Robotica (2022), **40**, pp. 2275–2294
doi:10.1017/S0263574721001648

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

An improved inverse kinematics solution for 6-DOF robot manipulators with offset wrists

Xing Zhou[1,2], Yaoqi Xian[1,*], Yuanhao Chen[1], Tongshu Chen[1], Lin Yang[3], Simin Chen[1] and Jian Huang[1]

[1]Foshan Institute of Intelligent Equipment Technology, Foshan 528000, China, [2]School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430000, China, and [3]Huashu Robot Co., Ltd., Foshan 528000, China
*Corresponding author. E-mail: xianyaoqi@gmail.com

Received: 31 August 2021; **Revised:** 30 September 2021; **Accepted:** 20 October 2021; **First published online:** 14 January 2022

Keywords: offset wrist, inverse kinematics, robot manipulator

Abstract

Efficiently solving inverse kinematics (IK) of robot manipulators with offset wrists remains a challenge in robotics due to noncompliance with Pieper criteria. In this paper, an improved method to solve the IK for 6-DOF robot manipulators with offset wrists is proposed. This method is based on the Newton iteration technique, but it does not require a selection of initial estimation of joint variables. The solution is divided into two parts: the first part is to reconstruct a simplified structure with analytical IK solution, and the second part is to obtain a numerical solution by iteration. Further, a robot manipulator HSR-BR606 with an offset wrist is used as an example to specifically elaborate the mathematical procedure of the method and to investigate the algorithm in terms of accuracy, efficiency, and application of motion planning. A comparative experiment is conducted with a typical IK algorithm, which demonstrates a higher accuracy and shorter calculation time of the proposed method. The mean calculation time for a single IK solution required for this algorithm is only 4% of the comparison algorithm.

Highlights

- An improved IK method for 6-DOF robot manipulators with an offset wrist is proposed.
- The method does not require a selection of initial estimation of joint variables.
- The proposed algorithm is simple, highly efficient, and suitable for real-time control.

1. Introduction

In the kinematics of robot manipulators, the forward kinematics (FK) function is straightforward and unique, while the solution of the inverse kinematics (IK) is complicated due to its nonlinear and coupled equations. In fact, the inverse problem (i.e., mapping the pose of the end effector from a Cartesian space to a joint space) is a problem of real practical interest since the motion is usually dealt with in Cartesian space. Therefore, solving the IK efficiently has been one of the basic challenges in robotics. [1] For the IK solution of a 6-DOF articulated robot manipulator, previous research generally focuses on the Euler as well as the spherical wrist structure (Fig. 1(a)) satisfying the Pieper criterion that the three adjacent joint axes of the robot intersect at a common point or the three axes are parallel (e.g., classical Puma series and Stanford robots). [2] Such kinds of wrist structures have been completely solved analytically [3–5] However, the robot manipulators with Euler or spherical wrists also have their own limitations and are unable to meet the practical requirements. For instance, the conventional robot manipulators with Euler wrist structure are not satisfactory in terms of load capacity and flexibility, which are difficult to cater for the craft requirements or the production of certain industries. [6] With the extension of

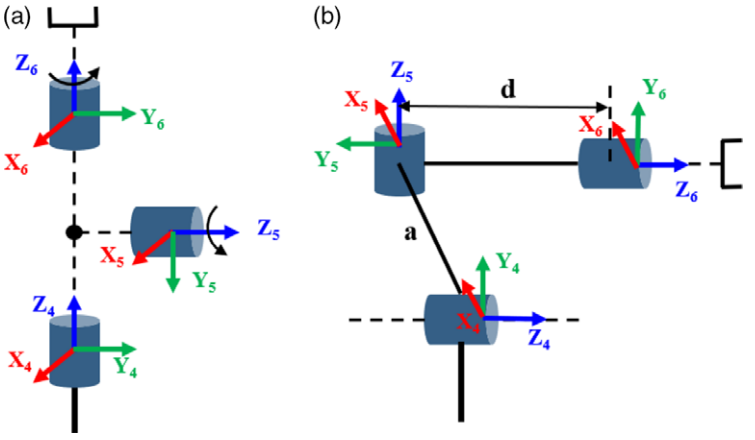

Figure 1. *The structure of wrist (a) Euler wrist (b) offset wrist.*

application fields, robot manipulators with offset wrists have been designed and adopted,[7–9] and the sketch of its structure as shown in Fig. 1(b). Because of its better dexterity, the offset wrist structure is suitable for many specific tasks such as welding, material handling, and machine tending; besides, this structure is allowed to form hollow wrist structure to lay the cables inside that is especially important for painting. Furthermore, the singularity set is different, in particular wrist singularities are eliminated. [10] However, this type of robot manipulator possesses a special geometric structure failing to conform to Pieper criterion in practical applications, which results in being hard to find closed-form solutions (i.e., analytical solution) for the IK. [11]

There have been lots of methods carried out to calculate the IK solutions utilizing numerical techniques. Jacobian-based inverse methods are the most commonly used method so far, such as the Newton-Raphson, the Jacobian Pseudo-inverse, the Jacobian Transpose, the predictor-corrector, and the damped least-squares methods. [1,12–15] The efficiency of some of these methods is determined by the selection of initial value and the singularity of the robot structure. Although some of the methods are modified to deal with singularity, those methods need to pay the corresponding cost (accuracy or computation time). For instance, the greatest advantage of using Jacobian transpose is to avoid inverse operation; however, the convergence rate of the iteration is slower and the output torque for manipulator controlled by this method at the joints far from the end effector is usually larger. Other methods that do not need to take Jacobian matrix and singularity into considerations are gradient-based nonlinear algorithms treating the IK problem as an equivalent minimization problem. [14,16] However, the efficiency of the methods decreases significantly as the nonlinearity and constraints increase. [17] On the other hand, heuristic and metaheuristics techniques have proposed to address the IK problems for robot manipulators with offset wrists. [18–22] Although this kind of intelligent optimization algorithms have the potential to solve the IK problem with singularity robustness (i.e., the property of providing us with continuous and feasible solutions even at or in the neighborhood of singular points), the local convergence rate is generally expected to be slow and it is unsatisfactory from the perspective of real-time control. Recently, Shi et al. [23] developed a method based on Adaboost Neural Network to solve the IK of robot with offset wrist, and this method indicated a good performance in both accuracy and stability. Xu et al. [17] proposed a hierarchical iterative algorithm to deal with the IK problem, which combined heuristic initial estimation and analytical calculation. Metin Toz [22] proposed a meta-heuristic optimization algorithm, but solving the IK problem of serial robot manipulators with offset wrists is just one of the applications of the method without detailed discussion of the IK problem. Li et al. [24] came up with a novel IK method for 6-DOF robots with nonspherical wrist inspired by the idea of virtual wrist center and implement a comparative study with the NR method.

This paper comes up with an improved method to solve the IK for the 6-DOF robot manipulators with offset wrists based on the Newton iteration technique. This method integrates wrist reconstruction by

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

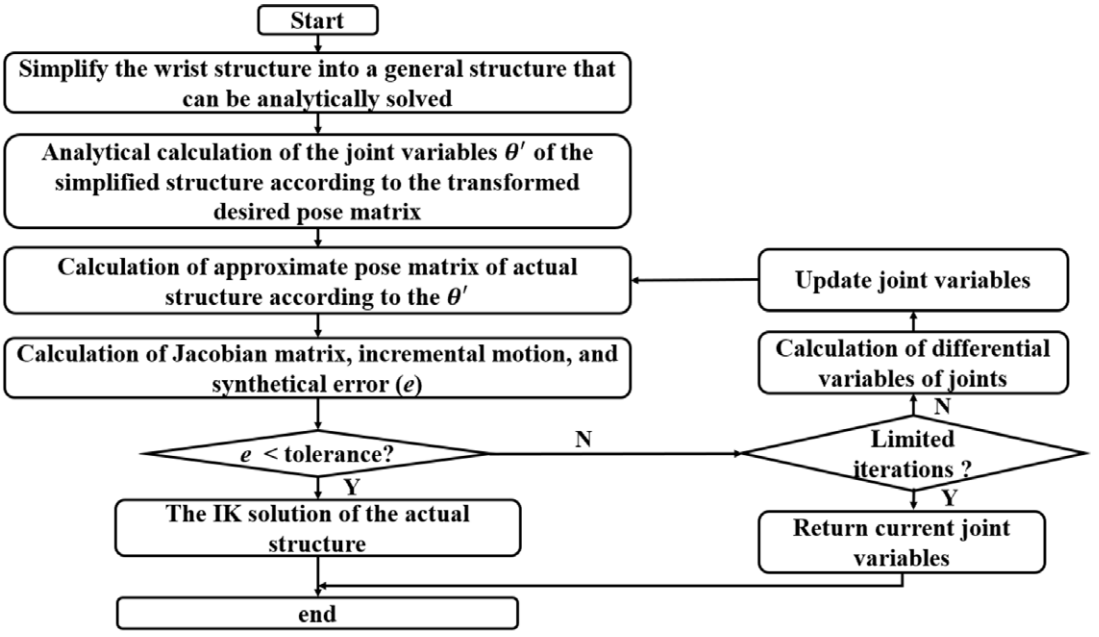

Figure 2. *Framework of the algorithm.*

translating the certain coordinate frames and the Newton iteration method. The input of the algorithm only requires the homogeneous matrix of the desired pose instead of both the matrix and the initial estimation of the joint variables as previous conventional algorithms. This algorithm is simple and suitable for real-time control. The paper is organized in the following manner. Section 2 introduces the framework of the algorithm along with elaborates the simplified reconstruction process of a specific robot manipulator with an offset wrist and the corresponding deduction procedure for the IK. After that, Section 2.4 carries out simulation and comparison experiments to investigate the performance of the proposed method.

2. Methodology

The method proposed in this paper involves two main steps to obtain the IK solution of the 6-DOF robot manipulators with offset wrists. The algorithm framework is depicted in Fig. 2. In the first step, the offset wrist structure is simplified into a common structure conforming the Pieper criterion via translating the relevant coordinate frames of the joints. The IK solution of the simplified structure can be analytically achieved after giving desired pose. The second step is to solve the IK problem iteratively based on the approximate pose that the joint coordinates of the simplified structure are the input to the FK of the actual structure. Specifically, the Jacobian matrix is used to establish the relationship between the differential Cartesian space motion and the joint variables, where the Cartesian space motion comprises positional and orientational components, and the approximate orientation associated with the desired orientation is expressed by the equivalent angle-axis representation. Differential increment of the joint variables can be mapped by the differential motion in Cartesian space through the Jacobian matrix. Then the new approximate pose is generated with differential increment of joint variables, and it will be compared with the desired pose again. The resulting synthetical error between the desired pose and the approximate pose is determined by the position error and the orientation error. If the synthetical error fails to meet the requirement of the error tolerance, the iteration will be repeated until the error of IK solution satisfies the accuracy or reaches the maximum number of iterations. As an example, the kinematics of a 6-DOF manipulator with an offset wrist is studied in this paper.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

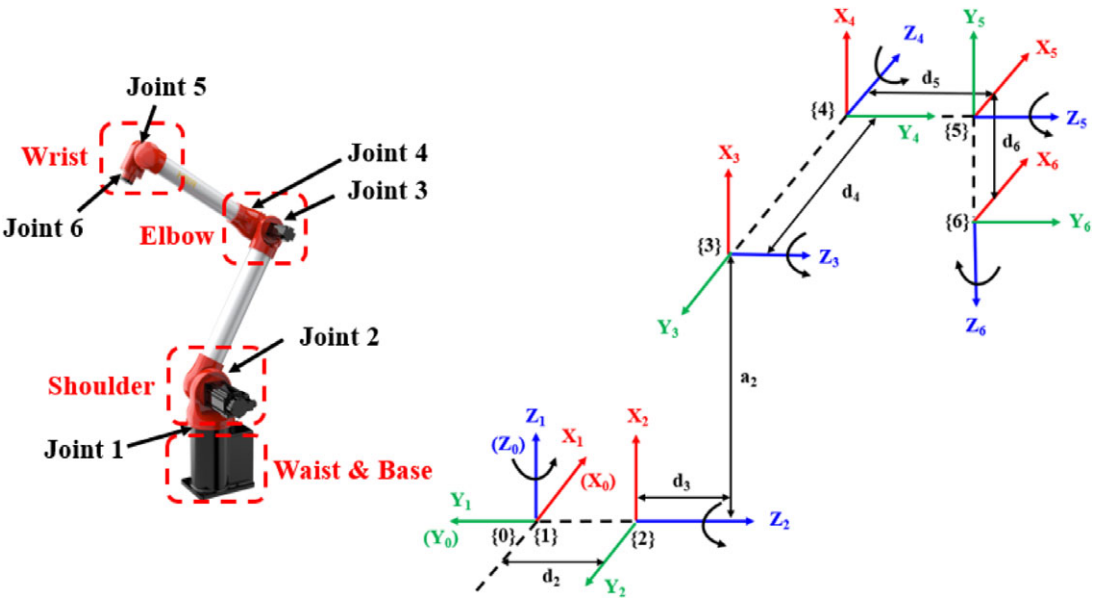

Figure 3. *Structure and coordinate frames of the HSR-BR606 robot manipulator.*

2.1. Simplification of wrist structure

The structure of the robot manipulator investigated herein is the HSR-BR606 developed by Huashu Robot Co., Ltd. and Foshan Institute of Intelligent Equipment Technology. The specific structure and the corresponding coordinate frames based on the modified Denavit-Hartenberg model [25] are illustrated in Fig. 3. The offset structure allows the robot to work in a narrow space constraint such as factory for intensive stamping and inside the machining center. In order to be more intuitive, the three views and axonometric drawing of the robot are shown in Fig. 4.

As discussed in Section 1, it is difficult to find a closed-form solution of the IK for such kind of robot manipulator. To obtain the numerical solutions of the IK, it is necessary to search a reasonable initial estimation of the joint configuration. Heuristic techniques may be suitable to deal with the estimation, but implementation of the real-time control is not satisfactory due to the large amount of computation for each estimation. According to the coordinate frames shown in Fig. 3, it can be easily simplified into an Euler wrist structure by translating the frames {5} and {6} at a distance of $-d_5$ along the Y axis of the frame {4}. It can be seen in Fig. 5 that the Z axes (joint axes) of the frame {4}{5}{6} intersect at a common point called virtual wrist center [26] after translation. The position and orientation of the center can be decoupled and thus the analytical solution of the IK can be obtained in this case.

2.2. Forward kinematics

The end pose in Cartesian coordinate can be mapped from the known robot manipulator configuration by the FK. In the IK solution method, the FK is used to identify the current pose according to the estimated joint variables. Kinematics structure of the robotic manipulators in this paper is described by the modified DH (MDH) convention, and the parameters with respect to the coordinate frames of the HSR-BR606 in Figs. 3 and 5 are illustrated in Table I. The specific values of parameter inside are $d_2 = 96$, $a_2 = 726$, $d_3 = 126.5$, $d_4 = 630.5$, $d_5 = 91$, $d_6 = 122$, where α_{i-1}, a_{i-1}, and d_i are structure parameters. α_{i-1} and a_{i-1}, respectively, are the rotation and translation between Z_i and Z_{i-1} about the X_{i-1} axis. d_i is the distance from the X_{i-1} axis the X_i axis along the Z_i axis. The structure parameters are known and the joint variables θ_i (i=1, 2, 3, 4, 5, 6) vary with the motion. Table II shows the rotation limits of each joint, and the constraints will be considered in the simulation that the group of angles are selected within the ranges.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

Table I. *The MDH parameters of the HSR-BR606.*

i	θ_i (rad)	a_{i-1} (mm)	α_{i-1} (rad)	d_i (mm)
1	θ_1	0	0	0
2	θ_2	0	$\pi/2$	d_2
3	θ_3	a_2	0	d_3
4	θ_4	0	$\pi/2$	d_4
5	θ_5	0	$-\pi/2$	d_5
6	θ_6	0	$\pi/2$	d_6

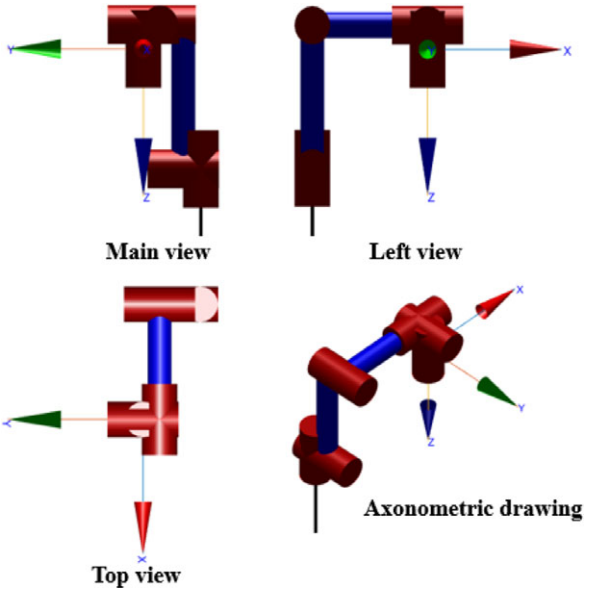

Figure 4. *The three views and axonometric drawing of the robot manipulator.*

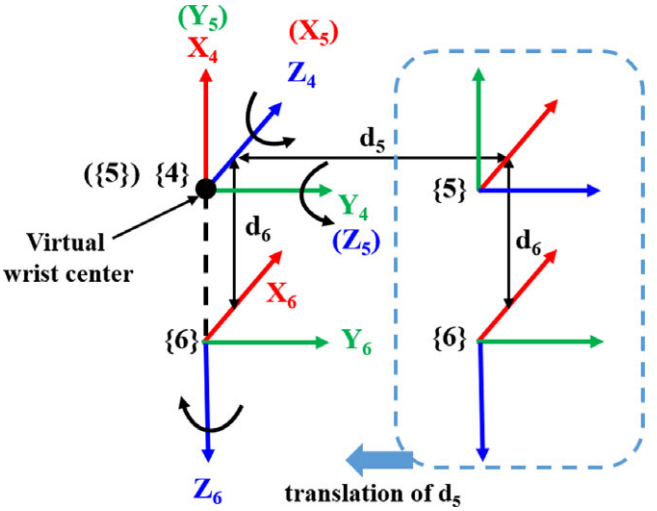

Figure 5. *The changes in wrist structure before and after simplification.*

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

Table II. *Joint ranges of the HSR-BR606.*

	θ_1(deg)	θ_2(deg)	θ_3(deg)	θ_4(deg)	θ_5(deg)	θ_6(deg)
min	–155	2	–218.5	–90	–96.5	–180
max	155	178	38.5	90	96.5	180

According to the MDH convention, the homogeneous matrix transforming link coordinate frame $\{i\}$ with respect to frame $\{i\text{-}1\}$ is expressed as [2]

$$\begin{aligned} {}^{i-1}_{i}T(\theta_i) &= Rot_X(\alpha_{i-1})Trans_X(a_{i-1})Rot_Z(\theta_i)Trans_Z(d_i) \\ &= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

The end pose homogeneous matrix of robot in the base coordinates can be expressed as

$${}^{0}_{6}T = {}^{0}_{1}T\,(\theta_1)\,{}^{1}_{2}T\,(\theta_2)\,{}^{2}_{3}T\,(\theta_3)\,{}^{3}_{4}T\,(\theta_4)\,{}^{4}_{5}T\,(\theta_5)\,{}^{5}_{6}T\,(\theta_6) = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \tag{2}$$

where R represents a 3×3 rotation matrix and P is a 3×1 position vector of transformation matrix ${}^{0}_{6}T$.

2.3. Inverse kinematics

In the proposed method, the IK involves solution of the simplified wrist structure and the actual offset wrist structure.

2.3.1. IK solution for simplified structure

The 6-DOF robot manipulator with simplified structure that satisfies the Pieper criterion can be obtained, when $d_5 = 0$. In this case, the end position of simplified mechanism is still not exactly the position of virtual wrist center. It can be seen in Fig. 5 that there is a position translation (d_6, along the Z of $\{6\}$) between them. The pose of the virtual wrist center is obtained by using the pose of the end coordinate frame ($[X\ Y\ Z\ A\ B\ C]$). Assuming the pose matrix of frame $\{6\}$ is as following:

$${}^{0}_{6}T = Rot_z(A)Rot_Y(B)Rot_X(C)Trans(X,Y,Z) = \begin{bmatrix} n & o & a & P \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & X \\ n_y & o_y & a_y & Y \\ n_z & o_z & a_z & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $n = [n_x\ n_y\ n_z]^T$, $o = [o_x\ o_y\ o_z]^T$ and $a = [a_x\ a_y\ a_z]^T$ are called normal vector, sliding vector, and approaching vector.

Then translating the origin of the frame $\{6\}$ to the virtual wrist center the pose matrix is expressed as

$$\begin{aligned} &Rot_z(A)Rot_Y(B)Rot_X(C)Trans(X,Y,Z)Trans(0,\ 0,\ -d_6) \\ &= \begin{bmatrix} n_x & o_x & a_x & X \\ n_y & o_y & a_y & Y \\ n_z & o_z & a_z & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & X - a_x d_6 \\ n_y & o_y & a_y & Y - a_y d_6 \\ n_z & z & a_z & Z - a_z d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4}$$

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

The position of virtual wrist center is determined by the former three joints, and it can be obtained from Eqs. (1)–(3):

$$ {}^{0}P_{4org} = {}_{1}^{0}T\,(\theta_1)\,{}_{2}^{1}T\,(\theta_2)\,{}_{3}^{2}T\,(\theta_3)\,{}^{3}P_{4org} = \begin{bmatrix} X - a_x d_6 \\ Y - a_y d_6 \\ Z - a_z d_6 \\ 1 \end{bmatrix} \tag{5} $$

where ${}^{0}P_{4org}$ and ${}^{3}P_{4org}$ are position vectors of transformation matrices ${}_{4}^{0}T$ and ${}_{4}^{3}T$, respectively:

$$ \begin{bmatrix} X - a_x d_6 \\ Y - a_y d_6 \\ Z - a_z d_6 \end{bmatrix} = \begin{bmatrix} (d_4 s_{23} + a_2 c_2) c_1 + (d_2 + d_3) s_1 \\ (d_4 s_{23} + a_2 c_2) s_1 + (d_2 + d_3) c_1 \\ a_2 s_2 - d_4 c_{23} \end{bmatrix} \tag{6} $$

where $s_i = \sin\theta_i,\ c_i = \cos\theta_i, s_{ij} = \sin\left(\theta_i + \theta_j\right), c_{ij} = \cos\left(\theta_i + \theta_j\right)$.

The former three joint variables (θ_1, θ_2, θ_3) can be obtained from Eq. (6). The last three joints variables (θ_4, θ_5, θ_6) only affect the orientation of the end. The difference between the orientation of coordinate frame {3} and coordinate frame {6} depends on the last three joints (θ_4, θ_5, θ_6).

$$ {}_{6}^{3}R = {}_{3}^{0}R^{-1}\,{}_{6}^{0}R = \begin{bmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_6 s_4 - c_5 s_6 & c_4 s_5 \\ c_6 s_5 & -s_5 s_6 & -c_5 \\ c_4 s_6 + c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & s_4 s_5 \end{bmatrix} \tag{7} $$

where ${}_{m}^{n}R$ is the rotation matrix of $\{m\}$ with respect to $\{n\}$.

If ${}_{6}^{3}R\,(2,\ 3) \neq \pm 1$, the last three joint variables θ_4, θ_5, and θ_6 can be obtained by Eq. (7), and the analytical solution is as following:

$$ \begin{aligned} \theta_5 &= \operatorname{acos}\left(-{}_{6}^{3}R\,(2,\ 3)\right) \\ \theta_4 &= \operatorname{atan2}\left(\frac{{}_{6}^{3}R\,(3,\ 3)}{s_5}, \frac{{}_{6}^{3}R\,(1,\ 3)}{s_5}\right) \\ \theta_6 &= \operatorname{atan2}\left(\frac{-{}_{6}^{3}R\,(2,\ 2)}{s_5}, \frac{{}_{6}^{3}R\,(2,\ 1)}{s_5}\right) \end{aligned} \tag{8} $$

2.3.2. IK solution for actual structure

Assuming the desired pose matrix (${}_{6}^{0}T_d$) of the coordinate frame {6} of the robot manipulator with offset wrist is given as following:

$$ {}_{6}^{0}T_d = \begin{bmatrix} n_{xd} & o_{xd} & a_{xd} & X_d \\ n_{yd} & o_{yd} & a_{yd} & Y_d \\ n_{zd} & o_{zd} & a_{zd} & Z_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9} $$

The pose matrix of the coordinate system {6} of the simplified structure can be obtained by translating the desired pose matrix along the Y axis of the coordinate frame {6} of actual structure in negative direction of the d_5.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

$$
{}^{0}_{6}T_{d}Trans\ (0,\ -d_5,\ 0) = \begin{bmatrix} n_{xd} & o_{xd} & a_{xd} & X_d \\ n_{yd} & o_{yd} & a_{yd} & Y_d \\ n_{zd} & o_{zd} & a_{zd} & Z_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -d_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$
$$
= \begin{bmatrix} n_{xd} & o_{xd} & a_{xd} & X_d - o_{xd}d_5 \\ n_{yd} & o_{yd} & a_{yd} & Y_d - o_{yd}d_5 \\ n_{zd} & o_{zd} & a_{zd} & Z_d - o_{zd}d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}
$$

The pose of coordinate frame {6} in Eq. (10) is equivalent to the end pose of the simplified structure in Section 2.3.1. The joint variables $\boldsymbol{\theta_s} = [\theta_{s1}\ \theta_{s2}\ \theta_3\ \theta_{s4}\ \theta_{s5}\ \theta_{s6}]$ of the robot manipulator with no offset wrist (i.e., simplified structure) can be obtained analytically by Eqs. (6) and (8). The joint variables are the approximate solutions of the IK for the robot manipulator with an offset wrist (i.e., actual structure). The joint variables $\boldsymbol{\theta_s}$ are then used in the iteration process. The approximate pose ${}^{0}_{6}T_s$ of the actual structure can be obtained by substituting the $\boldsymbol{\theta_s}$ into the FK and the homogeneous matrix is as following:

$$
{}^{0}_{6}T_s = {}^{0}_{1}T\ (\theta_{s1})\ {}^{1}_{2}T\ (\theta_{s2})\ {}^{2}_{3}T\ (\theta_{s3})\ {}^{3}_{4}T\ (\theta_{s4})\ {}^{4}_{5}T\ (\theta_{s5})\ {}^{5}_{6}T\ (\theta_{s6}) \tag{11}
$$

The synthetical error between the approximate pose and the desired pose can be defined as

$$
e = \|dP\|_2 + \|dAng\|_2 \tag{12}
$$

where dP is an incremental displacement and $dAng$ is an incremental rotation.

$$
dP = \begin{bmatrix} {}^{0}_{6}T_d\ (1,\ 4) - {}^{0}_{6}T_s\ (1,\ 4) \\ {}^{0}_{6}T_d\ (2,\ 4) - {}^{0}_{6}T_s\ (2,\ 4) \\ {}^{0}_{6}T_d\ (3,\ 4) - {}^{0}_{6}T_s\ (3,\ 4) \end{bmatrix} \tag{13}
$$

In term of the rotation increment between the approximate pose and the desired pose, the equivalent angle-axis representation is adopted. Based on the Rodrigues' rotation formula, the equivalent rotation matrix is [2]

$$
R\ (K, \varphi) = {}^{0}_{6}T_s(1{:}3,\ 1{:}3)^{-1}\,{}^{0}_{6}T_d\ (1{:}3,\ 1{:}3)
$$
$$
= \begin{bmatrix} k_xk_x\nu + \cos\varphi & k_xk_y\nu - k_z\sin\varphi & k_xk_z\nu + k_y\sin\varphi \\ k_xk_y\nu + k_z\sin\varphi & k_yk_y\nu + \cos\varphi & k_yk_z\nu - k_x\sin\varphi \\ k_xk_z\nu - k_y\sin\varphi & k_yk_z\nu + k_x\sin\varphi & k_zk_z\nu + \cos\varphi \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{14}
$$

where $\nu = 1 - \cos\varphi$, and $K = [k_x,\ k_y,\ k_z]$ is a unit vector passing through the origin of the coordinate system {6} of the robot manipulator with offset wrist. φ is the angle of rotation around the K vector. The desired orientation will be achieved after rotating the approximate orientation about K vector by the angle φ.

The vector K and the angle φ can be solved from the given rotation matrix [27]:

$$
\varphi = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \tag{15}
$$

$$
K = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} = \frac{1}{2\sin\varphi} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \tag{16}
$$

Thus, the incremental rotation $dAng$ is expressed as

$$
dAng = {}^{0}_{6}T_s\ (1{:}3, 1{:}3)\ K\varphi \tag{17}
$$

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

Combining the position increment dP and rotation increment $dAng$ into a vector $dX = [dP\ dAng]^T$ expressed in world coordinate, the dX is the differential motion (6x1) corresponding to the infinitesimal motion from approximate pose to desired pose.

If the synthetical error e is within the reasonable tolerance, the iteration will be ceased and the numerical IK solution of the robot manipulator with offset wrist can be obtained. Otherwise, new possible joint variables for next iteration will be generated based on the differential variables of the joint configuration $d\theta$, which is calculated by the Jacobian matrix and the differential motion dX.

$$d\theta = J^{-1}dX \tag{18}$$

The Jacobian matrix of the actual structure with offset wrist is expressed as following:

$$\mathrm{J} = \begin{bmatrix} J_{pi} & \cdots & J_{pn} \\ \vdots & \ddots & \vdots \\ J_{Oi} & \cdots & J_{On} \end{bmatrix}$$

$$\begin{bmatrix} J_{pi} \\ J_{Oi} \end{bmatrix} = \begin{bmatrix} {}^0Z_i \left({}^0P_n - {}^0P_i\right) \\ {}^0Z_i \end{bmatrix} \mathrm{i} = 1, 2, 3, \cdots, 6,\ n = 6 \tag{19}$$

where ${}^0Z_i$ is the vector representation of the Z axis of in the coordinate frame $\{i\}$ with respect to the base coordinate frame $\{0\}$. ${}^0P_i$ is the vector representation of the origin of the coordinate frame $\{i\}$ with respect to the base coordinate frame $\{0\}$.

The new possible joint variables θ can be obtained by Eq. (20). Substituting the new joint variables into Eq. (2), the current pose of the robot manipulator with offset wrist can be obtained and the synthetical error between the current pose and the desired pose will be calculated to decide whether the joint variables need to refresh again or not.

$$\theta = \theta + d\theta \tag{20}$$

2.4. Inverse kinematics algorithm flow

The key steps for the proposed IK algorithm are as followed:

(1) Translating the frame {6} of 6-DOF robot manipulator with offset wrist in its Y axis by d_5 to obtain the pose of frame{6} of 6-DOF robot with no offset wrist. The joint variables of simplified structure can be obtained by Eqs. (3)–(8);

(2) Using the joint variables of simplified structure as the input of Eq. (2) to get the pose matrix of the frame {6} of 6-DOF robot manipulator with offset wrist.

(3) Calculating the position increment dP and rotation increment $dAng$ by Eqs. (13)–(17) and obtaining the synthetical errors e by Eq. (12).

(4) Judging whether the synthetical error e is within the pre-set criteria or not. If the synthetical error is within the reasonable error range, the iteration is stopped and a current value of the joint variables as the iterative numerical solution of the IK of the 6-DOF robot manipulator with offset wrist is returned. If the synthetical error is out of range, the $d\theta$ is calculated by the Jacobian matrix of the 6-DOF robot manipulator with offset wrist and then get new joint variables for next iteration.

(5) Repeating step (2)-step (4) until the synthetical error is within the pre-set criteria or the number of iteration reaches the maximum setting number.

The generalized pseudocode description is as following:

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

The pseudocode of algorithm

Input: desired homogenous matrix of the end pose (T_d)
Initial: Tolerance $\boldsymbol{\varepsilon}$, MDH parameters, setting limit ranges of each joint
$T_d\_N = T_d * Trans() * Trans()$ % Translate the frames to a virtual point
$T_d\_N \rightarrow \boldsymbol{\theta}_S$ % analytical solution of simplified structure
$T_s = FK(\boldsymbol{\theta}_S)$ % FK function is based on actual structure
$[K, \varphi]$ = Equivalent angle-axis representation (T_s, T_d)
$[K, \varphi] \rightarrow$ dAng
[dP, dAng] $\rightarrow$ dX; $\boldsymbol{e}$=norm(dP)+norm(dAng)
while e > $\boldsymbol{\varepsilon}$ or within the maximum number of iteration
  $J = J\_OffsetWrist(\boldsymbol{\theta}_S)$
  $d\boldsymbol{\theta}_S = J \setminus dX$
  $\boldsymbol{\theta}_S = \boldsymbol{\theta}_S + d\boldsymbol{\theta}_S$
  repeat steps 5∼8
end while
return $\boldsymbol{\theta} = \boldsymbol{\theta}_S$

3. Simulations and discussion

In this section, the proposed algorithm is tested and discussed in terms of accuracy, efficiency and application for motion planning. To ensure the reproducibility of experiments and consistency, the model of the robot manipulator to implement the validation of the algorithm is the same as mentioned above, and the structure parameters are summarized in Section 2.2. After evaluating the performance of the proposed algorithm, a comparative experiment is conducted between the algorithm and the improved Jacobian transpose (IJT) algorithm developed by Corke et al. [28] The programming and experiments are developed and carried out on a CPU of Intel Core i5-7400 3.00 GHz with 16 GB RAM in MATLAB 2019b software.

3.1. Accuracy and efficiency

In order to investigate the accuracy and efficiency for the proposed algorithm, the first experimental scheme is that the joint coordinates vary from Pose A (20° 90° 30° 10° –90° 0°) to Pose B (–80° 50° 15° –80° 0° 10°) as illustrated in Fig. 6, and 1000 sets of joint angles between Poses A and B are interpolated by a quintic (5th order) polynomial in joint space. After interpolation, the corresponding homogeneous matrices can be obtained by FK, and the joint variables are solved via the proposed IK algorithm.

The distribution of the error during convergence can be seen in Fig. 7 where the convergence criteria is set to 1×10^{-6} in this test. The synthetical error (e) includes position error (dP) and orientation error ($dAng$) as mentioned in Eq. (12). In other word, the synthetical error can be considered as a spatial displacement contributed by the incremental displacement and incremental rotation. Figure 7 shows that the synthetic errors of 1000 test poses are within a reasonable range and most of the errors are much lower than the convergence criteria. For the first two hundred groups of poses, the synthetical errors are mainly dominated by the position error. In sequence number of trajectory points from 200 to 323, the orientation errors are dominant, and the peak of norm ($dAng$) reaches 9.698×10^{-7}. For the points from 700 to 850, the contribution of position error and orientation error is relatively obvious, and the orientation component is slightly larger than the position component. The average synthetical error is 1.09×10^{-7}, and its standard deviation is 2.07×10^{-7}.

Figure 8 shows the differences between the joint angles solved by the algorithm and the interpolated joint angles. It can be seen in Fig. 8(a) that large differences appear in trajectory points of sequence number around 830 at about Joint 4 and Joint 6, which have a greater influence on the orientation error of the robot. Figure 8(b) is the enlargement of the gray region (points 1–400) in Fig. 8(a). The differences

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

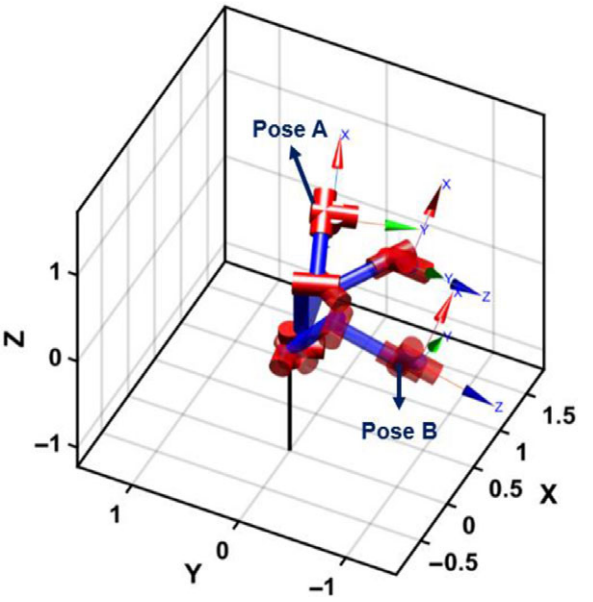

Figure 6. *The poses of the robot manipulator in the simulation. Starting from Pose A (20° 90° 30° 10° –90° 0°) to end Pose B (–80° 50° 15° –80° 0° 10°).*

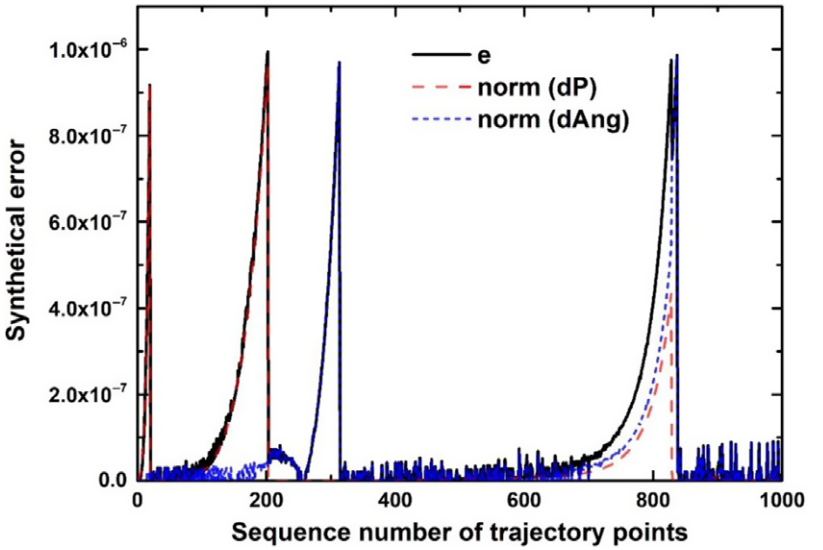

Figure 7. *The distribution of the error and its components during convergence.*

of all joint angles are within $\pm 3 \times 10^{-6}$ rad. The peaks in Fig. 7 can be explained by the accumulation of the angle difference of each joint in Fig. 8. In other words, Fig. 8 shows the detailed information of the synthetical errors in terms of the difference between calculation angle and target angle of each joint. Usually, the first three joints cooperate to determine the end position, and the last three joints determine the orientation. Because of the offset structure, the rotation of Joint 5 affects both the position and orientation of the end frame. For instance, the peaks of Joint 3 and Joint 5 of sequence number 200 in Fig. 8(b) result in the peak of synthetical error of the same sequence number in Fig. 7. The mean, standard deviation, maximum, and minimum of differences for each joint are listed in detail in Table III.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

Table III. *Statistical analysis of differences of the IK solution for each joint.*

	Joint 1 (10^{-6} rad)	**Joint 2 (10^{-6} rad)**	**Joint 3 (10^{-6} rad)**	**Joint 4 (10^{-6} rad)**	**Joint 5 (10^{-6} rad)**	**Joint 6 (10^{-6} rad)**
Min	–0.116	–1.44	–2.71	–2.02	–0.857	–13.5
Max	1.17	1.04	2.27	14.6	1.67	1.95
Mean	0.0186	0.0476	0.134	0.223	0.0932	0.208
Std	0.0921	0.153	0.369	1.17	0.229	1.08

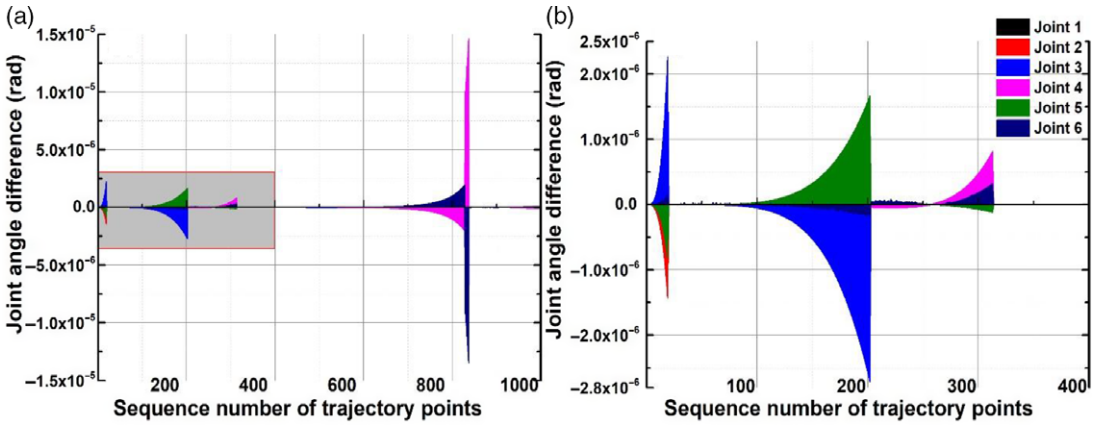

Figure 8. *The differences between the joint angles solved by algorithm and the interpolated joint angles.*

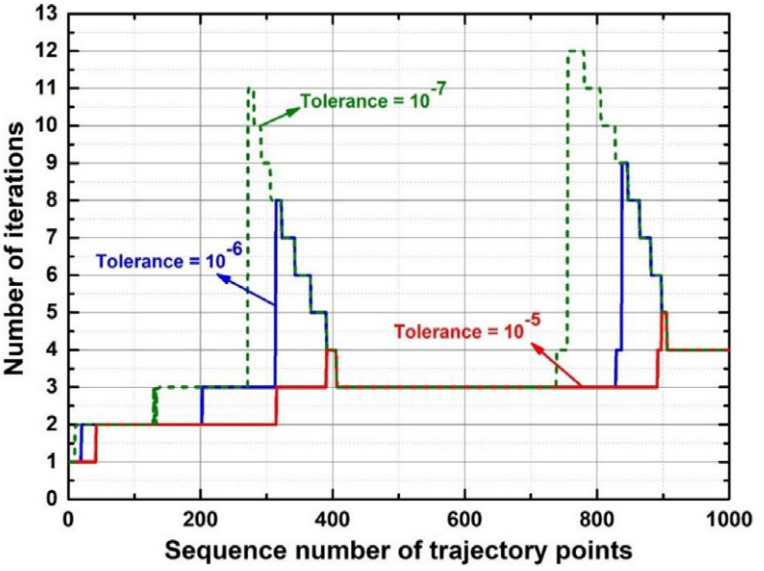

Figure 9. *The number of iterations for different tolerance.*

To investigate the computational efficiency, the number of iterations, and computation time are also considered in the same experiment. As shown in Fig. 9, when the convergence accuracy (i.e., tolerance) is set to 1×10^{-7}, 1×10^{-6}, and 1×10^{-5}, the maximum number of iterations is 12, 9, and 5, respectively. Figure 10 shows the calculation time of the algorithm in terms of a single joint configuration and the joint configurations from pose A to pose B. It turns out that the calculation time is unstable at the beginning

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

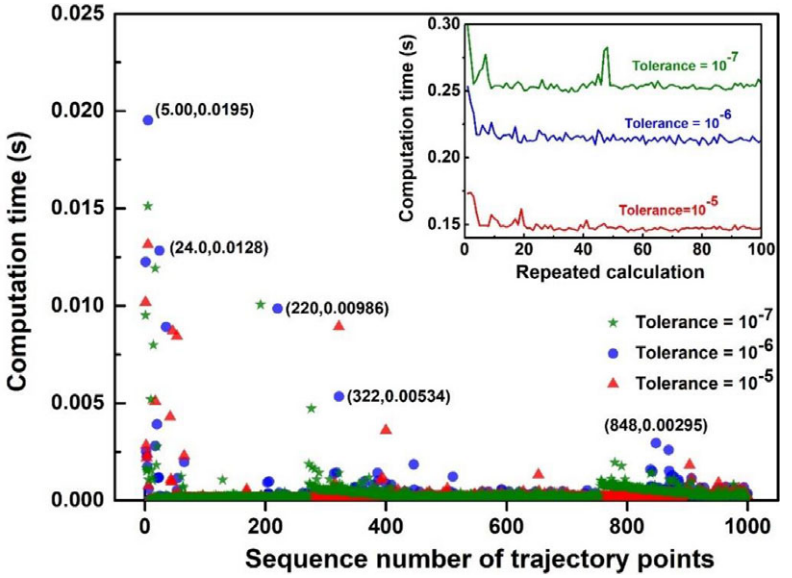

Figure 10. *The computation time of single joint configuration for different tolerance and the computation time of 1000 joint configurations for different tolerance (inset graph).*

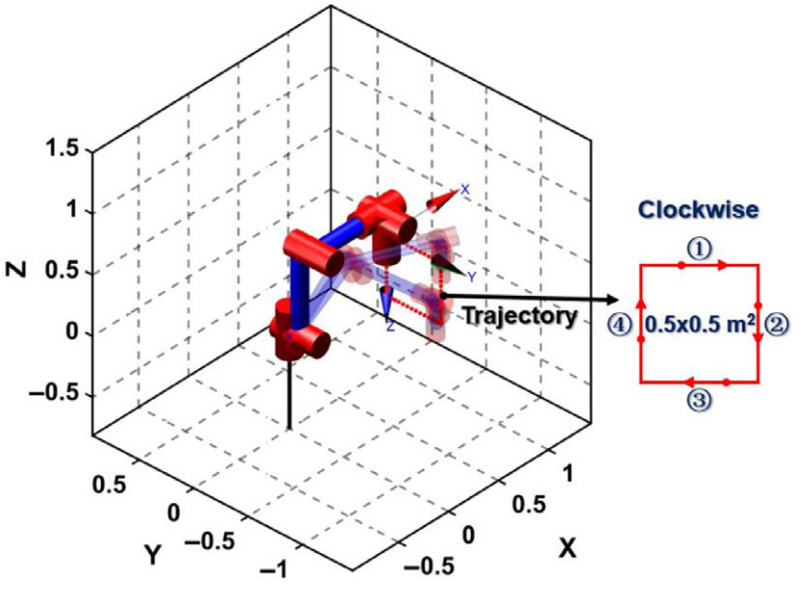

Figure 11. *The motion trajectory for rectangular path.*

and the maximum time consumption of the algorithm is 0.0195 s in test 5 for tolerance of 1×10^{-6}. The content of the brackets next to some points refer to the specific computation time for blue circle points in corresponding test number, which is used as a reference for neighboring points. For calculation of a single joint configuration, the mean time is 0.393, 0.346, and 0.263 ms for tolerance of 1×10^{-7}, 1×10^{-6}, and 1×10^{-5}. An inset graph in the upper right corner illustrates the computation time for the 100 repeated calculations of the 1000 joint configurations for different tolerance. It can be seen in Fig. 10 that the IK solutions from pose A to pose B take up to 0.3, 0.25, and 0.18 s for tolerance of 1×10^{-7}, 1×10^{-6}, and 1×10^{-5}, respectively. The mean time consuming is 0.255, 0.149, and 0.215 s.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

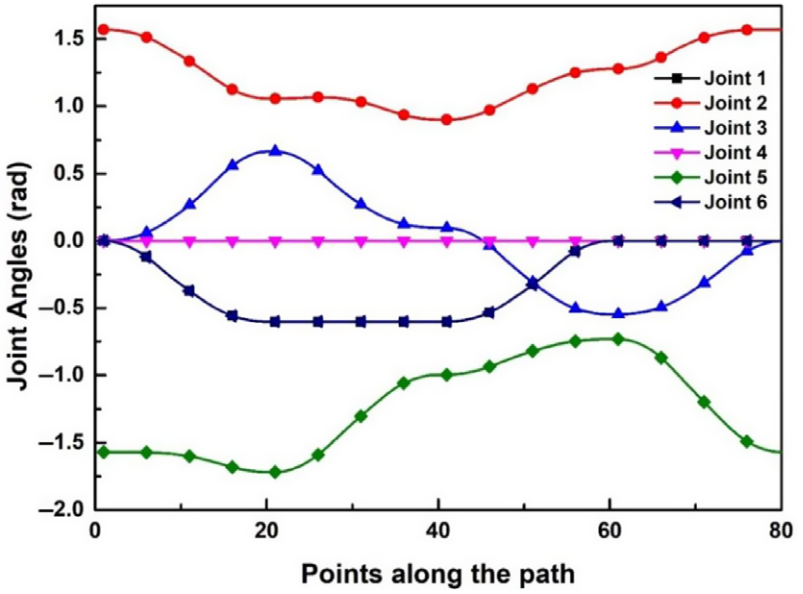

Figure 12. *The joint trajectories for the rectangular path.*

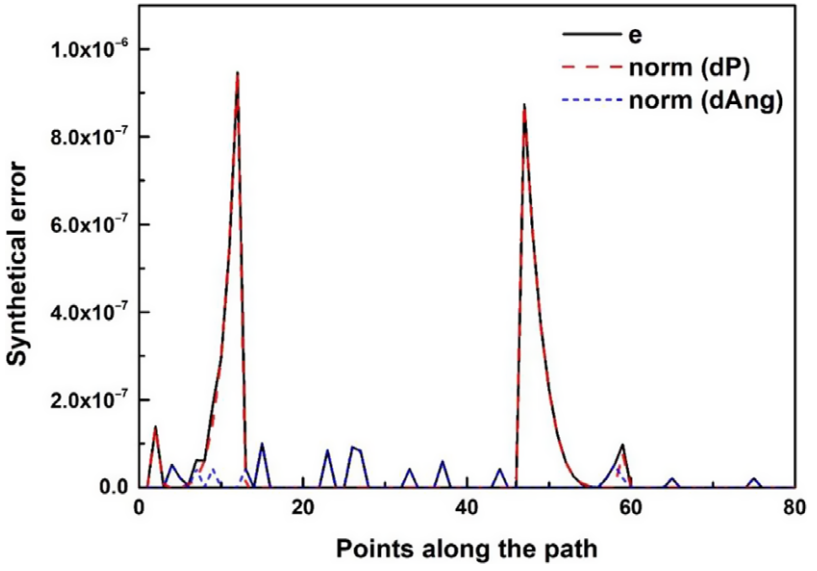

Figure 13. *The distribution of the error along the path.*

3.2. Motion planning

In this section, the performance of algorithm to track a trajectory planed in Cartesian coordinate is discussed. The trajectory is a rectangle and the motion of the end of robot manipulator starts and ends with the initial pose (0° 90° 0° 0° –90° 0°) as illustrated in Fig. 11. The rectangle with 0.5 m sides totally consists of 80 points described by homogeneous matrices and each side has 20 points. In the simulation, the end moves clockwise along the sides and the convergence accuracy is set to 1×10^{-6}. Figure 12 shows six joint trajectories changing with the points of rectangular path. The trajectories are continuous and smooth. The synthetical error and its components during the convergence can be seen in Fig. 13. Overall, the synthetical errors are much smaller than the convergence criterion except the two peaks caused by the position errors. The orientation errors are below 1×10^{-7}.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

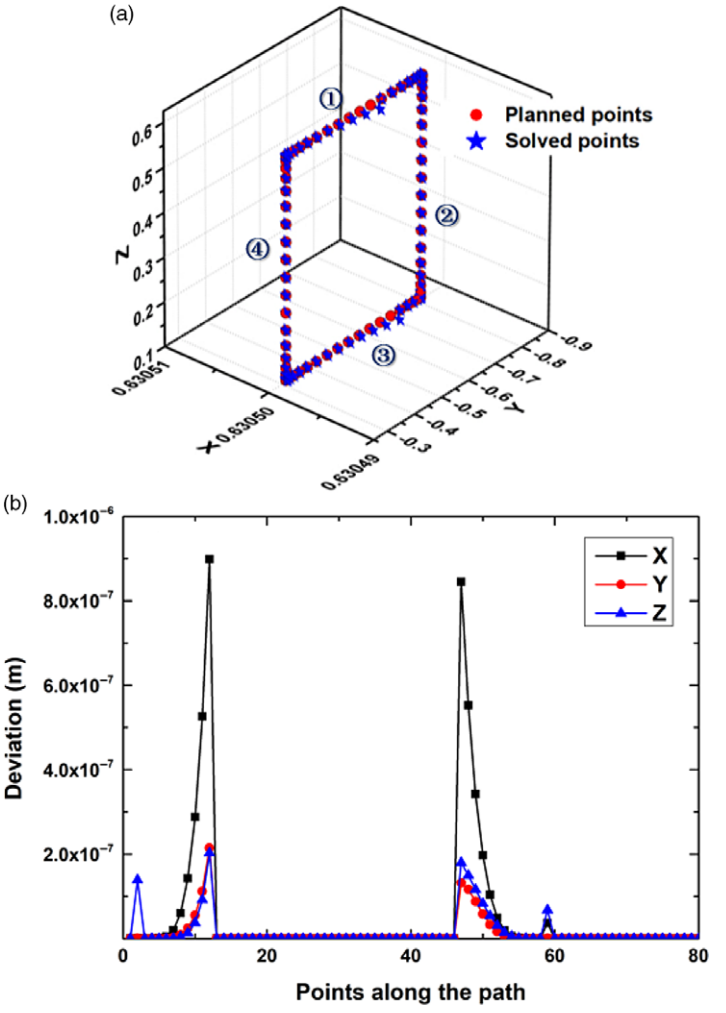

Figure 14. *Actual tracking of the rectangular path. (a) The comparison between the actual trajectory and the planned trajectory. (b) The deviation of each axis between actual track and planned track.*

To be more intuitive, Fig. 14(a) shows the actual tracking of the rectangular path. Red circle points are the planned points, and blue star shaped points are the points solved by the proposed method. The motion planning sequence in Fig. 14(a) is marked by 1, 2, 3, and 4 corresponding to the Fig. 11. In general, the proposed algorithm can track all planned points within a synthetical error of 1×10^{-6} as mentioned above. Figure 14(b) specifically illustrates the deviation of each axis (X-Y-Z axis of Cartesian coordinates) between actual tracking points and planned points, where the large errors, respectively, appear in point 12 and point 47 in x-axis direction in sequences 1 and 4, and the corresponding deviations are 8.989×10^{-7} and 8.452×10^{-7} m. The peaks in Fig. 13 correspond to the large errors in Fig. 14. The deviation in Fig. 14(b) shows that the position error peaks of Fig. 13 is mainly caused by the errors in the x-axis direction.

For efficiency, the number of iterations and computation time are investigated as well. Figure 15 shows that the maximum number of iterations is 4, which is less than the first experimental scheme discussed in Section 3.1 because of the relatively simple trajectory in the Cartesian space. Figure 16 shows the computation time for motion planning of a rectangular. To obtain a reasonable estimation, the calculation is repeatedly implemented for 100 times. The range of the calculation time is between

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

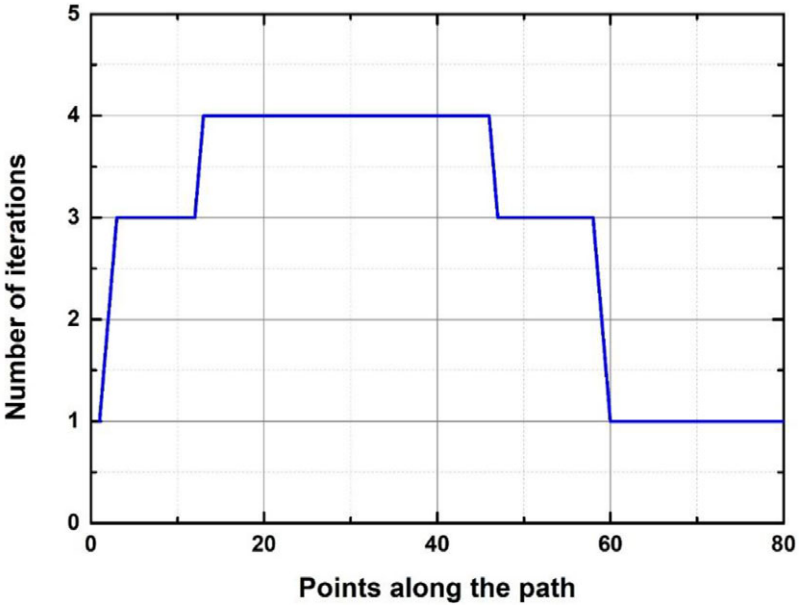

Figure 15. *The number of iterations for motion planning of a rectangular.*

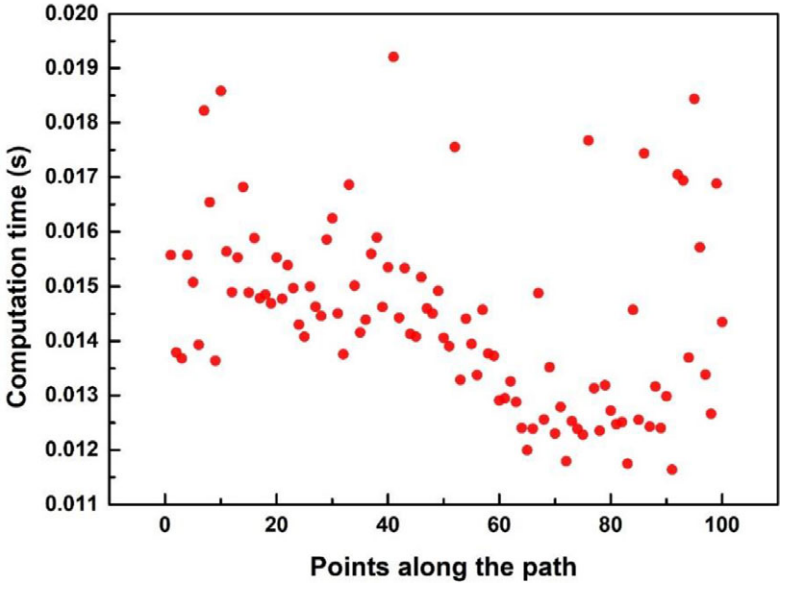

Figure 16. *The computation time for motion planning of a rectangular (a hundred repetitions of calculations).*

11 and 20 ms, and the mean and standard deviation are 14.43 and 1.66 ms, which satisfy the real-time control for the robot manipulator.

3.3. Comparing with the IJT algorithm

To valid the effectiveness of the proposed algorithm, a general numerical approach (IJT algorithm) for the IK from ref. [28] is used as a reference. The IJT algorithm is developed based on the Jacobian

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

Table IV. *Statistical analysis of synthetical errors for the proposed and IJT algorithm.*

Algorithm	Mean(10^{-6})	Max(10^{-6})	Std(10^{-6})
The proposed algorithm	0.067	0.946	0.173
The IJT algorithm	0.242	0.959	0.265

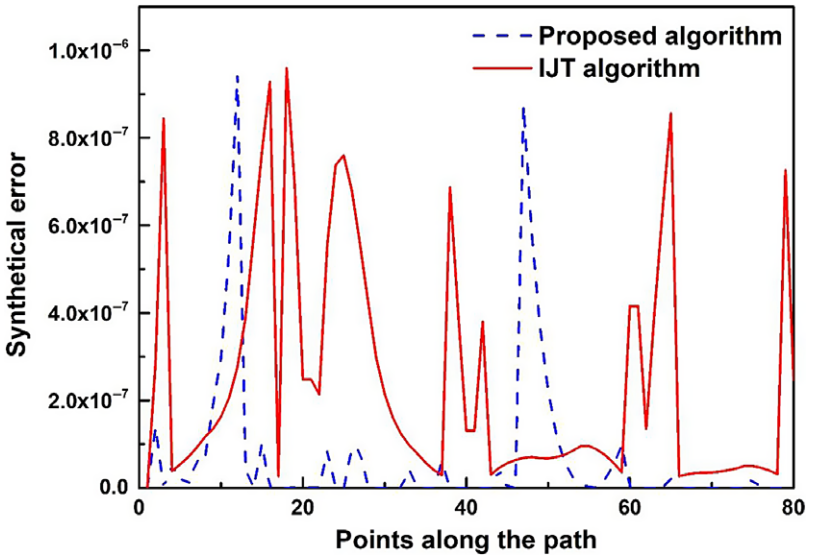

Figure 17. *The distribution of synthetical errors of the proposed and IJT algorithm.*

transpose and the Levenberg–Marquardt optimization algorithm which are suitable for any manipulator configuration. The comparison scheme is that both methods are used to solve the IK of the rectangular trajectory illustrating in Fig. 11. The initial estimation of the joint variables of the IJT algorithm is the joint variables of the previous pose, which aims to speed up the convergence. Figure 17 shows the distribution of the synthetical errors by using the IJT algorithm to track the rectangular path. Compared to the proposed algorithm, the IJT algorithm has more peaks, which indicates a relatively lower accuracy. Specifically, Table IV shows the statistical analysis of the synthetical errors of the two methods. It can be seen that the accuracy of the algorithm proposed in the paper is three times higher than that of the IJT algorithm.

Figure 18 shows the comparison of the calculation time regarding to a single point and a complete rectangular path for the two algorithms. It can be seen in Fig. 18(a) that there are 4 valleys corresponding to points 1, 21, 41, and 61 for the IJT algorithm. The pose of point 1 is the same as the initial pose, while points 21, 41, and 61 are the same corner points as the previous poses (i.e., the poses of points 20, 40, and 60). For calculating the IK of a complete rectangular path (a total of 80 points), the calculation time of the IJT algorithm fluctuates between 2.075 and 2.175 s, while the proposed algorithm is between 0.013 and 0.018 s. The statistical analysis of the calculation time of the two algorithms is shown in Table V. Figure 19 shows the number of iterations of the two algorithms for each planning point. The valleys here are corresponding to the valleys as shown in Fig. 18(a), and the reason of the formation is the same as mentioned above. The maximum number of iterations of the IJT algorithm is 8 and the algorithm proposed is only half of it. Note that the number of iterations for the proposed algorithm is independent to the previous pose, and it depends on the pose of the simplified structure of the robot manipulator.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

Table V. *Statistical analysis of computation time for the proposed and IJT algorithm.*

Algorithm	Mean time for calculating a single point (s)	Mean time for calculating a complete path (s)
The proposed algorithm	0.00119	0.01438
The IJT algorithm	0.02846	2.09050

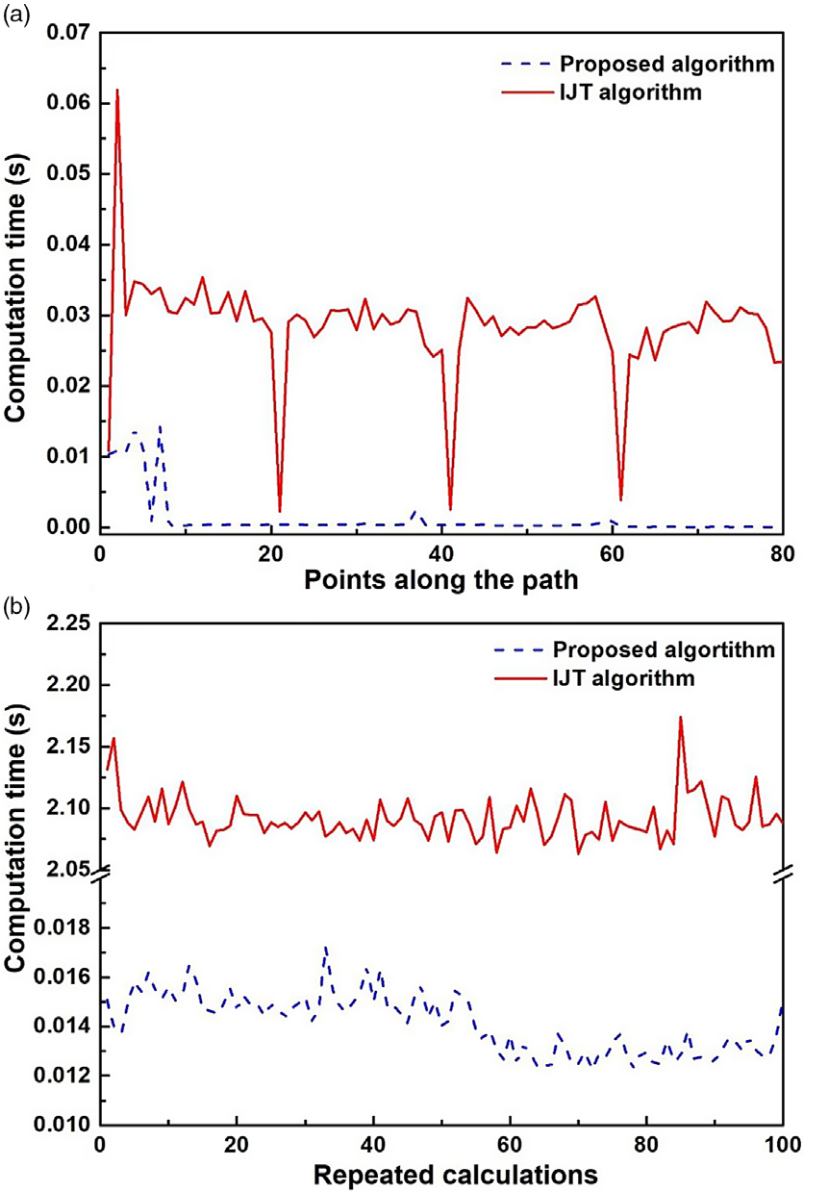

Figure 18. *The computation time of two algorithms (a) The computation time for each single point. (b) The computation time for motion planning of a rectangular (a hundred repetitions of calculations).*

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

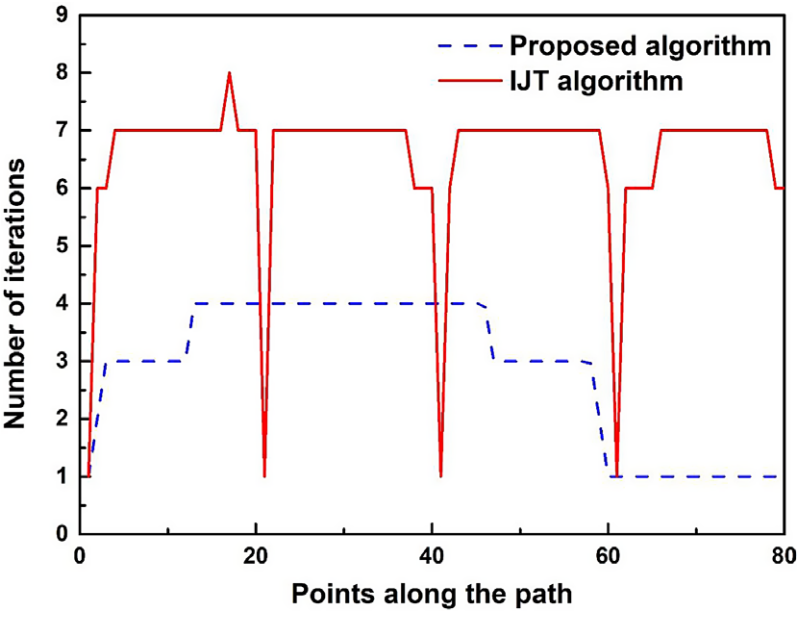

Figure 19. *The number of iterations for the two algorithms to track the same rectangular trajectory.*

4. Conclusion

This paper presents an improved IK solution for the 6-DOF robot manipulator with an offset wrist. By translating certain coordinate frames depending on the structure of the robot manipulator, the analytical solution of the simplified reconstructed structure is presented for the IK. This solution is served as an approximate solution for the actual offset structure, and the exact solution satisfying the convergence accuracy can be obtained by iteration. Further, a robot manipulator HSR-BR606 with an offset wrist is used as an example to specifically elaborate the mathematical procedure of the algorithm and to carry out the simulation calculation of the algorithm. Simulation experiments are carried out to investigate the performance of the algorithm in terms of accuracy, efficiency, and motion planning. A comparative experiment with the IJT algorithm is also conducted. The results show that the proposed algorithm has the advantages of high accuracy with short calculation time compared to the IJT algorithm. The proposed method is simple and efficient, which facilitates the real-time performance and as well as reduces the computational burden. It is believed that this method can be extended to other kinds of robot manipulators with offset structures.

Acknowledgments. The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors acknowledge the financial support provided by the Key Project of Foshan Kernel Technology (Project No. 1920001001367) and the Science and Technology Project of Guangdong Province (Project No. 2019B090919001).

References

[1] A. Aristidou and J. Lasenby, Inverse kinematics: A review of existing techniques and introduction of a new fast iterative solver, Technical Report, 2009.
[2] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. (Pearson Education, London, 2005).
[3] D. L. Pieper, *The Kinematics of Manipulators Under Computer Control* (Department of Mechanical Engineering, Stanford University, Stanford, 1969).
[4] R. P. Paul and C. N. Stevenson, "Kinematics of robot wrists," *Int. J. Rob. Res.* **2**(1), 31–38 (1983).
[5] S. KuCuk and Z. Bingul, "The Inverse Kinematics Solutions of Industrial Robot Manipulators," *Proceedings of the IEEE International Conference on Mechatronics, 2004*. ICM'04 (IEEE, 2004) pp. 274–279.

https://doi.org/10.1017/S0263574721001648 Published online by Cambridge University Press

[6] S. Kucuk and Z. Bingul, "Inverse kinematics solutions for industrial robot manipulators with offset wrists," *Appl. Math. Model.* **38**(7–8), 1983–1999 (2014).
[7] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero and J. Perez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access* **5**(1), 26754–26773 (2017).
[8] T.-M. Wang, Y. Tao and H. Liu, "Current researches and future development trend of intelligent robot: A review," *Int. J. Autom. Comput.* **15**(1), 525–546 (2018).
[9] A. Hentout, M. Aouache, A. Maoudj and I. Akli, "Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017," *Adv. Rob.* **33**(15–16), 764–799 (2019).
[10] C. Trinh, D. Zlatanov, M. Zoppi and R. Molfino, "A Geometrical Approach to the Inverse Kinematics of 6r Serial Robots with Offset Wrists," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (American Society of Mechanical Engineers, 2015) pp. V05CT08A016.
[11] Y. Wei, S. Jian, S. He and Z. Wang, "General approach for inverse kinematics of nR robots," *Mech. Mach. Theory* **75**(1), 97–106 (2014).
[12] J. Angeles, "On the numerical solution of the inverse kinematic problem," *Int. J. Rob. Res.* **4**(2), 21–37 (1985).
[13] A. Balestrino, G. De Maria and L. Sciavicco, "Robust control of robotic manipulators," *IFAC Proc. Vol.* **17**(2), 2435–2440 (1984).
[14] A. A. Goldenberg, J. A. Apkarian and H. W. Smith, "A new approach to kinematic control of robot manipulators," *J. Dyn. Syst. Meas. Control* **109**(2), 97–103 (1987).
[15] S. Chiaverini and B. Siciliano, "Review of damped least squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Trans.* Control Syst. *Technol.* **2**(2),123–134 (1994).
[16] Y. Feng, W. Yao-nan and Y. Yi-min, "Inverse kinematics solution for robot manipulator based on neural network under joint subspace," *Int. J. Comput. Commun. Control* **7**(3), 459–472 (2014).
[17] J. Xu, K. Song, Y. He, Z. Dong and Y. Yan, "Inverse kinematics for 6-DOF serial manipulators with offset or reduced wrists via a hierarchical iterative algorithm," *IEEE Access* **6**(1), 52899–52910 (2018).
[18] R. Köker, C. Öz, T. Çakar and H. Ekiz, "A study of neural network based inverse kinematics solution for a three-joint robot," *Rob. Auton. Syst.* **49**(3–4), 227–234 (2004).
[19] P. Karlra and N. R. Prakash, "A Neuro-Genetic Algorithm Approach for Solving the Inverse Kinematics of Robotic Manipulators," *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)* (IEEE, 2003) pp. 1979–1984.
[20] P. Kalra, P. Mahapatra and D. Aggarwal, "An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots," *Mech. Mach. Theory* **41**(10), 1213–1229 (2006).
[21] H. N. Ghafil and K. Jármai, "Optimization Algorithms for Inverse Kinematics of Robots with MATLAB Source Code," In: *Vehicle and Automotive Engineering* (Springer, 2020) pp. 468–477.
[22] M. Toz, "Chaos-based Vortex Search algorithm for solving inverse kinematics problem of serial robot manipulators with offset wrist," *Appl. Soft Comput.* **89**(1), 106074 (2020).
[23] Q. Shi and J. Xie, "A Research on Inverse Kinematics Solution of 6-DOF Robot with Offset-Wrist Based on Adaboost Neural Network," 2017 IEEE *International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)* (IEEE, 2017) pp. 370–375.
[24] J. Li, H. Yu, N. Shen, Z. Zhong, Y. Lu and J. Fan, "A novel inverse kinematics method for 6-DOF robots with non-spherical wrist," *Mech. Mach. Theory* **157**(1), 104180 (2021).
[25] B. Siciliano and O. Khatib, *Springer Handbook of Robotics* (Springer, Switzerland, 2016).
[26] L. Wu, X. Yang, D. Miao, Y. Xie and K. Chen, "Inverse Kinematics of a Class of 7R 6-DOF Robots with Non-Spherical Wrist," *2013 IEEE International Conference on Mechatronics and Automation* (IEEE, 2013) pp. 69–74.
[27] O. Bottema and B. Roth, *Theoretical Kinematics* (Courier Corporation, North Chelmsford, 1990).
[28] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in Matlab* (Springer, Switzerland, 2011).

Cite this article: X. Zhou, Y. Xian, Y. Chen, T. Chen, L. Yang, S. Chen and J. Huang (2022). "An improved inverse kinematics solution for 6-DOF robot manipulators with offset wrists", Robotica **40**, 2275–2294. https://doi.org/10.1017/S0263574721001648