Research Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2017, Vol. 9(8) 1–11
© The Author(s) 2017
DOI: 10.1177/1687814017714985
journals.sagepub.com/home/ade
SAGE

The inverse kinematics of a 7R 6-degree-of-freedom robot with non-spherical wrist

Xuhao Wang, Dawei Zhang and Chen Zhao

Abstract

The 7R 6-degree-of-freedom robots with hollow non-spherical wrist have been proven more suitable to spray painting. However, the inverse kinematics of this robot is still imperfect due to the coupling between position and orientation of the end-effector. In this article, a reliable numerical iterative algorithm for the inverse kinematics of a 7R 6-degree-of-freedom robot is proposed. Based on the geometry of the robot, the inverse kinematics is converted into a one-dimensional iterative research problem. Since the Jacobian matrix is not utilized, the proposed algorithm possesses good convergence, even for singular configurations. Moreover, the multiple-solution problem in the inverse kinematics is also discussed. By introducing three robot configuration indicators which are prespecified by a user, the correct solution could be chosen from all the possible solutions. In order to verify the accuracy and efficiency of the proposed algorithm, several simulations are implemented on a practical 7R 6-degree-of-freedom painting robot. The result shows that the proposed algorithm is more advantageous for a continuous trajectory.

Keywords

Inverse kinematics, iterative algorithm, 7R 6-degree-of-freedom robot, non-spherical wrist

Date received: 17 January 2017; accepted: 21 May 2017

Academic Editor: Yangmin Li

Introduction

In robotics, inverse kinematics is one of the most fundamental research areas. It is necessary for structure design, trajectory planning, and dynamic analysis of robots. There are mainly two types of inverse kinematic techniques, namely, analytical methods and numerical methods. Analytical solutions exist only for some special robots, including robot manipulators with three adjacent axes intersecting at one point or parallel to each other.[1] As an example, the robot with a spherical wrist is widely used to ensure the existence of analytical solution. However, in some industrial applications such as painting, the 7R 6-degree-of-freedom (DOF) robot with hollow non-spherical wrist has been proved to be advantageous due to wider range of motion of the wrist. As shown in Figure 1, the wrist of 7R 6-DOF robot has four revolute joints, the second and third of which are coupled with the relation: $\theta_6 = -\theta_5$. Figure 2 is the configuration of the 7R 6-DOF robot.

Because of the non-spherical wrist, the analytical solution of the 7R 6-DOF robot is non-existent. In order to get inverse kinematics of the robots with non-spherical wrist, Tsai and Morgan[2] proposed a higher dimensional approach with eight second-degree equations. Complementing this approach, Raghavan and

Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, School of Mechanical Engineering, Tianjin University, Tianjin, China

Corresponding author:
Chen Zhao, Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, School of Mechanical Engineering, Tianjin University, Tianjin 300072, China.
Email: zhaochen@tju.edu.cn

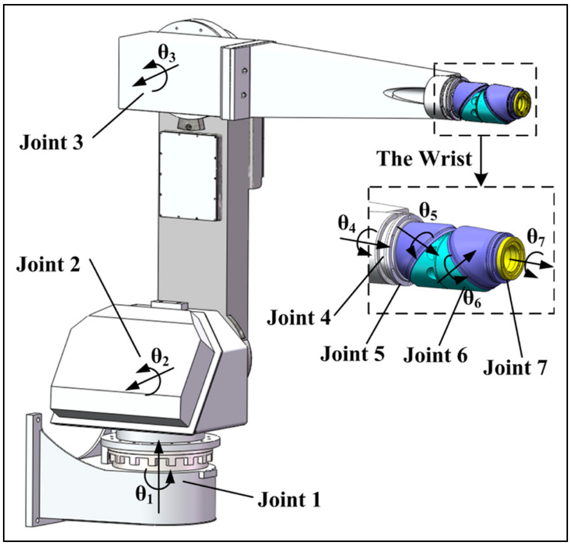

Figure 1. Prototype of a 7R 6-DOF painting robot.

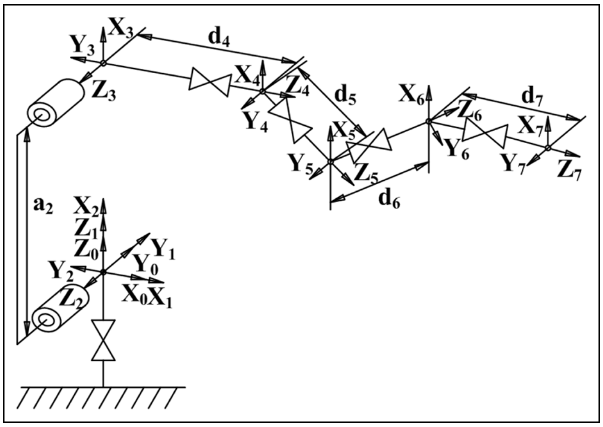

Figure 2. Configuration of the 7R 6-DOF robot.

Roth[3] used dialytic elimination to derive a 16-degree polynomial. Manocha and Canny[4] proposed symbolic preprocessing and matrix computations to convert the inverse kinematics to an eigenvalue problem. In recent years, literatures[5–8] have studied the inverse kinematics of general 6R robots. However, the problem is that these methods are limited to 6R robots. For the 7R 6-DOF robot, Wu et al.[9] proposed a two-step method: the approximate analytical solutions are first derived through solving the 7R robot with spherical wrist by introducing a virtual wrist center, and the Levenberg–Marquardt (LM) method is used to calculate the exact solutions. This is an interesting approach, but a complex polynomial system needs to be solved in the first step which is time consuming. In some references, the methods based on heuristic search techniques such as neural network solution,[10,11] genetic algorithms,[12,13] and simulated annealing[14] are developed for the solution of inverse kinematics problem. These methods converse kinematics problem into an equivalent minimization problem and generally suffer from low accuracy and are time consuming.

In the absence of analytical solution, the numerical iterative techniques are potential ways for the inverse kinematics. Several types of numerical iterative techniques have been developed. The first type, such as Newton–Raphson approach[15] and the damped least-squares (DLS) approach,[16,17] is based on the inverse Jacobian matrix. The main drawback is that these methods fail to converge when singularity exists and are sensitive to the initial solution. The second type, such as the steepest decent approach,[18] employs the gradient-based non-linear algorithm for solving an equivalent minimization problem. Because this method does not use Jacobian matrix, the singularity problem is avoided. However, the high non-linearity of the kinematic equations makes the gradient vector of the objective function computational time consuming. As an alternative approach, Pashkevich[19] converted the inverse kinematics of robots with offset and reduced wrist into an efficient one-dimensional search problem. This algorithm has good convergence even for singularity configuration and is computationally efficient. Kucuk and Bingul[20] proposed a similar algorithm for the inverse kinematics of the 6-DOF robots that cannot be solved in closed form. These methods are limited to several special kinds of 6R 6-DOF robots. And, the multiple-solution problem is not investigated. Following the idea of Pashkevich,[19] a reliable iterative algorithm for the inverse kinematics of a 7R 6-DOF painting robot is presented in this article. Based on the geometrical properties of the robot, the inverse kinematics is converted into an efficient one-dimensional search problem about the fifth joint angle θ_5. The multiple-solution problem is also investigated by introducing three robot configuration indicators which are prespecified by a user. In order to verify the accuracy and efficiency of the proposed algorithm, several simulations are implemented.

The rest of the article is organized as follows: In section "The inverse kinematics algorithm," the geometry of the 7R 6-DOF robot is analyzed and a reliable iterative inverse kinematics algorithm is presented. Section "The multiple-solution problem and resolution" discusses the multiple-solution problem, and three configuration indicators are defined to find the correct solution from all the possible solutions. In section "Simulation and discussion," three simulations are implemented to illustrate the accuracy and efficiency of the proposed algorithm. The results are presented and discussed. Section "Conclusion" is the conclusion.

The inverse kinematics algorithm

The most popular real-time inverse kinematics algorithm is based on the geometrical properties of the

Table 1. The DH parameters of the 7R 6-DOF robot.

*i*th	a_{i-1}(mm)	α_{i-1}(°)	d_i(mm)	θ_i(°)	φ_i(°)
1	0	0	0	θ_1	0
2	0	90	0	θ_2	90
3	1000	0	0	θ_3	0
4	0	90	900	θ_4	0
5	0	−35	80	θ_5	0
6	0	70	80	θ_6	0
7	0	−35	100	θ_7	0

DH: Denavit–Hartenberg; DOF: degree of freedom.

robot. For serial industrial robot, Denavit–Hartenberg (DH) parameters are widely used to describe the geometry of manipulators. Following this notation, a coordinate frame system is attached to each link of the robot, as shown in Figure 2. The corresponding DH parameters of the 7R 6-DOF robot are shown in Table 1, where φ_i denotes the initial value of joint angle in original state shown in Figure 2. The transformation matrix relating joint i to joint $i-1$ could be given by equation (1)

$$ {}^{i-1}_{\ \ i}\mathbf{T} = \begin{bmatrix} c_i & -s_i & 0 & a_{i-1} \\ s_i c\alpha_{i-1} & c_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s_i s\alpha_{i-1} & c_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

where $s_i = \sin\theta_i$, $c_i = \cos\theta_i$, $c\alpha_i = \cos\alpha_i$, $s\alpha_i = \sin\alpha_i$, parameters a_i, α_i, and d_i are link length, link twist, and link offset, respectively. Then, the forward kinematics of the manipulator could be formulated by

$$ {}^0_1\mathbf{T}(\theta_1){}^1_2\mathbf{T}(\theta_2){}^2_3\mathbf{T}(\theta_3){}^3_4\mathbf{T}(\theta_4){}^4_5\mathbf{T}(\theta_5){}^5_6\mathbf{T}(\theta_6){}^6_7\mathbf{T}(\theta_7) = \mathbf{T}_{\text{end}} \quad (2) $$

where $\mathbf{T}_{\text{end}}$ is the configuration of the end-effector (EE) with respect to the base frame. For the inverse kinematics, $\mathbf{T}_{\text{end}}$ is known and described by equation (3), where $\mathbf{n}$, $\mathbf{o}$, and $\mathbf{a}$ are three unit orientation vectors, and $\mathbf{p}$ is the position vector of $\mathbf{T}_{\text{end}}$

$$ \mathbf{T}_{\text{end}} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

As a result, the problem of inverse kinematics corresponds to calculating the joint angles θ_i with the matrix equation (2). For the robots with spherical wrist, the inverse kinematics can be divided into two lower dimension problems. First, the first three joint angles $\theta_i (i = 1, 2, 3)$ are derived according to position vector of the wrist center. Second, the last three joint angles $\theta_i (i = 4, 5, 6)$ are derived according to the orientation of EE. However, as for the 7R 6-DOF robot, the wrist center is non-existent. In order to derive the inverse kinematics, first, the geometry of the 7R 6-DOF robot is analyzed.

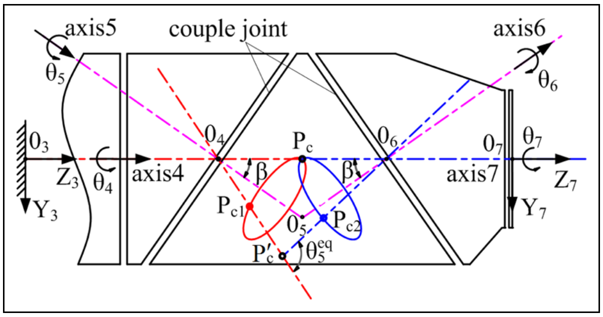

Figure 3. Configuration of the 4R 3-DOF wrist.

As shown in Figure 1, the 7R 6-DOF robot consists of seven revolute joints. Configuration of the first three joints, that is, R⊥R || R is widely used in industrial robots. The last four joints construct the non-spherical wrist, that is, the 4R 3-DOF wrist, which is the main difference. In order to have a further understanding, the geometrical properties of the 4R 3-DOF wrist are investigated separately, as shown in Figure 3. Because the couple joint is the main difference, we first keep Joint 4 and Joint 7 at the initial position, that is, $\theta_4 = \theta_7 = 0$. As the result of rotating the couple joint over an angle θ_5, the axis 4 and axis 7 revolve around the axis 5 and axis 6, respectively. And the axis 4 and axis 7 intersect at a changing point P'_c. It is remarkable that the position of P'_c with respect to P_c (i.e. the distance between P'_c and P_{c1}) is only determined by the fifth joint angle θ_5. In other words, if θ_5 and the configuration of EE $\mathbf{T}_{\text{end}}$ are given, the position of P'_c could be derived as equation (4), where $\mathbf{p}$ is the position vector of $\mathbf{T}_{\text{end}}$, $\mathbf{a}$ is the orientation vector of its z axis, $d_7^{eq} = d_7 + d$ is the distance between P'_c and O_7, d is the distance between P'_c and O_4 or between P'_c and O_6. Thus, if we take point P'_c as a virtual wrist center, the first three joint angles $\theta_i (i = 1, 2, 3)$ can be derived. As the key point, the relationship between θ_5 and d is constructed as following

$$ \mathbf{p}_{pc'} = \mathbf{p} - d_7^{eq}\mathbf{a} \quad (4) $$

As shown in Figure 3, the parameter d can be derived in triangle $O_6P'_cP_c$ as

$$ d = \frac{d_6 \cos(\beta)}{\cos(\theta_5^{eq}/2)} \quad (5) $$

where θ_5^{eq} is the angle between axis 4 and axis 7 (i.e. the bending angle of the wrist), which can be derived by analyzing the orientation of axis 7 with respect to the (x_4, y_4, z_4) coordinate frame. As shown in Figure 4, the orientation of axis 7 with respect to the (x_4, y_4, z_4) coordinate frame can be expressed as

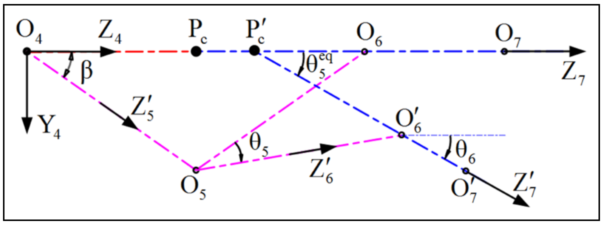

Figure 4. The bending angle of the 4R 3-DOF wrist.

$$
\begin{aligned}
{}^4\mathbf{a}_7 = {} & R(x,\ -\beta)R(z,\theta_5)R(x,2\beta)R(z,\theta_6) \\
& R(x,\ -\beta)R(z,\theta_7)^4\mathbf{a}_4
\end{aligned} \tag{6}
$$

where $R(x,\ -\beta)$ denotes rotation $-\beta$ about the x axis, $R(z,\theta_5)$ denotes rotation θ_5 about the z axis, ${}^4\mathbf{a}_4 = [0,0,1]^{\mathrm{T}}$ is the orientation vector of axis 4 with respect to the $(x_4,\ y_4,\ z_4)$ coordinate frame. The third element of ${}^4\mathbf{a}_7$ describes the projection along z axis of orientation vector of axis 7, which can also be derived with θ_5^{eq}. Thus, we obtain equation (7) for solving θ_5^{eq}

$$
c_\beta(c_\beta c_{\alpha5} - s_\beta s_\beta c_5) - s_\beta(c_{\alpha5}s_\beta c_5^2 + c_\beta s_{\alpha5}c_5 + s_\beta s_5^2) = c_{5eq} \tag{7}
$$

Then, θ_5^{eq} can be derived by substituting $c_{5eq} = (1 - x^2/1 + x^2)$ into equation (7), where $x = \tan(\theta_5^{eq}/2)$

$$
\theta_5^{eq} = 2sign(\theta_5)\mathrm{atan}\sqrt{\frac{1-A}{1+A}} \tag{8}
$$

where $\mathrm{A} = c_{\alpha6}(c_{\alpha4}c_{\alpha5} - s_{\alpha4}s_{\alpha5}c_5) - s_{\alpha6}(c_{\alpha5}s_{\alpha4}c_5^2 + c_{\alpha4}s_{\alpha5}c_5 + s_{\alpha4}s_5^2)$.

As the summary of the above analysis, if θ_5 is appointed, θ_5^{eq}, d, and the position of $\mathrm{P'_c}$ can be derived with equations (8), (5), and (4), respectively. Thus, the inverse kinematics can be converted into a one-dimensional search problem about θ_5. Figure 5 depicts the flowchart of the new proposed algorithm (NA). NA works as follows: (1) an initial value is randomly selected for the fifth joint angle θ_5; (2) the position of point $\mathrm{P'_c}$ is derived with equation (4); (3) the first three joint angles $\theta_i(i = 1,2,3)$ are derived analytically according to position of $\mathrm{P'_c}$ (see Appendix 1); (4) the last three joint angles $\theta_i(i = 4,5,7)$ are derived analytically according to the orientation of EE (see Appendix 2); (5) if increment of θ_5 is no larger than a preset threshold (i.e. $\Delta\theta_5 \leq \varepsilon$), the iteration is finished and the final solution is obtained; else, go back to step 2 with the new computed θ_5. Because the Jacobian matrix is not utilized, the proposed algorithm possesses good convergence, even for singular configurations. This is one of the main advantages of the proposed algorithm over the Jacobian matrix–based methods.

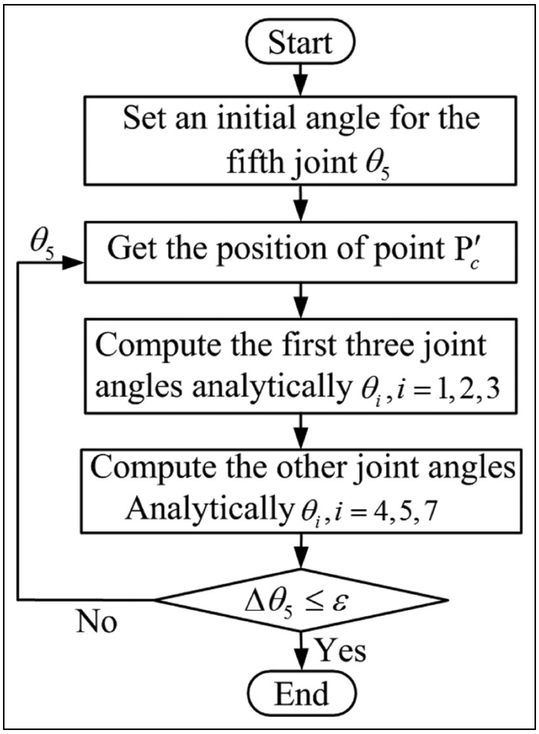
Start
Set an initial angle for the fifth joint θ_5
θ_5
Get the position of point P'_c
Compute the first three joint angles analytically $\theta_i, i=1,2,3$
Compute the other joint angles Analytically $\theta_i, i=4,5,7$
$\Delta\theta_5 \leq \varepsilon$
No
Yes
End

Figure 5. The flowchart of the new proposed algorithm (NA).

For comparison, the Newton–Raphson approach is also reviewed to solve the inverse kinematics of this robot. The Newton–Raphson approach is a well-known iterative method for the inverse kinematics. And the iterative formula of joint angles could be described as

$$
\boldsymbol{\theta} = \boldsymbol{\theta} + d\boldsymbol{\theta} = \boldsymbol{\theta} + J^{-1}\mathbf{e} \tag{9}
$$

where $\mathbf{e}$ denotes the configuration error of EE, J is the Jacobian matrix. It is notable that the initial value of the joint angles $\boldsymbol{\theta}_{in} = [\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_7]$ is selected randomly. And if increment of $d\boldsymbol{\theta}$ is no larger than a preset threshold (i.e. $\|d\boldsymbol{\theta}\| \leq \varepsilon_1$), the iteration is finished. It is notable that the Newton–Raphson approach is a Jacobian matrix–based method. So, it will fail to converge when singularity exists. For the 7R 6-DOF robot, the singularities could be derived by equation (10). However, the identification of the singularities of the 7R 6-DOF robot is really a difficult work. Therefore, for simplicity, two singularity conditions of the 7R 6-DOF robot are directly given as $\theta_5 = 0°$ or $\theta_5 = 180°$ without the detail derivation.

$$
\det(JJ^{\mathrm{T}}) = 0 \tag{10}
$$

The multiple-solution problem and resolution

As stated previously, the first three joint angles $\theta_i(i = 1,2,3)$ are derived according to position of $\mathrm{P'_c}$ in NA. With a special configuration of the first three joints, that is, $\mathrm{R}\perp\mathrm{R} \parallel \mathrm{R}$, four solutions can be obtained.[21] Then, the multiple-solution problem arises.

In order to deal with this problem, two configuration indicators (*Arm* and *Elbow*) are introduced for the first and the third joint angle, respectively:

> *Arm* = 1 (left): $\theta_1 \in [-\mathrm{pi}/2, \mathrm{pi}/2)$; *Arm* = 0 (right): $\theta_1 \in [-\mathrm{pi}, -\mathrm{pi}/2) \cup [\mathrm{pi}/2, \mathrm{pi})$.
> *Elbow* = 1 (up): $\theta_3 \in [-\mathrm{pi}/2, \mathrm{pi}/2)$; *Elbow* = 0 (down): $\theta_3 \in [\mathrm{pi}/2, \mathrm{pi}^*3/2)$.

According to the value of the two configuration indicators, four robot configurations corresponding to the four solutions of the first three joint angles can be defined. Then, by appointing the value of the two indicators, one solution can be determined from the possible four solutions. In order to obtain all possible inverse kinematic solutions of the 7R 6-DOF robot for a given $\mathbf{T}_{\mathrm{end}}$, the multiple-solution result from the last four joints (i.e. the 4R 3-DOF wrist) are analyzed separately. From section "The inverse kinematics algorithm," if θ_5 and the configuration of EE $\mathbf{T}_{\mathrm{end}}$ are given, the position of P'_{c} could be derived with equation (4). Then, four solutions corresponding to four robot configurations can be obtained for the first three joints and one solution for the last four joints. And, the closer $\theta_7(°)$ is to the accurate value, the smaller the error between the calculated configuration of EE and the desired one. As a special case, if θ_5 is equal to the exact value, the error between the calculated configuration of EE and the desired one would be zero. Thus, θ_5 is discretized in its working space, that is, $[-180°, 180°]$ with a small increment of $\Delta\theta_5 = 0.1°$. All the discrete points are used to derive the position of P'_{c} and then the four inverse solutions. By analyzing the error between the calculated configurations of EE and the desired one, the multiple-solution result from the last four joints can be obtained.

As an example, we take $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_7] = [30°, 60°, -60°, -30°, 60°, -60°, 30°]$ as the target joint angle. Through the forward kinematic equations, the desired configuration of EE is described as equation (11). The position error between the calculated configurations of EE and the desired one is shown in Figure 6. Here, four-solution result from the first three joints is depicted by corresponding four robot configurations for simplicity. It is obvious that two values of θ_5 makes the position error be zero for each robot configuration, and the two values are opposite to each other. In other words, there are two solutions for the last four joints.

$$\mathbf{T}_{\mathrm{end}} = \begin{bmatrix} 0.371157 & 0.540588 & 0.754988 & 567.728363 \\ 0.157481 & -0.837933 & 0.522559 & 343.242487 \\ 0.915118 & -0.075056 & -0.396137 & -183.093234 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

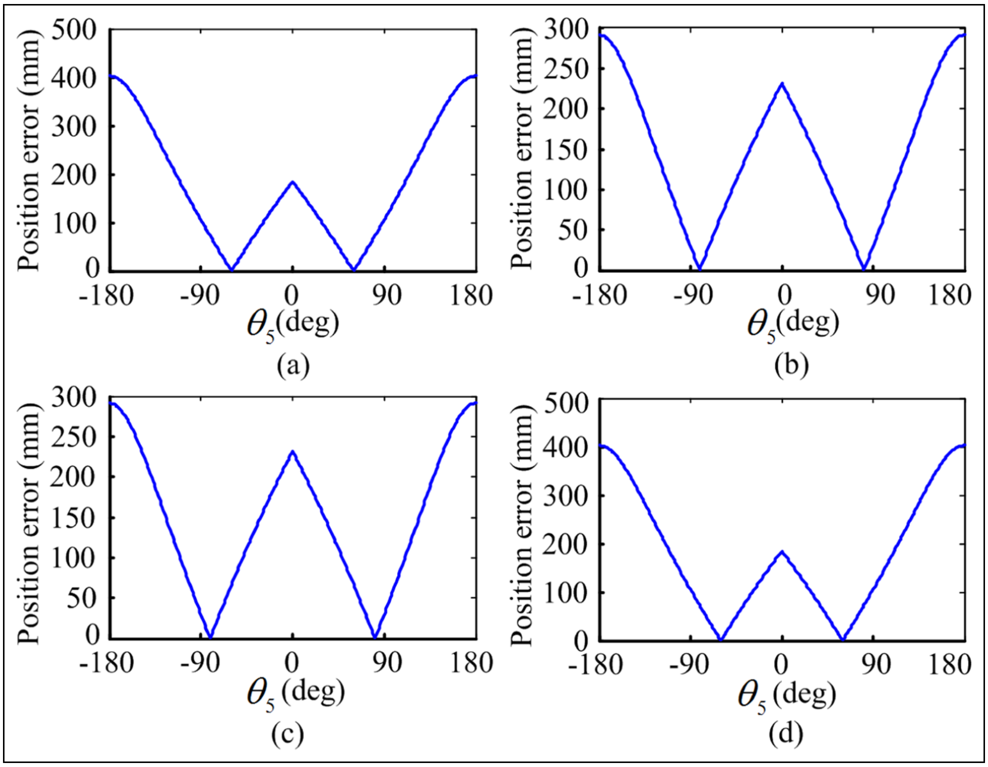

Figure 6. Position errors of EE: (a) Configuration A, (b) Configuration B, (c) Configuration C, and (d) Configuration D.

Table 2. The inverse kinematic solutions of the 7R 6-DOF robot.

Configuration, *i*	θ_1(°)	θ_2(°)	θ_3(°)	θ_4(°)	θ_5(°)	θ_7(°)
1	60.0000	60.0000	0.0000	−30.0000	60.0000	30.0000
2	60.0000	60.0000	0.0000	−159.3775	−60.0000	159.3775
3	60.0360	−27.5107	178.4662	−178.6218	21.6300	−177.6485
4	60.0360	−27.5107	178.4662	19.1657	−21.6300	−15.4359
5	−119.9640	207.5107	1.5338	1.3782	21.6300	−177.6485
6	−119.9640	207.5107	1.5338	−160.8343	−21.6300	−15.4359
7	−120.0000	120.0000	180.0000	150.0000	60.0000	30.0000
8	−120.0000	120.0000	180.0000	20.6225	−60.0000	159.3775

DOF: degree of freedom.

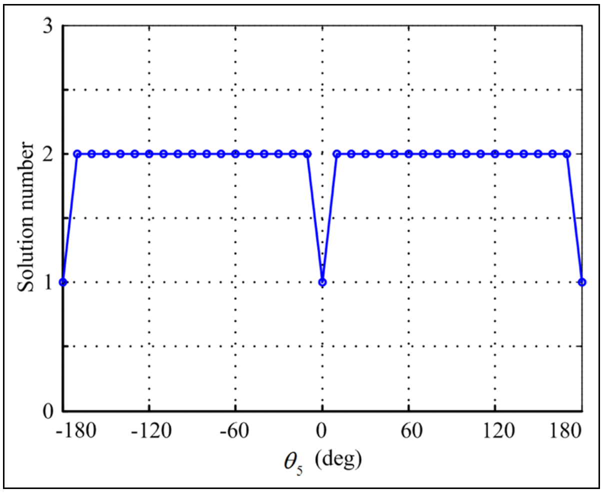

Figure 7. The number of solutions for the last four joints.

Without the loss of generality, θ_5 is discretized in its working space, that is, $[-180°, 180°]$ with increment of $\Delta\theta_5 10°$. The discrete points along with randomly selected other joint angles (i.e. θ_i, $i = 1, 2, 3, 4, 7$) are taken as the target joint angles. The number of solutions for the last four joints is depicted in Figure 7. Except for two special cases (i.e. $\theta_5 = \pm 180°$ and $\theta_5 = 0°$), there are two solutions for the last four joints. Thus, in some degree, we can get the conclusion that there are at most eight inverse kinematic solutions (four-solution result from the first three joints and two-solution result from the last four joints) for a given $\mathbf{T}_{\text{end}}$.

In order to deal with the multiple-solution problem result from the last four joints, one more configuration indicator (*Wrist*) is introduced for the fifth joint. And by appointing the value of the three indicators (*Arm, Elbow*, and *Wrist*), the correct solution can be determined from the eight possible solutions (see Appendices 1 and 2):

$Wrist = 1$ (positive): $\theta_5 \in [0, \text{pi})$; $Wrist = 0$ (negative): $\theta_5 \in [-\text{pi}, 0)$.

Simulation and discussion

In order to illustrate the accuracy and efficiency of the inverse kinematics algorithm proposed in this article, three simulations are implemented on a practical 7R 6-DOF painting robot as shown in Figure 1. The DH parameters of the robot are shown in Table 1. The first simulation aims to validate the accuracy of the algorithm by calculating the eight inverse kinematic solutions with a given configuration of EE. The accuracy of the algorithm is also assessed in the whole usable workspace including the singularities of the robot. The last simulation aims to illustrate the efficiency of the algorithm, when it is tested in an off-line programming operation.

Simulation 1

Take $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_7] = [60°, 60°, 0°, -30°, 60°, 30°]$ as the target joint angle. Through the forward kinematic equations, the configuration of EE is described as

$$\mathbf{T}_{\text{end}} = \begin{bmatrix} -0.253609 & 0.9073303 & 0.3353118 & 733.50553 \\ -0.537657 & -0.4203879 & 0.7308889 & 1297.25391 \\ 0.8041186 & 0.0050774 & 0.5944472 & 482.878011 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

According to the definition of the three robot configuration indicators (*Arm, Elbow*, and *Wrist*), the given target joint angle belongs to *Configuration 1* (i.e. $Arm = 1$, $Elbow = 1$, $Wrist = 1$). By rationally appointing the value of the three indicators (i.e. $Arm = 1$, $Elbow = 1$, and $Wrist = 1$), the correct solution can be obtained with the proposed inverse kinematics algorithm. As shown in Table 2, the first solution is the obtained correct solution and the deviation is less than 0.0001°, which illustrates the accuracy of the proposed algorithm. The position errors of initial selected joint angles versus the convergence times of this configuration are illustrated in Figure 8. It can be observed from Figure 8 that as the time progresses, the position error

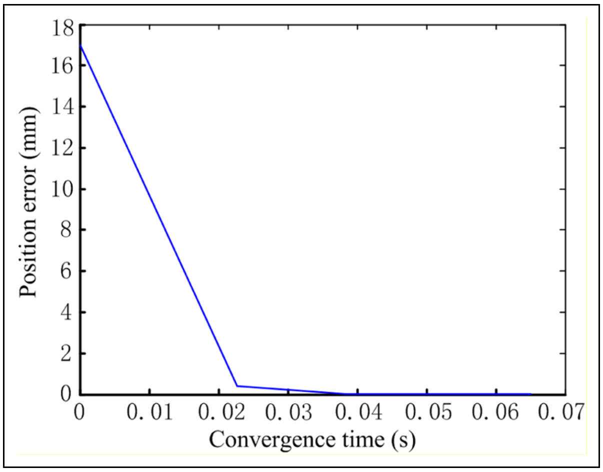

Figure 8. The position errors of initial selected joint angles versus the convergence times.

Table 3. The computation time of the proposed algorithm and Newton–Raphson approach.

Method	The initial value $\boldsymbol{\theta}_{in}$(°)	Computation time (ms)
The proposed method	Randomly selected	65
Newton–Raphson approach	[60, 60, 0, −30, 30, 30]	63
	[0, 60, 0, −30, 30, 30]	104
	[0, 60, 0, −30, 0, 30]	Failure

asymptotically converges to zero, and the convergence time demonstrates that the proposed algorithm is effective for solving the inverse kinematics problem. Additionally, by setting the value of the three indicators equal to other seven possible values (see Appendix 3), all the possible inverse kinematic solutions can be obtained, as shown in Table 2. The corresponding postures for the eight inverse kinematic solutions are shown in Figure 9. The algorithm is performed on a desktop computer platform (Pentium i7 2.8 GHz, 8 GB RAM, MATLAB software program).

In order to further illustrate the performance of the proposed algorithm, the Newton–Raphson approach is also used to solve this inverse kinematics problem. For simplicity, the computation times of the two methods are summarized in Table 3. From Table 3, the proposed algorithm takes about 65 ms on an average. It is notable that three different initial values are chosen for the Newton–Raphson approach. When the initial value is chosen as $\boldsymbol{\theta}_{in} = [60°, 60°, 0°, -30°, 30°, 30°]$, the computation time of the Newton–Raphson approach is 63 ms which is slightly smaller than that of the proposed method. However, when the initial value is changed to $\boldsymbol{\theta}_{in} = [0°, 60°, 0°, -30°, 30°, 30°]$, the computation time will increase to 104 ms. The Newton–Raphson approach even fails when the initial value is chosen as $\boldsymbol{\theta}_{in} = [0°, 60°, 0°, -30°, 0°, 30°]$, because the Jacobian matrix is singular. So, we conclude that the proposed algorithm is more stable and efficient than the Newton–Raphson approach.

Simulation 2

In this simulation, the accuracy of the algorithm is tested in the whole usable workspace including the singularities of the robot. For the 7R 6-DOF robot studied, the usable workspace is shown in Table 4. Then, 500 groups of target joint angles are randomly selected

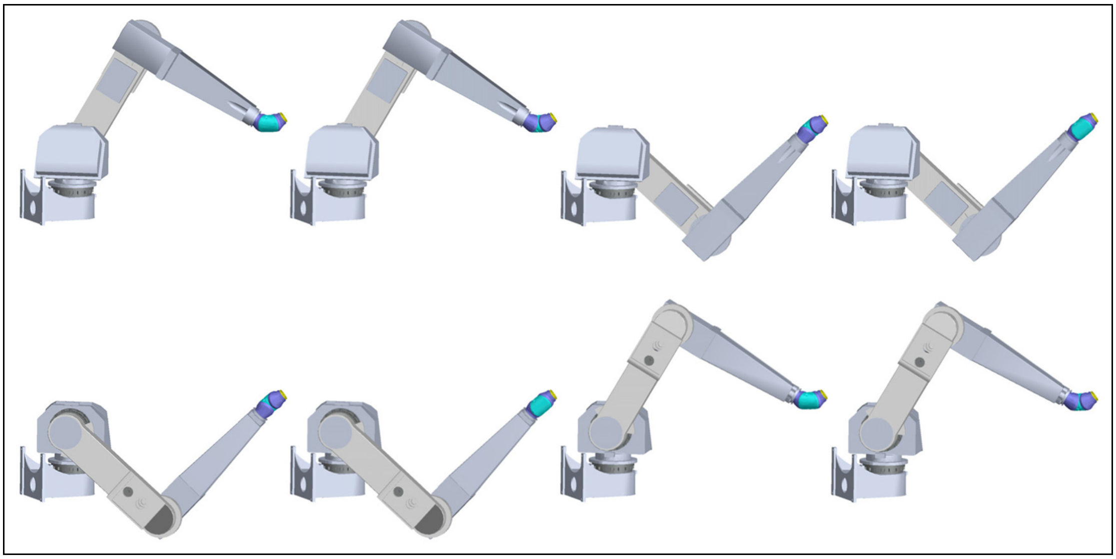

Figure 9. Postures for the eight inverse kinematic solutions of the 7R 6-DOF robot.

Table 4. The usable workspace of the 7R 6-DOF robot.

*i*th	$\theta_1(°)$	$\theta_2(°)$	$\theta_3(°)$	$\theta_4(°)$	$\theta_5(°)$	$\theta_7(°)$
Range of joint angle	[−70, 70]	[−80, 135]	[−70, 70]	[−360, 360)	[−180, 180]	[−360, 360)

DOF: degree of freedom.

Table 5. The solution errors and computation time for simulation 2.

Configurations	Average position error (mm)	Maximum position error (mm)	Average computation time (s)
Randomly selected	0.003	0.01	0.061
Singular Configuration 1	0.009	0.009	0.043
Singular Configuration 2	0.004	0.004	0.069

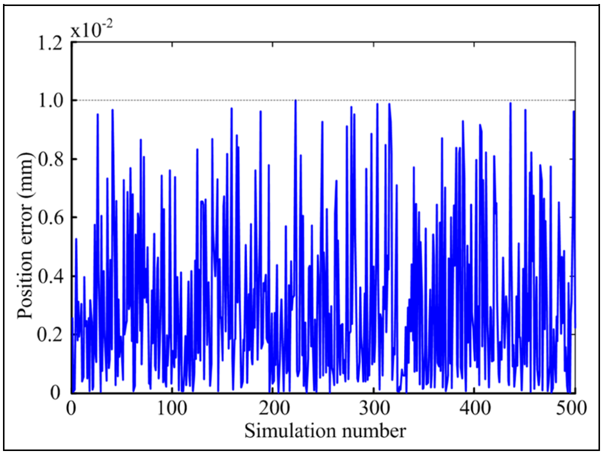

Figure 10. Accuracy of the inverse kinematic solutions in the usable workspace.

in the usable workspace. In addition, two singular configurations, that is, $\boldsymbol{\theta}_{a1} = [60°, -30°, 60°, -30°, 0°, 30°]$ and $\boldsymbol{\theta}_{a2} = [60°, -30°, 60°, -30°, 180°, 30°]$, are selected as the target joint angles. Through the forward kinematic equations, the configurations of EE are derived. The proposed algorithm is used to derive the inverse kinematic solutions. The corresponding solution errors, that is, the position errors of EE, are depicted in Figure 10. For simplicity, the solution errors and computation time for all the configurations are also summarized in Table 5.

From Figure 10 and Table 5, the inverse kinematic solutions for all 500 randomly selected configurations and two singular configurations are derived with the proposed algorithm. The average and maximum position errors for the 500 randomly selected configurations are 0.003 and 0.01 mm, respectively, which illustrate the accuracy of the algorithm in the whole usable workspace. As for the two singular configurations, the position errors are 0.009 and 0.004 mm, respectively. It is notable that the computation time is influenced by the randomly selected initial solution. However, the computation time is less than 0.07 s, which illustrates that the proposed inverse kinematics algorithm can converge to the accurate solution rapidly. In other words, the proposed algorithm is efficient even for singular configurations.

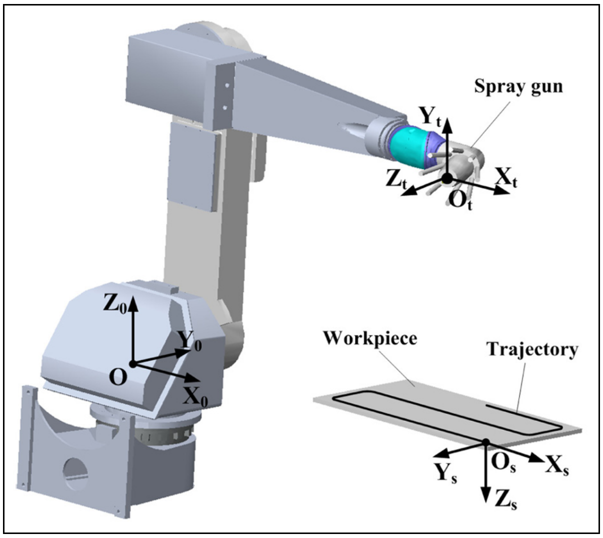

Figure 11. Graphic simulator of the 7R 6-DOF robot and the trajectory.

Simulation 3

The new proposed inverse kinematics algorithm is implemented in an off-line programming operation to painting. The graphic simulator of a 7R 6-DOF painting robot and the task are shown in Figure 11. The trajectory of EE is a classical raster trajectory in the plane *x-o-y* on the surface of a workpiece. The orientation of EE is constant, with the *x* axis parallel to that of the

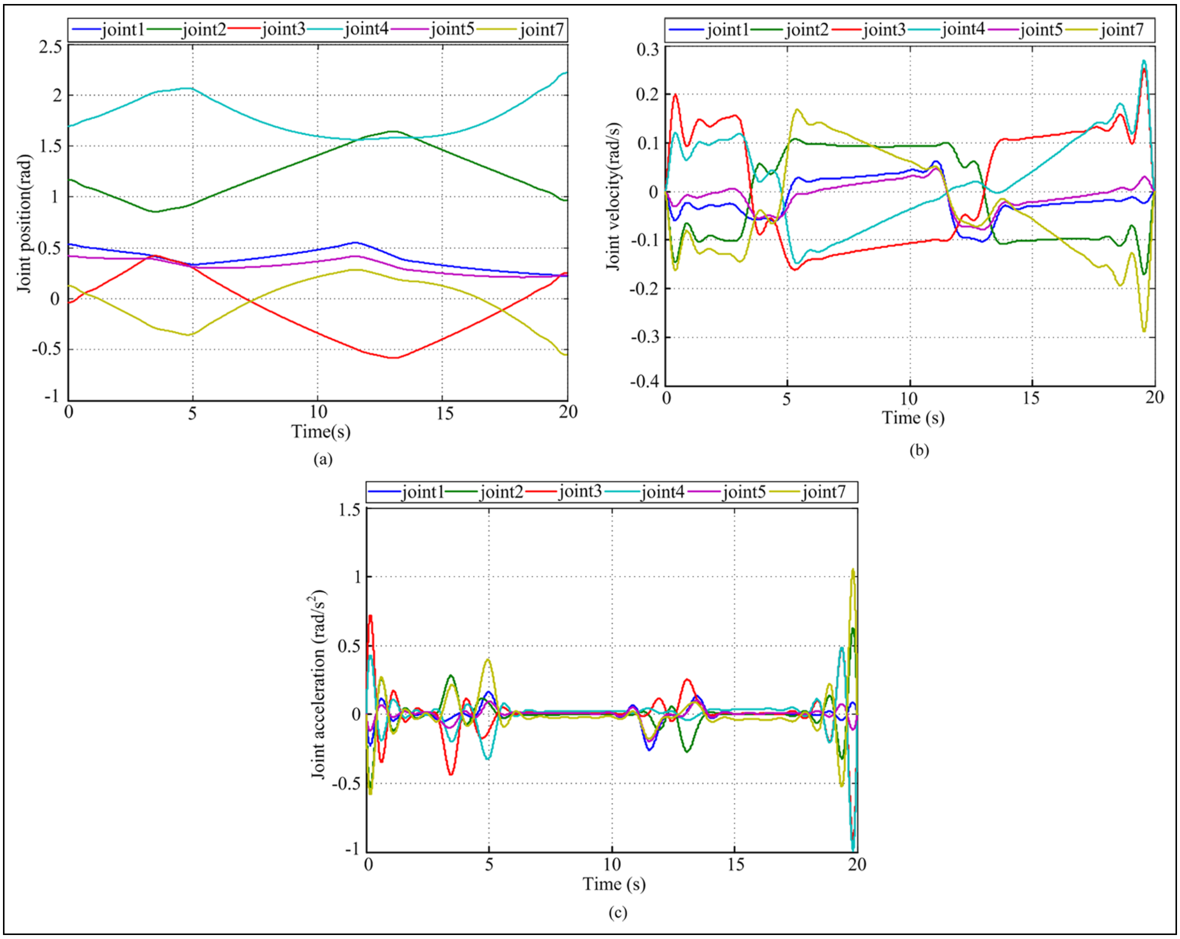

Figure 12. Simulation results of the 7R 6-DOF robot: (a) joint position, (b) joint velocity, and (c) joint acceleration.

base frame, and the z and y axes opposite to that of the base frame, respectively. The duration of the motion is determined as 20 s. For trajectory tracking, first, discrete points are sampled from the trajectory. The inverse kinematic solution for each sample point is derived. Later on, a further interpolation in joint space is implemented to make the trajectory smooth enough. It is remarkable that the three robot configuration indicators are appointed according to *Configuration 1* (i.e. *Arm* = 1, *Elbow* = 1, and *Wrist* = 1). And the calculated joint positions, velocities, and accelerations of the 7R 6-DOF robot are given in Figure 12.

It is obvious that the off-line programming operation is completed with the NA. The algorithm takes 0.009 s on an average on a desktop computer platform (Pentium i7 2.8 GHz, 8 GB RAM, MATLAB software program), which illustrates the efficiency. By comparing simulation 2 and simulation 3, the computing time in simulation 3 is shorter than that in simulation 2. The reason is that the solution of the last step is taken as the initial value of the next step when the robot is following a continuous trajectory. In other words, we should only randomly select the initial value for the original point of the continuous trajectory. Thus, in some degree, we can get the conclusion that the proposed algorithm is more advantageous for a continuous trajectory. And by presetting the value of the three indicators (*Arm, Elbow*, and *Wrist*) according to the joint working range restrictions, the optimal solution can be selected from eight possible solutions, which is another main advantage of the proposed inverse kinematics algorithm.

Conclusion

A new and efficient numerical iterative algorithm for the inverse kinematics of a 7R 6-DOF robot is proposed and studied in this article. In contrast to the classical methods, the inverse kinematics is converted into an efficient one-dimensional search problem about the fifth joint angle θ_5. Since the Jacobian matrix is not utilized, the proposed algorithm possesses good convergence, even for singular configurations. In addition, the multiple-solution problem is also investigated. There are at most eight inverse kinematic solutions (four-solution result from the first three joints and two-solution result from the last four joints) for a given configuration of EE. By defining three robot configuration indicators (*Arm, Elbow*, and *Wrist*) which are prespecified by a user, the correct solution can be chosen from all the possible solutions.

The accuracy and efficiency have been evaluated with three simulations. In the first simulation, eight solutions are got for a given configuration of EE. By rationally appointing the value of the three indicators, the correct solution can be derived. The solution error is less than 0.0001 degree, which illustrates the accuracy of the proposed algorithm. Additionally, in order to further illustrate the performance of the proposed algorithm, the Jacobian matrix–based method, that is, the Newton–Raphson approach is also used to solve the inverse kinematics problem. The results show that the proposed algorithm is more stable and efficient than the Newton–Raphson approach. The accuracy of the algorithm is also assessed in the whole usable workspace of the robot; the proposed algorithm possesses good convergence, even for singular configurations. In the third simulation, the algorithm is implemented in an off-line programming operation. The results show that the proposed algorithm is efficient for off-line programming. And the proposed algorithm is more advantageous for a continuous trajectory.

Declaration of conflicting interests

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

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Tianjin Science and Technology Committee (grant no. 15ZXZNGX00200).

References

1. Pieper DL. *The kinematics of manipulators under computer control*. PhD Thesis, Stanford University, Stanford, CA, 1968.
2. Tsai LW and Morgan AP. Solving the kinematics of the most general six and five degree of freedom manipulators by continuation methods. *J Mech Transm* 1985; 107: 189–200.
3. Raghavan M and Roth B. Inverse kinematics of the general 6R manipulator and related linkages. *J Mech Design* 1993; 115: 502–508.
4. Manocha D and Canny JF. Efficient inverse kinematics for general 6R manipulators. *IEEE T Robotic Autom* 1994; 10: 648–657.
5. Husty ML, Pfurner M and Schröcker HP. A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mech Mach Theory* 2007; 42: 66–81.
6. Liu SG, Zhu SQ and Wang XY. Real-time and high-accurate inverse kinematics algorithm for general 6R robots based on matrix decomposition. *Chin J Mech Eng* 2008; 44: 304–309.
7. Qiao SG, Liao QZ, Wei SM, et al. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions. *Mech Mach Theory* 2010; 45: 193–199.
8. Wei YH, Jian SQ, He S, et al. General approach for inverse kinematics of nR robots. *Mech Mach Theory* 2014; 75: 97–106.
9. Wu L, Yang XD, Miao DJ, et al. Inverse kinematics of a class of 7R 6-DOF robots with non-spherical wrist. In: *Proceedings of the IEEE international conference on mechatronics and automation*, Takamatsu, Japan, 4–7 August 2013. New York: IEEE.
10. Köker R, Cakar T and Sari Y. A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators. *Eng Computation* 2014; 30: 641–649.
11. Asadi-Eydivand M, Ebadzadeh MM, Solati-Hashjin M, et al. Cerebellum-inspired neural network solution of the inverse kinematics problem. *Biol Cybern* 2015; 109: 561–574.
12. Chapelle F and Bidaud P. Closed form solutions for inverse kinematics approximation of general 6R manipulators. *Mech Mach Theory* 2004; 39: 323–338.
13. Kalra P, Mahapatra PB and Aggarwal DK. An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. *Mech Mach Theory* 2006; 41: 1213–1229.
14. Köker R. A neuro-simulated annealing approach to the inverse kinematics solution of redundant robotic manipulators. *Eng Computation* 2013; 29: 507–515.
15. Khatib O. A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J Robot Autom* 1987; 3: 43–53.
16. Chiaverini S, Siciliano B and Egeland O. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE T Contr Syst T* 1994; 2: 123–134.
17. Xu WF, Zhang JT, Liang B, et al. Singularity analysis and avoidance for robot manipulators with nonspherical wrists. *IEEE T Ind Electron* 2016; 63: 277–290.
18. Wolovich WA and Elliott H. A computational technique for inverse kinematics. In: *Proceedings of the 23rd IEEE conference on decision and control*, Las Vegas, NV, 12–14 December 1984. New York: IEEE.
19. Pashkevich A. Real-time inverse kinematics for robots with offset and reduced wrist. *Control Eng Pract* 1997; 5: 1443–1450.
20. Kucuk S and Bingul Z. Inverse kinematics solutions for industrial robot manipulators with offset wrists. *Appl Math Model* 2014; 38: 1983–1999.
21. Lee CSG and Ziegler M. Geometric approach in solving inverse kinematics of PUMA robots. *IEEE T Aero Elec Sys* 1984; 20: 695–706.

Appendix 1

The algorithm to derive the first three joint angles

The position vector of P′$_c$ with respect to the base frame can be expressed as

$$\mathbf{p}_{\mathrm{pc}'} = [p_x, p_y, p_z]^{\mathrm{T}}$$

Then, two values of θ_1 can be expressed as

$$\theta_{11} = \operatorname{atan2}(p_y, p_x),\ \theta_{12} = \begin{cases} \theta_1 - pi, & \theta_1 > 0 \\ \theta_1 + pi, & \theta_1 \leq 0 \end{cases}$$

According to the value of robot configuration indicator *Arm*, the correct θ_1 can be derived

$$\text{If } Arm = 1, \quad \theta_1 = \begin{cases} \theta_{11} \text{ if } \theta_{11} \in [-\mathrm{pi}/2, \mathrm{pi}/2) \\ \theta_{12} \qquad \text{else} \end{cases}$$

$$\text{If } Arm = 0, \quad \theta_1 = \begin{cases} \theta_{11} \text{ if } \theta_{11} \in [-\mathrm{pi}, -\mathrm{pi}/2) \cup [\mathrm{pi}/2, \mathrm{pi}) \\ \theta_{12} \qquad \text{else} \end{cases}$$

Two values of θ_3 are expressed as

$$\theta_{31} = 2\operatorname{atan}((1 + \sqrt{1 - k^2})/k) + \mathrm{pi}/2,$$
$$\theta_{32} = 2\operatorname{atan}((1 - \sqrt{1 - k^2})/k) + \mathrm{pi}/2$$

where $k = (p_x + p_y + p_z - d_4 - a_2)/(2a_2d_4)$. According to the value of robot configuration indicator *Elbow*, the correct θ_3 can be derived

$$\text{If } Elbow = 1, \quad \theta_3 = \begin{cases} \theta_{31} \text{ if } \theta_{31} \in [-\mathrm{pi}/2, \mathrm{pi}/2) \\ \theta_{32} \qquad \text{else} \end{cases}$$

$$\text{If } Elbow = 0, \quad \theta_3 = \begin{cases} \theta_{31} \text{ if } \theta_{31} \in [\mathrm{pi}/2, \mathrm{pi}^*3/2) \\ \theta_{32} \qquad \text{else} \end{cases}$$

Then, θ_2 can be derived as

$$\theta_2 = \operatorname{atan2}(p_z a_2 c_3 + k_2(a_2 s_3 + d_4), k_1 + k_2 a_2 c_3) - \theta_3$$

where $k_1 = -p_z d_4 - p_z a_2 s_3$, $k_1 = -p_z d_4 - p_z a_2 s_3$, and $k_2 = p_x c_1 + p_y s_1$.

Appendix 2

The algorithm to derive the last three joint angles

We rearranged the forward kinematic equations as

$$\begin{aligned} &{}^3_4\mathbf{T}(\theta_4){}^4_5\mathbf{T}(\theta_5){}^5_6\mathbf{T}(\theta_6) \\ &= {}^2_3\mathbf{T}(\theta_3)^{-1}{}^1_2\mathbf{T}(\theta_2)^{-1}{}^0_1\mathbf{T}(\theta_1)^{-1}\mathbf{T}_{\mathrm{end}} = {}^3\mathbf{T}_{\mathrm{end}} \end{aligned}$$

When first three joint angles $\theta_i (i = 1, 2, 3)$ are obtained, ${}^3\mathbf{T}_{\mathrm{end}}$ is known and can be expressed as

$${}^3\mathbf{T}_{\mathrm{end}} = \begin{bmatrix} n_{x73} & o_{x73} & a_{x73} & p_{x73} \\ n_{y73} & o_{y73} & a_{y73} & p_{y73} \\ n_{z73} & o_{z73} & a_{z73} & p_{z73} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, two values of θ_5 are obtained as

$$\begin{aligned} \theta_{51} &= 2\operatorname{atan}((A - C \\ &+ \sqrt{((C - A)^2 - (A - B + C)(A + B + C)))}/(A - B + C)) \\ \theta_{52} &= 2\operatorname{atan}((A - C \\ &- \sqrt{((C - A)^2 - (A - B + C)(A + B + C)))}/(A - B + C)) \end{aligned}$$

where $A = s_{\alpha 6} s_{\alpha 4}(c_{\alpha 5} - 1)$, $B = s_{\alpha 6} c_{\alpha 4} s_{\alpha 5} + c_{\alpha 6} s_{\alpha 4} s_{\alpha 5}$, and $C = s_{\alpha 6} s_{\alpha 4} - c_{\alpha 6} c_{\alpha 4} c_{\alpha 5} - a_{y73}$.

According to the value of robot configuration indicator *Wrist*, the correct θ_5 can be derived

$$\text{If } Wrist = 1, \quad \theta_5 = \begin{cases} \theta_{51} \text{ if } \theta_{51} \in [0, \mathrm{pi}) \\ \theta_{52} \qquad \text{else} \end{cases}$$

$$\text{If } Wrist = 0, \quad \theta_5 = \begin{cases} \theta_{51} \text{ if } \theta_{51} \in [-\mathrm{pi}, 0) \\ \theta_{52} \qquad \text{else} \end{cases}$$

Then, θ_4 can be derived as

$$\begin{aligned} \theta_4 = \operatorname{atan2}(&(p_{z73}E - p_{x73}F)/(p_{z73}^2 + p_{x73}^2), \\ &(p_{x73}E + p_{z73}F)/(p_{z73}^2 + p_{x73}^2)) \end{aligned}$$

where E and F are functions of θ_5 and the elements of ${}^3\mathbf{T}_{\mathrm{end}}$.

Similarly, we can get θ_7 as

$$\theta_7 = \operatorname{atan2}(s_7, c_7)$$

where s_7 and c_7 are functions of θ_4, θ_5, and the elements of ${}^3\mathbf{T}_{\mathrm{end}}$.

Appendix 3

Definition of the eight configurations

Configuration 1: Arm = 1, *Elbow* = 1, *Wrist* = 1;
Configuration 2: Arm = 1, *Elbow* = 1, *Wrist* = 0;
Configuration 3: Arm = 1, *Elbow* = 0, *Wrist* = 1;
Configuration 4: Arm = 1, *Elbow* = 0, *Wrist* = 0;
Configuration 5: Arm = 0, *Elbow* = 1, *Wrist* = 1;
Configuration 6: Arm = 0, *Elbow* = 1, *Wrist* = 0;
Configuration 7: Arm = 0, *Elbow* = 0, *Wrist* = 1;
Configuration 8: Arm = 0, *Elbow* = 0, *Wrist* = 0.