefore, in an interpolation period, the displacement changes in x and y directions are as follows:

$$dx = \frac{\Delta x}{h} = \frac{x_t - x_0}{h} = -0.1 \text{ mm} \tag{21}$$

$$dy = \frac{\Delta y}{h} = \frac{y_t - y_0}{h} = -0.2 \text{ mm}. \tag{22}$$

Calculate the path point $P_b(x_b, y_b)$ of the manipulator from the initial position $P_0(x_0, y_0)$ to the target position $P_t(x_t, y_t)$.

$$\begin{cases} x_b = x_0 + b \cdot dx \\ y_b = y_0 + b \cdot dy \end{cases} \quad (b = 1, 2, 3, \dots, h) \tag{23}$$

The first path point $P_1(x_1, y_1)$ is determined, where $x_1 = x_0 + dx$ and $y_1 = y_0 + dy$. Next, $q_m = (\theta_{m1}, \theta_{m2}, \theta_{m3})$ and $P_m(x_m, y_m)$ can be obtained through iterative calculations from the initial position $P_0(x_0, y_0)$ to the first path point $P_1(x_1, y_1)$. Further, q_m and $P_m(x_m, y_m)$ are assigned to q_0 and $P_0(x_0, y_0)$, respectively, and then iterated to the second path point $P_2(x_2, y_2)$, where $x_2 = x_1 + dx$, $y_2 = y_1 + dy$. The iterative calculations and solutions will be repeated until $D = \sqrt{(x_m - x_t)^2 + (y_m - y_t)^2} \leq e$.

In the process of iteration from the initial position $P_0(x_0, y_0)$ to the target position $P_t(x_t, y_t)$, the joint angles corresponding to each path point can be calculated. The joint

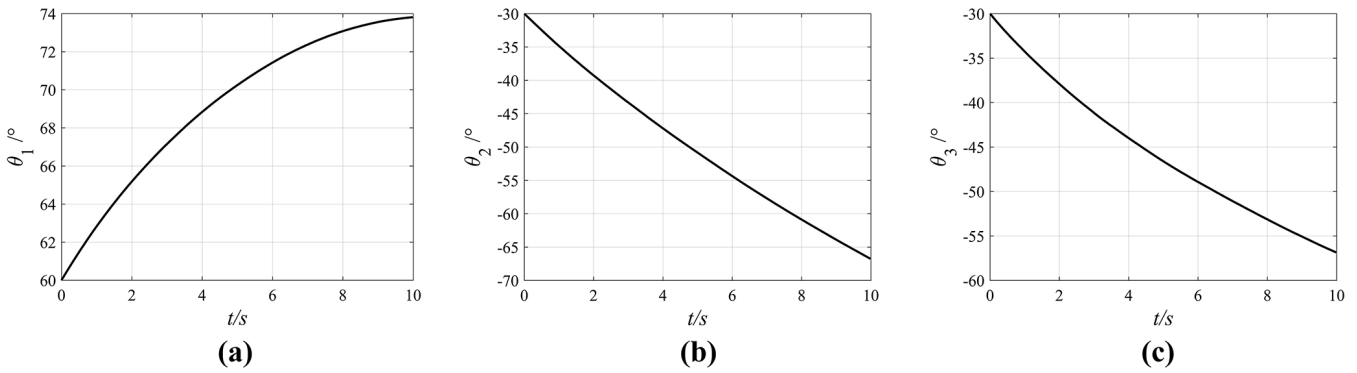


Figure 8. Motion curves of the three joints ($k_1 = k_2 = k_3 = 1$). (a) The first joint. (b) The second joint. (c) The third joint.

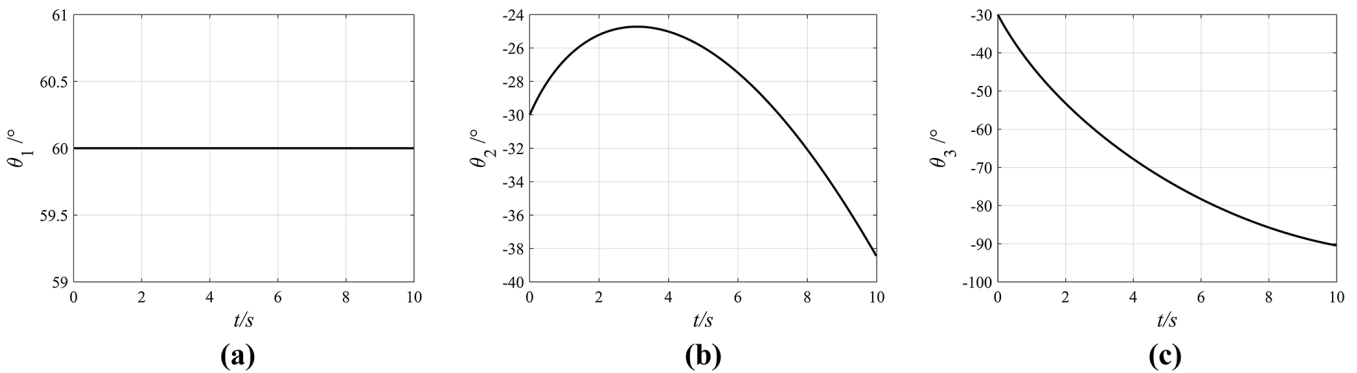


Figure 9. Motion curves of the three joints ($k_1 = 0, k_2 = k_3 = 1$). (a) The first joint. (b) The second joint. (c) The third joint.

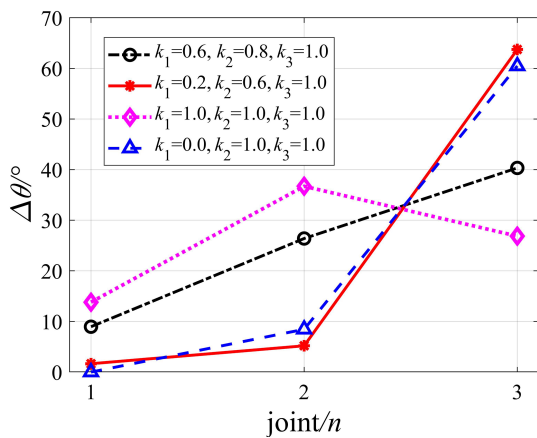


Figure 10. Variations of joint motions with different motion priorities.

motion curves of the entire process are obtained after the iteration is completed, as shown in Fig. 6, where the abscissa represents the motion time and the ordinate represents the corresponding joint angles.

Figure 6 shows that the angle variations of the three joints are $\Delta\theta_1 = 8.96$, $\Delta\theta_2 = 26.39$, and $\Delta\theta_3 = 40.33^\circ$, which comply with the setting of the motion priorities of the three

joints, and the manipulator can move smoothly in the joint space.

Change the motion priority of each joint, by setting $k_1 = 0.2, k_2 = 0.6, k_3 = 1, k_1 = k_2 = k_3 = 1$, and $k_1 = 0, k_2 = k_3 = 1$ while keeping other parameters unchanged. The effect of the joint motion priorities on joint motions is obtained through simulations. The joint motion curves under different motion priorities are shown in Figs. 7, 8, and 9.

After analysing the joint motion curves in the four cases, the angle variation of each joint under different motion priorities can be obtained, as shown in Fig. 10, where the abscissa represents the three joint motions of the manipulator and the ordinate represents the magnitude of the angle variation of each joint. The specific values are shown in Table 1. In this Table, θ_{01}, θ_{02} , and θ_{03} represent the initial joint angles; θ_{t1}, θ_{t2} , and θ_{t3} represent the joint angles corresponding to the target position; and $\Delta\theta_1, \Delta\theta_2$, and $\Delta\theta_3$ represent the angle variations of three joints, $\Delta\theta_i = |\theta_{ti} - \theta_{0i}|$ ($i = 1, 2, 3$).

The effect of the joint motion priority on joint motions can be obtained according to Table 1 and Fig. 11. Figure 11a and b show that the motion priority of the first joint decreases by $\Delta k_1 = 0.4$, and the variation of joint motion changes from 8.96 to 1.65° with a decrease of 81.58%; the motion priority of the second joint decreases by $\Delta k_2 = 0.2$, and the corresponding variation of joint motion changes from 26.39 to

Table 1. Comparisons of joint motions with different motion priorities.

Priorities of joint motions	Joint 1			Joint 2			Joint 3		
	θ_{01}	θ_{t1}	$\Delta\theta_1$	θ_{02}	θ_{t2}	$\Delta\theta_2$	θ_{03}	θ_{t3}	$\Delta\theta_3$
$k_1 = 0.6, k_2 = 0.8, k_3 = 1$	60°	68.96°	8.96°	-30°	-56.39°	26.39°	-30°	-70.33°	40.33°
$k_1 = 0.2, k_2 = 0.6, k_3 = 1$	60°	58.35°	1.65°	-30°	-35.19°	5.19°	-30°	-93.73°	63.73°
$k_1 = 1, k_2 = 1, k_3 = 1$	60°	73.81°	13.81°	-30°	-66.71°	36.71°	-30°	-56.86°	26.86°
$k_1 = 0, k_2 = 1, k_3 = 1$	60°	60.00°	0°	-30°	-38.45°	8.45°	-30°	-90.47°	60.47°

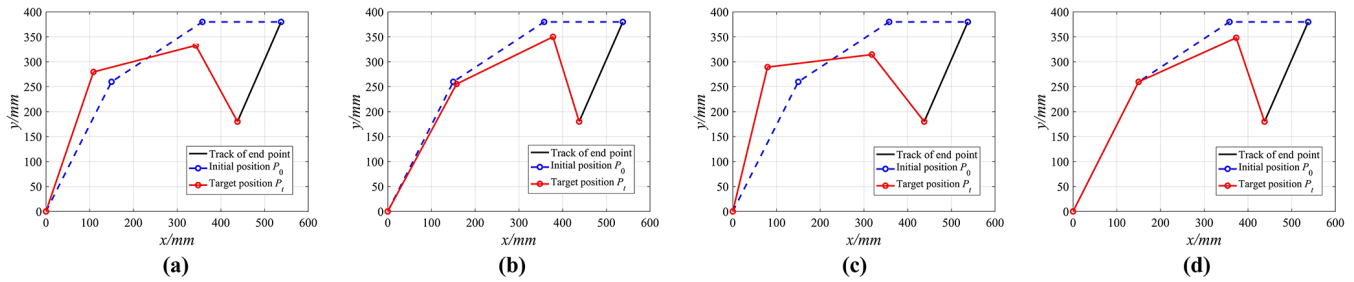


Figure 11. Motion trajectories of the manipulator with different motion priorities. (a) $k_1 = 0.6, k_2 = 0.8, k_3 = 1.0$. (b) $k_1 = 0.2, k_2 = 0.6, k_3 = 1.0$. (c) $k_1 = k_2 = k_3 = 1$. (d) $k_1 = 0, k_2 = k_3 = 1$.

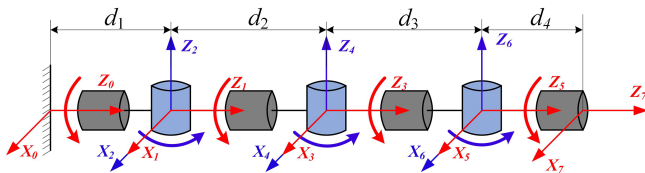


Figure 12. D–H model of a manipulator with 7 DOFs.

Table 2. D–H parameters of a manipulator with 7 DOFs.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1
2	$\pi/2$	0	0	θ_2
3	$-\pi/2$	0	d_2	θ_3
4	$\pi/2$	0	0	θ_4
5	$-\pi/2$	0	d_3	θ_5
6	$\pi/2$	0	0	θ_6
7	$-\pi/2$	0	d_4	θ_7

5.19° with a decrease of 80.33 %, while the variation of the third joint motion changes from 40.33 to 63.73° with an increase of 58.02 %. Figure 11a and c show that the motion priority of the first joint increases by $\Delta k_1 = 0.4$, and the variation of joint motion changes from 8.96 to 13.81° with an increase of 54.13 %; the motion priority of the second joint increases by $\Delta k_2 = 0.2$, and the corresponding variation of joint motion changes from 26.39 to 36.71° with an increase of 39.11 %, while the variation of the third joint motion changes from 40.33 to 26.86° with a decrease of 33.34 %.

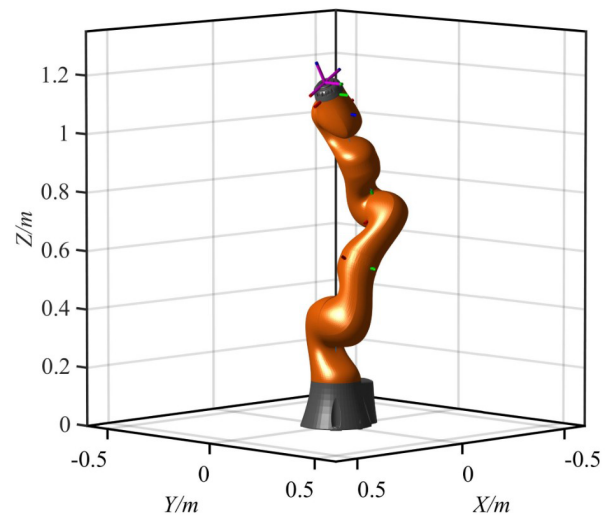


Figure 13. Initial position of iiwa manipulator.

Figure 11d shows that when the motion priority of the first joint is 0 ($k_1 = 0$), the first joint will remain stationary, and the corresponding motion is completed by the second joint and the third joint.

3.2 Simulation verification of a manipulator with 7 DOFs

To verify the universality of the algorithm, a redundant manipulator with 7 DOFs (KUKA iiwa) is used for simulation verification. First, the model of the redundant manipulator is obtained using the D–H method, as shown in Fig. 12, where

Table 3. Setting of joint motion priorities.

Group	k_1	k_2	k_3	k_4	k_5	k_6	k_7
1	1	1	1	1	1	1	1
2	0.4	0.5	0.6	0.7	0.8	0.9	1
3	1	0.5	1	1	1	1	1
4	0	1	1	1	1	1	1
5	1	1	0	1	1	1	1

$d_1 = 340$, $d_2 = 400$, $d_3 = 400$, and $d_4 = 126.6$ mm. The D–H parameters are shown in Table 2.

Set the initial joint values as $q_0 = (0, \pi/6, 0, -\pi/3, 0, 0, 0)$, and the coordinates of the manipulator end can be obtained by the forward kinematics, $P_0 = [63.301142.5]$ mm, as shown in Fig. 13.

The manipulator will take $t = 10$ s to move from the initial position to the target position $P_t = [263.3 - 400542.5]$. Set the allowable error as $e = 0.01$ mm, and calculate the basic perturbation angle $\delta\theta = 3.92 \times 10^{-5}^\circ$.

The interpolation period is set as $\Delta t = 10$ ms, and thus the number of interpolations is $h = 1000$. Therefore, the displacement changes in x , y , and z directions are $dx = 0.2$, $dy = -0.4$, and $dz = -0.6$ mm, respectively. The simulations are divided into five groups for comparisons, and the joint motion priorities of each group are set as shown in Table 3.

Using the motion priorities of the five groups while keeping other parameters unchanged, the joint motion curves under different motion priorities are shown in Figs. 14, 15, 16, 17, and 18.

The angle variation of each joint under different motion priorities can be obtained, as shown in Table 4. In this table, θ_{0i} represents the initial joint angles, θ_{ti} represents the joint angles corresponding to the target position, and $\Delta\theta_i$ represents the angle variations of seven joints, $\Delta\theta_i = |\theta_{ti} - \theta_{0i}| (i = 1, 2, \dots, 7)$.

Because the joint motion priorities are the same in Group 1, the motion angle of each joint is relatively averaged. In Group 2, the motion priorities of big joints are relatively low, while the motion priorities of small joints are relatively high. Then, the motion amplitudes of the first and the second joints decrease significantly. Comparing Groups 1 and 3 shows that the motion priority of the second joint is reduced to 0.5, and the corresponding angle variation is decreased from 35.47 to 7.15°. In Groups 4 and 5, the motion priorities of the first joint and the third joint are set as 0, so the corresponding joints remain stationary.

Solving inverse kinematics of redundant manipulators with 3 DOFs and 7 DOFs shows consistent simulation results. The higher the relative joint motion priority, the bigger the joint motion angle, and in contrast, the lower the relative joint motion priority, the smaller the joint motion angle. When the joints have the same priority, the motion angle of

each joint will be relatively averaged. In particular, when the motion priority of a joint is zero, the joint motion angle is zero, too. Therefore, the motion range of a joint can be controlled by changing the motion priority. Hence, the joint perturbation method can be used to deal with joint limit avoidance when calculating the inverse kinematics solution of a redundant manipulator. For a manipulator with n DOFs, assume that the motion range limit of joint i is $[q_{\min,i}, q_{\max,i}]$, then the joint angles q_i satisfy the following conditions:

$$q_{\min,i} \leq q_i \leq q_{\max,i}. \tag{24}$$

Normally, the constraint interval of a joint angle is symmetrical; that is $q_{\max,i} = -q_{\min,i}$. Then, the motion priority of joint i can be defined as

$$k_i = \frac{\min\{|q_{\max,i} - q_i|, |q_i - q_{\min,i}|\}}{\left(\frac{q_{\max,i} - q_{\min,i}}{2}\right)}. \tag{25}$$

Therefore, k_i approaches 0 when q_i approaches the joint limit; $k_i = 1$ when $q_i = \frac{q_{\max,i} + q_{\min,i}}{2}$.

3.3 Simulation experiment based on CoppeliaSim

Take the manipulator with 7 DOFs, KUKA iiwa, to perform the same motion task of Sect. 3.2 in CoppeliaSim, of which simulation results are widely acknowledged to have a high consistency with actual experiments. The joint motion ranges of the manipulator are shown in Table 5, and the real-time joint motion priorities are calculated according to formula (25). The joint curves are obtained as shown in Fig. 19a. All joints do not exceed their motion ranges.

The joint curves were imported into CoppeliaSim as the driving function to complete the experimental tasks, as shown in Fig. 19b. During the movement, the manipulator ran smoothly and completed the task of a linear trajectory.

3.4 Real-time characteristic analyses

The Jacobian pseudo-inverse method is widely used to solve the inverse kinematics of redundant manipulators. The real-time characteristic comparisons will be conducted between it and the proposed method. The inverse kinematics solution of the redundant manipulator with n DOFs is

$$\dot{q} = J^+ \dot{p} + \alpha(I - J^+ J)z, \tag{26}$$

where \dot{q} is the joint velocity, \dot{p} is the velocity of the manipulator, J^+ is the generalized inverse matrix of J , α is the scalar coefficient, $z \in R^{n \times 1}$ is an arbitrary vector, and $\alpha(I - J^+ J)z$ is the null space vector which represents the self-motion of the manipulator in the null space and satisfies $\alpha J(I - J^+ J)z = 0$.

All the conditions and motion tasks of the two manipulators (3 DOFs and 7 DOFs) are the same as those of the two in Sect. 3.1 and 3.2 respectively. In this paper, the minimum

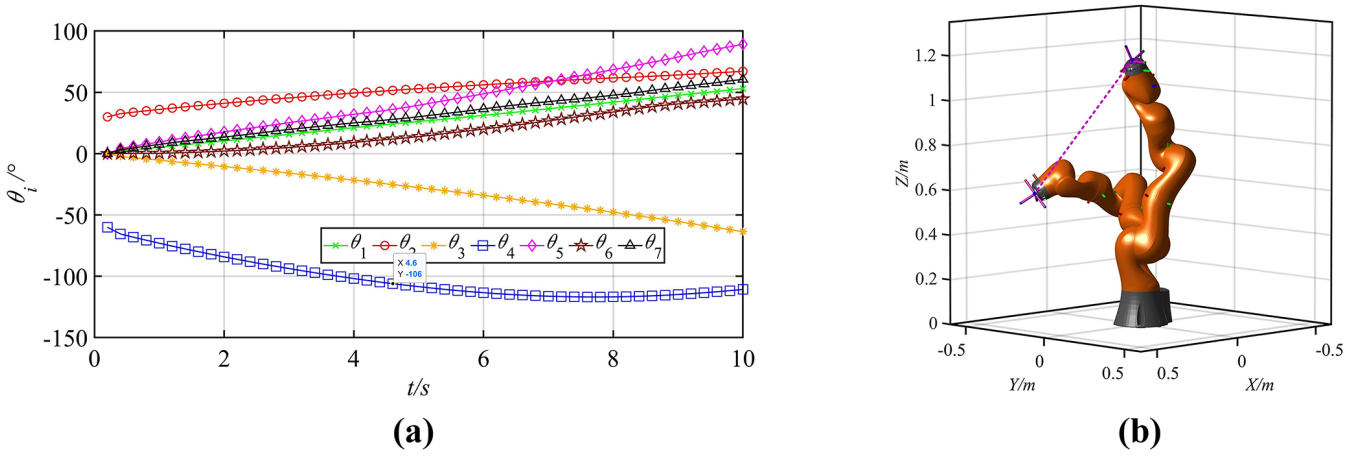


Figure 14. Motion curves of the joints and the manipulator ($k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = k_7 = 1.0$). (a) Joint motion curves. (b) Trajectory of the manipulator.

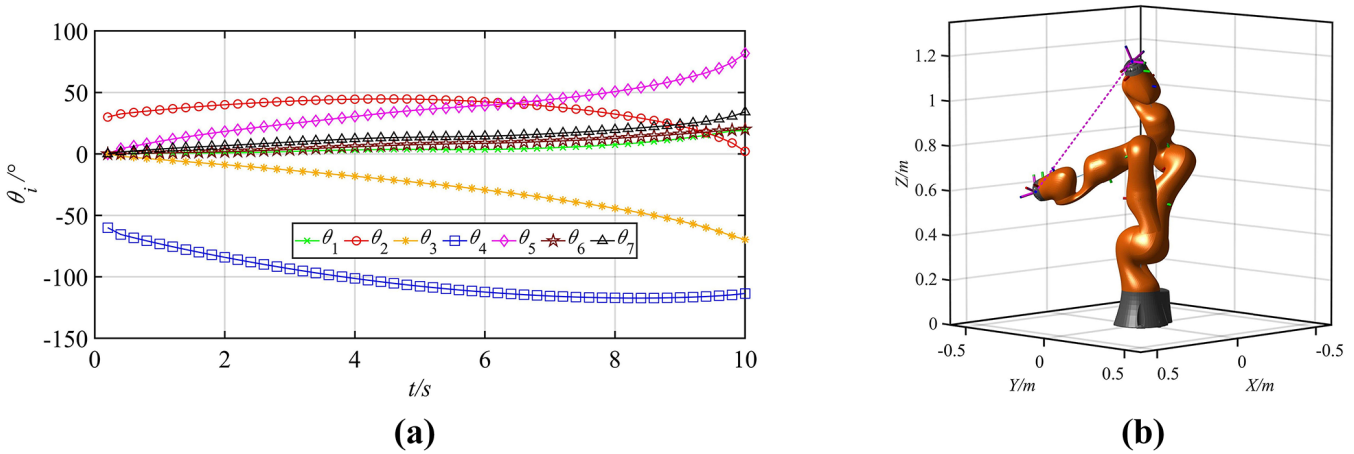


Figure 15. Motion curves of the joints and the manipulator ($k_1 = 0.4, k_2 = 0.5, k_3 = 0.6, k_4 = 0.7, k_5 = 0.8, k_6 = 0.9, k_7 = 1.0$). (a) Joint motion curves. (b) Trajectory of the manipulator.

norm solution of joint velocity, $\dot{q} = J^+ \dot{p}$, is adopted, and the joint motion curves are obtained as shown in Figs. 20 and 21.

For the planar manipulator with 3 DOFs and the space manipulator with 7 DOFs, different solution accuracy values ($e = 0.01$ and $e = 0.001$ mm, respectively) are set. The solving times of the two methods are calculated with MATLAB, and the results are shown in Table 6. According to Table 6, both methods can reach the microsecond level, and the solving time of the perturbation method t_j is shorter than that of the Jacobian pseudo-inverse method t_m . So, the joint perturbation method has better efficiency and real-time performance than the Jacobian pseudo-inverse method. However, with the improvement of the solution accuracy and the increase of the degrees of freedom, the calculation times of the both methods increase significantly, and the preponderance of the joint perturbation method over the Jacobian pseudo-inverse method has a downward trend.

4 Conclusions

The inverse kinematics solution of redundant manipulators usually takes a considerable amount of time to calculate. Aiming at improving the solving efficiency of inverse kinematics, this paper first defines the concepts of joint motion priority and joint perturbation coefficient matrix, then proposes an algorithm based on joint perturbation to solve the inverse solution of redundant manipulators. The feasibility and real-time characteristics of the algorithm were verified through simulations of two types of redundant manipulators with 3 DOFs and 7 DOFs, respectively. The following conclusions are obtained:

1. The magnitude of motion priority determines the level of joint motion priority. When a redundant manipulator performs a specific task, a higher joint motion priority leads to a bigger joint motion angle, while a lower joint motion priority results in a smaller joint motion angle.

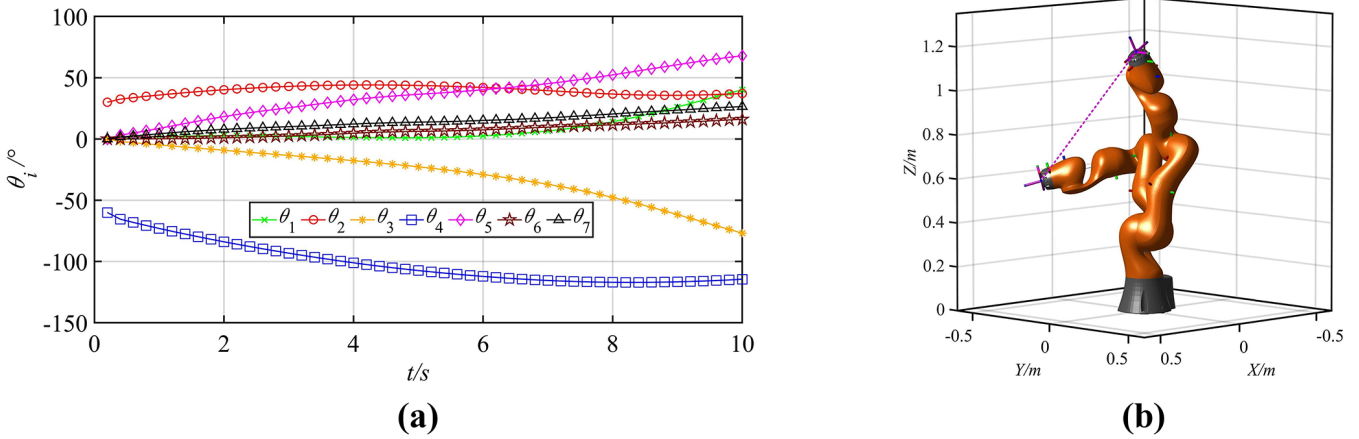


Figure 16. Motion curves of the joints and the manipulator ($k_2 = 0.5, k_1 = k_3 = k_4 = k_5 = k_6 = k_7 = 1.0$). (a) Joint motion curves. (b) Trajectory of the manipulator.

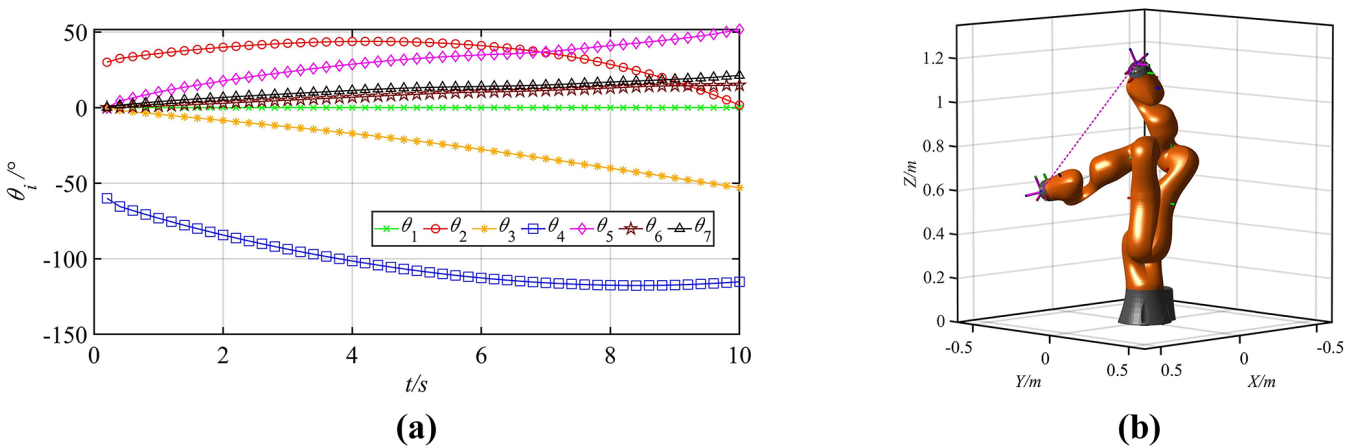


Figure 17. Motion curves of the joints and the manipulator ($k_1 = 0, k_2 = k_3 = k_4 = k_5 = k_6 = k_7 = 1.0$). (a) Joint motion curves. (b) Trajectory of the manipulator.

In particular, when the joint motion priority is zero, the joint motion angle is also zero. Therefore, the motion of a joint can be controlled by setting different joint motion priorities, which can be used in obstacle avoidance and joint limit avoidance.

2. The proposed algorithm has universality and has no restrictions on the structure and DOFs of manipulators. Moreover, the solving process of this algorithm only involves the forward kinematics, which can avoid the pseudo-inverse calculations of the Jacobian matrix and significantly reduce the solving complexity, so it has a better real-time performance than the Jacobian pseudo-inverse method. The method can be used in real applications to speed up the work cycle and improve production efficiency.
3. However, the proposed algorithm also has certain limitations, and the calculation load will increase exponen-

tially with the increase of the degrees of freedom of a manipulator. In the future, we will continue to study the algorithm in the optimization of joint motion priorities and experimental verifications.

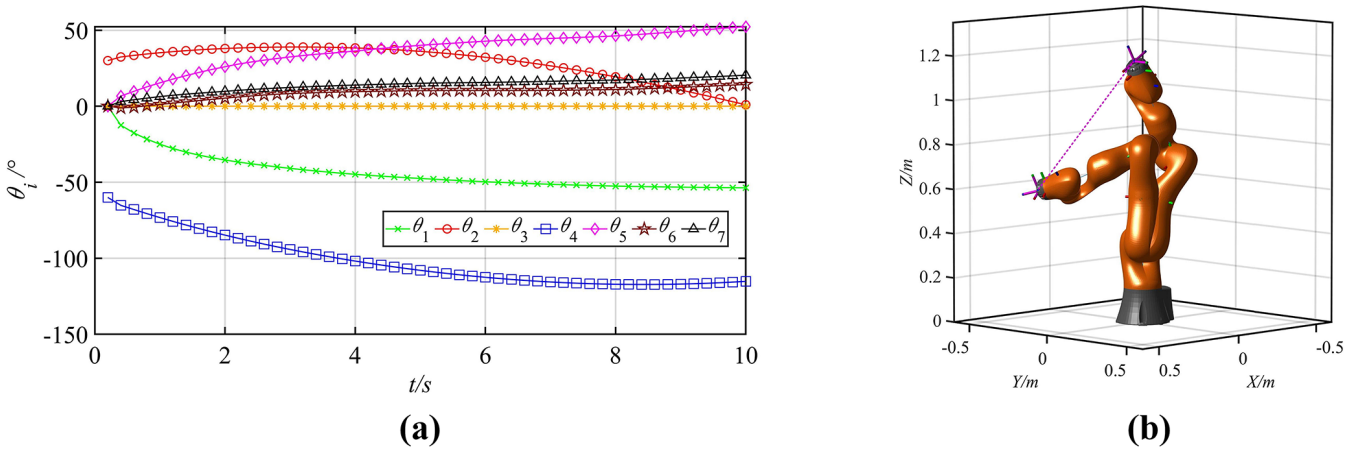


Figure 18. Motion curves of the joints and the manipulator ($k_3 = 0, k_1 = k_2 = k_4 = k_5 = k_6 = k_7 = 1.0$). (a) Joint motion curves. (b) Trajectory of the manipulator.

Table 4. Comparisons of joint motions with different motion priorities.

Group	Angle variation of each joint $\Delta\theta_i = \theta_{ti} - \theta_{0i} $						
	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_4$	$\Delta\theta_5$	$\Delta\theta_6$	$\Delta\theta_7$
1	54.01°	35.47°	63.63°	48.15°	97.70°	50.87°	63.58°
2	18.93°	28.31°	69.75°	53.45°	82.19°	20.08°	34.27°
3	40.27°	7.15°	76.99°	54.43°	68.01°	16.13°	26.45°
4	0°	28.50°	53.09°	55.24°	51.78°	14.87°	21.23°
5	53.68°	29.13°	0°	55.18°	52.37°	14.34°	20.44°

Table 5. The joint motion ranges of the manipulator.

Joint i	1	2	3	4	5	6	7
Motion range (°)	(-170, 170)	(-120, 120)	(-170, 170)	(-120, 120)	(-170, 170)	(-120, 120)	(-175, 175)

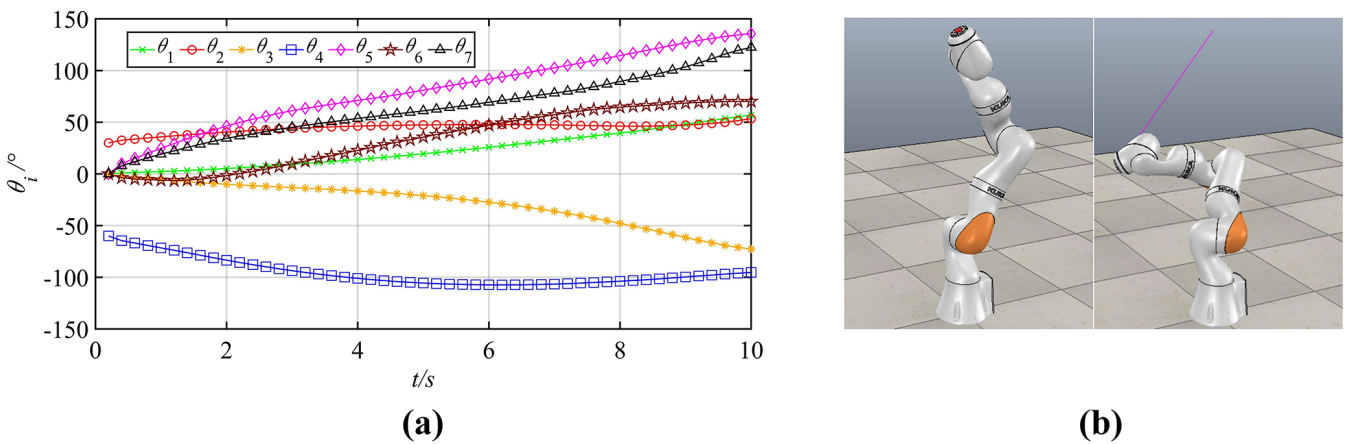


Figure 19. Joint motion curves and simulation process. (a) Joint motion curves. (b) Initial and target poses.

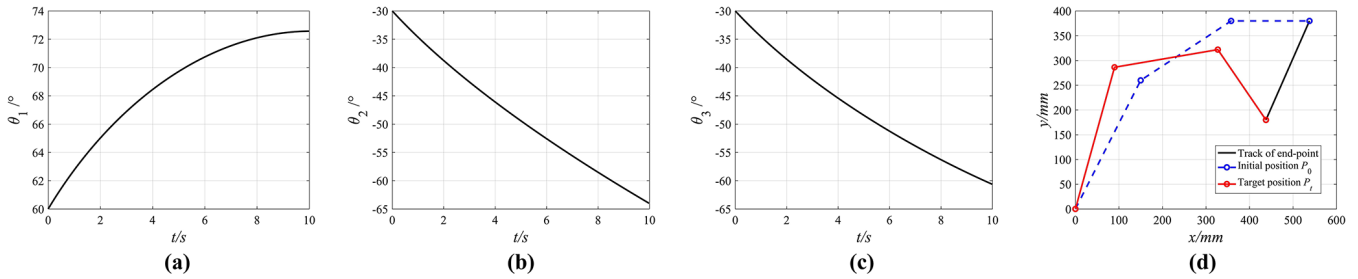


Figure 20. Motion curves of the joints and the manipulator (3 DOFs). (a) The first joint. (b) The second joint. (c) The third joint. (d) Trajectory of the manipulator.

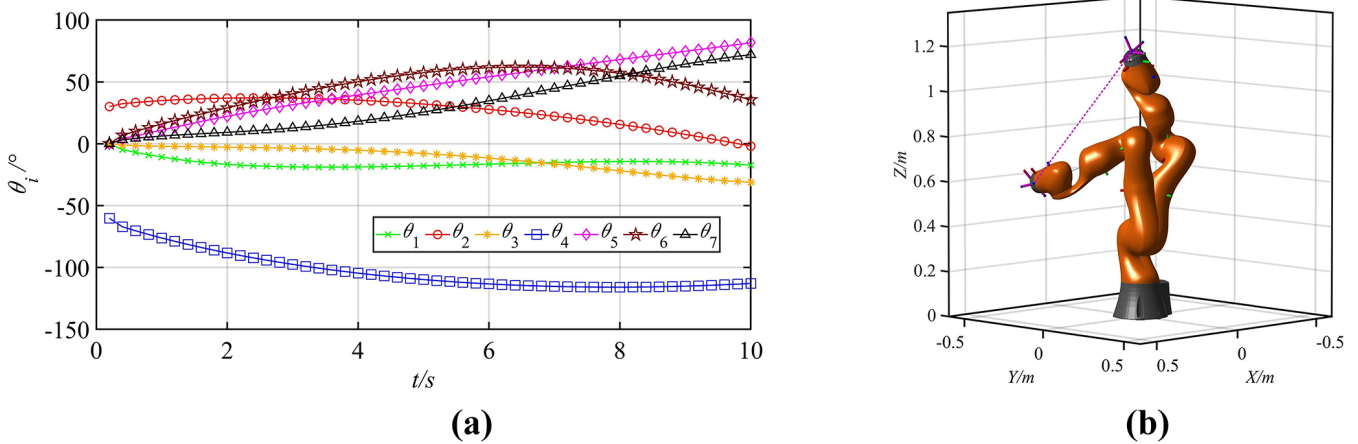


Figure 21. Motion curves of the joints and the manipulator (7 DOFs). (a) Joint motion curves. (b) Trajectory of the manipulator.

Table 6. Solving time for the Jacobian pseudo-inverse method and the perturbation method.

Group	Solution accuracy e in mm	Solving time t_m in ms (pseudo-inverse)	Solving time t_j in ms (joint perturbation)	$\frac{t_m - t_j}{t_m} \times 100\%$
Three DOFs	0.01	0.17	0.04	76.47 %
	0.001	0.48	0.15	68.75 %
Seven DOFs	0.01	26.2	17.7	32.44 %
	0.001	55.7	46.3	16.87 %

Data availability. All the data used in this paper can be obtained upon request from the corresponding author.

Author contributions. QX contributed to the conception of the study, performed the validation, and wrote the original draft. QZ contributed to the funding acquisition and project administration. QX and QZ jointly completed data curation, review, and editing.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. Thanks are due to Xinyang Tian for assistance with the simulation and to Xiangzhen Chen for valuable discussion. Thanks are expressed to the research platform provided by the Laboratory of Complex Mechanism and Intelligent Control.

Financial support. This research has been supported by the Beijing Natural Science Foundation, Haidian Original Innovation Joint Fund Project (grant no. L172015).

Review statement. This paper was edited by Daniel Condurache and reviewed by Giuseppe Carbone, Miguel Díaz-Rodríguez, and one anonymous referee.

References

- Akbaripour, H. and Masehian, E.: Semi-lazy probabilistic roadmap: a parameter-tuned, resilient and robust path planning method for manipulator robots, *Int. J. Adv. Manuf. Technol.*, 89, 1401–1430, <https://doi.org/10.1007/s00170-016-9074-6>, 2017.
- Aouache, M., Maoudj, A., and Akli, I.: Human-robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017, *Adv. Robotics*, 33, 764–799, <https://doi.org/10.1080/01691864.2019.1636714>, 2019.
- Ayusawa, K. and Nakamura, Y.: Fast inverse kinematics algorithm for large DOF system with decomposed gradient computation based on recursive formulation of equilibrium, in: *Proc. IROS, Vilamoura, Portugal*, 3447–3452, <https://doi.org/10.1109/IROS.2012.6385780>, 2012.
- Damas, B. J. and Santos, V.: An online algorithm for simultaneously learning forward and inverse kinematics, in: *Proc. IROS, Vilamoura, Portugal*, 1499–1506, <https://doi.org/10.1109/IROS.2012.6386156>, 2012.
- Dasgupta, B., Gupta, A., and Singla, E.: A variational approach to path planning for hyper-redundant manipulators, *Robot. Auto. Syst.*, 2, 194–201, <https://doi.org/10.1016/j.robot.2008.05.001>, 2009.
- Faria, C., Ferreira, F., Erhagen, W., Sérgio, M., and Estela, B.: Position-based kinematics for 7-dof serial manipulators with global configuration control, joint limit and singularity avoidance, *Mech. Mach. Theory.*, 121, 317–334, <https://doi.org/10.1016/j.mechmachtheory.2017.10.025>, 2018.
- Hayashibe, M., Suzuki, N., and Hashizume, M.: Robotic surgery setup simulation with the integration of inverse-kinematics computation and medical imaging, *Comput. Meth. Prog. Bio.*, 1, 63–72, <https://doi.org/10.1016/j.cmpb.2006.04.010>, 2006.
- Jia, Q. X., Zhang, L., Chen, G., and Sun, H. X.: Pre-impact trajectory optimization of redundant space manipulator with multi-target fusion, *J. Astronautics*, 6, 639–647, <https://doi.org/10.3873/j.issn.1000-1328.2014.06.004>, 2014.
- Kalra, P., Mahapatra, P. B., and Aggareal, D. K.: An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots, *Mech. Mach. Theory*, 41, 1213–1229, <https://doi.org/10.1016/j.mechmachtheory.2005.11.005>, 2006.
- Korein, J. U. and Badler, N. I.: Techniques for generating the goal-directed motion of articulated structures, *IEEE Comput. Graph.*, 9, 71–81, <https://doi.org/10.1109/MCG.1982.1674498>, 1982.
- Jiokou Kouabon, A. G., Melingui, A., Mvogo Ahanda, J. J. B., Lakhali, O., Coelen, V., Kom, M., and Merzouki, R.: A Learning Framework to inverse kinematics of high DOF redundant manipulators, *Mech. Mach. Theory*, 153, 103978, <https://doi.org/10.1016/j.mechmachtheory.2020.103978>, 2020.
- Kuhlemann, I., Schweikard, A., Jauer, P., and Ernst, F.: Robust inverse kinematics by configuration control for redundant manipulators with seven DOF, in: *Proc. International Conference on Control, Automation and Robotics, Hong Kong*, 49–55, <https://doi.org/10.1109/ICCAR.2016.7486697>, 2016.
- Kumar, S., Sukavanam, N., and Balasubramanian, R.: An optimization approach to solve the inverse kinematics of redundant manipulator, *Int. J. Inf. Syst. Sci.*, 4, 414–423, <https://doi.org/10.13140/2.1.2666.9125>, 2010.
- Liu, W., Chen, D., and Steil, J.: Analytical inverse kinematics solver for anthropomorphic 7-dof redundant manipulators with human-like configuration constraints, *J. Intell. Robot. Syst.*, 86, 63–79, <https://doi.org/10.1007/s10846-016-0449-6>, 2017.
- Lopezfranco, C., Hernandezbarragan, J., and Alanis, A. Y.: Inverse kinematics of mobile manipulators based on differential evolution, *Int. J. Adv. Robot. Syst.*, 1, 1–22, <https://doi.org/10.1177/1729881417752738>, 2018.
- Lv, Y. and Liu, F.: Application Analysis of Generalized and Modified D-H Method in Kinematic Modeling, *Comput. Syst. Appl.*, 5, 197–202, 2016.
- Maciejewski, A. A. and Klein, C. A.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *Int. J. Robot. Res.*, 3, 109–117, <https://doi.org/10.1177/027836498500400308>, 1985.
- Mohamed, H. A., Yahya, S., Moghavvemi, M., and Yang, S. S.: A new inverse kinematics method for three dimensional redundant manipulators, in: *Proc. IEEE ICCAS-SICE, Fukuoka, Japan*, 1557–1562, 2009.
- Omcrcen, D. and Zlajpah, L.: Compensation of velocity and/or acceleration joint saturation applied to redundant manipulator, *Robot. Auto. Syst.*, 55, 337–344, <https://doi.org/10.1016/j.robot.2006.10.001>, 2007.
- Park, H., Kwak, B., and Bae, J.: Inverse Kinematics Analysis and COG Trajectory Planning Algorithms for Stable Walking of a Quadruped Robot with Redundant DOFs, *J. Bionic Eng.*, 15, 610–622, <https://doi.org/10.1007/s42235-018-0050-8>, 2018.
- Tian, X. Y., Xu, Q. H., and Zhan, Q.: An analytical inverse kinematics solution with joint limits avoidance of 7-DOF anthropo-

- morphic manipulators without offset, *J. Frankl. Inst.*, 358, 1252–1272, <https://doi.org/10.1016/j.jfranklin.2020.11.020>, 2020.
- Tolani, D., Goswami, A., and Badler, N. I.: Real-time inverse kinematics techniques for anthropomorphic limbs, *Graph. Models*, 5, 353–388, <https://doi.org/10.1006/gmod.2000.0528>, 2000.
- Umar, A., Shi, Z., and Wang, W.: A novel mutating pso based solution for inverse kinematic analysis of multi degree-of-freedom robot manipulators, in: *Proc. ICAICA*, Dalian, China, 459–463, <https://doi.org/10.1109/ICAICA.2019.8873449>, 2019.
- Xia, Y. and Wang, J.: A dual neural network for kinematic control of redundant robot manipulators, *IEEE Trans. Syst. Man Cybern.*, 1, 147–154, <https://doi.org/10.1109/3477.907574>, 2001.
- Xu, H., Xie, X., and Zhuang, J.: Global Time-energy optimal planning of industrial robot trajectories, *J. Mech. Eng.*, 9, 19–25, <https://doi.org/10.3901/JME.2010.09.019>, 2010.
- Zaplana, I. and Basanez, L.: A novel closed-formed solution for the inverse kinematics of redundant manipulators through workspace analysis, *Mech. Mach. Theory*, 121, 829–843, <https://doi.org/10.1016/j.mechmachtheory.2017.12.005>, 2018.
- Zeng, Z., Chen, Z., Shu, G., and Chen, Q.: Optimization of analytical inverse kinematic solution for redundant manipulators using GA-PSO algorithm, *IEEE Industrial Cyber-Physical Systems*, 5, 446–451, <https://doi.org/10.1109/ICPHYS.2018.8390746>, 2018.
- Zu, D., Wu, Z., and Tan, D.: Efficient inverse kinematic solution for redundant manipulator, *China, J. Mech. Eng.-En.*, 6, 71–75, <https://doi.org/10.3901/JME.2005.06.071>, 2005.