


RESEARCH ARTICLE

A novel inverse kinematics for solving repetitive motion planning of 7-DOF SRS manipulator

Jingdong Zhao, Zichun Xu , Liangliang Zhao*, Yuntao Li, Liyan Ma and Hong Liu

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, Heilongjiang, China

*Corresponding author. E-mail: zhaoliangliang0619@126.com

Received: 20 April 2022; Revised: 19 July 2022; Accepted: 5 September 2022; First published online: 6 October 2022

Keywords: inverse kinematics, redundant manipulators, joint angle drift, repetitive motion planning

Abstract

The repetitive motion planning movements of the redundant manipulator will cause oscillations and unintended swings of joints, which increase the risk of collisions between the manipulator and its surroundings. Motivated by this phenomenon, this paper presents an inverse kinematics algorithm for the spherical-revolute-spherical manipulator to solve the paradox raised by joint-drift and control the pose with no swing of the elbow. This algorithm takes the joint Cartesian positions set as the intermediary and divides the inverse solution process into two mapping processes within joint limits. Simulations are executed to evaluate this algorithm, and the results show this algorithm is applicable to repetitive motion planning and is capable of producing superior configurations based on its real-time ability and stable solve rate. Experiments using the 7-degree-of-freedom spherical-revolute-spherical manipulator demonstrate the effectiveness of this algorithm to remedy the joint-drift and elbow swing compared to Kinematics and Dynamics Library and TRAC-IK.

1. Introduction

Redundant manipulators that process more degrees-of-freedom (DOFs) than required are attracting increasing attention with their significant advantages in terms of obstacle avoidance [1] and dynamic scenarios [2]. With the feature of kinematic redundancy, redundant manipulators have been widely used in various related fields [3, 4]. The forward kinematics function of redundant manipulators can be expressed as

$$\mathbf{r}(t) = f(\Theta(t)), \quad (1)$$

where $\mathbf{r}(t) \in \mathbb{R}^m$ and $\Theta(t) \in \mathbb{R}^n$. $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ implements the transformation from n -dimensional configuration space (C-space) to m -dimensional task space. Conversely, $f(\cdot)^{-1}$, which is difficult to deduce, denotes the inverse kinematics (IK) of redundant manipulators. Generally, the IK algorithm for redundant manipulators is designed to solve the kinematics redundancy and to satisfy the given secondary task. For repetitive motion planning (RMP) tasks, where the end-effector (EE) repetitively follows the given closed path in Cartesian space or C-space, we first want the manipulator to return to the initial position both in C-space and Cartesian space after each cyclic task, that is, joint-drift-free (JDF) motion. Klein and Huang [5] first explain the joint-drift phenomenon (JDP) that occurred during the RMP, which means a closed path in the task space rather than in C-space is generated due to the non-integrability of the Pfaffian constraint. JDP may cause damage to manipulators [6] and even some unpredictable problems [7], such as joint failure or joint lock. Meanwhile, readjusting configurations will reduce efficiency [8]. In addition, we also want to eliminate needless movements of some joints during repetitive tip rolling, which may cause collisions between the other joints and surroundings. Thus, in addition to the basic criteria of real-time and solving stability, an IK algorithm suitable for the RMP task is indispensable.

Although numerous analyses have been performed in the past decades, there is no universal IK method with all advantages. Analytical methods, numerical methods, and intelligent methods are the three main types of IK algorithms currently available. Some of the most widely used numerical approaches are Jacobian-based methods

$$\dot{\mathbf{r}}(t) = J(\Theta)_{m \times n} \dot{\Theta}(t), \tag{2}$$

in which $\dot{\mathbf{r}}(t)$ and $\dot{\Theta}(t)$ are the velocities of *EE* and joints, respectively. $J(\Theta)_{m \times n} = \partial f(\Theta(t)) / \partial \Theta$ is the Jacobian matrix. For redundant manipulators, Jacobian transpose [9], Jacobian pseudo-inverse [10], damped least squares [11], and singular value decomposition [12] are developed. Equation (2) can be justified with the pseudo-inverse method, which is given by

$$\dot{\Theta}(t) = J^\dagger(\Theta(t))_{n \times m} \dot{\mathbf{r}}(t), \tag{3}$$

where $J^\dagger(\Theta(t))_{n \times m}$ is the *Moore-Penrose generalized inverse* of $J(\Theta)_{m \times n}$, and $J^\dagger(\Theta(t))_{n \times m} = J^T(JJ^T)^{-1}$ when $J(\Theta)_{m \times n}$ has full row rank r . The generic Jacobian pseudo-inverse IK solver, Orocos Kinematics and Dynamics Library (KDL), is realized based on Eq. (3), whose solve rate is not satisfactory due to joint limits. Thus, TRACLAB built the TRAC-IK solver based on KDL with random restarts and nonlinear optimization to improve the computation performance and scalability [13]. However, the traditional pseudo-inverse algorithms will lead to abrupt changes in joint angles with small movements of *EE* when approaching singularities. Meanwhile, the manipulability [14]

$$w = \sqrt{\det J(\Theta)_{m \times n} J^T(\Theta)_{n \times m}} \tag{4}$$

will drop significantly, which is one of the most commonly used indexes to measure the superiority of configurations. Moreover, during the motion planning, the neighboring pose should be taken as the initial seed for the next solution, which means the JDF motion and continuity of configurations cannot be realized at the same time using the traditional pseudo-inverse algorithms. Due to the lack of internal pose control, JDP and elbow swing of the 7-DOF manipulator are then caused. Abbasi *et al.* [15] solve this problem by using an augmented Jacobian method to maintain the arm pose. In addition, using $(I - J^\dagger J)$ to project onto the nullspace of $J(\Theta)_{m \times n}$, Eq. (3) can be reformulated as

$$\dot{\Theta}(t) = J^\dagger \dot{\mathbf{r}}(t) + (I - J^\dagger J) \varphi, \tag{5}$$

and appropriate selection of $\varphi \in \mathbb{R}^n$ can accomplish some tasks with special demands. Some extensions [16–18] are proposed based on Eqs. (2)–(5) to realize JDF motion. They are, however, accompanied by the time cost consumed by the tuning process. The Cyclic Coordinate Descent method [19], which is a sub-category of heuristic methods [20], is realized by iteratively updating each joint angle and may yield unnatural configurations with discontinuities in joint trajectories and oscillations that are not suitable for redundant manipulators.

The analytical methods [21, 22] use the geometrical relationships between different links to deduce the parametric expression of different joint angles. These methods can deduce more than one IK solution and usually do not suffer from the problem of singularities relative to numerical methods. Analytical methods can be applied with some additional tasks [23]. However, it is hard to obtain analytical solutions and the computation efficiency drops significantly when applied to redundant manipulators; for example, the IK solver in [24] takes 50 – 70 ms to compute one IK solution for Baxter. The backbone curve methods [25] are suitable for redundant manipulators, nevertheless, accompanied by the sacrifices of real-time action and *EE* orientation.

The neural network methods [26], which have the advantages of generality and low dependence on kinematics function models [27, 28], are the most important components of intelligent methods. Moreover, neural networks can be the solvers of optimization issues to reduce the complexity of the IK problem and achieve some secondary tasks [29]. For instance, Xie *et al.* [30] use the gradient descent method to construct the dynamic neural network with velocity compensation to complete high-precision RMP tasks. Li *et al.* [31] formulate the sparse optimization problem to solve the redundancy resolution with the consideration of joint failure. Nevertheless, pre-training the network and constructing the training set have a significant impact on prediction accuracy and take unpredictable long. The tuning process

remains a challenge for these methods. In addition, some hybrid methods have been utilized to solve the IK of manipulators. The combination of genetic algorithm and particle swarm [32] is impressive with its 100% solve rate, but with a higher computation cost relative to TRAC-IK. Further research in ref. [33] extends the algorithm in ref. [32] with a variety of constraints and improves its scalability and computation efficiency.

In reference to previous works, our aim is to design an IK algorithm for the spherical-revolute-spherical (SRS) manipulator that solves the RMP issues with low computational cost and complexity while also satisfying the high solve rate and real-time applications. The main contributions of this paper are the following:

- (1) The proposed algorithm remedies the JDP without a complex tuning process and enables internal pose control, that is, the elimination of elbow swing.
- (2) Some classifications are considered to boost the second mapping process, in which the nonlinear constrained optimization (NLOpt) problems are formulated to reduce the calculation complexity. This algorithm is suitable for real-time applications and provides a high solve rate.
- (3) This algorithm can deduce superior solutions with higher manipulability while inducing only *EE* position error without pose error.
- (4) Comparative simulations and experiments are performed on the KUKA LBR iiwa 14 R820 manipulator to demonstrate the effectiveness and advantages of this algorithm in RMP and internal pose control.

The remainder of this paper is organized as follows: Section 2 explains the two mapping processes of the proposed algorithm, in which the theoretical analyses are presented to demonstrate the advantages. Section 3 first presents the quantitative test results of the proposed algorithm and then performs the simulations to demonstrate its effectiveness in RMP and the ability to deduce superior solutions with higher manipulability. To verify simulation results, the experiments of RMP are performed with the KUKA LBR iiwa 14 R820 in Section 4. Finally, Section 5 gives the conclusions of this article and presents future work.

2. Algorithm

In this section, the proposed algorithm and relevant theoretical analyses are explained in detail to solve the IK of the SRS manipulator and remedy the problems encountered in RMP. Some important notations used in the proposed algorithm are listed in Table I, and the abbreviations frequently used in this paper are shown in Table II.

2.1. Mapping from task space to joint positions set

To prevent the JDP and algorithmic singularity induced by the Jacobian-based algorithm, the IK problem is solved by two mapping processes instead of directly calculating the joint angles. For the first mapping $h(\cdot) : \mathbb{R}^m \rightarrow \mathcal{S}_p$, the Cartesian position of the most important elbow joint (i.e., \mathbf{P}_3) can be derived by spatial iteration with the consideration of link length and joint limits. As shown in Fig. 1, \mathbf{P}_1 and \mathbf{P}_5 can be excluded from the iteration due to the fixed base and desired *EE* pose. Thus, the determined target (i.e., \mathbf{P}_{nt}) is for \mathbf{P}_4 , and the iteration process is given by the following two phases:

$$\mathbf{S}_{1p} \triangleq \begin{cases} {}^i\mathbf{P}'_4 = \mathbf{P}_{nt} = \mathbf{P}_{des} - R_{des} [{}^n_{n-1}x, {}^n_{n-1}y, {}^n_{n-1}z]^T \quad (n = 7) \\ {}^i\mathbf{P}'_3 = {}^i\mathbf{P}'_4 + \widehat{{}^i\mathbf{P}'_4{}^{i-1}\mathbf{P}_3} \cdot l_3, \theta_6^{\min} \leq \theta_6 \leq \theta_6^{\max} \text{ or } {}^i\mathbf{P}'_3 = {}^i\mathbf{P}'_4 + R_{u_1}(\Delta\varphi_6) \widehat{{}^i\mathbf{P}'_4{}^{i-1}\mathbf{P}_3} \cdot l_3 \\ {}^i\mathbf{P}'_2 = {}^i\mathbf{P}'_3 + \widehat{{}^i\mathbf{P}'_3{}^{i-1}\mathbf{P}_2} \cdot l_2, \theta_4^{\min} \leq \theta_4 \leq \theta_4^{\max} \text{ or } {}^i\mathbf{P}'_2 = {}^i\mathbf{P}'_3 + R_{u_2}(\Delta\varphi_4) \widehat{{}^i\mathbf{P}'_3{}^{i-1}\mathbf{P}_2} \cdot l_2, \end{cases} \quad (6)$$

in which $\mathbf{u}_1 = -\widehat{l}_4 \times \widehat{{}^i\mathbf{P}'_4{}^{i-1}\mathbf{P}_3}$ and $\mathbf{u}_2 = \widehat{{}^i\mathbf{P}'_4{}^{i-1}\mathbf{P}_3} \times \widehat{{}^i\mathbf{P}'_3{}^{i-1}\mathbf{P}_2}$, and

Table I. Mathematical notations.

Notation	Meaning
S_{1p}, S_{2p}	Sets of joint Cartesian positions in different phases
S_p	Set of joint Cartesian positions after iteration
C_{1-7}	Set of all possible IK solutions
E	Subset of C_{1-7} that meets the Cartesian error constraint
P_i	Cartesian position vector of the i -th joint in Fig. 1
\vec{l}_i	Direction vector of the i -th link in Fig. 1
A_n	Transformation matrix of frame Σ_n with respect to frame Σ_{n-1}
$T_{des}, R_{des}, P_{des}$	The desired homogeneous matrix and its rotation matrix and position vector
$\Theta_{init}, \Theta_{des}$	The initial and desired joint angle vectors of the manipulator
q_0	The initial state vector for each iteration of the proposed algorithm
$h(\cdot)$	Mapping from task space to S_p
$g(\cdot)$	Mapping from S_p to Θ_{des}
w, w_{avg}	Manipulability and its average value

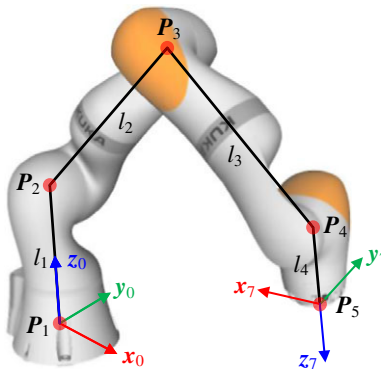


Figure 1. Schematic of the 7-DOF SRS manipulator, where P_1 and P_5 indicate the base and end-effector positions, respectively, and P_i ($i = 2, 3, 4$) indicate the shoulder, elbow, and wrist positions in order.

$$S_{2p} \triangleq \begin{cases} {}^iP'_2 = [0, 0, \widehat{z}_0 \cdot l_1]^T \\ {}^iP'_3 = {}^iP'_2 + \widehat{{}^iP'_2 P'_3} \cdot l_2, \theta_2^{\min} \leq \theta_2 \leq \theta_2^{\max} \text{ or } {}^iP'_3 = {}^iP'_2 + R_{u_3}(\Delta\varphi_2) \widehat{{}^iP'_2 P'_3} \cdot l_2 \\ {}^iP'_4 = {}^iP'_3 + \widehat{{}^iP'_3 P'_4} \cdot l_3, \theta_4^{\min} \leq \theta_4 \leq \theta_4^{\max} \text{ or } {}^iP'_4 = {}^iP'_3 + R_{u_4}(\Delta\varphi_4) \widehat{{}^iP'_3 P'_4} \cdot l_3, \end{cases} \quad (7)$$

in which $u_3 = \widehat{z}_0 \times \widehat{{}^iP'_2 P'_3}$ and $u_4 = \widehat{{}^iP'_2 P'_3} \times \widehat{{}^iP'_3 P'_4}$. When θ_i ($i = 2, 4, 6$) exceed joint limits, $\Delta\varphi_i$ can be given by

$$\Delta\varphi_i = \begin{cases} \theta_i^{\max} - \theta_i & \text{if } \theta_i^{\max} < \theta_i, \\ \theta_i^{\min} - \theta_i & \text{if } \theta_i < \theta_i^{\min}. \end{cases}$$

$R_u(\theta)$ represents the rotation around the axis u by an angle θ , which is given as

$$R_u(\theta) = (\cos \theta) I + (\sin \theta) [u]_{\times} + (1 - \cos \theta)(u \otimes u),$$

where $[u]_{\times}$ is the cross-product matrix of u . ${}^iP'_n$ and ${}^iP''_n$ ($2 \leq n \leq 4$) denote the position of P_n in Fig. 1 after the first and second phases of the i -th iteration, respectively. The superscript $i-1$ indicates the joint positions after the last iteration, or the initial joint positions when $i = 1$. l_n is the length of the n -th link. P_{des} and R_{des} are the position vector and rotation matrix of T_{des} , respectively. ${}^n_{n-1}x$, ${}^n_{n-1}y$, and ${}^n_{n-1}z$ are the

Table II. *Abbreviations.*

Abbreviation	Name
EE	End-effector
JDF	Joint-drift-free
IK	Inverse kinematics
DOF	Degree-of-freedom
C-space	Configuration space
JDP	Joint-drift phenomenon (or phenomena)
RMP	Repetitive motion planning
SRS	Spherical-revolute-spherical
KDL	Orocos Kinematics and Dynamics Library

x , y , and z coordinates of the origin of frame Σ_{n-1} in frame $\Sigma_n (1 \leq n \leq 7)$. \widehat{z}_n is the z -axis unit vector of frame Σ_n , and $\widehat{l}_4 = R_{des} [{}^n_{n-1}x, {}^n_{n-1}y, {}^n_{n-1}z]^T (n = 7)$ is the desired direction vector of the end link.

Due to the determined direction vector of the end link and the fixed base, ${}^iP'_4$ and ${}^iP''_2$ should be corrected at the beginning of each phase. Equations (6)–(7) are cycled through until the distance between ${}^iP'_4$ and P_m meets the Cartesian error requirement and the angles between adjacent links are within joint limits. The *EE* position is subsequently deduced by $P_5 = {}^iP'_4 + \widehat{l}_4 \cdot l_4$. Set $S_P \triangleq \{P_3, P_4 \in S_{2p}, P_1, P_2, P_5\}$, containing all joint positions, is obtained after all iterations are completed. The advantages of the proposed algorithm are presented below, including theoretical analyses demonstrating its applicability to RMP.

(1) JDF motion: When there are no external disturbances, each iteration phase and the final configuration are predetermined before the first phase with the given initial state vector $q_0 \in \mathbb{R}^7$, P_m , and the Cartesian error. The initial joint positions (i.e., ${}^0P_3, {}^0P_4$) for the iteration of $h(\cdot)$ can be obtained by forward kinematics with q_0 . Once the above variables are determined, Eqs. (6)–(7) can guarantee the uniqueness of $h(\cdot)$, that is, S_P is unique for each IK query.

(2) Continuity of links' configurations $\{\vec{l}_1, \vec{l}_2, \vec{l}_3, \vec{l}_4\}$: For motion planning tasks, the Cartesian path can be interpolated as consecutive path points. Meanwhile, the pose of the end link can be guaranteed because P_5 is not involved in iterations. In addition, since \vec{l}_1 is a constant vector, the key to iteration is to find the elbow position (i.e., P_3). Assuming P_t and $\underline{P}_t = P_t + \overrightarrow{\Delta P}$ are two consecutive path points that are treated as the targets for P_4 , for the iteration of \underline{P}_t , the position change of P_3 in different phases can be summarized as

$$\begin{cases} \underline{{}^iP'_3} = (P_t + \overrightarrow{\Delta p}) + \frac{{}^{i-1}P_3 - (P_t + \overrightarrow{\Delta p})}{\|{}^{i-1}P_3 - (P_t + \overrightarrow{\Delta p})\|_2} l_3 \\ \underline{{}^iP''_3} = \underline{{}^iP''_2} + \frac{{}^iP'_3 - {}^iP''_2}{\|{}^iP'_3 - {}^iP''_2\|_2} l_2, \end{cases} \tag{8}$$

where the underline is used to indicate the iteration of \underline{P}_t . Using Eq. (8) to compare with the iteration process of P_t , the variation can be quantified as

$$\begin{cases} \underline{{}^iP'_3} - {}^iP'_3 = \overrightarrow{\Delta P} + l_3 \left(\frac{{}^{i-1}P_3 - (P_t + \overrightarrow{\Delta P})}{\|{}^{i-1}P_3 - (P_t + \overrightarrow{\Delta P})\|_2} - \frac{{}^{i-1}P_3 - P_t}{\|{}^{i-1}P_3 - P_t\|_2} \right) \\ \underline{{}^iP''_3} - {}^iP''_3 = l_2 \left(\frac{{}^iP'_3 - {}^iP''_2}{\|{}^iP'_3 - {}^iP''_2\|_2} - \frac{{}^iP'_3 - {}^iP''_2}{\|{}^iP'_3 - {}^iP''_2\|_2} \right), \end{cases} \tag{9}$$

where ${}^i\mathbf{P}'_2 = {}^i\mathbf{P}'_2$. Since i starts from 1, the limit of Eq. (9) as $\overrightarrow{\Delta\mathbf{P}}$ approaches $\mathbf{0}$ is given by

$$\begin{cases} \lim_{\overrightarrow{\Delta\mathbf{P}} \rightarrow \mathbf{0}} ({}^i\mathbf{P}'_3 - {}^i\mathbf{P}'_3) = \mathbf{0} \\ \lim_{\overrightarrow{\Delta\mathbf{P}} \rightarrow \mathbf{0}} ({}^i\mathbf{P}''_3 - {}^i\mathbf{P}''_3) = \mathbf{0}. \end{cases} \tag{10}$$

As shown in Eq. (10), the minor movement of \mathbf{P}_4 in Cartesian space does not lead to the abrupt movement of \mathbf{P}_3 with the same \mathbf{q}_0 . Equations (6)–(7) are the theoretical basis for generating smooth trajectories since they can be used to deduce continuous links' configurations $\{\vec{l}_1, \vec{l}_2, \vec{l}_3, \vec{l}_4\}$ for continuous path points.

(3) Guaranteed *EE* pose: \mathbf{P}_5 is excluded from the iteration to improve efficiency and induce only position error with no orientation error of *EE*, which eliminates the effect of repeated rotation of the 7th joint on other links compared to the pseudo-inverse algorithms. When performing repetitive tip rolling motions, such as screwing tasks, the elbow joint does not swing due to the repetitive rotation of the 7th joint, which reduces the risk of collisions and improves the safety of operations.

(4) Superior configurations: Generally, a higher w implies better singularity avoidance and adaptability during execution. In contrast with pseudo-inverse algorithms, Eqs. (6)–(7) can deduce the desired and natural configurations with higher w . Furthermore, the solution process of the proposed algorithm is completed without using $J(\Theta)_{m \times n}$, which compensates for the algorithmic singularity of the pseudo-inverse algorithms.

2.2. Mapping from joint positions set to the desired joint angle vector

An algorithmic solution implementing the mapping $g(\cdot) : \mathcal{S}_p \rightarrow \Theta_{des}$ is given in this subsection, in which $\Theta_{des} \in \mathbb{R}^7$ is the desired IK joint angle vector. Note that the solution uniqueness and continuity derived from $h(\cdot)$ can be retained in the process of $g(\cdot)$. $|\theta_i| (i = 2, 4, 6)$ are first calculated by

$$|\theta_i| = \left| \pi - \arccos \left(\frac{l_j^2 + l_{j+1}^2 - (\|\mathbf{P}_{j+2} - \mathbf{P}_j\|_2)^2}{2l_j l_{j+1}} \right) \right|, \tag{11}$$

where $i = 2j$. $|\theta_i| (i = 2, 4, 6)$ will appear in the denominator of some equations and terminate calculations when one of them approaches or equals to zero. Hence, the realization of $g(\cdot)$ can be classified into the following cases:

(1) θ_2, θ_4 , and θ_6 are not equal to zero: θ_1 is first calculated by

$$(A_1 A_2 A_3 [0, 0, 0, 1]^T)_{1,1} = \mathbf{P}_{3,x}, \tag{12}$$

where the subscripts i, j indicate the value of the i -th row and j -th column of the corresponding matrix, and x indicates the x coordinate of the corresponding joint Cartesian position. A_n is the transformation matrix of frame Σ_n with respect to frame Σ_{n-1} . However, because the sign of θ_2 cannot be determined temporally, the combinations of θ_1 and θ_2 should be filtered with the following equations

$$\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}, \tag{13}$$

$$|1 - \widehat{\mathbf{z}}_i \cdot \widehat{\mathbf{l}}_{i/2+1}| \leq \varepsilon_1 \quad (i = 2, 4, 6), \tag{14}$$

where $\varepsilon_1 = 10^{-3}$ is the defined tolerance. The solution set $\mathbf{C}_{1,2}$ is defined as

$$\mathbf{C}_{1,2} = \left\{ (\theta_1, \theta_2) \mid \text{Eqs. (11) - (14)} \right\}. \tag{15}$$

Given the set $\mathbf{C}_{1,2}$, θ_3 is calculated by

$$(A_3 A_4 A_5 [0, 0, 0, 1]^T)_{1,1} = (A_2^{-1} A_1^{-1} [\mathbf{P}_4, 1]^T)_{1,1}. \tag{16}$$

Thus, the set $C_{3,4}$ is defined as

$$C_{3,4} = \left\{ (\theta_3, \theta_4) \mid \text{Eqs. (11), (13), (14), and (16)} \right\}. \tag{17}$$

Similarly, θ_5 is calculated by

$$(A_5 A_6 A_7 [0, 0, 0, 1]^T)_{1,1} = (A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} T_{des} [0, 0, 0, 1]^T)_{1,1}, \tag{18}$$

which is used to define the set

$$C_{5,6} = \left\{ (\theta_5, \theta_6) \mid \text{Eqs. (11), (13), (14), and (18)} \right\}. \tag{19}$$

Finally, $|\theta_7|$, which is the angle between \widehat{x}_6 and \widehat{x}_7 , is given by

$$|\theta_7| = |\arccos(\widehat{x}_6 \cdot \widehat{x}_7)|, \tag{20}$$

where \widehat{x}_6 and \widehat{x}_7 are the first three rows of the first columns of $\prod_{n=1}^6 A_n$ and T_{des} , respectively. The set C_{1-7} , which includes all possible IK solutions, is expressed as

$$C_{1-7} = \left\{ \Theta \mid \text{Eqs. (15), (17), (19), and (20)} \right\}. \tag{21}$$

(2) $|\theta_2| \neq 0$, $|\theta_4| = 0$, and $|\theta_6| = 0$: $C_{1,2}$ is derived in the same way as in the first case. Due to $|\theta_4| = |\theta_6| = 0$, the analytical expressions for the remaining joint angles are difficult to obtain. Thus, the remaining solution processes can be formulated as the NLopt problem, which is given by

$$\begin{aligned} & \min \left[(\theta_3 - \Theta_{init,3})^2 + (\theta_5 - \Theta_{init,5})^2 + (\theta_7 - \Theta_{init,7})^2 \right] \\ & \text{s.t. } |\theta_3 + \theta_5 + \theta_7| = \left| \arccos\left((A_2^{-1} A_1^{-1} T_{des})_{1,1} \right) \right| \text{ and Eq. (13),} \end{aligned} \tag{22}$$

in which $\Theta_{init,i}$ denotes the i -th value of the initial joint angle vector of the manipulator (i.e., Θ_{init}). The NLopt problem can be solved by the Sequential Least Squares Quadratic Programming algorithm. In this case, C_{1-7} is defined as $C_{1-7} = \{\Theta \mid \text{Eqs. (15), (22), and } \theta_4 = \theta_6 = 0\}$.

(3) $|\theta_2| = 0$, $|\theta_4| \neq 0$, and $|\theta_6| = 0$: The solutions to $|\theta_1 + \theta_3|$ and $|\theta_5 + \theta_7|$ are formulated as

$$\begin{aligned} & \min \left[(\theta_1 - \Theta_{init,1})^2 + (\theta_3 - \Theta_{init,3})^2 \right] \\ & \text{s.t. } |\theta_1 + \theta_3| = \left| \arccos\left(\frac{T_{des,1,3}}{s_4} \right) \right| \text{ and Eq. (13),} \end{aligned} \tag{23}$$

where s_i is the abbreviation of $\sin \theta_i$, and

$$\begin{aligned} & \min \left[(\theta_5 - \Theta_{init,5})^2 + (\theta_7 - \Theta_{init,7})^2 \right] \\ & \text{s.t. } |\theta_5 + \theta_7| = \left| \arccos\left(\frac{T_{des,3,1}}{-s_4} \right) \right| \text{ and Eq. (13).} \end{aligned} \tag{24}$$

Equation (14) is utilized to eliminate some wrong combinations of $(\theta_1 + \theta_3)$ and θ_4 . C_{1-7} is thus given by $C_{1-7} = \{\Theta \mid \text{Eqs. (11), (14), (23), (24), and } \theta_4 = \theta_6 = 0\}$.

(4) $|\theta_2| = 0$, $|\theta_4| = 0$, and $|\theta_6| \neq 0$: Using Eq. (11), $|\theta_7|$ can be calculated by

$$|\theta_7| = \left| \arccos\left(\frac{T_{des,3,1}}{-s_6} \right) \right|. \tag{25}$$

Then, $C_{6,7}$ is defined as

$$C_{6,7} = \left\{ (\theta_6, \theta_7) \mid \text{Eqs. (11), (25), and } |\widehat{z}_5 \cdot \widehat{l}_4| \leq \varepsilon_1 \right\}. \tag{26}$$

The solution to $\theta_1, \theta_3,$ and θ_5 is formulated as

$$\begin{aligned} &\min \left[(\theta_1 - \Theta_{init,1})^2 + (\theta_3 - \Theta_{init,3})^2 + (\theta_5 - \Theta_{init,5})^2 \right] \\ &\text{s.t. } |\theta_1 + \theta_3 + \theta_5| = \left| \arccos \left((T_{des} A_7^{-1} A_6^{-1})_{1,1} \right) \right| \text{ and Eq. (13).} \end{aligned} \tag{27}$$

Then, we can obtain $C_{1-7} = \{\Theta \mid \text{Eqs. (26), (27), and } \theta_2 = \theta_4 = 0\}$.

(5) $|\theta_2| \neq 0, |\theta_4| \neq 0,$ and $|\theta_6| = 0$: The procedures in the first case are repeated until the sets $C_{1,2}$ and $C_{3,4}$ are obtained. Then, the solution to θ_5 and θ_7 can be written as

$$\begin{aligned} &\min \left[(\theta_5 - \Theta_{init,5})^2 + (\theta_7 - \Theta_{init,7})^2 \right] \\ &\text{s.t. } |\theta_5 + \theta_7| = \left| \arccos \left((A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} T_{des})_{1,1} \right) \right| \text{ and Eq. (13).} \end{aligned} \tag{28}$$

Similarly, C_{1-7} is thus defined as $C_{1-7} = \{\Theta \mid \text{Eqs. (15), (17), (28), and } \theta_6 = 0\}$.

(6) $|\theta_2| = 0, |\theta_4| \neq 0,$ and $|\theta_6| \neq 0$: Using Eq. (11) and T_{des} , $|\theta_5|$ should be calculated first by

$$|\theta_5| = \left| \arccos \left(\frac{c_4 c_6 - T_{des,3,3}}{s_4 s_6} \right) \right|, \tag{29}$$

where c_i is the abbreviation of $\cos \theta_i$. Given Eq. (29), $C_{5,6}$ is defined as

$$C_{5,6} = \left\{ (\theta_5, \theta_6) \mid \text{Eqs. (11), (13), and (29)} \right\}. \tag{30}$$

Then, θ_7 is obtained by solving

$$s_4 s_5 s_7 - (c_4 s_6 + c_5 c_6 s_4) c_7 = T_{des,3,1} \tag{31}$$

with the constraints of Eqs. (14), (30), and $|1 + \widehat{y}_3 \cdot \widehat{l}_2| \leq \varepsilon_1$. After deriving the set

$$C_{4,5,6,7} = \left\{ (\theta_4, \theta_5, \theta_6, \theta_7) \mid \text{Eqs. (11), (13), (30), and (31)} \right\}, \tag{32}$$

the solution to θ_1 and θ_3 is formulated as

$$\begin{aligned} &\min \left[(\theta_1 - \Theta_{init,1})^2 + (\theta_3 - \Theta_{init,3})^2 \right] \\ &\text{s.t. } |\theta_1 + \theta_3| = \left| \arccos \left((T_{des} A_7^{-1} A_6^{-1} A_5^{-1} A_4^{-1})_{1,1} \right) \right| \text{ and Eq. (13).} \end{aligned} \tag{33}$$

Finally, C_{1-7} is defined as $C_{1-7} = \{\Theta \mid \text{Eqs. (32), (33), and } \theta_2 = 0\}$.

(7) $|\theta_2| \neq 0, |\theta_4| = 0,$ and $|\theta_6| \neq 0$: Given Eq. (11), θ_7 can be calculated first with

$$|\theta_7| = \left| \arccos \left(\frac{(A_2^{-1} A_1^{-1} T_{des})_{3,1}}{-s_6} \right) \right| \tag{34}$$

and $C_{1,2}$, which is similarly defined as in the first case. Then, $C_{6,7}$ is defined as

$$C_{6,7} = \left\{ (\theta_6, \theta_7) \mid \text{Eqs. (11), (34), and } |\widehat{z}_5 \cdot \widehat{l}_4| \leq \varepsilon_1 \right\}. \tag{35}$$

Thus, the solution to θ_3 and θ_5 can be formulated as

$$\begin{aligned} &\min \left[(\theta_3 - \Theta_{init,3})^2 + (\theta_5 - \Theta_{init,5})^2 \right] \\ &\text{s.t. } |\theta_3 + \theta_5| = \left| \arccos \left((A_2^{-1} A_1^{-1} T_{des} A_7^{-1} A_6^{-1})_{1,1} \right) \right| \text{ and Eq. (13).} \end{aligned} \tag{36}$$

C_{1-7} is then defined as $C_{1-7} = \{\Theta \mid \text{Eqs. (15), (35), (36), and } \theta_4 = 0\}$.

(8) $|\theta_2| = 0, |\theta_4| = 0,$ and $|\theta_6| = 0$: The solution to the remaining joint angles can be formulated as

$$\begin{aligned} &\min \left[(\theta_1 - \Theta_{init,1})^2 + (\theta_3 - \Theta_{init,3})^2 + (\theta_5 - \Theta_{init,5})^2 + (\theta_7 - \Theta_{init,7})^2 \right] \\ &\text{s.t. } |\theta_1 + \theta_3 + \theta_5 + \theta_7| = \left| \arccos (T_{des,1,1}) \right| \text{ and Eq. (13).} \end{aligned} \tag{37}$$

Hence, C_{1-7} is given as $C_{1-7} = \{\Theta \mid \theta_2 = \theta_4 = \theta_6 = 0 \text{ and Eq. (37)}\}$.

No matter which case is performed, C_{1-7} will eventually be defined to obtain Θ_{des} . Traversing C_{1-7} with

$$T_{temp}(\Theta) = A_1 A_2 \cdots A_7 = \begin{bmatrix} R_{temp} & P_{temp} \\ 0 & 1 \end{bmatrix}, \tag{38}$$

each $T_{temp}(\Theta)$ can be used to compare with T_{des} to derive the set $E \subseteq C_{1-7}$, which is defined as

$$E = \left\{ \Theta : (\varepsilon_{rot} + \varepsilon_{pos}) \leq \varepsilon_{tol}, \Theta \in C_{1-7} \right\}. \tag{39}$$

The rotation error (i.e., ε_{rot}) and position error (i.e., ε_{pos}) can be calculated by

$$\varepsilon_{rot} = \arccos\left(\frac{\text{tr}(R_{temp}^{-1}R_{des})-1}{2}\right) \text{ and } \varepsilon_{pos} = \text{dist}(P_{temp}, P_{des}) = \|P_{temp} - P_{des}\|_2, \tag{40}$$

respectively. Meanwhile, Θ_{des} can be found by

$$\Theta_{des} = \arg \min_{\Theta \in E} \|\Theta - \Theta_{init}\|_1. \tag{41}$$

Some joint angle elements in Θ and Θ_{init} can be omitted to reduce the variation in other joints and obtain other desired solutions. Although $|\theta_i| (i = 2, 4, 6)$ are rounded to a given number of places, C_{1-7} in the first case has the most elements, and the multi-threaded solvers are required to boost the computational process to obtain Θ_{des} . For other cases, multithreading may reduce the efficiency of computation. The procedure of the proposed algorithm is outlined in Algorithm 1, where MultiThread is programmed to realize Eqs. (38)–(40) with multithreading, and MultiCase implements the solutions to Cases 2–8.

Algorithm : Solving process of the proposed algorithm

Input: $q_0, \Theta_{init}, T_{des}, \varepsilon_1 = 10^{-3}$, and ε_{tol} ;

Output: Θ_{des} ;

- 1: $P_{nt} \leftarrow$ determined target for iteration
 - 2: **if** The target is reachable **then**
 - 3: **while** $\|{}^iP''_4 - P_{nt}\|_2 > \varepsilon_{tol}$ **do**
 - 4: Equations (6)–(7)
 - 5: **if** $\|{}^iP''_4 - P_{nt}\|_2 \leq \varepsilon_{tol}$ **then**
 - 6: $S_P \leftarrow$ Eq. (7)
 - 7: **end if**
 - 8: **end while**
 - 9: $|\theta_i| (i = 2, 4, 6) \leftarrow$ Eq. (11)
 - 10: **if** $\forall |\theta_i| \neq 0$ **then**
 - 11: $C_{1-7} \leftarrow$ Eq. (21)
 - 12: $E \leftarrow$ MultiThread(C_{1-7})
 - 13: **else if** $\exists |\theta_i| = 0$ **then**
 - 14: $C_{1-7} \leftarrow$ MultiCase($\theta_2, \theta_4, \theta_6$)
 - 15: **for each** $\Theta \in C_{1-7}$ **do**
 - 16: **if** Eq. (40) is satisfied **then**
 - 17: PushBack(Θ, E)
 - 18: **end if**
 - 19: **end for**
 - 20: **end if**
 - 21: $\Theta_{des} \leftarrow$ Eq. (41)
 - 22: **end if**
-

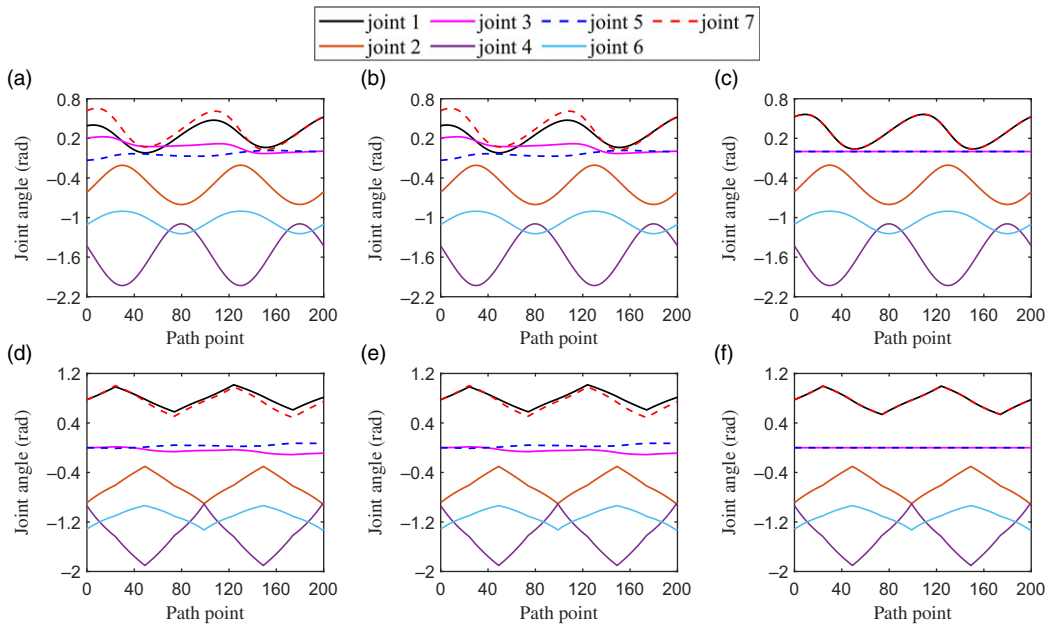


Figure 2. The joint trajectories synthesized by (a), (d) KDL, (b), (e) TRAC-IK, and (c), (f) the proposed algorithm, respectively, when the tip of the KUKA manipulator traces the circle (first row) and square paths (second row) twice.

3. Simulation and performance

The performance of the proposed algorithm is quantified in this section by applying it to the KUKA LBR iiwa 14 R820. Meanwhile, KDL and TRAC-IK are adopted as baselines for comparisons. The joint-drift and screwing simulations are performed using the proposed algorithm on the KUKA manipulator. The ability of the proposed algorithm to deduce superior solutions is also demonstrated.

In this section, the initial state for each iteration of the proposed algorithm is $\mathbf{q}_0 = [0, 0, 0, 0, 0, 0, 0]$ rad, that is, ${}^0\mathbf{P}_i = [0, 0, \sum_{n=1}^{i-1} l_n]^T$ ($i = 3, 4$). Before performing the simulations below, the solve rate and average solve time of the proposed algorithm are first verified by a quantitative test with 10,000 random samples, which are constructed by reachable manipulator configurations. When the Cartesian error is 10^{-6} , the results show that the solve rate is 99.93% and the average solve time is 0.59 ms, which satisfy our requirements for stability and real-time ability of the IK algorithm. All simulations in this section are performed with the 10^{-6} Cartesian error constraint and implemented in C++ on a computer with an Intel Core i3-9100 CPU @ 3.6 GHz \times 4 and 15 GB RAM.

3.1. Joint-drift simulations

In this subsection, the joint-drift simulations are performed on the KUKA manipulator using KDL, TRAC-IK, and the proposed algorithm, respectively. The EE of the KUKA manipulator must trace the given circle and square paths repeatedly, where the radius of the circle is 0.14 m and the side length of the square is 0.2 m. Both paths have been interpolated to 100 path points. Figure 2 illustrates the joint trajectories synthesized by the above three algorithms when tracing the given circular (Fig. 3(a)) and square (Fig. 3(c)) paths twice, in which the initial joint angles are $\Theta_{init} = [0.526, -0.609, 0, -1.431, 0, -1.102, 0.526]$ rad and $\Theta_{init} = [0.777, -0.888, 0, -0.936, 0, -1.316, 0.777]$ rad, respectively. As shown in Fig. 2(a), (b), (d), and (e), the drifts of joints 1, 3, 5, and 7 can be clearly observed when using KDL and TRAC-IK. Contrastively, the joint trajectories deduced by the proposed algorithm all

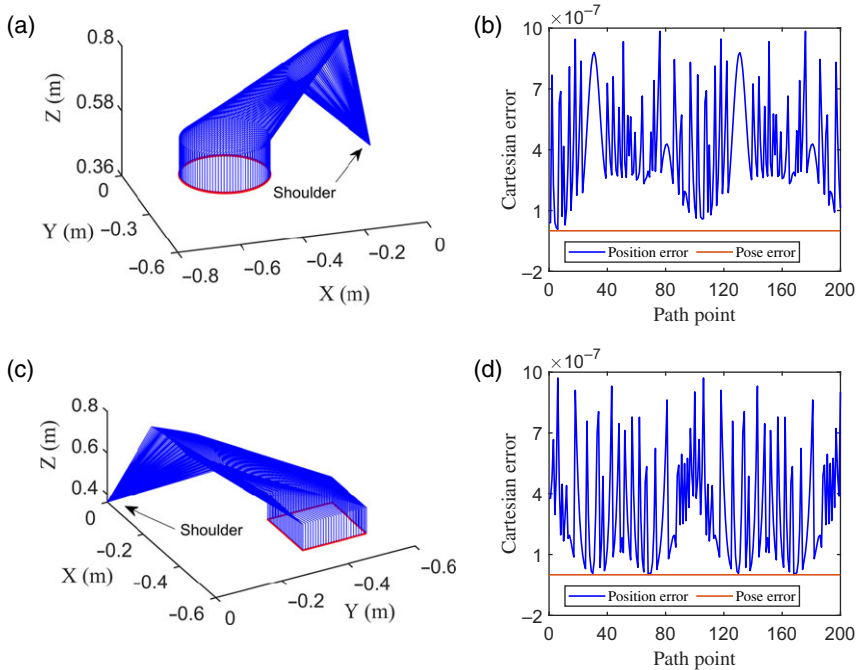


Figure 3. (a), (c) Manipulator motion trajectory profiles and the (b), (d) Cartesian errors generated by the proposed algorithm when tracing the given circular (first row) and square (second row) paths twice.

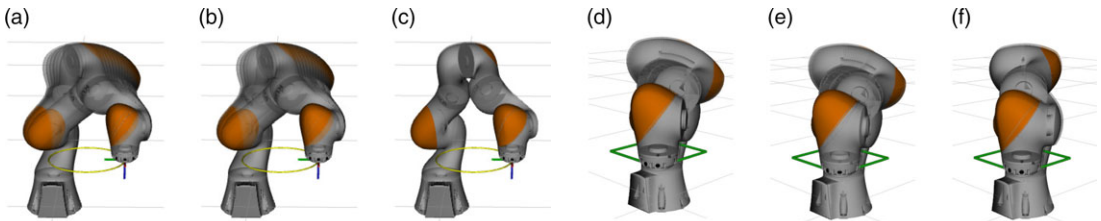


Figure 4. The configurations of the KUKA manipulator synthesized by (a), (d) KDL, (b), (e) TRAC-IK, and (c), (f) the proposed algorithm, respectively, when passing the same point in the 20-cycle circular and square tracing tasks.

return to their initial states in Fig. 2(c) and (f), which indicates the JDF motion is realized. Moreover, Fig. 3(a) and (c) illustrate the tracking performance of the proposed algorithm, in which the *EE* can successfully trace the specified paths (red lines) with good continuity of configurations (blue lines). Using Eq. (40) to generate variation curves of the position error and pose error, it can be observed from Fig. 3(b) and (d) that the maximum value of position error is less than 10^{-6} and the pose error is always zero, both of which meet the error constraint and demonstrate the theoretical analyses in Section 2.1.

As the tip of the KUKA manipulator continues to trace the paths, some of the joint angles drift to their fixed values until the path is closed in C-space when using KDL and TRAC-IK. Figure 4 displays the whole joint-drift process induced by KDL and TRAC-IK, and the joint angle repeatability guaranteed by using the proposed algorithm. Table III lists the detailed simulation data, in which the joint-drift ($\|\Theta(20) - \Theta(0)\|_2$) synthesized by the proposed algorithm in different tracing tasks is 4.491×10^{-7} and 9.177×10^{-8} rad, which are significantly smaller than those of KDL and TRAC-IK. For the circular and square tracing tasks with 2000 path points, the average solution time of the proposed algorithm is 0.243 and 0.239 ms, respectively. Notably, the JDF motion is realized by the proposed algorithm with only

Table III. Comparisons of joint-drifts among KDL, TRAC-IK, and the proposed algorithm when the tip traces the two different paths twenty times.

Joint-drifts (rad)	KDL		TRAC-IK		Proposed algorithm	
	Circle	Square	Circle	Square	Circle	Square
$ \theta_1(20) - \theta_1(0) $	0.544	0.279	0.544	0.275	0	0
$ \theta_2(20) - \theta_2(0) $	0.172	0.076	0.172	0.074	0	0
$ \theta_3(20) - \theta_3(0) $	0.896	0.656	0.896	0.647	1.490×10^{-8}	7.885×10^{-8}
$ \theta_4(20) - \theta_4(0) $	0	8.900×10^{-5}	0	8.700×10^{-5}	0	0
$ \theta_5(20) - \theta_5(0) $	0.623	0.536	0.623	0.528	4.489×10^{-7}	4.696×10^{-8}
$ \theta_6(20) - \theta_6(0) $	0.126	0.066	0.126	0.064	0	0
$ \theta_7(20) - \theta_7(0) $	0.418	0.245	0.418	0.242	0	0
$\ \Theta(20) - \Theta(0)\ _2$	1.306	0.930	1.306	0.917	4.491×10^{-7}	9.177×10^{-8}

Table IV. The effects of position error on joint-drift when the proposed algorithm is applied to different tracing tasks.

Position error	Task	$\ \Theta(20) - \Theta(0)\ _2$ (rad)
10^{-3}	Circle	4.409×10^{-7}
	Square	0
10^{-4}	Circle	4.409×10^{-7}
	Square	0
10^{-5}	Circle	4.425×10^{-7}
	Square	4.891×10^{-8}
10^{-6}	Circle	4.491×10^{-7}
	Square	9.177×10^{-8}

three parameters, that is, q_0 , P_m , and Cartesian error requirement, which omit the time-consuming tuning process. In summary, the simulation results substantiate the effectiveness of the proposed algorithm when applied to the KUKA manipulator to perform real-time JDF motion.

Finally, since the proposed algorithm does not induce pose error, the effects of position error on joint-drift (i.e., $\|\Theta(20) - \Theta(0)\|_2$) are verified. As shown in Table IV, position error causes a minor change in joint-drift within the computation accuracy. This effect is insignificant when compared to the change in magnitude of the position error.

3.2. Screwing simulation

The 2F-85 ROBOTIQ gripper and KUKA manipulator are used to perform the screwing simulation. Meanwhile, the connector is designed to connect the KUKA flange and the ROBOTIQ gripper, as shown in Fig. 5(a). The initial joint angle vector is $\Theta_{init} = [0.527, -0.609, 0, -1.430, 0, -1.102, -3.0]$ rad. Due to the joint limit of θ_7 , the height of EE needs to be adjusted to reset θ_7 during the screwing process. Hence, the above process is divided into the following stages:

- (1) EE is first planned to finish the motion in which the gripper rotates and drops in height until reaching the limit of θ_7 .
- (2) EE moves up to reset θ_7 and then returns to the height at the end of the first stage.
- (3) Repeat the motion of the first stage.

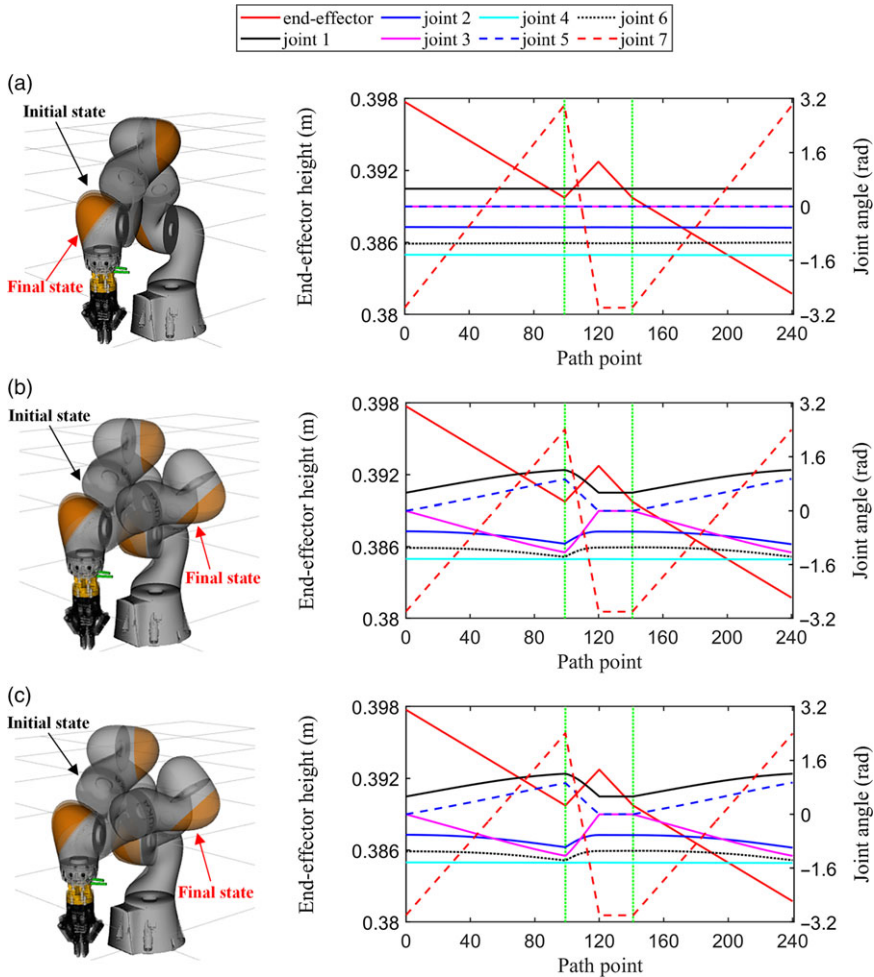


Figure 5. Screwing simulation results using (a) the proposed algorithm, (b) TRAC-IK, and (c) KDL, respectively. For each row, the first column shows the initial and final configurations, and the second column shows the joint trajectories and the variations of EE height relative to the base.

To complete the solution for each path point, the timeout is 15 ms and the running type of TRAC-IK is “Distance,” which means TRAC-IK runs the full timeout and then returns the solution that minimizes the sum of squared error from the seed. As shown in Fig. 5, the green dotted lines indicate the different stages, from which the variations of EE height clearly show the screwing processes. In contrast with TRAC-IK and KDL, the proposed algorithm can rotate the 7th joint significantly as desired and with little variation in other joint angles during the process. Both KDL and TRAC-IK induce an intense swing of the elbow joint, which increases the risk of colliding with nearby objects and people. Moreover, for 664 path points, the average solution time of the proposed algorithm is 0.223 ms.

3.3. Manipulability performance evaluation

In this subsection, the simulation path points in Cartesian space are obtained by calculating the linear path between the target configuration in C-space and the zero position using forward kinematics, where the target configuration is selected randomly in the low manipulability region of the KUKA manipulator. The average values of manipulability w_{avg} are obtained by solving the path with the three algorithms and

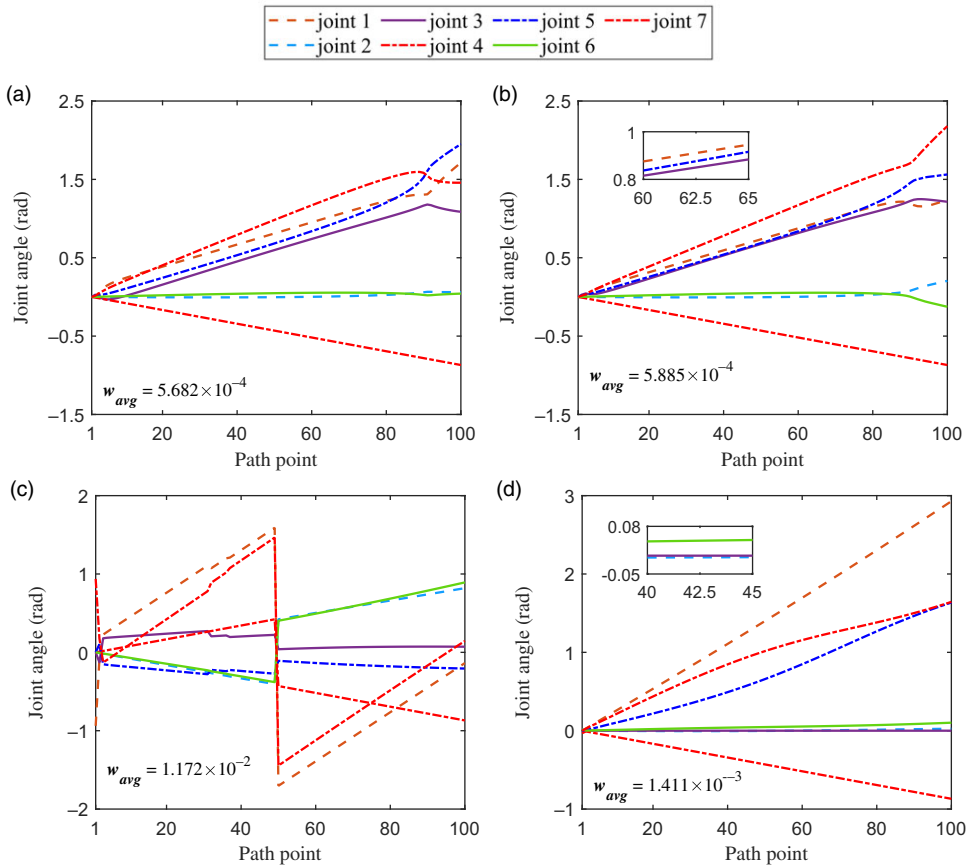


Figure 6. Joint trajectories derived from manipulability simulations using (a) KDL, TRAC-IK with the running types (b) “Distance” and (c) “Manip1”, and (d) the proposed algorithm, respectively.

used as the measure indexes. Moreover, both the running types “Manip1” and “Distance” of TRAC-IK are used to solve the path, in which “Manip1” means TRAC-IK runs the full timeout and then returns the solution that maximizes w . The timeout is 10 ms to guarantee the solution stability of KDL and TRAC-IK. The target configuration is $\Theta_{end} = [0.144, -0.027, 2.805, -0.869, 1.610, 0.112, 1.649]$ rad. As shown in Fig. 6(c), although the max w_{avg} is derived, the oscillatory joint trajectories are generated when using the “Manip1” type of TRAC-IK, which is not applicable to continuous motion. Figure 6(d) indicates that the proposed algorithm can generate smooth trajectories with a higher $w_{avg} = 1.411 \times 10^{-3}$, which is approximately triple that of KDL (Fig. 6(a)) and the “Distance” type of TRAC-IK (Fig. 6(b)). The average solution time of the proposed algorithm is 5.543 ms, and the simulation results substantiate the above statements that the proposed algorithm can produce superior configurations relative to KDL and TRAC-IK.

4. Experiment verification

The experiments in this section are designed to verify the simulation results in Sections 3.1 and 3.2, as well as the practical feasibility of the proposed algorithm. Thus, the relevant procedures of the experiments are basically the same as those of simulations. Notably, the proposed algorithm is implemented on the KUKA LBR iiwa 14 R820 manipulator with the 10^{-6} Cartesian error constraint to solve all path points in real time.



Figure 7. The experiments for tracking the circle (first three rows) and square (last three rows) paths are performed on the KUKA manipulator using (a), (d) KDL, (b), (e) TRAC-IK, and (c), (f) the proposed algorithm, respectively. The KUKA manipulator begins with the initial point (first column), moves counterclockwise through the path points (middle three columns), and finally gets back to the start point (last column).

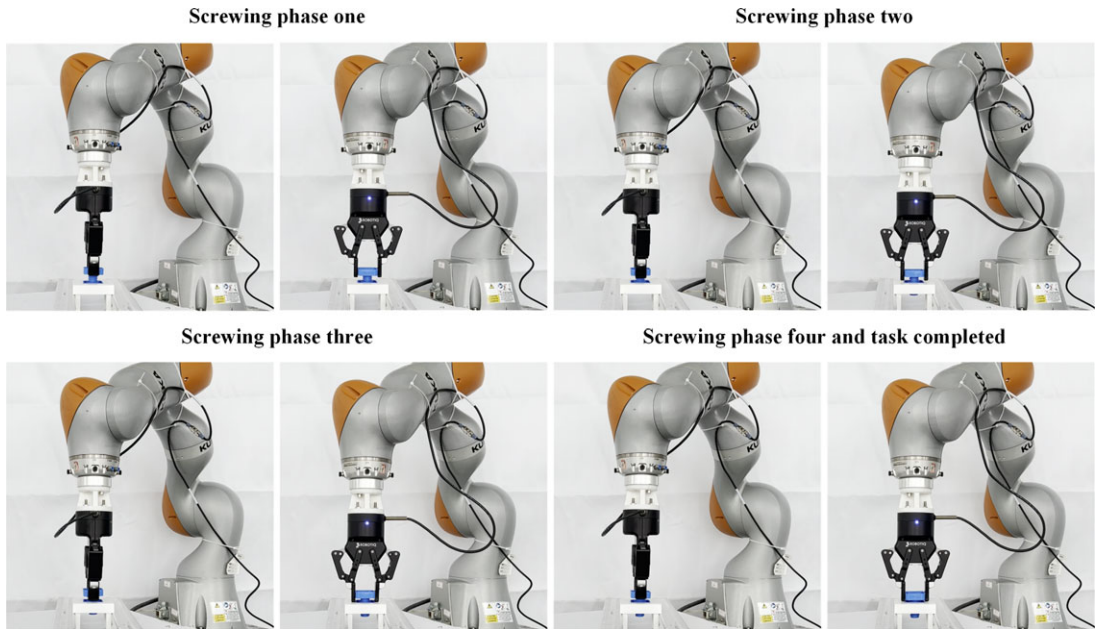


Figure 8. Snapshots of the screwing assembling experiment using the proposed algorithm with the KUKA manipulator and the 2F-85 ROBOTIQ gripper. The 7th joint rotates from -3 rad to 1.714 rad, and the EE moves down during each screwing process, after which the 7th joint must be reset to -3 rad, and the EE moves upwards as required.

4.1. Experiment A: Circular and square path tracking

In order to verify the reliability of joint-drift simulation results, the physical experiments are performed on the KUKA manipulator using the above three baselines. For the circular and square path tracing, the initial joint angles of the KUKA manipulator are $\Theta_{init} = [0.527, -0.609, 0, 1.430, 0, -1.102, 0.527]$ rad and $\Theta_{init} = [0.777, -0.888, 0, 0.936, 0, -1.316, 0.777]$ rad, respectively. The initial state for each iteration of the proposed algorithm is $\mathbf{q}_0 = [0, 0, 0, 0, 0, 0, 0]$ rad. As shown in Fig. 7(c) and (f), the tip of the KUKA manipulator can track the red paths accurately and steadily when using the proposed algorithm. Meanwhile, the continuity of configurations can be guaranteed. After 20-cycle tracking tasks, the initial and final configurations of the KUKA manipulator are identical. On the contrary, as shown in the last columns of Fig. 7(a), (b), (d), and (e), the JDP caused by KDL and TRAC-IK can be clearly observed. Thus, the feasibility of the proposed algorithm to generate JDF motion with guaranteed Cartesian precision is experimentally verified. All the experimental processes can be found in the accompanying video (See [Supplementary materials](#)).

4.2. Experiment B: Screwing assembling

This subsection presents the screwing experiment performed by the KUKA manipulator and 2F-85 ROBOTIQ gripper to validate the simulation results derived from the proposed algorithm. The initial joint angles of the KUKA manipulator are $\Theta_{init} = [0.527, -0.609, 0, 1.430, 0, -1.102, -3.0]$ rad, and the initial state for each iteration of the proposed algorithm is $\mathbf{q}_0 = [0, 0, 0, 0, 0, 0, 0]$ rad. In this experiment, the allowable motion range of the seventh joint is set from -3 to 1.714 rad. As shown in Fig. 8, the screwing experiment is divided into four phases. When the seventh joint reaches 1.714 rad in the second and fourth columns of Fig. 8, the EE should be raised to reset the seventh joint and the gripper should be released. The KUKA manipulator finished the whole screwing process with minor variations in other joints and no swing of the

elbow, which is presented in detail in the accompanying video (See [Supplementary materials](#)). In addition, it can be observed that the bolt can be accurately screwed into the hole with the guarantee of *EE* precision by the proposed algorithm.

5. Conclusion

In this paper, an IK algorithm to solve the redundancy resolution and the RMP of the SRS manipulators has been presented. The proposed algorithm first implements the mapping from the task space to the joint Cartesian positions and then deduces the desired joint angles from the second mapping. Beyond that, the proposed algorithm can be applied to real-time applications and provide a high solve rate, which has been demonstrated by a quantitative IK test on the KUKA manipulator. In contrast with KDL and TRAC-IK, simulations and experiments have been performed on the KUKA manipulator to verify that the proposed algorithm can remedy the joint-drift problems and elbow swing. Meanwhile, the *EE* accuracy and the continuity of the trajectory can be guaranteed simultaneously. Moreover, when compared to the other two baselines, the proposed algorithm can generate superior configurations with higher manipulability.

Future work will focus on extending this algorithm to more types of manipulators. A more computationally efficient algorithm that does not need classification during the second mapping is worth investigating when applied to hyper-redundant manipulators. Some constraints on obstacle avoidance can be imposed in the first mapping to take advantage of the kinematic redundancy of redundant manipulators. Notably, the initial value has a significant impact on solution time for different IK queries, which is worth optimizing and further discussion.

Supplementary materials. To view supplementary material for this article, please visit <https://doi.org/10.1017/S0263574722001370>.

Authors' contributions. None.

Financial support. This work has been supported by the National Natural Science Foundation of China [Project Number: 92148203] and Self-Planned Task [NO. SKLRS202201A01] of State Key Laboratory of Robotics and System.

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

Ethical considerations. None.

References

- [1] L. Zhao, J. Zhao and H. Liu, "Solving the inverse kinematics problem of multiple redundant manipulators with collision avoidance in dynamic environments," *J. Intell. Robot. Syst.* **101**(2), 30 (2021).
- [2] L. Zhao, Z. Jiang, Y. Sun, J. Zhao and H. Liu, "Collision-free kinematics for hyper-redundant manipulators in dynamic scenes using optimal velocity obstacles," *Int. J. Adv. Robot. Syst.* **18**(1), 1–17 (2021).
- [3] R. K. Malhan, A. M. Kabir, B. Shah and S. K. Gupta, "Identifying Feasible Workpiece Placement with Respect to Redundant Manipulator for Complex Manufacturing Tasks," *In: IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2019) pp. 5585–5591.
- [4] Z. Mu, T. Liu, W. Xu, Y. Lou and B. Liang, "A hybrid obstacle-avoidance method of spatial hyper-redundant manipulators for servicing in confined space," *Robotica* **37**(6), 998–1019 (2019).
- [5] C. A. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst. Man Cybern.* **13**(2), 245–250 (1983).
- [6] Z. Li, F. Xu, D. Guo, P. Wang and B. Yuan, "New P-type RMPC scheme for redundant robot manipulators in noisy environment," *Robotica* **38**(5), 775–786 (2020).
- [7] Z. Li, C. Li, S. Li and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE Trans. Ind. Inform.* **16**(12), 7469–7478 (2019).
- [8] Z. Zhang, T. Fu, Z. Yan, L. Jin, L. Xiao, Y. Sun, Z. Yu and Y. Li, "A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators," *IEEE/ASME Trans. Mech.* **23**(2), 679–689 (2018).

- [9] P. Chiacchio and B. Siciliano, “A closed-loop jacobian transpose scheme for solving the inverse kinematics of nonredundant and redundant wrists,” *J. Robot. Syst.* **6**(5), 601–630 (1989).
- [10] J. Wang, Y. Li and X. Zhao, “Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm,” *Int. J. Adv. Robot. Syst.* **7**(4), 37–47 (2010).
- [11] C. W. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Trans. Syst. Man Cybern.* **16**(1), 93–101 (1986).
- [12] Y. Nakamura and H. Hanafusa, “Inverse kinematic solutions with singularity robustness for robot manipulator control,” *ASME J. Dyn. Syst. Meas. Cont.* **108**(3), 163–171 (1986).
- [13] P. Beeson and B. Ames, “TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics,” **In: IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)**, IEEE (2015) pp. 928–935.
- [14] T. Yoshikawa, “Manipulability of robotic mechanisms,” *Int. J. Robot. Res.* **4**(2), 3–9 (1985).
- [15] V. Abbasi, B. Azria, E. Tabarah, V. Menon, E. Phillips and M. Bedirian, “Improved 7-DOF Control of ISS Robotic Manipulators,” **In: Space OPS, 2004 Conference** (2004) pp. 407.
- [16] H. Lu, L. Jin, J. Zhang, Z. Sun, S. Li and Z. Zhang, “New joint-drift-free scheme aided with projected ZNN for motion generation of redundant robot manipulators perturbed by disturbances,” *IEEE Trans. Syst. Man Cybern. Syst.* **51**(9), 5639–5651 (2019).
- [17] L. Jin, Z. Xie, M. Liu, K. Chen, C. Li and C. Yang, “Novel joint-drift-free scheme at acceleration level for robotic redundancy resolution with tracking error theoretically eliminated,” *IEEE/ASME Trans. Mech.* **26**(1), 90–101 (2020).
- [18] D. Guo, Z. Li, A. H. Khan, Q. Feng and J. Cai, “Repetitive motion planning of robotic manipulators with guaranteed precision,” *IEEE Trans. Ind. Inform.* **17**(1), 356–366 (2020).
- [19] L.-C. Wang and C.-C. Chen, “A combined optimization method for solving the inverse kinematics problems of mechanical manipulators,” *IEEE Trans. Robot. Autom.* **7**(4), 489–499 (1991).
- [20] A. Aristidou, J. Lasenby, Y. Chrysanthou and A. Shamir, “Inverse kinematics techniques in computer graphics: A survey,” *Comput. Graph. Forum* **37**(6), 35–58 (2018).
- [21] W. Xu, L. Yan, Z. Mu and Z. Wang, “Dual arm-angle parameterisation and its applications for analytical inverse kinematics of redundant manipulators,” *Robotica* **34**(12), 2669–2688 (2016).
- [22] B. Ma, Z. Xie, Z. Jiang and H. Liu, “Precise semi-analytical inverse kinematic solution for 7-DOF offset manipulator with arm angle optimization,” *Front. Mech. Eng.* **16**(3), 435–450 (2021).
- [23] J. Oh, H. Bae and J.-H. Oh, “Analytic Inverse Kinematics Considering the Joint Constraints and Self-Collision for Redundant 7DOF Manipulator,” **In: First IEEE International Conference on Robotic Computing (IRC)**, IEEE (2017) pp. 123–128.
- [24] A. Sinha and N. Chakraborty, “Geometric Search-based Inverse Kinematics of 7-DoF Redundant Manipulator with Multiple Joint Offsets,” **In: International Conference on Robotics and Automation (ICRA)**, IEEE (2019) pp. 5592–5598.
- [25] G. S. Chirikjian and J. W. Burdick, “A modal approach to hyper-redundant manipulator kinematics,” *IEEE Trans. Robot. Autom.* **10**(3), 343–354 (1994).
- [26] E. Oyama, N. Y. Chong, A. Agah and T. Maeda, “Inverse Kinematics Learning by Modular Architecture Neural Networks with Performance Prediction Networks,” **In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)**, IEEE (2001) pp. 1006–1012.
- [27] G. M. Marconi, R. Camoriano, L. Rosasco and C. Ciliberto, “Structured prediction for CRiSP inverse kinematics learning with misspecified robot models,” *IEEE Robot. Autom. Lett.* **6**(3), 5650–5657 (2021).
- [28] Z. Xie, L. Jin, X. Luo, B. Hu and S. Li, “An acceleration-level data-driven repetitive motion planning scheme for kinematic control of robots with unknown structure,” *IEEE Trans. Syst. Man Cybern. Syst.* **52**(9), 5679–5691 (2021).
- [29] A. A. Hassan, M. El-Habrouk and S. Deghedie, “Inverse kinematics of redundant manipulators formulated as quadratic programming optimization problem solved using recurrent neural networks: A review,” *Robotica* **38**(8), 1495–1512 (2020).
- [30] Z. Xie, L. Jin, X. Du, X. Xiao, H. Li and S. Li, “On generalized RMP scheme for redundant robot manipulators aided with dynamic neural networks and nonconvex bound constraints,” *IEEE Trans. Ind. Inform.* **15**(9), 5172–5181 (2019).
- [31] Z. Li, C. Li, S. Li, S. Zhu and H. Samani, “A sparsity-based method for fault-tolerant manipulation of a redundant robot,” *Robotica* **40**(10), 3396–3414 (2022).
- [32] S. Starke, N. Hendrich, S. Magg and J. Zhang, “An Efficient Hybridization of Genetic Algorithms and Particle Swarm Optimization for Inverse Kinematics,” **In: IEEE International Conference on Robotics and Biomimetics (ROBIO)**, IEEE (2016) pp. 1782–1789.
- [33] S. Starke, N. Hendrich and J. Zhang, “Memetic evolution for generic full-body inverse kinematics in robotics and animation,” *IEEE Trans. Evol. Comput.* **23**(3), 406–420 (2019).