IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 4, 2019, accepted January 3, 2020, date of publication January 15, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966768

An Efficient and Accurate Inverse Kinematics for 7-DOF Redundant Manipulators Based on a Hybrid of Analytical and Numerical Method

MINGHE JIN, QIANG LIU, BIN WANG, AND HONG LIU
State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 15001, China

Corresponding author: Qiang Liu (14s008022@hit.edu.cn)

This work was supported in part by the Foundation Innovative Research Groups of the National Natural Science Foundation of China under Grant 51521003, and in part by the Self-Planned Task of the State Key Laboratory of Robotics and System (HIT) under Grant SKLRS201805B.

ABSTRACT This paper proposes an efficient and accuracy inverse kinematic algorithm for 7-DOF redundant manipulators with obstacles avoidance and singularities avoidance based on the hybrid of analytical and numerical method (IK-HAN). Specially, the paper focuses on how to solve the inverse kinematics problem accurately and efficiently for a novel configuration, i.e. SSRMS-type manipulator. First, the elbow orientation is introduced and the algebraic relationship between the elbow orientation and joint angles is derived. Second, the optimization algorithm is designed to find the optimal elbow orientation based on Particle Swarm Optimization. To improve the efficiency, the equivalent optimization model based on the azimuth angle is investigated. Third, optimal models are developed to avoid obstacles and singularities and improve manipulability in the constraint domain. Moreover, how to employ optimization resolution to solve the inverse kinematics problem is discussed. Finally, the validity of the algorithm is verified via kinematics simulations and the result illustrates that the algorithm performs well in accuracy, stability and efficiency.

INDEX TERMS Elbow orientation, obstacles avoidance, redundant manipulators, real-time inverse kinematics, singularities avoidance.

I. INTRODUCTION

The redundant manipulators have great advantages in dealing with complicated problems than 6-DOF robots, such as obstacles avoidance [1], singularities handing [2], arm angle limits [3] minimum base disturbance [4], so they are applied in many fields including aerospace docking [5] bionic arms [6] and medical robots [7]. However, the kinematic control of redundant manipulators is a big challenge because manipulators more than 6 DOF results in an infinite number of solutions in the joint space which map the same pose in the Cartesian space. Analytical solutions cannot be figured out directly owing to the one-to-many relationship space.

Some additional constraints must be exploited to pick out the optimal solutions of redundant manipulators. The constraints can be meet by adding instantaneous cost functions. There are various methods to optimize cost functions including weighted least norm (WLN) [8], nonlinear programming problem (NLP) [9] and the quadratic programming (QP) [10]. WLN is effective for the velocity limits, however, other tasks cannot be guaranteed well except the main task. For multiply tasks, the general weighted least-norm (GWLN) method was proposed [11], [12] which is a method of modified WLN. GWLN introduces a concept of virtual joints and assume that each subtask could be formulated in terms of cost functions about virtual joints. However, the performances of GWLN subtasks are determined by weights and the Jacobian's singularity and meanwhile the computing efficiency is low. It is not suitable for the real-time projections. The Jacobian's pseudo inverse-based technologies are adopted broadly to estimate joint velocities of redundant manipulators to replace the real solutions [13]. The velocity damping method is a classical application of pseudoinverse of Jacobi matrixes. This method limits joint velocities but massive iterations are need to get accuracy solutions [14]. The pseudoinverse of Jacobian is also used to figure out higher order inverse kinematics [9] which can generate high order smooth trajectories and eliminate vibrations of elastic components of the robot. The higher

The associate editor coordinating the review of this manuscript and approving it for publication was Saeid Nahavandi.

order inverse kinematics (IK) method adopts NLP to minimize the terminal time and obtains the optimal solutions, but this algorithm, on the expense of high computational load, is not suitable for the real-time projects.

Although in most literatures the joint limits have been avoided by changing joint velocities so far, it is difficult for velocity-based method to plan positions exactly under region constraints. In fact, the engineering robots control effector velocity by changing step data derived from positions. Therefore, a position-based inverse solution approach of kinematics is necessary. The arm angle proposed by Kreutz-Delgado [15] is broadly utilized to solve the position-based inverse kinematics, especially for the redundant manipulator with 7 revolute joints (R) configured as S-R-S(spherical joint). The arm angle specifies the orientation of arm plane and is equivalent to another kinematic constrain to the manipulator, so the unique inverse resolution can be determined. However, the algorithm exists singularities when the arm plane is parallel with the axis connecting shoulder and wrist. On another hand, the arm angle, as a parameter defined dynamically, is difficult to determine intuitively. Analogously, a redundant angle was introduced to replace the arm angle as a new kinematic constraint and they have the same fundamental principle [16]. However, this method also has the same drawbacks to the arm angle and both angles are defined between the same dynamic plane and a static vector. Reference [17] analyzed singularities of the algorithm and analytical methods of inverse kinematics for joint limit avoidances. Another popular manipulator for 7R is Space Station Remote Manipulator System (SSRMS) [18], which is a topological configuration of SRS but has more complex structures - additional link offsets at the shoulder, elbow and wrist parts. However, in previous studies, the most approaches to inverse redundant solution are designed for SRS and are not available to SSRMS. For this, given the configuration characteristics of the SRS, the shoulder and wrist of SSRMS are reconstructed as a virtual spherical-joint and inverse solutions for SSRMS are figured out by the same method to SRS [19]. SSRMS-type manipulators are reconstructed as a new 6-DOF manipulator by locking one joint. And then the inverse kinematic analysis of SSRMS can be performed by closing the loop with the hypothetical joint. However, it is difficult for this method to guarantee global optimal performance of manipulators, such as flexibility and manipulability [20], [21]. Patchaikani [1] utilized a Single Network Adaptive Critic (SNAC) to build an optimal controller for closed-loop error discrete-time dynamics. The controller formulates the inputs and cost function by the TS fuzzy system and resolve the real-time optimal redundancy resolutions. However, the accuracy of the resolutions is subject to training data, computing time and the initial position of the end effector.

Trajectory tracking is the primary task for redundant manipulators. In addition, if robots meet with obstacles, additional tasks should be performed with the primary task simultaneously. In the respect of obstacles avoidance, plenty of researchers have applied the artificial potential field methods to avoid collisions in the last decades. However, the collision-free trajectory in the Cartesian space is subject to the actuation which is positive to distance from desire position and also subject to the repulsion which is negative to the distance from obstacles. If the actuation and repulsion are matched badly with each other, the algorithm easily steps into the local optimum and meanwhile the manipulator is lead to the potential basin and trapped in self-lock [10]. A method of internal planning in joint space was proposed to avoid local optimum for SRS [13]. This method proposes a virtual force concept, which adjusts the redundant joints to avoid obstacles and self-lock, but there are the angle abrupt changes. A three new screw-based inverse kinematics (IK) sub-problems and their corresponding models are presented and could be applied to find the IK solutions for manipulators with rotational joints without angle abrupt changes [22]. Given that the most of collisions happen in arm links, the pose of arm links are considered as the key factor to avoid obstacles. The redundant angle can be utilized to find the angle range of collisions free but the collision detection algorithm needs be carried out many times for determining an optimal angle just in one pose, so it weaken the real-time performance of the algorithm [16]. Considering constraints including angle limits, collisions avoidance, minimum disturbance and so on, the inverse kinematics problem will become more complex and constraints cannot be expressed in the same space. For this purpose, the Neuro Networks (NN) are widely adopted to solve this problem because of its generalization, versatility, simplification and independence of models [23], [24]. However, NN needs large amounts of data to train the model and improve learning rate and accuracy, so it is bad at the accuracy, the manipulability and the computation cost even worse in time-varying surroundings [10], [25]. The null-space projection of Jacobian can guarantee the trajectory of end-effector well and avoid obstacles by changing joints in the null-space [26], [27], but also constraints the subtask performance seriously and breaks down the high position accuracy when any joint is close to limits [11].

The studies of inverse kinematics about SRS is adequate, but about SSRMS is few. In the studies of inverse kinematics, SRS is simplified as a two-link robot with the known pose, while SSRMS is only simplified as a six-link robot owing to joint offsets. That is to say that SSRMS has more unknown parameters to be determined and more complex structures than SRS. So, approaches to solve the inverse kinematics of SRS are not available to SSRMS. This paper introduces a novel notation - the elbow orientation (EO) as an additional control parameter and proposes a hybrid of analytical and numerical method (HAN) to solve the real-time inverse kinematics for SSRMS accurately and efficient. The proposed algorithm is applied to a self-developed SSRMS-type manipulator which is used in space activities including assembling, repairing, upgrading a satellite or space station. The elbow orientation is a three-dimension vector defined in the static frame and represents the pose of redundant manipulator. The inverse kinematics solution is determined uniquely

after the elbow orientation is picked out by the optimal algorithm. To improve efficiency of the algorithm, a one-dimension optimal model is established to replace the optimization for the elbow orientation and then the numerical solutions are obtained. Additionally, the obstacles model and free-singularity scheme are introduced into the algorithm to avoid obstacles and singularities in real time. The main highlights in this paper are

1) The efficiency and accuracy of IK-HAN. Most researches on IK only focus on the theory studies and ignore that the efficiency and accuracy are vital for engineering applications.

2) A concept based on a hybrid of analytical and numerical method is applied to solve the inverse kinematics for SSRMS-type manipulators. It is the key to improve the efficiency of the proposed IK-HAN.

The rest of this paper is organized as following. Section II analyzes the kinematics model of SSRMS manipulator and derives the analytical solution of the inverse kinematics based on known EO. Section III proposes a novel optimization algorithm which improves the computation efficiency. At first, the optimization model for EO is transformed to be the model for the azimuth angle. And then the PSO-based optimization algorithm is designed to search the optimal numerical solution of EO. The additional subtasks case and its combination are addressed in Section IV. The multiply-task optimization model is established and employed to solve the cases for the feasible combination of multiple subtasks. The novel inverse kinematics algorithm based on the analytical and numerical method (IK-HAN) is proposed. The simulations are illustrated in Section V, where the accuracy, the convergence and the consuming time are analyzed and compared. As a realistic project, the inverse kinematics verified by the simulation are be used in the self-developed SSRMS-type manipulator which completes space activities in the future. Finally, Section VI concludes the paper.

II. ANALYSIS OF INVERSE KINEMATICS

A. SSRMS CONFIGURATION

SSRMS is a topological configuration of SRS but has additional link offsets at the shoulder, elbow and wrist parts respectively. Its joints are arranged as following

1) The shoulder consists of three joints, i.e. shoulder roll, yaw and pitch joints.
2) The elbow is pitch joint.
3) The wrist consists of three joints, i.e. wrist roll, yaw and pitch joints.

The joint layout of SSRMS-type manipulators is display as (Roll-Yaw-Pitch)-Pitch-(Pitch-Yaw-Roll), as shown in Fig. 1. Analyzing the configuration, we can see that three joints of the shoulder or wrist have link offsets respectively and their axes are not intersected at one point. Axes of middle adjacent joints (i.e. the 3rd, 5th joints and the elbow joint) are parallel with each other. Although the SSRMS-type manipulator loses spherical joints and its inverse kinematics solutions is difficult, it has joints of lager movement range and can avoid mechanical interference.

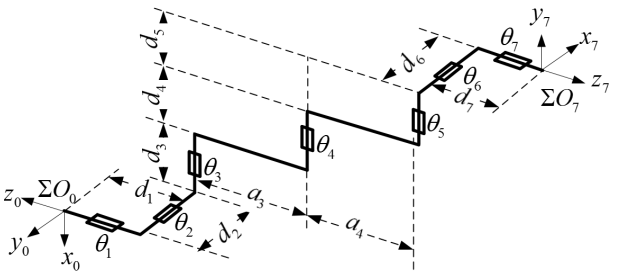


FIGURE 1. The configuration of SSRMS manipulator.

B. MODEL ANALYSIS

The modified D-H parameters of the SSRMS-type manipulator are presented in Table 1. The base frame ΣO_0 is fixed to the ground and subsequent variables are expressed in ΣO_0. The frames from ΣO_1 to ΣO_6 are established according to the modified D-H standard and the frame ΣO_7 is attached to the end effector so that the position and orientation of the target is convenient to be expressed. The pose (position and orientation) of ΣO_7 with respect to ΣO_0 is known all the time. Fig. 1 shows the D-H frames and the home configuration in which each joint is zero off-set.

TABLE 1. The modified D-H parameters of the SSRMS-type manipulator.

i.	${}^{i-1}T_i$	a_{i-1}	α_{i-1}	d_i	θ_i
1	${}^{0}T_1$	0	-90°	c_0	0
2	${}^{1}T_2$	0	90°	c_1	0
3	${}^{2}T_3$	0	-90°	c_2	−90°
4	${}^{3}T_4$	c_3	0	c_4	0
5	${}^{4}T_5$	c_5	0	c_6	90°
6	${}^{5}T_6$	0	-90°	c_7	0
7	${}^{6}T_7$	0	90°	c_8	0

The arm plane is defined as a plane passing through the midpoint of link 'a_4' and perpendicular to the axes of 4th joint, shown as Fig. 2. By analysis, we can know that the arm plane can rotate for a fixed pose of the end effector and this rotation forms self-motion and results in the redundancy and multiply solutions [17]. However, the rotation is about a dynamic axis and its position cannot be derived by the known information. Motivated by the inspiration for human arm motions, this paper adopts the elbow orientation as a redundant parameter to control the manipulator moving.

C. ELBOW ORIENTATION (EO)

$\boldsymbol{z}_i$ is a z-coordinates unit vector in the D-H frame and represents the orientation of the i^{th} joint. We define the z-coordinate of ΣO_4 as the elbow orientation (EO) denoted as $\boldsymbol{e}_o$, the z-coordinate of ΣO_7 as the end effector orientation denoted as $\boldsymbol{e}_e$ and the z-coordinate of ΣO_1 denoted

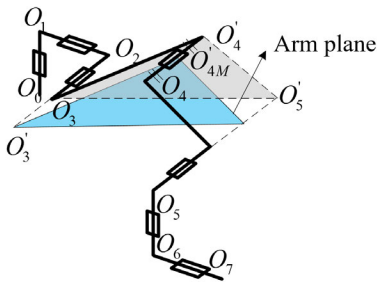


FIGURE 2. The arm plane passes through O_{4M} and is perpendicular to the axes of 4th joint, where O_{4M} is the midpoint of link 'a_4'. The plane $\triangle O_3 O_4' O_5'$ is used to figure out the interior angles.

as $\boldsymbol{e}_b$, The EO indicates the orientation of arm plane and is independent of positions of the shoulder and the wrist, that is O_3, O_5'. For a fixed shoulder in space along with a given pose of the wrist, the manipulator can be fully determined if only the EO is specified. So, it is chosen as an additional control parameter to figure out unique joints. The specific geometrical relationships between the EO and the position of joints are shown as following.

At any time, three geometric relationships are always obeyed. First, in the shoulder joint,

$$\boldsymbol{z_1} = \boldsymbol{e_b} \tag{1}$$

and z_2 is always perpendicular to its adjacent joints $\boldsymbol{z_1}, \boldsymbol{z_3}$, so

$$\boldsymbol{z_2} = [\![\boldsymbol{z_3} \times \boldsymbol{z_1}]\!] \tag{2}$$

where $[\![*]\!]$ is defined as the operation of unitizing vectors. Second, the axes of 3rd, 5th joint and elbow are always parallel with each other and we can know

$$\begin{aligned} \boldsymbol{z_3} &= \boldsymbol{e_o}, \\ \boldsymbol{z_5} &= \boldsymbol{e_o}. \end{aligned} \tag{3}$$

Third, in the wrist joint, $\boldsymbol{z_6}$ is perpendicular to its adjacent joints $\boldsymbol{z_5}, \boldsymbol{z_7}$, so

$$\boldsymbol{z_6} = [\![\boldsymbol{z_5} \times \boldsymbol{z_7}]\!]. \tag{4}$$

The position of the point $\boldsymbol{O_i}$ with respect to the base frame is denoted as $\boldsymbol{p_{Oi}}$,

$$\boldsymbol{p_{01}} = a_0 * \boldsymbol{z_1} \tag{5}$$

$$\boldsymbol{p_{02}} = a_1 * \boldsymbol{z_2} + \boldsymbol{p_{01}} \tag{6}$$

$$\boldsymbol{p_{03}} = a_2 * \boldsymbol{z_3} + \boldsymbol{p_{02}} \tag{7}$$

The position of the end effector is known and denoted as $\boldsymbol{p_{07}}$, the position of latter three joints can be computed, as

$$\boldsymbol{p_{06}} = -a_8 * \boldsymbol{e_e} + \boldsymbol{p_{07}} \tag{8}$$

$$\boldsymbol{p_{05}} = -a_7 * \boldsymbol{z_6} + \boldsymbol{p_{06}} \tag{9}$$

To compute the position of the elbow, O_4', O_5' are introduced and establish a plane $O_3 O_4' O_5'$ parallel with the arm plane, where the line $O_4' O_5'$ is parallel with the link a_5, shown as Fig.2.

From now, we define $\boldsymbol{p_{xy}}$ as the vector $\overrightarrow{O_x O_y}$, such as $\boldsymbol{p_{34'}}, \boldsymbol{p_{4'5'}}, \boldsymbol{p_{35'}}$ as the vectors $\overrightarrow{O_3 O_4'}, \overrightarrow{O_4' O_5'}, \overrightarrow{O_3 O_5'}$ respectively and define p_{xy} as the magnitude of $\boldsymbol{p_{xy}}$. In $O_3 O_4' O_5'$, obviously,

$$p_{34'} = a_3, \tag{10}$$

$$p_{4'5'} = a_5, \tag{11}$$

and

$$\boldsymbol{p_{35'}} = \boldsymbol{p_{35}} + \boldsymbol{p_{55'}}$$

where $\boldsymbol{p_{35}} = \boldsymbol{p_{05}} - \boldsymbol{p_{03}}$ and $\boldsymbol{p_{55'}}$ is the vector of $\overrightarrow{O_5 O_5'}$ parallels with the EO. So,

$$\boldsymbol{p_{35'}} = \boldsymbol{p_{05}} - \boldsymbol{p_{03}} + (a_4 + a_6) * \boldsymbol{e_o}. \tag{12}$$

Fig.2 shows the initial configuration and the initial pose and the orientation of the end effector z_7 is express in the base frame ΣO_0.

D. INVERSE KINEMATICS

1) FORWARD KINEMATICS

Based on the D-H notation, the pose of the i^{th} D-H frame in the $(i-1)^{th}$ D-H frame can be denoted as the 4×4 homogeneous transformation matrix ${}^{i-1}T_i$

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where θ, α, d, a are listed in the Table 1 and Fig.1 is referred to the home configuration where all links are parallel with the arm plane. The forward kinematics ${}^0T_7$ expresses the pose of the end effector with respect to the base frame of the articulated mechanism consisting of serial multi-links

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7, \tag{14}$$

2) INVERSE KINEMATICS

The method of picking up the optimal EO is analyzed in Section III. Given that EO and the six constraints with the end pose forming a total of seven constraints, the configuration is fixed. So, all joint angles can be derived with the help of the analysis in Section II-C.

In the base frame, we define z_i as the positive rotation direction of the i^{th} joint. θ_1 is formed by the rotation of z_2 around z_1, so

$$|\theta_1| = < -\boldsymbol{e_b}, \boldsymbol{z_2} >, \tag{15}$$

where $< \boldsymbol{a}, \boldsymbol{b} >$ is defined as the angle norm between the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and can be calculated as follow

$$< \boldsymbol{a}, \boldsymbol{b} > = arccos \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}| \cdot |\boldsymbol{b}|}. \tag{16}$$

The rotation direction of θ_1 is

$$sgn_1 = [\![-\boldsymbol{e_b} \times \boldsymbol{z_2} \cdot \boldsymbol{z_1}]\!]. \tag{17}$$

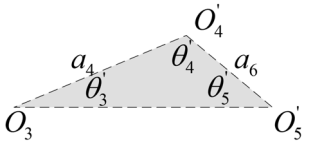


FIGURE 3. The plane $\triangle O_3 O_4' O_5'$ is used to figure out θ_4' and $p_{34'}, p_{4'5'}$.

So, $\theta_1 = sgn_1 * |\theta_1|$ and same as below. Similarly, θ_2 is also expressed as

$$\begin{aligned} |\theta_2| &= < z_1, \boldsymbol{e_o} >, \\ sgn_2 &= [\![z_1 \times \boldsymbol{e}_o \cdot z_2]\!]; \end{aligned} \tag{18}$$

θ_6 is

$$\begin{aligned} |\theta_6| &= < \boldsymbol{e_o}, \boldsymbol{e_e} >, \\ sgn_6 &= [\![\boldsymbol{e}_o \times \boldsymbol{e_e} \cdot z_6]\!]. \end{aligned} \tag{19}$$

In $\triangle O_3 O_4' O_5'$ shown in Fig.3, according to (10), (11), (12), the interior angles θ_3', θ_4', can be computed as

$$\begin{aligned} \theta_3' &= arccos \frac{a_3^2 - a_5^2 + p_{35'}^2}{2 a_3 p_{35'}}, \\ \theta_4' &= arccos \frac{a_3^2 + a_5^2 - p_{35'}^2}{2 a_3 a_5}. \end{aligned} \tag{20}$$

According to (7), (12) and (19), the position of O_4' in. the base frame can be derived as

$$\boldsymbol{p_{04'}} = \boldsymbol{p_{03}} + a_3 cos\theta_3' [\![\boldsymbol{p_{35'}}]\!] + a_3 a_5 sin\theta_4' / p_{35'} [\![\boldsymbol{e_o} \times \boldsymbol{p_{35'}}]\!]. \tag{21}$$

Given (7), (12), (21), we can obtain

$$\boldsymbol{p_{34'}} = \boldsymbol{p_{04'}} - \boldsymbol{p_{03}}, \tag{22}$$

and

$$\boldsymbol{p_{4'5'}} = \boldsymbol{p_{03}} + \boldsymbol{p_{35'}} - \boldsymbol{p_{04'}}. \tag{23}$$

θ_3 is formed by the rotation of link a_3 around $\boldsymbol{e}_o$, so

$$\begin{aligned} |\theta_3| &= < z_2, \boldsymbol{p_{34'}} >, \\ sgn_3 &= [\![z_2 \times \boldsymbol{p_{34'}} \cdot \boldsymbol{e_o}]\!]. \end{aligned} \tag{24}$$

Similarly, according to (19), (22), (23),

$$\begin{aligned} |\theta_4| &= \pi - \theta_4', \\ sgn_4 &= [\![\boldsymbol{p_{34'}} \times \boldsymbol{p_{4'5'}} \cdot \boldsymbol{e_o}]\!] \end{aligned} \tag{25}$$

and according to (8), (9), (23),

$$\begin{aligned} |\theta_5| &= < \boldsymbol{p_{4'5'}}, \boldsymbol{p_{56}} >, \\ sgn_5 &= [\![\boldsymbol{p_{4'5'}} \times \boldsymbol{p_{56}} \cdot \boldsymbol{e_o}]\!]. \end{aligned} \tag{26}$$

where $\boldsymbol{p_{56}} = \boldsymbol{p_{06}} - \boldsymbol{p_{05}}$.

So far, $\theta_1 - \theta_6$ have been derive as above and z_7 has been coincident with the desired pose after the former six joints rotate their corresponding angles, that is, θ_1 to θ_6, respectively. The last operation is to rotate the current end effector around z_7 to get to the desired pose. We define the desired x-coordinate of ΣO_7 as $\boldsymbol{x}_{d7}$ and the current x-coordinate of ΣO_7 as $\boldsymbol{x}_7$. According to (14), we can obtain

$$\boldsymbol{x_7} = {}^0T_7(\boldsymbol{\theta}) \cdot [0; 0; 1; 0], \tag{27}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, 0]$. According to (27), θ_7 is

$$\begin{aligned} |\theta_7| &= < \boldsymbol{x_7}, \boldsymbol{x}_{d7} >, \\ sgn_7 &= [\![\boldsymbol{x_7} \times \boldsymbol{x_{d7}} \cdot z_7]\!]; \end{aligned} \tag{28}$$

III. EO OPTIMIZATION

A. CONSTRAINT MODEL OF EO

For the fixed end effector pose, any self-motion corresponds a unique EO, so there are numerous EOs satisfying tip constraints. However, not all the orientation vectors can be considered as the EO which achieves the tip pose. To find a reasonable configuration of the self-motion, an optimization model need be established to pick up the optimal EO. The searching ranges of redundant parameters are easy to be specified for one-dimension parameters such as the arm angle, redundant angel, but it is difficult for the EO because the EO distributes irregularly in the Cartesian space. A novel optimal model is introduced to solve these problems. Two aspects are discussed about the model, the constraint model and the cost function.

1) CONSTRAINT MODEL

In the Cartesian space, a 3-dimension vector can be optimized by adjusting each component iteratively in the global set so that the cost function can approach to the expected value. Although this algorithm is simple and intuitive, it has a large computation cost and deteriorates the real-time performance of the inverse kinematics. In this paper, the constraint model is established according to the geometry relationship and simplifies the searching range from 3-dimension to 1-dimension. In addition, this model is able to specify the searching set of the EO precisely and improve the searching efficiency.

$\boldsymbol{p_{16\perp}}$ is defined as the projection perpendicular to the arm plane of $\boldsymbol{p_{16}}$, so $\boldsymbol{p_{16\perp}}$ is parallel with $\boldsymbol{e_o}$ and

$$p_{16\perp} = \boldsymbol{p_{16}} \cdot \boldsymbol{e_o}. \tag{29}$$

According to the configuration of the manipulator, we can know that link a_2, a_4, a_6 are parallel with $\boldsymbol{e_o}$, so

$$p_{16\perp} = a_2 + a_4 + a_6. \tag{30}$$

The equivalent condition of existence of $\boldsymbol{e_o}$ is derived by substituting (30) into (29)

$$\boldsymbol{p_{16}} \cdot \boldsymbol{e_o} = D, \tag{31}$$

where $D = a_2 + a_4 + a_6$ and $\boldsymbol{e_o}$ is a unit vector

$$e_o = 1. \tag{32}$$

Constraint conditions (31), (32) are combined as the constraint model.

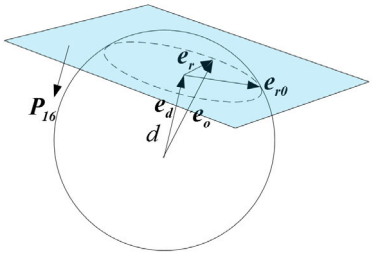


FIGURE 4. Equivalent optimal model based on the azimuth angle.

2) AZIMUTH ANGLE

In terms of geometry, the constraint model is considered as the intersection of the unit sphere's surface and the constraint plane whose orientation vector is $\boldsymbol{p_{16}}$. The distance d is measured between the plane and the origin, shown in Fig.4. $\boldsymbol{e_o}$ can be decomposed as $\boldsymbol{e_d}$ and $\boldsymbol{e_r}$,

$$\boldsymbol{e_o} = \boldsymbol{e_r} + \boldsymbol{e_d}, \tag{33}$$

and $\boldsymbol{e_d}$ is parallel with $\boldsymbol{p_{16}}$

$$\boldsymbol{e_d} = d \cdot [\![\boldsymbol{p_{16}}]\!], \tag{34}$$

where $d = D/p_{16}$ is the distance between the origin and the constraint plane.

To reduce optimal dimensions and improve the efficiency of the constraint model, the azimuth angle φ is introduced to express $\boldsymbol{e_r}$. The azimuth angle φ represents the angle between $\boldsymbol{e_r}$ and its initial value $\boldsymbol{e_{r0}}$. Assuming $\boldsymbol{e_o} = [e_{ox}, e_{oy}, e_{oz}]$, we define dynamically the initial value of $\boldsymbol{e_r}$ as $\boldsymbol{e_{r0}} = [e_{oy}, -e_{ox}, 0]$. If $\boldsymbol{p_{16}}$ is fixed, the $\boldsymbol{e_d}$ is a constant vector and $\boldsymbol{e_r}$ is the function about the azimuth angle

$$\boldsymbol{e_r}(\varphi) = L_\varphi(\boldsymbol{e_{r0}}), \tag{35}$$

where $L_\varphi(*)$ is the operator achieving a rotation from $\boldsymbol{e_{r0}}$ to the desired orientation $\boldsymbol{e_r}$ through the angle φ. Here, the quaternion algebra is introduced to solve (35) and define the unit quaternion $\boldsymbol{\eta}(\varphi) = [\eta_0, \boldsymbol{\eta_v}] = [\![[\cos(\varphi/2), \sin(\varphi/2) \cdot \boldsymbol{e_d}]]\!]$. According to the quaternion theory [28], for the vector $\boldsymbol{e_{r0}}$, the rotation can be achieved by

$$L_\varphi(\boldsymbol{e_{r0}}) = \boldsymbol{\eta} \otimes \boldsymbol{e_{r0}} \otimes \boldsymbol{\eta}^*, \tag{36}$$

where $\boldsymbol{\eta}^*$ is the conjugate of $\boldsymbol{\eta}$ and $\otimes$ represents the quaternion product. The product acts on the quaternion $\boldsymbol{\eta}$ as $\boldsymbol{\eta} \otimes = \eta_0 \mathbf{1_4} + \boldsymbol{\Omega}(\boldsymbol{\eta_v})$, where $\boldsymbol{\Omega}(\boldsymbol{\eta_v}) = \begin{bmatrix} -[\boldsymbol{\eta_v}\times] & \boldsymbol{\eta_v} \\ -\boldsymbol{\eta_v}^T & 0 \end{bmatrix}$. Further, (36) can be expressed as

$$L_\varphi(\boldsymbol{e_{r0}}) = \boldsymbol{A}(\varphi) * \boldsymbol{e_{r0}}, \tag{37}$$

where $\boldsymbol{A}(\varphi) = (2\eta_0^2 - 1)\,\mathbf{1_3} + 2\eta_0[\boldsymbol{\eta_v}\times] + 2\boldsymbol{\eta_v}\boldsymbol{\eta_v}^T$. Substitute (34), (37) into (33), $\boldsymbol{e_o}$ can be obtained

$$\boldsymbol{e_o} = d \cdot [\![\boldsymbol{p_{16}}]\!] + \boldsymbol{A}(\varphi) \cdot \boldsymbol{e_{r0}}, \tag{38}$$

where $\varphi \in [0, 2\pi)$.

The constraint model can be express by (38) equivalent to simultaneous constraint conditions (31), (32). The mapping relationship between $\boldsymbol{e_o}$ and $\boldsymbol{\varphi}$ is established, so the optimal problem of $\boldsymbol{e_o}$ is transformed as the optimal problem of φ.

B. OPTIMIZATION ALGORITHM

Because of the iteration of $T(\varphi)$ in (35), analytic solutions of the optimal problem are unavailable and an optimization algorithm need be chosen to solve the problem. There are plenty of methods applied to optimize the redundant parameter, including NN [29], the least square problem (LSP) [30] and the predictive task-scaling technique (PTS) [31]. However, it is difficult for NN and LSP to obtain the high accuracy and the low complexity simultaneously. For PTS, it is hard to cope with the noise happened in the high order sensors in engineer projects. This paper proposes a novel algorithm based on particle swam optimization (PSO) and the algorithm can guarantee the accuracy and the efficiency.

1) PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) was a discipline rising in the 1990's and proposed by Kennedy and Eberhart [32]. The basic concept of PSO is illustrated in Fig.5. The scheme of PSO to approach to the target is shown as Fig.6. PSO has many advantages including fast convergence, high efficiency, no limitation to the scale of population. In addition, PSO is good at global searching and suitable for non-linear or multi-extremum problems.

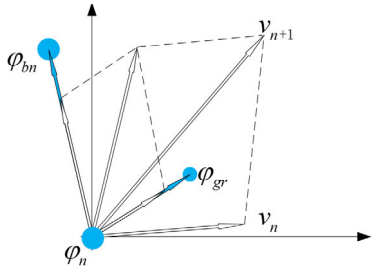
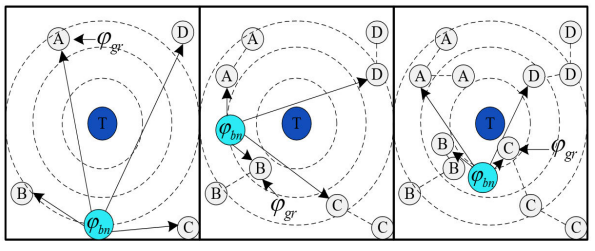


FIGURE 5. Illustration of the swarm's random feeding behavior.



Particles: A, B, C, D; Target: T; Global best: φ_{bn}; Local best: φ_{gr}

FIGURE 6. Schematic diagram of PSO.

According to the analysis in Section III-B, the design variable is the azimuth angle φ. Suppose the position and the velocity of the n^{th} particle is denoted as φ_n and v_n respectively and their updates can be expresses as

$$\begin{aligned} v_{n+1} &= wv_n + h_1 r_1(\varphi_{bn} - \varphi_n) + h_2 r_2(\varphi_{gr} - \varphi_n) \\ \varphi_{n+1} &= \varphi_{n+1} + v_{n+1} \end{aligned} \tag{39}$$

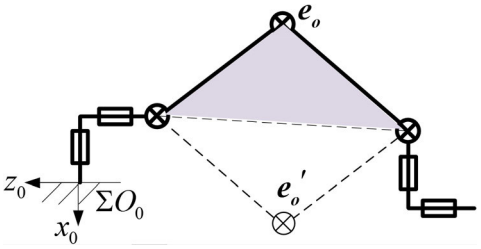


FIGURE 7. The configuration jump of the self-motion. During the self-motion, there always exist two configurations satisfying the fixed EO.

where h_1, h_2 are acceleration constants and r_1, r_2 are random distributed values between [0,1]. φ_{bn}, φ_{gr} are the local optimization of the n^{th} particle and the global optimization after the r^{th} generation respectively and w is the inertial weight factor which can change the searching range. According to (39), the updated velocity of the n^{th} particle is composed of the inertial velocity, the velocity increasement towards the global optimization and the velocity increasement towards the local optimization.

2) FITNESS FUNCTION

The fitness function is applied in PSO to describe the quality of particle and lead the swarm to get to the target, so it can be designed to minimize the cost function. To smooth the trajectory in the joint space and avoid the configuration jump as shown in Fig.7, a good choice is

$$\Gamma = \lambda_1 \left(\varphi_n - \varphi_{j-1}\right)^2, \tag{40}$$

where φ_j is the optimal azimuth angle corresponding the previous (j^{th}) joint command $\boldsymbol{\theta}_j$.

More generally, other functions can be achieved by add cost functions to (39), so the general fitness function is

$$\Gamma = \lambda_1 \left(\varphi_n - \varphi_{j-1}\right)^2 + \sum_{k>1} \lambda_k X_k, \tag{41}$$

where X_k is the k^{th} additional task and λ_k is the corresponding weight and $X_1 = \left(\varphi_n - \varphi_{j-1}\right)^2$. The analysis about X_k is shown in Section III.

Inverse Kinematics Using EO.

IV. ADDITIONAL TASKS

Using the self-motion of EO, the SSRMS-type redundant manipulator can deal with more complicated problems than smoothing trajectory and limiting joints' jump. Notice that the SSRMS-type manipulator is different from the hyper-redundant manipulators and is aimed to guide the end-effector to accurately track and capture in the position and orientation in space services. So, at least 6 joints are employed to achieve the desired pose of the end-effector. The only remained degree of freedom, EO, is applied to control the arm plane to avoid obstacles. The problems is formulated as additional tasks X_k and substituted into the fitness function (41) to evaluate the quality of each particle in the swarm. According to the quality, the particle will be driven to the target stepwise. In this paper, the fitness function is designed to avoid obstacles and singularities and then optimizes EO in the manipulability.

A. OBSTACLES AVOIDANCE

To avoid the collision between obstacles and the whole manipulator, this paper establishes an obstacle avoidance model. By analyzing the configuration of SSRMS, we can find the different part of SSRMS have different function, such as the shoulder responsible for orientation of the arm plane, the elbow responsible for locating the tip and the wrist responsible for orientating the tip. The workspaces of the shoulder and the wrist are small, when the elbow pose is fixed. The collision always happens in the wrist and adjacent links a_3, a_4, so they are considered as objects of obstacles avoidance in this paper. Because of the complex outlines of link bodies and obstacles, this paper adopts a minimum-volume enclosing method which can achieve the obstacle avoidances simply and efficiently and do not deteriorate the real-time performance. Briefly speaking, the method lumps the complex surface of the object into the single and regular geometry intuitively.

For the obstacles, the Lowner-John ellipsoid method is adopted to formulate the obstacle model establishment as a convex optimization problem [33]. In this paper, we apply $\varepsilon^m(\boldsymbol{c}, \boldsymbol{C})$ to represent the Lowner-John ellipsoid, where m is the dimension and $\boldsymbol{c}, \boldsymbol{C}$ are the center coordinate and the characteristic matrix of the Lowner-John ellipsoid respectively. To simplify the obstacle model, three main axes of the Lowner-John ellipsoid are constrained to be the same length and the ellipsoid degenerated into a sphere, denoted as,

$$\varepsilon^3(\boldsymbol{p^b}, \boldsymbol{\Upsilon}), \tag{42}$$

where $\boldsymbol{p^b}$ is the position of the sphere center and $\boldsymbol{\Upsilon}$ is a diagonal matrix. The computation of $\varepsilon^3(\boldsymbol{p^b}, \boldsymbol{\Upsilon})$ can be found in reference [33].

For the manipulator, a cube is employed to cover the objects of obstacles avoidance, shown as Fig.7. The cube's subfaces are parallel with the arm plane and one of them is composed of O_3O_5' and the corresponding height in $\triangle O_3O_4'O_5'$. The distance between the two subfaces is a_4. The proposed model can avoid solving the inverse cosine function during iterations and improve the real-time performance.

The obstacles and manipulator bodies have been simplified as regular geometries. The obstacles avoidance can be regarded that the arm plane keeps enough distances from each sphere when the arm plane moves following the manipulator simultaneously. The distance between the i^{th} sphere and the arm plane is denoted as d_i

$$d_i = \frac{\boldsymbol{p_{34}} - \boldsymbol{p_{45}}}{|\boldsymbol{p_{34}} - \boldsymbol{p_{45}}|}\boldsymbol{p^b}, \tag{43}$$

and the safety distance is denoted as d_s,

$$d_s = (\xi_0 + \xi), \tag{44}$$

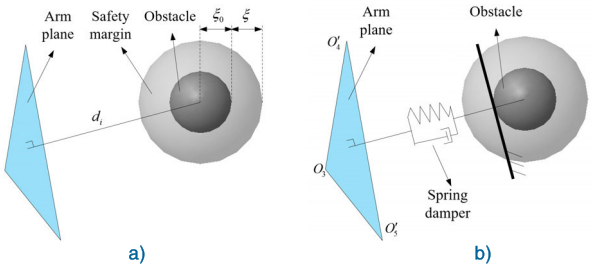


FIGURE 8. a) Model of the obstacle avoidance. b) The equivalent model by a spring damper.

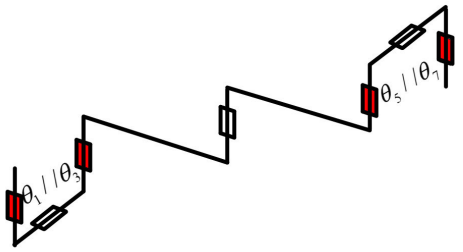


FIGURE 10. The gimbal lock happens when $\theta_1//\theta_3$ or $\theta_5//\theta_7$. The freedom degree is lost when the gimbal lock happens.

where ξ_0 is the collision distance and ξ is the safety margin, shown as Fig.8. According to (41), if $d_i \geq d_s$, the additional task does not work, that is

$$X_2 = 0; \tag{45}$$

If $d_i \leq \xi_0$, the solution is infeasible and the additional task is defined as

$$X_2 = INF, \tag{46}$$

else that

$$X_2(d_i) = 1/(d_i - \xi_0)^2 - 1/(\xi - \xi_0)^2, \tag{47}$$

where $INF = +\infty$ and is assigned as a large value. The performance criterion of X_2 is shown in Fig.9.

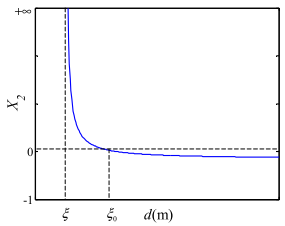


FIGURE 9. Fitness value of X_2 with varying distance between the obstacle and $\triangle O_3O_4'O_5'$.

B. SINGULARITY AVOIDANCE

The singularity avoidance can improve the manipulability and motion stability, so it is significant to analyze the singularity. There are two kinds of singularities totally including the algorithm singularity and the configuration singularity.

1) CONFIGURATION SINGULARITY

SSRMS-type manipulator is composed of three parts including the shoulder, the elbow and the wrist. In the shoulder part, three orthogonal rotation axes are composed to achieve the function of the sphere joint. While two of them are parallel with each other, the shoulder loses one degree of freedom and this phenomenon is called as ''gimbal lock'' [34], shown as Fig.10. The same problem also happens to the wrist part. In addition, the algorithm generates singularity when $\boldsymbol{e}_o$ is parallel with $\boldsymbol{e}_b$ or $\boldsymbol{e}_e$, because there are infinite solutions to (2) and (3). This paper adopts θ_o, θ_e rather than traditional ''manipulability'' [28] to indicate the singular degree of the manipulator and

$$\theta_o = < \boldsymbol{e}_b, \boldsymbol{e}_o >,$$
$$\theta_e = < \boldsymbol{e}_e, \boldsymbol{e}_o > .$$

To avoid the gimbal lock and the algorithm singularity, $\boldsymbol{e}_o$ need to be far from $\boldsymbol{e}_b$ and $\boldsymbol{e}_e$, so the third additional task is introduced

$$X_3 = 1/\theta_o + 1/\theta_e. \tag{48}$$

However, the computation of angles weakens the efficiency owing to solving the inverse cosine function. Actually, X_3 works in the fitness function (41) only if θ_o or θ_e is close to zero. So, (48) is simplified as

$$X_3 = \frac{1}{|\boldsymbol{e}_b - \boldsymbol{e}_o|} + \frac{1}{|\boldsymbol{e}_e - \boldsymbol{e}_o|}. \tag{49}$$

Furthermore, the optimal $\boldsymbol{e}_o$ is to be perpendicular to $\boldsymbol{e}_b$ and $\boldsymbol{e}_e$ simultaneously so the fourth is defined as

$$X_4 = [(\boldsymbol{e}_b + \boldsymbol{e}_e) \cdot \boldsymbol{e}_o]^2 . \tag{50}$$

2) ALGORITHM SINGULARITY

According to (2), (3), when $\boldsymbol{e}_b$ is parallel with $\boldsymbol{e}_e$, it can be derived

$$z_2 = \pm z_6. \tag{51}$$

When the singularity happens, the arm plane can rotate freely around $\overrightarrow{O_3O_5}$. All arm planes have the same fitness value and the optimization algorithm is invalid. For this problem, (40) is adopted to be added to the (41) to determine the unique EO and avoid configuration jumps.

C. WRAPPING UP

Above all, this section shows the optimal method of obstacles avoidance and singularity avoidance. These requirements are formulated as the additional tasks X_1, X_2, X_3, X_4 listed in (40), (46), (49), (50) respectively. Each task plays different role in the fitness function (41). X_1 is used to improve the dynamic performance of the manipulator motion including smoothing the trajectory and eliminating vibrations. X_2 is activated to specify a new steady EO by regulating the objective value of the fitness function when the manipulator oversteps the safety margin. And then the manipulator is led to

reach to the updated EO. X_3 is activated to deteriorate the objective value only if the gimbal lock happens. When X_2, X_3 do not work, X_4 can improve the manipulability by regulating the EO to approach to the optimum but the regulating effect depends on the weight λ_4. The fitness function (41) combines four additional tasks with four weights to find the optimal solution. The flow framework and more details of the inverse kinematic algorithm for the SSRMS-type manipulator are shown as Fig.11.

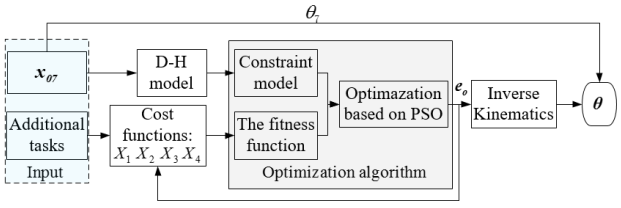


FIGURE 11. The inverse kinematic algorithm for 7-dof redundant robots based on a hybrid of analytical and numerical method. θ_7 can be computed directly by x_{07}; Optimization algorithm flow is shown in Fig.12.

The flow diagram of the optimization algorithm based on PSO is shown in Fig.12.

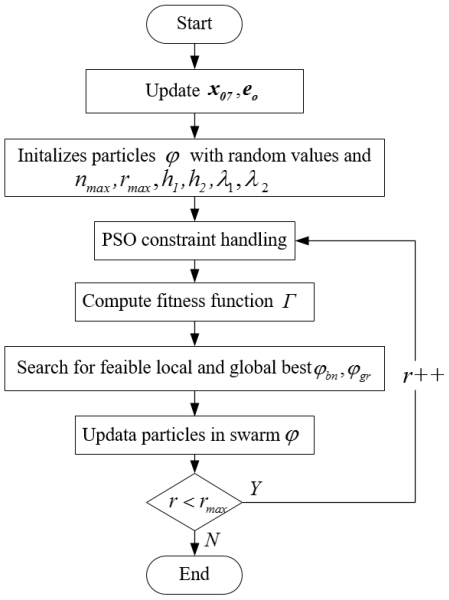


FIGURE 12. The flow diagram of the optimization algorithm.

V. NUMERICAL SIMULATIONS

To verify the availability of the proposed algorithm of the inverse kinematics, simulations are conducted on a SSRMS-type manipulator in this section. As a realistic project, the proposed algorithm is be applied to a self-developed SSRMS-type manipulator which is used in space activities including assembling, repairing, upgrading a satellite or space station. This manipulator has the same configuration as shown in Fig.1. The manipulator parameters are given as follow: $c_0 = 0.21$, $c_1 = 0.2515$, $c_2 = 0.24$, $c_3 = 1.5$, $c_4 = 0.2265$, $c_5 = 1.5$, $c_6 = 0.24$, $c_7 = 0.2515$, $c_7 = 0.6245$ and all units are meter. The involved parameters of the PSO algorithm are list as follow: $n_{max} = 15$, $r_{max} = 15$, $\varphi_{max} = 2\pi$, $\varphi_{min} = 0$, $h_1 = 1.532$, $h_2 = 1.49$, $w = 0.79$, $\lambda_1 = 0.09$, $\lambda_2 = 0.89$, $\lambda_3 = 0.01$, $\lambda_4 = 0.009$. To verify the effectiveness of the algorithm, this paper shows motion performances of the manipulator in three cases including Free Motion, Static Avoiding Obstacles and Dynamic Avoiding Obstacles. In these cases, the manipulator end-effector is commanded to move from its initial pose $\boldsymbol{x}_{\mathbf{07}}^{\boldsymbol{ini}} = \left[\boldsymbol{p}_{\mathbf{07}}^{ini}, \boldsymbol{\eta}_{\mathbf{07}}^{ini}\right] = [0.5969, 0.4214, -1.7, 0, 0.53, -0.848, 0]$ to the target pose, where the corresponding configuration of $\boldsymbol{x}_{\mathbf{07}}^{ini}$ is $\boldsymbol{\theta}^{ini}$ and $\boldsymbol{\theta}^{ini} = [0.23, 1.57, 0.66, -2.41, 0.18, -1.34, 0.45]$, shown as Fig.13.

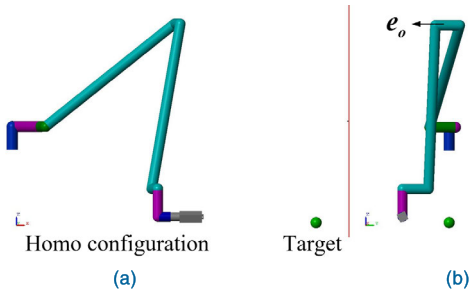


FIGURE 13. The home configuration and the target position: (a) is front view which shows home configuration of the SSRMS manipulator and the target; (b) is side view which shows EO and the target.

The motion of the target differs among three cases. In addition, the computing time of the algorithm is listed and compared with other methods proposed in [11].

A. FREE MOTION

In this case, the aim is to test the convergence, accuracy and performance of the additional multi-task model consisting of X_1, X_3, X_4. The manipulator freely moves towards the target fixed at $\boldsymbol{x}^t = \left[\boldsymbol{p}^t, \boldsymbol{\eta}^t\right] = [0.625, 0.255, -3.422, 0, 0.5299, -0.848, 0]$ without the constraints of obstacles, that is, X_2 invalid in (41). The desired poses $\boldsymbol{x}_{\mathbf{07}}^{d}$ of the end effector are generated by the linear difference between the current pose and the target pose. As this paper focuses on the inverse kinematics rather than the trajectory planning, we assume without loss of generality that the desired pose in each moment has been known. Led by the desired pose in each moment, the manipulator searches the optimal solution of the inverse kinematics in the feasible joint space.

Fig.14(a), (b) displays the dynamic configurations of the manipulator in two views and Fig.14(c), (d) display the optimal values of EO and the solutions of inverse kinematics respectively. The end effector gets to and then keep at the target at $\boldsymbol{t_s} = 35$s. Constrained by X_1, current posture is updated as the closest posture according to the optimized EO.

According to Fig.14(c), the EO variation is small and stable and that indicates the adjustment of the manipulator configuration is stable without overturns. Fig.14(d) illustrates that the

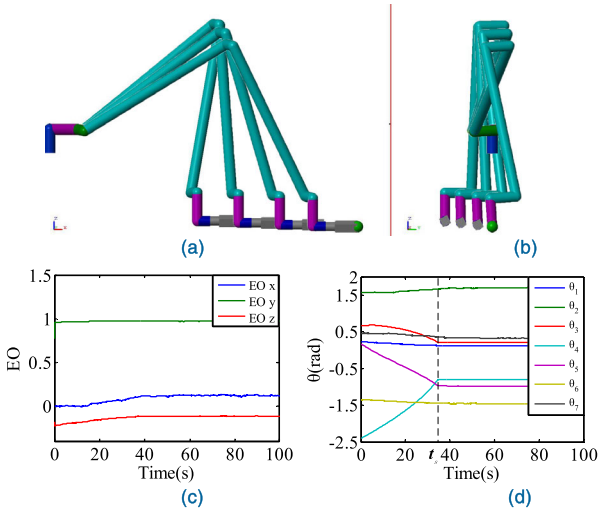


FIGURE 14. Simulation solutions of the inverse kinematics for SSRMS manipulator. (a) shows the configuration motion in front view; (b) shows the configuration motion in the side view; (c) shows optimal EO optimized by X_1; (d) shows joint inverse kinematics solution corresponding EO.

joints motion is smooth without the joint jumps. The smooth joint trajectory is vital for the manipulator to achieve dynamic control and suppress the noise of higher-order parameters, such as velocity and acceleration.

The convergence index is introduced to describe the performance of the inverse kinematics algorithm and defined as

$$c = \frac{\Gamma - \Gamma^{opt}}{\Gamma^{ini} - \Gamma^{opt}}, \tag{52}$$

where Γ^{ini} is the initial fitness evaluation and Γ^{opt} is the global optimal fitness evaluation. Equation (52) is a feature scaling technique which normalize the fitness evaluation as convergence indexes which ranges from 1 to 0 corresponding from the initial value to the steady value [35]. The technique does not change the convergence tendency of fitness evaluations. The convergence index represents the velocity of a fitness evaluation determined by a randomly given EO approaching to the optimal fitness evaluation. During the 100s simulation, the inverse kinematics algorithm executes totally 50000 optimizations, generates 5000 local optimal fitness values and picks out 1000 optimal solutions. The local optimal fitness values are normalized as convergence indexes according to (52) and collected to make a statistic, shown as Fig.15. According to this figure, the convergence index after the 12^{th} generation is within 0.2%, so the inverse kinematic algorithm is of good fast convergency.

The position error of the proposed inverse kinematics is defined as

$$\Delta \boldsymbol{e}_p = \boldsymbol{p}_{07}^{d} - \boldsymbol{p}_{07}^{e}, \tag{53}$$

where $\boldsymbol{p}_{07}^{d}$ is the desired position and $\boldsymbol{p}_{07}^{e}$ is the executive position of the end effector. The error between the desired orientation $\boldsymbol{\eta}_{07}^{d}$ and the executive orientation $\boldsymbol{\eta}_{07}^{e}$ is

$$\Delta \boldsymbol{\eta} = \boldsymbol{\eta}_{07}^{d} \otimes \boldsymbol{\eta}_{07}^{e\,*}.$$

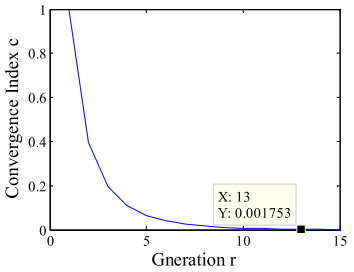


FIGURE 15. The convergence index of the local fitness evaluations.

However, the error is a quaternion which indicates the coupling orientation relationship between two frames. To express the orientation error intuitively, the axis angle $\Delta\eta_0$ is extracted to be defined the orientation error

$$\Delta e_\eta = arccos(\Delta\eta_0). \tag{54}$$

Although the orientation error loses the information of the rotation axis, it can represent the error magnitude between two frames. The position error and the orientation error are shown in Fig.16(a) and Fig.16(b) respectively. According to the figures, we can know that $\Delta e_{px_{max}} = 0.257\ mm$, $\Delta e_{py_{max}} = 2\times10^{-12} mm$, $\Delta e_{pz_{max}} = 0.006\ mm$, $\left|\Delta \boldsymbol{e_p}\right|_{max} = 0.257\ mm$ and $\Delta e_{\eta_{max}} = 1.19\times10^{-12^\circ}$. For the position, the error mainly happens in the period of dynamic approaching, that is $0 \sim t_s$. During this period, the inverse kinematics algorithm with fixed parameters is employed to deal with varying conditions, so the steady solution is always varying and needs to be searched again starting from the previous solution. However, during the period of the fixed pose, that is $t_s \sim 100$s, the searching initial value is approximately same as the steady solution, so the steady solution can be updated more precisely on the basis of the previous searching solution. For the orientation, similarly, owing to the little variation of end effector posture, the steady solution is updated based on the accurate initial value and hardly varies.

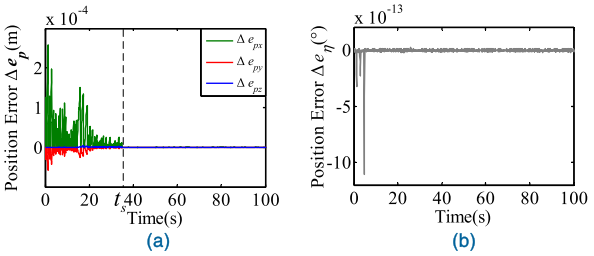


FIGURE 16. Errors of the position and orientation. (a) shows component error of the position; (b) shows the angle error between the target normal vector and z_7.

B. DYNAMIC AVOIDING OBSTACLES

In this case, the multi-task model consisting of X_1, X_2, X_3, X_4 is employed to avoid obstacles dynamically. During the obstacle avoidance process, the manipulator moves towards the dynamic target. The trajectory of the dynamic target is

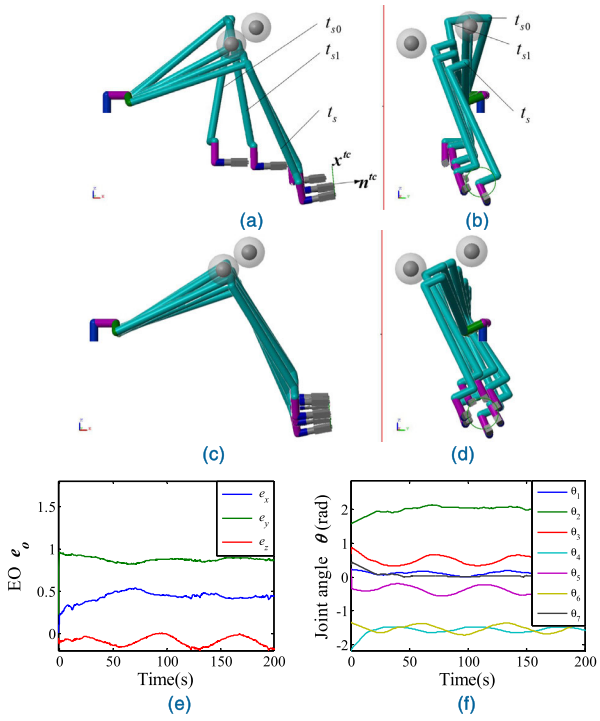


FIGURE 17. Simulation solutions of the inverse kinematics for SSRMS manipulator. (a), (b) show the configurations of free motion t_{s0}, obstacles avoidance t_{s1} and capturing the target t_s in two views; (c), (d) shows the varying configurations with the moving target; (f) shows joint inverse kinematics solutions corresponding EO shown as (e); $t_{s0} = 0.5s$; $t_{s1} = 10s$; $t_s = 24.9s$.

defined as $\boldsymbol{x}^{\boldsymbol{t}}(t) = \left[\boldsymbol{p}^{\boldsymbol{t}}(\boldsymbol{t}), \boldsymbol{\eta}^{\boldsymbol{t}}\right]$. The shape of the dynamic target is a circle located in $\boldsymbol{x}^{\boldsymbol{tc}} = [0.8260, 0, -2.799]$ and orientated towards $\boldsymbol{n}^{\boldsymbol{tc}} = [-0.0995, 0, -0.9950]$. The position of the trajectory, $\boldsymbol{p}^{\boldsymbol{t}}(t)$

$$\begin{aligned} p_x^t &= -0.1760cos\left(f_\omega t\right) + 0.8260, \\ p_y^t &= -0.1769sin\left(f_\omega t\right), \\ p_z^t &= -0.0176cos\left(f_\omega t\right) - 2.7398, \end{aligned} \tag{55}$$

where $f_\omega = 0.0088$. The orientation of the trajectory $\boldsymbol{\eta}^{\boldsymbol{t}} = \text{dcm2quat}([[\boldsymbol{e_b} \times \boldsymbol{n}^{\boldsymbol{tc}} \times \boldsymbol{n}^{\boldsymbol{tc}}]], [[\boldsymbol{e_b} \times \boldsymbol{n}^{\boldsymbol{tc}}]], \boldsymbol{n}^{\boldsymbol{tc}}])$. Obstacles are fixed in $\boldsymbol{x}^{b1} = [-1.2297, 0.1267, -1.205]$ and $\boldsymbol{x}^{b2} = [-1.5297, 0.1067, -1.505]$ respectively and they are set on the way of freely moving trajectory. Compared with existing methods [1], [3], [11] obstacles are enough. Both safety margins are the same $\xi^1 = \xi^1 = 0.05$ and both collision distances $\xi_0^1 = \xi_0^2 = a_4/2$. The visual model and the approaching process are shown from two views as Fig.17. The manipulator freely moves during $0s \sim t_{s0}$ and then is optimized to avoid obstacles by the multi-tasks model consisting of X_1, X_2, X_3, X_4. During $t_{s0} \sim t_{s1}$, the optimal configuration is searched to lead the manipulator to avoid collisions and achieve minimum orientation variation. In this period, EO is adjusted continuously to lead the manipulator to get to the optimal configuration. During $t_{s1} \sim t_s$, the manipulator is approaching the initial position of the target trajectory. After that to $200s$, the manipulator tracks the trajectory and avoids obstacles simultaneously.

Notice that, the avoidance obstacle task X_4 does not work in each moment, it only works when the manipulator is beyond the safety margins of obstacles. During the whole simulation, the inverse kinematics algorithm executes totally 450000 optimizations, generates 30000 local optimal fitness values and picks out 2000 optimal solutions. Among them, there are 182 optimal solutions obtained by executing the avoidance obstacle task, so the whole process consists of the free motion and the avoidance obstacles. Their distributions are shown by the step function in Fig.18(a), where "0" represents the avoidance obstacles and "1" represents the free motion. The convergence indexes of two cases are figured out by (52) respectively and separately shown in Fig.18(b). According to (45), the fitness value extremely deteriorates when the collision happens, so the convergence index of the avoidance obstacles c_{oa} converges slower than the convergency index of the free motion c_{fm}. The slow convergence results in the low accuracy of the end effector and meanwhile the varying convergence accuracy results in vibration. The tracking error of the end effector are shown in Fig.19. According to the Fig.19 (a), the position error $\Delta e_{px}^{max} = -0.691mm$, $\Delta e_{py}^{max} = 0.109\ mm$, $\Delta e_{pz}^{max} = 0.343\ mm$, $\left|\Delta \boldsymbol{e_p}\right|_{max} = 0.731mm$. According to Fig.19(b), $\Delta e_{\eta}^{max} = 0.059°$. By analysis in Section IV-A, the accuracy of dynamic searching is lower than the stastic searching. Because the orientation of the end effector varies all the time, the orientation error of the avoidance obstacles is bigger compared with the free motion in Section IV-A. The error distributions are shown in Fig.20.

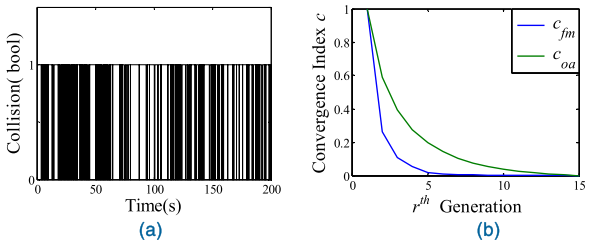


FIGURE 18. (a) shows the enabling frequency of the additional task X_2. (b) shows the convergence index of the local fitness evaluations, where c_{fm} is convergence index of the free motion and c_{oa} is the convergence index of the obstacles avoidance.

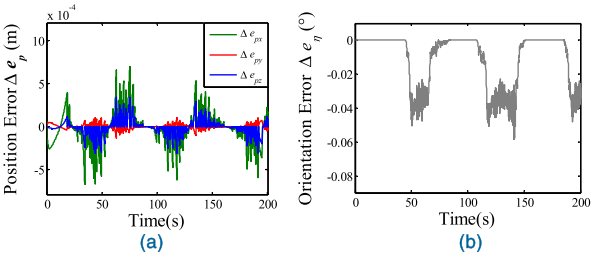


FIGURE 19. Errors of the position and orientation. (a) shows component error of the position; (b) shows the angle error between the target normal vector and z_7.

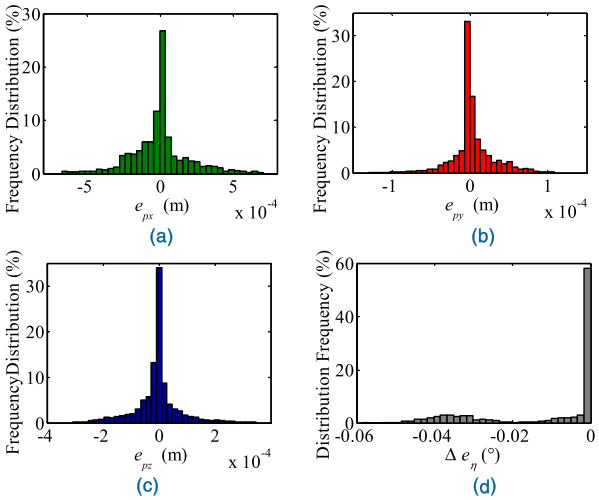


FIGURE 20. Frequency distributions of position errors and orientation errors. The position errors, shown as Fig (a), (b), (c) are distributed normally. Most errors of position and orientation distribute near zero.

The position error components e_{px}, e_{py}, e_{pz} display normal distributions whose standard deviations are small. About the orientation error, the frequency reduces quikly with the increasing of the error. The mean position error is 0.1315 *mm* and the mean orientation error is $-0.011°$.

The manipulability indicates the kinematics performance of the manipulator and contains translational and rotational velocity information. This paper adopts the Yoshikawa's method [36] to define the manipulability and it is shown in Fig.21. X_3, X_4 computed by (49), (50) as shown in Fig.21. Comparing these three variables in Fig.21, we can conclude that the manipulability is correlated positively with X_3 and negatively with X_4. According to Fig.21, the manipulability is correlated with the linear combination of X_3, X_4 and optimations to X_3, X_4 improve manipulability. The manipulator keeps resonable configurations during the whole process of approaching and tracking.

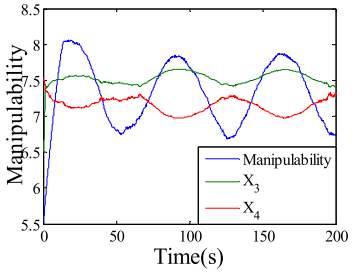


FIGURE 21. The Manipulability and the fitness values of additional tasks X_3, X_4. The manipulability is correlation with X_3, X_4.

C. ANALYSIS OF EFFICIENCY AND ACCURACY

The proposed inverse kinematics algorithm (IK-HAN) is designed in the Microsoft VS2010 software. The procedure is executed on the Win7 operating systems which is installed in a computer with an Inter(R) Core (TM)i5-6300HQ CPU 2.30GHz and Quad CPU kernels. The average consuming time of the proposed algorithm is 2.917ms measured by Multimedia Timer. It is vital for the proposed algorithm to be integrated into the underlying hardware, such as DSP or FPGA, to achieve the real-time kinematics control.

1) EFFICIENCY AND ACCURACY

According to past researches, the inverse kinematics for 7-dof manipulators can be classified in 4 categories including IK based on AI (IK-AI) [10], closed-loop optimization (IK-CLO) [1], [11], [12], Jacobian s pseudo-inverse (IK-JPI) [3], [37] and attaching constraints. According to the reference [37], the solution of the inverse kinematic problem in the velocity level has the following form

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{+}(\boldsymbol{\theta})\left[\dot{\boldsymbol{x}}^{\boldsymbol{d}} + K_p(\boldsymbol{x_d} - \boldsymbol{x})\right] + \left[\boldsymbol{I} - \boldsymbol{J}^{+}(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})\right]\dot{\boldsymbol{\theta}}_{\boldsymbol{0}}. \quad (56)$$

where $\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{x_d}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \boldsymbol{J}, \boldsymbol{J}^{+}$ denote the vector of joint angles, the position of the end-effector, the desired Cartesian point, the desired velocity of the end-effector, Jacobian matrix and Jacobian's pseudo-inverse, respectively, and $K_p > 0$. The $\dot{\theta}$ denotes as an arbitrary vector in the null space of the Jacobian matrix and to avoid joint angle limits, it is given as

$$\dot{\boldsymbol{\theta}} = \frac{\partial \vartheta}{\partial \boldsymbol{\theta}}, \quad \partial(\boldsymbol{\theta}) = \left\| \frac{\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}}{\boldsymbol{\theta}_M - \boldsymbol{\theta}_m} \right\|_p, \ i = 1, \ldots, 7$$

where $\boldsymbol{\theta}_M$ ($\boldsymbol{\theta}_m$) is the maximum (minimum) of the joint angles and $\bar{\boldsymbol{\theta}}$ is the middle value of the joint angles. According to reference [3], a real-time inverse kinematics based on RBF-NN (IK-RBF-NN) is proposed and its output function is

$$\boldsymbol{\theta} = \sum_{i=1}^{m} w_i \exp\left(-\frac{1}{2}\left(\frac{\|\boldsymbol{i} - \boldsymbol{o_i}\|}{\sigma_i}\right)^2\right), \quad i = 1, \ldots, 7. \quad (57)$$

with the cost function

$$f = \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{H}\boldsymbol{d}$$
$$\boldsymbol{c} = \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}, \quad \boldsymbol{d} = \boldsymbol{\Delta w}$$

where $\boldsymbol{i}, \boldsymbol{o}$ is the input vector, the output vector of the hidden layer and $\boldsymbol{w}, \boldsymbol{H}$ is the connecting weight and an arbitrary positive-definite matrix. The algorithm block diagram is shown in Fig.22 a). According to the reference [1], an adaptive close-loop system is proposed and its output function is

$$\boldsymbol{X}(k+1) = h(\boldsymbol{X}(k)) + g(\boldsymbol{X}(\boldsymbol{k}))\boldsymbol{U}(k), \quad (58)$$

with the cost function

$$f = \frac{1}{2}\sum_{k=0}^{\infty}(\boldsymbol{X}^{\boldsymbol{T}}(k)\boldsymbol{Q}\boldsymbol{X}(k) + \boldsymbol{U}^{T}(k)\boldsymbol{R}\boldsymbol{U}(k))$$
$$\boldsymbol{U}(k) = -\boldsymbol{R}^{-\boldsymbol{1}}g^{T}(X(k))\lambda^{*}(k+1)$$

where $\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{Q}, \boldsymbol{R}$ is the state, input, positive semidefinite matrix, positive-definite matrix respectively and $\boldsymbol{\lambda}^{*}$ is the optimal costate vector of the discrete-time system. The algorithm block diagram is shown in Fig.22 b).

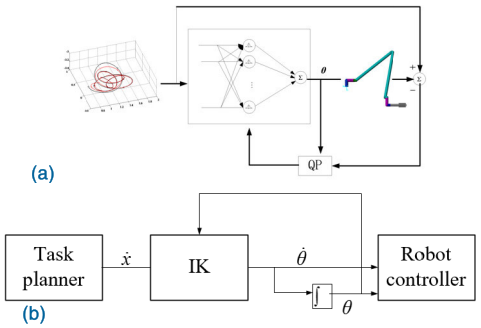


FIGURE 22. (a) The algorithm block diagram of IK-AI; (b) The algorithm block diagram of IK-CLO.

Then comparations are made as below between above three methods and the proposed inverse kinematics in this paper. The efficiency is defined as the consuming time of calculating the inverse solution between adjacent points in the planned trajectory. So that different planned trajectory will not affect the efficiency of the algorithm. The simulation results in [11] show the ability of IK-JPI that the efficiency is 3.6ms and the accuracy is 29mm. For IK-CLO in [3], the efficiency is at least 30ms and the accuracy is 1mm. Similarly, another method of IK-CLO in [1], the efficiency can be raised up to 75% of IK-JPI and the accuracy is 25mm. For IK-AI in [10], an RBF-NN is employed to solve the inverse kinematics in real time and consumes at least 100ms to calculate and the minimum position error is 3.41mm.

Above all, the analytical method performs well in efficiency and the numerical method does well in inverse solutions of the redundant manipulators. However, one can conclude that the accuracy is opposite to the efficiency and it is difficult to improve them simultaneously except adopting a hybrid of analytical and numerical method proposed in this paper.

2) COMPLEXITY ANALYSIS

According to computational complexity rules, we can figure out the complexity of IK-JPI, IK-CLO and proposed IK-HAN as following.

There are three sections in the complexity of IK-HAN. First, the computation of EO vector requires $N_{H1}n_H$ order flops for computing (1) $\sim$ (12), where $n_H = 3$ and N_{H1} is the number of firing rules. Owing to calculating trigonometric functions in the inverse kinematics, the local complexity will get up to $N_{H2}n_H\log(n_H)$, where N_{H2} is the calculation number of trigonometric functions [38]. The last part of complexity for optimization is $N_{Hd}N_{Hg}n_{AA}$, where $N_{Hd} = 10$, $N_{Hg} = 5$ are the iteration number and the population size of PSO and $n_{AA} = 1$ is the dimension of the azimuth angle. So, the complexity of IK-HAN is

$$(n_H \log(n_H), n_{AA}) = N_{H1}n_H + N_{H2}n_H \log(n_H) + N_{Hd}N_{Hg}n_{AA}, \quad (59)$$

For IK-CLO in [3], the computation of inverse solutions (X-2) requires at least $N_c(n+2n^2)$ order flops, where $n = 7$ is the degree of freedom and N_C is the number of optimizations.

TABLE 2. Computational performance.

Method	*Accuracy* *Position /mm*	*Efficiency /ms*	*Complexity*
IK-HAN	*0.731* *(0.059° for Orientation)*	2.917	$o(n_H)$
IK-JPI	*29*	*3.6*	$o(n^3)$
IK-CLO	*1*	*>30*	$o(n^2)$
RBF-NN	3.41	>100	~

Owing to the bad efficiency of RBF-NN, its complexity is not be compared.

So, the complexity of IK-CLO is

$$o_{CLO}(n^2) = N_c(n + 2n^2), \quad (60)$$

IK-JPI [11] involves computation of the minimum norm motion for the primary task and the self-motion for accomplishing the additional task. The redundancy resolution requires computation of Moore Penrose pseudoinverse. The computation of Moore Penrose pseudoinverse involves singular value decomposition (SVD) and matrix multiplications for inverse computation. The total computational cost of IK-JPI is

$$o_{JPI}(n^3) = N_J(3mn^2 + n^3 + 1.5m^2n - 0.5mn), \quad (61)$$

where m = 6 is the dimensions of the workspace and N_J is the number of corrections.

With the increment of degrees of freedom, the computational requirement increases rapidly for IK-CLO and IK-JPI while it is approximate linear for IK-HAN, which makes it a better approach for real-time implementation. More comparations about efficiency and accuracy of above four methods are listed in the Table 2.

The azimuth angle is optimized by the PSO in the constraints model in real time. It will be shown that the computational requirement of the proposed inverse kinematics is low since the 3-dimensional optimization for the EO is transformed as the 1-dimesional optimization for the azimuth angle.

VI. CONCLUSION

This paper has proposed a novel inverse kinematic algorithm for SSRMS-type 7-DOF redundant robots based on a hybrid of analytical and numerical method. The algebraic relationship between the joints and EO is studies. The proposed algorithm transforms the inverse kinematics problem as the optimization model of EO and the additional tasks are formulated as fitness evaluation functions. To improve the algorithm efficiency, EO is simplified as the unary function of the azimuth angle according to the geometric constraints and is optimized by employing PSO. With the algebraic relationship, the optimal joints solutions are obtained in real time. The simulations are executed on two cases and shows that the proposed algorithm performs well in accuracy, efficiency, avoiding obstacles, avoiding singularities, improving

manipulability. The proposed algorithm also can be applied to SRS-type manipulators. The further study will be required for the joint limits.

REFERENCES

[1] P. K. Patchaikani, L. Behera, and G. Prasad, "A single network adaptive critic-based redundancy resolution scheme for robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3241–3253, Aug. 2012.

[2] D. N. Nenchev, Y. Tsumaki, and M. Takahashi, "Singularity-consistent kinematic redundancy resolution for the S-R-S manipulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Apr. 2004.

[3] M. M. Schill and M. Buss, "Kinematic trajectory planning for dynamically unconstrained nonprehensile joints," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 728–734, Apr. 2018.

[4] K. Yoshida, K. Hashizume, and S. Abiko, "Zero reaction maneuver: Flight validation with ETS-VII space robot and extension to kinematically redundant arm," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nov. 2001.

[5] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Prog. Aerosp. Sci.*, vol. 68, no. 8, pp. 1–26, Jul. 2014.

[6] H. Kim, L. M. Miller, N. Byl, G. Abrams, and J. Rosen, "Redundancy resolution of the human arm and an upper limb exoskeleton," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 6, pp. 1770–1779, Jun. 2012.

[7] P. Berthet-Rayne, K. Leibrandt, G. Gras, P. Fraisse, A. Crosnier, and G.-Z. Yang, "Inverse kinematics control methods for redundant snake-like robot teleoperation during minimally invasive surgery," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2501–2508, Jul. 2018.

[8] T. Fung Chan and R. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.

[9] A. Reiter, A. Muller, and H. Gattringer, "On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1681–1690, Apr. 2018.

[10] H. Toshani and M. Farrokhi, "Real-time inverse kinematics of redundant manipulators using neural networks and quadratic programming: A Lyapunov-based approach," *Robot. Auto. Syst.*, vol. 62, no. 6, pp. 766–781, Jun. 2014.

[11] J. Xiang, C. Zhong, and W. Wei, "General-weighted least-norm control for redundant manipulators," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 660–669, Aug. 2010.

[12] S. Huang, J. Xiang, W. Wei, and M. Z. Q. Chen, "On the virtual joints for kinematic control of redundant manipulators with multiple constraints," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 65–76, Jan. 2018.

[13] Y.-J. Chen, M.-Y. Ju, and K.-S. Hwang, "A virtual torque-based approach to kinematic control of redundant manipulators," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1728–1736, Feb. 2017.

[14] P. Corke, *Robotics, Vision and Control—Fundamental Algorithms in MATLAB*. Berlin, Germany: Springer, 2011.

[15] K. Kreutzdelgado, M. Long, and H. Seraji, "Kinematic analysis of 7 DOF anthropomorphic arms," in *Proc. IEEE Int. Conf. Robot. Automat.*, Dec. 1990.

[16] M. Gong, X. Li, and L. Zhang, "Analytical inverse kinematics and self-motion application for 7-DOF redundant manipulator," *IEEE Access*, vol. 7, pp. 18662–18674, 2019.

[17] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1131–1142, Oct. 2008.

[18] E. L. P. Laryssa, O. Layi, O. Marius, K. Nara, and L. Aris, "International space station robotics: A comparative study of ERA, JEMRMS and MSS," in *Proc. 7th ESA Workshop Adv. Spac eTechnologies Robot. Autom.*, Noordwijk, The Netherlands, 2002, pp. 1–8.

[19] W. Xu, L. Yan, Z. Mu, and Z. Wang, "Dual arm-angle parameterisation and its applications for analytical inverse kinematics of redundant manipulators," *Robotica*, vol. 34, no. 12, pp. 2669–2688, Dec. 2016.

[20] C. D. Crane, J. Duffy, and T. Carnahan, "A kinematic analysis of the space station remote manipulator system (SSRMS)," *J. Robot. Syst.*, vol. 8, no. 5, pp. 637–658, Oct. 1991.

[21] W. Xu, Y. She, and Y. Xu, "Analytical and semi-analytical inverse kinematics of SSRMS-type manipulators with single joint locked failure," *Acta Astronautica*, vol. 105, no. 1, pp. 201–217, Dec. 2014.

[22] J. Xu, Z. Liu, Q. Cheng, Y. Zhao, Y. Pei, and C. Yang, "Models for three new screw-based IK sub-problems using geometric descriptions and their applications," *Appl. Math. Model.*, vol. 67, pp. 399–412, Mar. 2019.

[23] Y. Zhang, X. Lv, Z. Li, Z. Yang, and K. Chen, "Repetitive motion planning of PA10 robot arm subject to joint physical limits and using LVI-based primal–dual neural network," *Mechatronics*, vol. 18, no. 9, pp. 475–485, Nov. 2008.

[24] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 752–759, Feb. 2004.

[25] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, Jun. 2017.

[26] A. Colome and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 944–955, Apr. 2015.

[27] K. Tchon, "Optimal extended jacobian inverse kinematics algorithms for robotic manipulators," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1440–1445, Dec. 2008.

[28] A. Q. L. Keemink, G. Van Oort, M. Wessels, and A. H. A. Stienen, "Differential inverse kinematics of a redundant 4R exoskeleton shoulder joint," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 817–829, Apr. 2018.

[29] L. Jin, Y. Zhang, and S. Li, "Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2615–2627, Dec. 2016.

[30] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "OpenSoT: A whole-body control library for the compliant humanoid robot COMAN," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 6248–6253.

[31] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Predictive inverse kinematics for redundant manipulators with task scaling and kinematic constraints," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 278–285, Feb. 2019.

[32] R. C. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proc. Mhs95 6th Int. Symp. Micro Mach. Hum. Sci.*, 2002.

[33] E. Rimon and S. P. Boyd, "Obstacle collision detection using best ellipsoid fit," *J. Intell. Robot. Syst.*, vol. 18, no. 2, pp. 105–126, 1997.

[34] E. G. Hemingway and O. M. O'Reilly, "Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments," *Multibody Syst. Dyn.*, vol. 44, no. 1, pp. 31–56, Sep. 2018.

[35] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognit. Lett.*, vol. 22, no. 5, pp. 563–582, Apr. 2001.

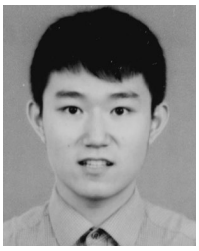
[36] P. H. Chang, "Analysis and control of robot manipulators with kinematic redundancy," Massachusetts Inst. Technol., Cambridge, MA, USA, 1988.

[37] J. Wang, S. Zhu, and Y. Gong, "Resolution enhancement based on learning the sparse association of image patches," *Pattern Recognit. Lett.*, vol. 31, no. 1, pp. 1–10, Jan. 2010.

[38] P. R. Brent, "Fast multiple-precision evaluation of elementary functions," *J. ACM*, vol. 23, no. 3, pp. 242–251, 1976.

MINGHE JIN received the B.S., M.S., and Ph.D. degrees from the College of Mechanical and Electronic Engineering from the Harbin Institute of Technology, Harbin, China, in 1993, 1996, 2000, respectively

He was with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, where he was a Research Associate. Since 2009, he has been a Professor with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His current research interests include redundant manipulators planning, dual-arm cooperative control, and electronic simulators.



QIANG LIU received the B.S. from the College of Mechanical Engineering and Automation, Northeast University, Shenyang, China, in 2014, and the M.S. from the College of Mechanical and Electronic Engineering, Harbin Institute of Technology, Harbin, China, in 2016, where he is currently pursuing the Ph.D. degree in control engineering of robotics and system.

His current research interests include multibodies dynamics, dual-arms cooperative control, and optimal control for multimanipulators.

BIN WANG received the B.S., M.S., and Ph.D. degrees from the College of Mechanical and Electronic Engineering, Harbin Institute of Technology, Harbin, China, in 1996, 1998, 2002, respectively.

He was with the of the State Key Laboratory of Robotics and System, Harbin Institute of Technology, where he has been a Research Associate, since 2004. His current research interests include visual servoing and robotics technology.

HONG LIU received B.S. and Ph.D. degrees from the College of Mechanical and Electronic Engineering from the Harbin Institute of Technology, Harbin, China, in 1986 and 1993, respectively.

He was with the the State Key Laboratory of Robotics and System, Harbin Institute of Technology, where he was a Research Associate, from 1993 to 1995. Since 2001, he has been the Research Chief of the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His current research interests include dexterous hand and space robot technology.