


RESEARCH ARTICLE

An iterative algorithm for inverse displacement analysis of Hyper-redundant elephant's trunk robot

Feifei Yuan¹, Yongjie Zhao^{1,*} , Yongxing Zhang¹, Xingwei Zhang¹ and Xinjian Lu²

¹Department of Mechatronics Engineering, Shantou University, Shantou 515063, China and ²Guangdong Goldenwork Robot Technology Ltd, Foshan 528000, China

*Corresponding author. Email: mejzha@stu.edu.cn

Received: 11 May 2021; **Revised:** 12 February 2022; **Accepted:** 28 February 2022; **First published online:** 28 March 2022

Keywords: hyper-redundant elephant's trunk robot, hybrid mechanism, joint constraint, inverse displacement, iterative algorithm

Abstract

This paper proposes an iterative algorithm to solve the inverse displacement for a hyper-redundant elephant's trunk robot (HRETR). In this algorithm, each parallel module is regarded as a geometric line segment and point model. According to the forward approximation and inverse pose adjustment principles, the iteration process can be divided into forward and backward iteration. This iterative algorithm transforms the inverse displacement problem of the HRETR into the parallel module's inverse displacement problem. Considering the mechanical joint constraints, multiple iterations are carried out to ensure that the robot satisfies the required position error. Simulation results show that the algorithm is effective in solving the inverse displacement problem of HRETR.

1. Introduction

Traditional industrial robots perform operational tasks in a specific environment using “teaching/playback.” However, developments in robotics have led to a gradual expansion of the field from traditional fixed-point tasks to autonomous operations in unstructured environments, such as deep-sea exploration and salvage [1, 2], space shuttle inspection and maintenance [3], nuclear reactor failure detection [4, 5], and *in vivo* testing [6] in medicine and healthcare. Traditional robots struggle to meet the demands of the above applications due to their mechanical structure [7, 8], workspace [9], and flexibility [10–12]. Scholars have been inspired by physiology [13, 14] and the motion mechanics [15] of natural organisms which, when applied to the study of HRETR, offer excellent operational performance and motion flexibility, enabling operation in complex environments. HRETRs are a form of a complex redundant mechanical system. Displacement analysis for HRETRs is a challenging problem due to their coupling structure, complex mathematical model, and the cost of calculations. Therefore, providing an inverse displacement algorithm with multiple solutions is of particular importance for the development of HRETR.

Some works have been done on the inverse kinematics of robots with redundant degrees of freedom (DOFs) robots. For traditional redundant robots, numerical iterative methods based on the Jacobian matrix and associated deformations [16, 17] are typically used to solve the inverse kinematics problem, such as the Jacobian pseudo-inverse [17], Jacobian transpose [18], singular value decomposition (SVD) [19, 20], and damped least squares (DLS) [21]. Although such methods can obtain the robot's smooth motion and stable posture, the process of calculation is complicated and time-consuming due to their large matrix operations. Chirikjian and Burdick [22, 23] proposed a modal approach based on the hyper-redundant manipulator's overall geometrical characteristics, which abstracted the backbone curve into a continuous, smooth spatial curve, and used the curve fitting to obtain the inverse kinematics solutions. Kim and Chirikjian [24] put forward two solutions for the inverse kinematics of a six DOFs robot based

on biopolymer segments. One is the extended elimination method [25], which can act on the biopolymer segment directly, the other is a heuristic algorithm based on the Lie group theory. In the second method, all the requirements can be met by solving the Jacobian matrix inverse iteration, as verified by numerical examples. Focusing on a hyper-redundant elephant's trunk robot (HRETR) consisting of six 3UPS-PRU parallel models in series, Zhao et al. [26] solved the inverse kinematics problem by using a parametric backbone curve to describe the geometric characteristics of the HRETR, but with a lengthy search time. Wang and Chen [27] proposed a combined optimization method for inverse kinematics, which is efficient and insensitive to the manipulator's initial and singular configurations, but motion discontinuity can sometimes occur. Tao et al. [28] proposed a modified inverse algorithm for improved tip following motion, to solve the problems of large computation complexity, exceeding joint limits, and large configuration deviations in a hyper-redundant snake-arm robot. Results show that the improved algorithm can guarantee the bending angles and do not exceed the given range at a reduced computational cost. Aristidou and Lasenby [29, 30] proposed an algorithm named forward and backward reaching inverse kinematics to solve the anthropometric robot's inverse kinematics problem, which is efficient, simplicity, convergence speed, and with a controlled performance. Anandhanarayanan and Ordóñez [31] introduced an efficient method of solving inverse kinematics for $2n+1$ DOFs (where n is the number of joints). This method combines an iterative algorithm and a numerical method, based on the Jacobian matrix, to enhance computational efficiency. It is demonstrated with examples of obstacle avoidance and joint limit avoidance. Neppalli et al. [32, 33] proposed a novel, analytical approach to solve inverse kinematics for multisection continuum robots, implemented these algorithms in the simulation and on a prototype of continuum robot and discussed its possible applications. Kaganov and Karpenko [34] introduced an automatic control system to control each section of the manipulator and the entire structure for the multisection trunk type manipulator, and this system can be used to react to changing uncertain conditions in real time. Manfred Husty et al. [35] presented a new algebraic parameterization method to achieve a direct mathematical description between the desired motion parameters of parallel mechanisms and numerical examples for the forward and inverse kinematics were given. Saglia et al. [36] investigated a parallel mechanism with a central strut and demonstrated the advantage of introducing the actuation redundancy to eliminate singularities and to improve dexterity. Gan et al. [37] presented a kinematics model for linear-actuated symmetrical spherical parallel manipulators (LASSPMs) which had advantages in solving the forward kinematic equations. Zhao et al. [38] had made some research about HRETR. The HRETR is a form of a complex redundant mechanical system, in which the joint space dimensions are much larger than the dimensions of the operation space in mathematics. As a result, multiple solutions exist for the inverse kinematics of an HRETR, and inverse displacement is a challenging problem. This paper proposes an iterative algorithm, which transforming the HRETR's inverse displacement into the inverse displacement of parallel modules to solve the complex inverse displacement problem.

This paper's structure is as follows: Section 2 presents the mechanical configuration of an HRETR and describes the structure. In Section 3, a kinematics model of the HRETR is established and an iterative algorithm for HRETR's inverse displacement is introduced by considering the joint constraints. Section 4 provides a numerical example. Finally, the conclusions are presented in Section 5.

2. System description

The mechanical configuration of an HRETR is shown in Fig. 1. The HRETR discussed in this paper is composed of six parallel modules with two degrees of rotational freedom. Each parallel module is composed of a base platform, moving platform, central fixed-length link, and three UPS chains in which U is the universal joint, P is the active prismatic joint, and S is the spherical joint. The first parallel module's central fixed-length link is connected to its own base platform vertically and fixedly and connected to the moving platform through the universal joint. The base platform of the first parallel module is the base platform for the whole HRETR. The first parallel module's moving platform is the base platform

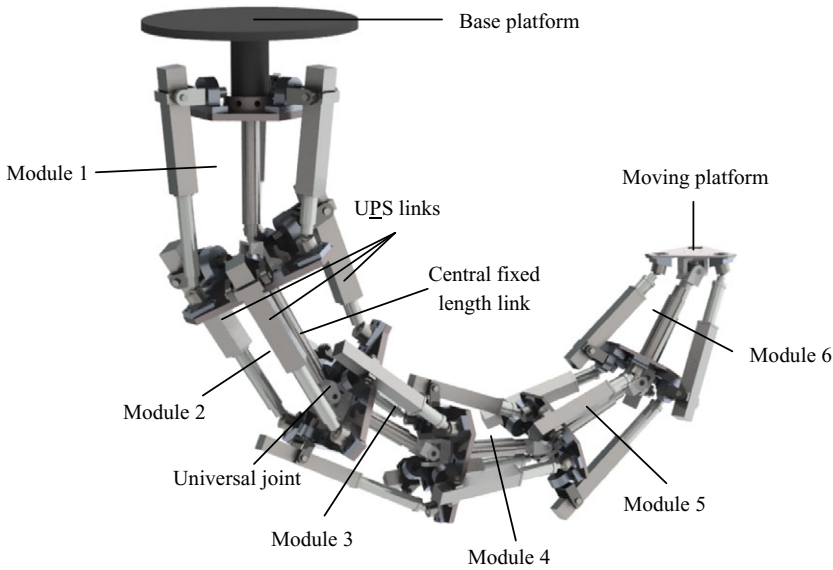


Figure 1. Hyper-redundant elephant's trunk robot (HRETR).

of the second parallel module, and the following parallel modules are connected as before. The moving platform of the last parallel module is the moving platform of the whole HRETR. The HRETR has 18 driving DOFs, and the flexible movement of the end moving platform in 3D space can be realized by coordinating and controlling each driving chain.

3. Inverse displacement model of HRETR

3.1. Inverse displacement of the parallel module

The inverse displacement model is established according to the k – th parallel module. The following coordinate systems are defined for the inverse displacement model: The reference coordinate system $O_{k-1} - x_{k-1}y_{k-1}z_{k-1}$ is attached to the center of the base platform O_{k-1} , and the moving coordinate system $O_k - x_ky_kz_k$ is located at the center of mass of the moving platform O_k . A rotation matrix ${}^{O_{k-1}}\mathbf{R}_{O_k}$ can describe the orientation of the moving platform. Thus, the rotation matrix can be obtained by:

$${}^{O_{k-1}}\mathbf{R}_{O_k} = \text{Rot}(z_{k-1}, \varphi_{zk})\text{Rot}(y'_{k-1}, \varphi_{yk})\text{Rot}(x''_{k-1}, \varphi_{xk}) \tag{1}$$

The transformation matrix ${}^k_{k-1}\mathbf{T}$, which represents the position and orientation of the moving platform for the base platform of the k – th parallel module, can be expressed as:

$${}^k_{k-1}\mathbf{T} = \begin{bmatrix} {}^{O_{k-1}}\mathbf{R}_{O_k} & \mathbf{r}_k \\ \mathbf{0} & 1 \end{bmatrix} \tag{2}$$

where \mathbf{r}_k donates the position vector of the k – th central link. Hence, the overall forward kinematics model of the HRETR can be obtained by multiplying the homogeneous transformation matrices as:

$${}^0_k\mathbf{T} = {}^0_1\mathbf{T}_2\mathbf{T} \cdots {}^{k-2}_{k-1}\mathbf{T}_i^{k-1}\mathbf{T} \tag{3}$$

where ${}^0_k\mathbf{T}$ is the homogeneous transformation matrix of the k – th moving coordinate system for the referenced coordinate system fixed on the base platform of the HRETR. The schematic diagram of the parallel module is shown in Fig. 2.

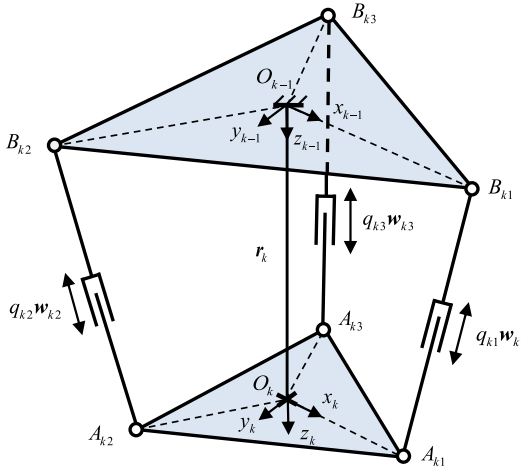


Figure 2. Schematic diagram of parallel module.

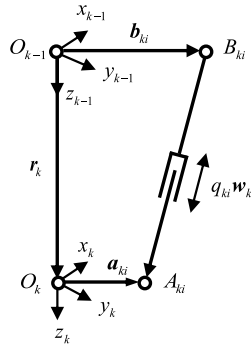


Figure 3. Vector diagram of a kinematics chain.

As shown in Fig. 3, \mathbf{a}_{ki} and \mathbf{b}_{ki} denote the position vector of the i – th connection point with the moving platform and the base platform of the k – th parallel module, respectively. The position equation associated with the i – th kinematics chain of the k – th parallel module can be described as:

$$\mathbf{r}_k + \mathbf{a}_{ki} = \mathbf{b}_{ki} + q_{ki}\mathbf{w}_{ki} \tag{4}$$

where $k = 1, 2, 3, \dots, 6, i = 1, 2, 3$.

The length associated with the i – th kinematics chain of the k – th parallel module can be written as:

$$q_{ki} = \|\mathbf{r}_k + \mathbf{a}_{ki} - \mathbf{b}_{ki}\| \tag{5}$$

Furthermore, the position vector \mathbf{r}_k can be obtained by the following inverse displacement algorithm.

3.2. Inverse displacement algorithm of HRETR

3.2.1 Algorithm description

Each of the HRETR’s parallel modules is abstracted as a geometric line segment and point model in the inverse displacement algorithm. The algorithm can be divided into forward iterations and backward iterations according to the forward approximation and inverse pose adjustment principle. The position vector \mathbf{r}_k of the central link can be obtained by the following algorithm and the inverse displacement of

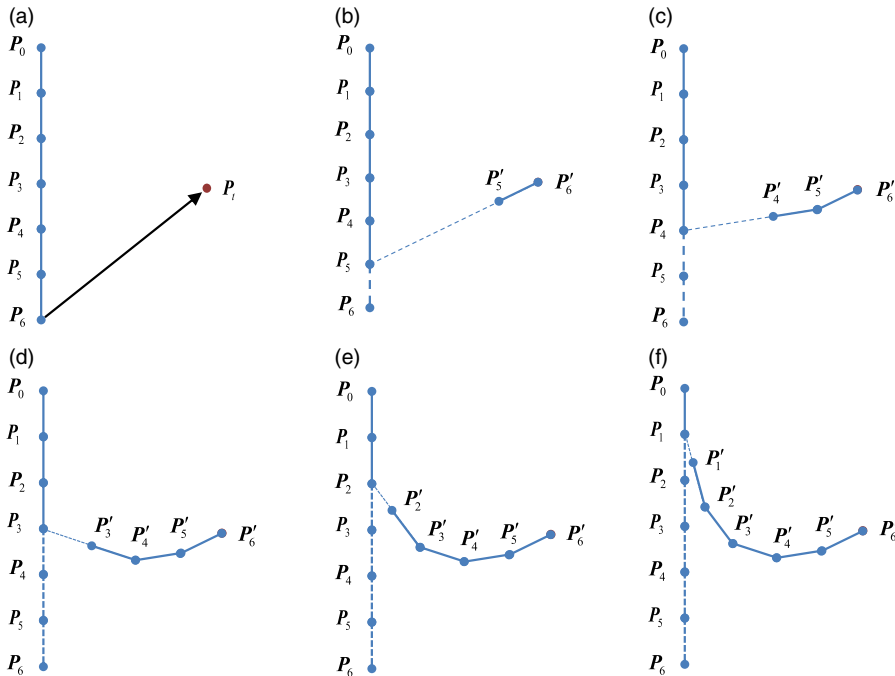


Figure 4. Forward iteration. (a) Move the moving platform P_6 to the target point P_t . (b) Find the joint P'_5 which lies on the line $\overline{P_5P_6}$. (c) Find the joint P'_4 which lies on the line $\overline{P_4P'_5}$. (d) Find the joint P'_3 which lies on the line $\overline{P_3P'_4}$. (e) Find the joint P'_2 which lies on the line $\overline{P_2P'_3}$. (f) Find the joint P'_1 which lies on the line $\overline{P_1P'_2}$.

the HRETR can be solved. In order to describe the position change process clearly, the UPS links are omitted. The forward iteration diagram and the flow chart are presented in Figs. 4 and 5.

Assuming that the initial configuration of the HRETR is at downward, as shown in Fig. 4(a), $P_0, P_1 \dots P_6$ denote the position coordinate of each of the robot's joint points: P_0 is the position of the center of the base platform in the first parallel module, P_k and P_{k-1} are the position of the center of the moving platform and base platform of the k -th parallel module, respectively, and d_k is the central fixed-length link of the k -th parallel module, as $d_k = \|P_k - P_{k-1}\|$. All central fixed-length links mentioned in this paper are of identical length d . P_t is the target point and r_t denotes the normal vector of the end platform. Since the central fixed-length link of the first parallel module is connected to its own base platform vertically and fixedly, thus, P_0 is not involved in the iterative process. The forward iteration determines whether the target point is within the maximum range that the robot can initially reach. The judgment method is as follows:

$$d_a = \sum_{k=1}^5 d_k \tag{6}$$

$$d_t = \|P_t - P_1\| \tag{7}$$

where d_a denotes the total length of the central fixed-length links of the last five parallel modules of the robot, d_t denotes the distance between the center of mass of the moving platform and the target point in the first parallel module. If $d_a < d_t$, the target point is not within the maximum range that the robot can reach. Otherwise, the target point is within the maximum range that the robot can reach.

The position error ε is given, and then d_f denoting the distance between the end of the robot P'_6 and the target point P_t is computed by $d_f = \|P_t - P'_6\|$. If $d_f > \varepsilon$, then start the forward iteration process is started. Otherwise, the iterative calculation will stop. The forward iteration process is as follows:

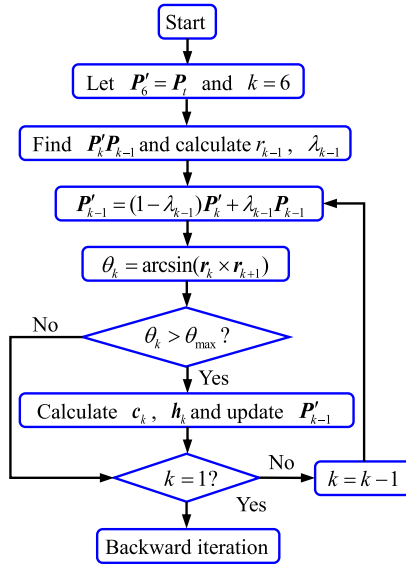


Figure 5. Forward iteration flow chart.

$P'_0, P'_1 \dots P'_6$ is the position of each point after forward iteration. Moving the end point of the robot to the target point is the first step, as shown in Fig. 4(b), namely $P'_6 = P_t$. The second step is finding the line $P'_6 P_5$ as shown in Fig. 4(c), and then calculating the length r_5 of the link $P'_6 P_5$ with Eq. (8) and the ratio of d_5 to r_5 with Eq. (9). Substituting r_5 and λ_5 into Eq. (10) and obtaining the coordinate P'_5 after updating the position is the third step. By repeating the above calculation process, P'_4, P'_3, P'_2, P'_1 can be obtained. P_0 is not involved in the iterative process, let $P'_0 = P_0$. Thus, the forward iteration is complete:

$$r_{k-1} = ||P'_k - P_{k-1}|| \tag{8}$$

$$\lambda_{k-1} = d_{k-1} / r_{k-1} \tag{9}$$

$$P'_{k-1} = (1 - \lambda_{k-1})P'_k + \lambda_{k-1}P_{k-1} \tag{10}$$

The end platform’s mass center P'_6 coincides with the target point P_t after the forward iteration, but the first parallel module’s base platform’s mass center no longer coincides with P_0 . Therefore, backward iteration is needed to ensure the first parallel module’s moving platform mass center is permanently fixed. Backward iteration is shown in Fig. 6.

As shown in Fig. 6(a), the first step of backward iteration is to find the joint P''_1 by moving P'_1 to the initial point P_1 :

$$r'_{k+1} = ||P'_{k+1} - P'_k|| \tag{11}$$

$$\lambda'_{k+1} = d_{k+1} / r'_{k+1} \tag{12}$$

$$P''_{k+1} = (1 - \lambda'_{k+1})P'_k + \lambda'_{k+1}P'_{k+1} \tag{13}$$

Then, connect P''_1 and P'_2 , calculate the distance r'_2 between the two points and the ratio λ'_2 of d to r'_2 via Eqs. (11) and (12), and obtain the P''_2 via Eq. (13). $P''_3, P''_4, P''_5, P''_6$ can be calculated using the same iteration method as shown in Fig. 6(b) to (f). The flow chart of backward iteration is presented in Fig. 7.

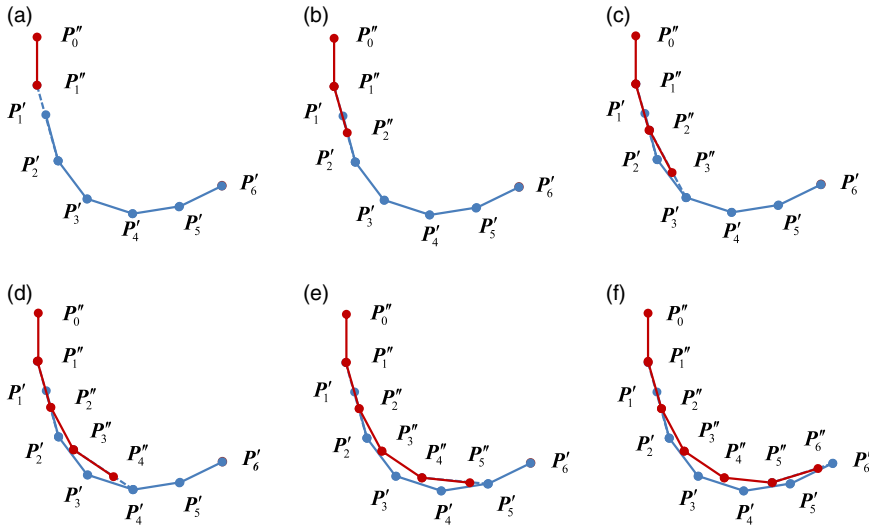


Figure 6. Backward iteration. (a) The configuration of the HRETR after forward iteration and P_1' is covered by P_1'' . (b) Find the joint P_2'' which lies on the line $\overline{P_1'P_2}$. (c) Find the joint P_3'' which lies on the line $\overline{P_2'P_3}$. (d) Find the joint P_4'' which lies on the line $\overline{P_3'P_4}$. (e) Find the joint P_5'' which lies on the line $\overline{P_4'P_5}$. (f) Find the moving platform P_6'' which lies on the line $\overline{P_5'P_6}$.

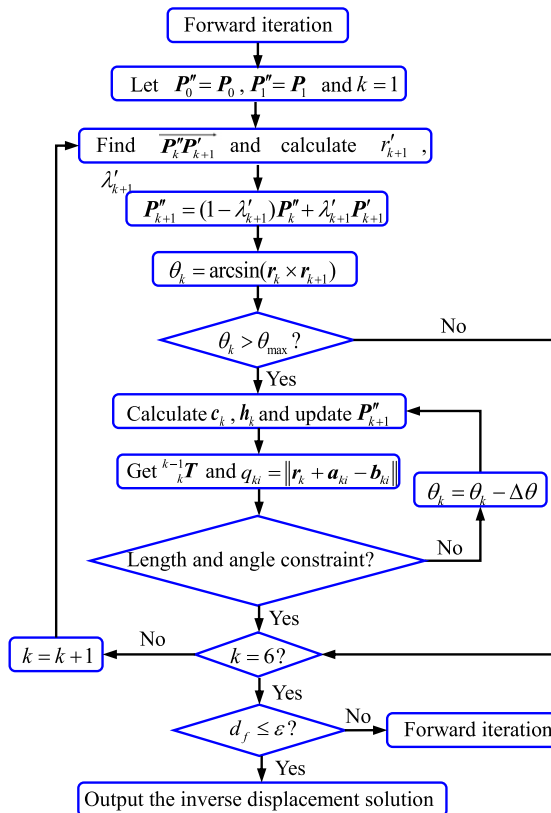


Figure 7 Flow chart of backward iteration.

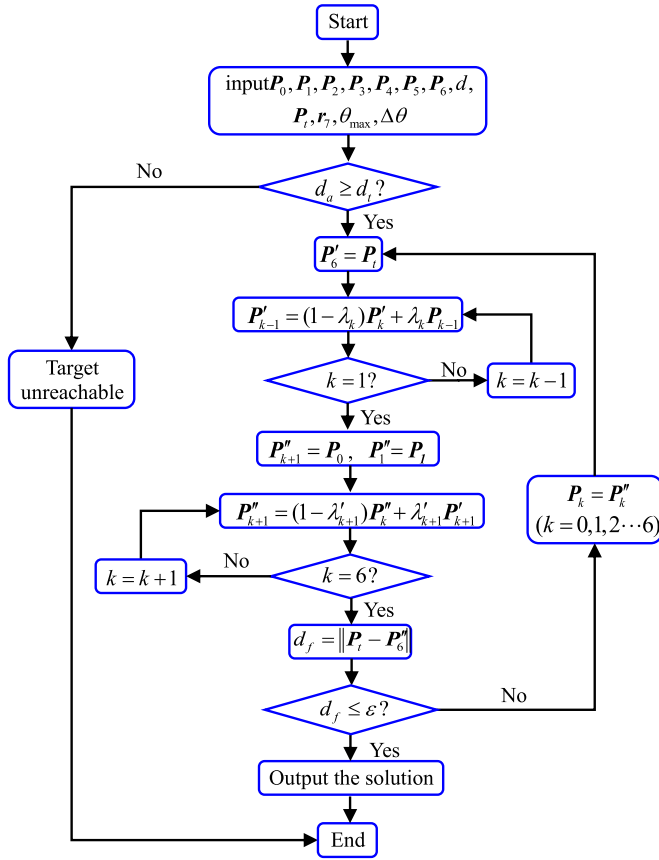


Figure 8. HRETR inverse displacement algorithm flow chart.

Following a complete iteration, it is necessary to check whether the robot’s end point is closer to the target point to determine whether to proceed with the next iteration. To achieve this, the distance d_f between the robot’s end point and the target point is calculated and compared with ε . If $d_f > \varepsilon$, the iterative calculation must be continued. Otherwise, P'_{k+1} meets the requirements and the algorithm is complete.

3.2.2 Constraint conditions

Constraint conditions on the mechanical relationships are introduced to the inverse displacement algorithm. According to the mechanical structure of the HRETR, the following physical parameters of each module must be constrained: length of the UPS links, angle of the universal joint of the parallel module’s central link, angle of the universal joint of the UPS links, and angle of the spherical joint of the UPS links.

(1) Length of the UPS links

The lengths of the HRETR’s UPS links can be obtained using Eq. (5). Considering the electric cylinder assembly, the constraint condition on the i – th UPS link of the k – th parallel module must be met

Table I. Radius of each platform of the HRETR (m)

Platform	1	2	3	4	5	6	7
Radius of each platform	0.200	0.175	0.150	0.125	0.100	0.075	0.050

Table II. Length parameters for the sleeve and push rod in the UPS link of each module (m)

	Module 1, 2	Module 3, 4, 5	Module 6
Sleeve length	0.156	0.142	0.140
Push rod length	0.162	0.142	0.152

and can be written as:

$$\begin{cases} q_{ki,\min} \leq q_{ki} \leq q_{ki,\max} \\ q_{ki,\min} = l_{bki} \\ q_{ki,\max} = l_{bki} + l_{rki} \end{cases} \tag{14}$$

where l_{bki} and l_{rki} are the length of the cylinder barrel and the length of the cylinder rod, respectively. If the condition in Eq. (14) is not satisfied, the angle θ_k between the two adjacent central links should be reduced by one step, $\theta_k = \theta_k - \Delta\theta$, and the lengths of the parallel module’s links recalculated until the constraint condition is met.

(2) Angle of the universal joint of the central link

The rotation angle θ_k of the moving platform can be regarded as the angle between the central links of two adjacent parallel modules, as shown in Eq. (15):

$$\theta_k = \arcsin(\mathbf{r}_k \times \mathbf{r}_{k+1}) \tag{15}$$

where \mathbf{r}_k and \mathbf{r}_{k+1} denote the vector of the central link of the $k - \text{th}$ parallel module and the $(k + 1) - \text{th}$ parallel module. Only when $\theta_k \leq \theta_{\max}$, the result can be considered as meeting the constraints. Otherwise, θ_k should be updated using the following methods until Eq. (15) is satisfied:

a. In the forward iteration, \mathbf{c}_k denotes a vector of the same length and opposite direction to \mathbf{r}_{k+1} , as shown in Eq. (16), and \mathbf{h}_k denotes the plane normal vector by the right-hand screw rule, according to vector \mathbf{r}_{k+1} and vector \mathbf{r}_k . When $\theta_k \leq \theta_{\max}$ is not satisfied, let \mathbf{h}_k be the rotation axis and rotate the vector \mathbf{v}_k by θ_{\max} according to the right-hand screw rule, to obtain the updated vector \mathbf{r}_k :

$$\mathbf{c}_k = -\mathbf{r}_{k+1} \tag{16}$$

$$\mathbf{h}_k = \mathbf{r}_{k+1} \times \mathbf{r}_k \tag{17}$$

b. In the backward iteration, \mathbf{c}_k is a vector of the same length and direction as \mathbf{r}_k , as shown in Eq. (18), and \mathbf{h}_k is the plane normal vector by the right-hand screw rule according to vector \mathbf{r}_k and vector \mathbf{r}_{k+1} . When $\theta_k \leq \theta_{\max}$ is not satisfied, set $\theta_k = \theta_{\max}$, let \mathbf{h}_k be the rotation axis, and rotate the vector \mathbf{v}_k by θ_k according to the right-hand screw rule. Calculate the updated vector \mathbf{r}_{k+1} and ensure the angle constraint of the universal joint is met. Otherwise, let $\theta_k = \theta_k - \Delta\theta$ and recalculate \mathbf{v}_k and \mathbf{m}_k until the updated vector \mathbf{r}_{k+1} meets the angle constraint of the universal joint:

$$\mathbf{c}_k = \mathbf{r}_k \tag{18}$$

$$\mathbf{h}_k = \mathbf{r}_k \times \mathbf{r}_{k+1} \tag{19}$$

Table III. Inverse displacement solutions for different values of position error ε

Position error ε (mm)		Module	Module	Module	Module	Module	Module
Iterative times p		1	2	3	4	5	6
Running time e (s)	Parameters						
$\varepsilon = 10^{-1}$	$\phi_x(^{\circ})$	-4.3084	3.0291	33.2667	14.1836	29.5842	39.6393
$p = 7$	$\phi_y(^{\circ})$	4.3207	11.2607	9.1340	-24.3022	28.2508	6.2064
$e = 0.1452664$	Center of mass of base platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$
	Center of mass of moving platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$	$\begin{bmatrix} 0.4000 \\ 0.4000 \\ 0.8999 \end{bmatrix}$
	Length of UPS links(m)	$\begin{bmatrix} 0.1885 \\ 0.1968 \\ 0.2195 \end{bmatrix}$	$\begin{bmatrix} 0.1732 \\ 0.2235 \\ 0.2087 \end{bmatrix}$	$\begin{bmatrix} 0.2018 \\ 0.2679 \\ 0.1464 \end{bmatrix}$	$\begin{bmatrix} 0.2516 \\ 0.1895 \\ 0.1790 \end{bmatrix}$	$\begin{bmatrix} 0.1631 \\ 0.2457 \\ 0.2114 \end{bmatrix}$	$\begin{bmatrix} 0.1934 \\ 0.2339 \\ 0.1805 \end{bmatrix}$
$\varepsilon = 10^{-3}$	$\phi_x(^{\circ})$	-4.3088	3.0293	33.3031	14.1395	29.5687	39.5940
$p = 15$	$\phi_y(^{\circ})$	4.3210	11.2614	9.1466	-24.2411	28.2671	6.2124
$e = 0.1738098$	Center of mass of base platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$
	Center of mass of moving platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$	$\begin{bmatrix} 0.4000 \\ 0.4000 \\ 0.9000 \end{bmatrix}$
	Length of UPS links(m)	$\begin{bmatrix} 0.1885 \\ 0.1968 \\ 0.2195 \end{bmatrix}$	$\begin{bmatrix} 0.1732 \\ 0.2235 \\ 0.2087 \end{bmatrix}$	$\begin{bmatrix} 0.2018 \\ 0.2680 \\ 0.1464 \end{bmatrix}$	$\begin{bmatrix} 0.2515 \\ 0.1896 \\ 0.1790 \end{bmatrix}$	$\begin{bmatrix} 0.1631 \\ 0.2457 \\ 0.2114 \end{bmatrix}$	$\begin{bmatrix} 0.1934 \\ 0.2338 \\ 0.1805 \end{bmatrix}$

Table III. Continued.

Position error ε (mm)		Module	Module	Module	Module	Module	Module
Iterative times p	Parameters	1	2	3	4	5	6
Running time e (s)							
$\varepsilon = 10^{-6}$	$\phi_x(^{\circ})$	-4.3087	3.0293	33.3045	14.1385	29.5685	39.5931
$p = 38$	$\phi_y(^{\circ})$	4.3209	11.2612	9.1472	-24.2397	28.2673	6.2125
$e = 0.2820901$	Center of mass of base platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$
	Center of mass of moving platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$	$\begin{bmatrix} 0.4000 \\ 0.4000 \\ 0.9000 \end{bmatrix}$
	Length of UPS links(m)	$\begin{bmatrix} 0.1885 \\ 0.1968 \\ 0.2195 \end{bmatrix}$	$\begin{bmatrix} 0.1732 \\ 0.2235 \\ 0.2087 \end{bmatrix}$	$\begin{bmatrix} 0.2018 \\ 0.2680 \\ 0.1464 \end{bmatrix}$	$\begin{bmatrix} 0.2515 \\ 0.1896 \\ 0.1790 \end{bmatrix}$	$\begin{bmatrix} 0.1631 \\ 0.2457 \\ 0.2114 \end{bmatrix}$	$\begin{bmatrix} 0.1934 \\ 0.2338 \\ 0.1805 \end{bmatrix}$
$\varepsilon = 10^{-10}$	$\phi_x(^{\circ})$	-4.3087	3.0293	33.3045	14.1385	29.5685	39.5931
$p = 70$	$\phi_y(^{\circ})$	4.3209	11.2612	9.1472	-24.2397	28.2673	6.2125
$e = 0.4908419$	Center of mass of base platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$
	Center of mass of moving platform	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.2000 \end{bmatrix}$	$\begin{bmatrix} 0.0150 \\ 0.0150 \\ 0.3989 \end{bmatrix}$	$\begin{bmatrix} 0.0582 \\ 0.0582 \\ 0.5893 \end{bmatrix}$	$\begin{bmatrix} 0.1698 \\ 0.1698 \\ 0.7121 \end{bmatrix}$	$\begin{bmatrix} 0.3091 \\ 0.3091 \\ 0.7468 \end{bmatrix}$	$\begin{bmatrix} 0.4000 \\ 0.4000 \\ 0.9000 \end{bmatrix}$
	Length of UPS links(m)	$\begin{bmatrix} 0.1885 \\ 0.1968 \\ 0.2195 \end{bmatrix}$	$\begin{bmatrix} 0.1732 \\ 0.2235 \\ 0.2087 \end{bmatrix}$	$\begin{bmatrix} 0.2018 \\ 0.2680 \\ 0.1464 \end{bmatrix}$	$\begin{bmatrix} 0.2515 \\ 0.1896 \\ 0.1790 \end{bmatrix}$	$\begin{bmatrix} 0.1631 \\ 0.2457 \\ 0.2114 \end{bmatrix}$	$\begin{bmatrix} 0.1934 \\ 0.2338 \\ 0.1805 \end{bmatrix}$

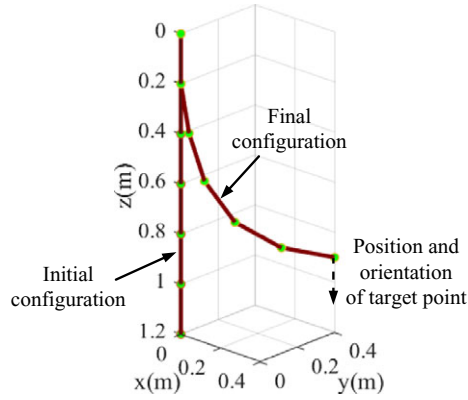


Figure 9. Configuration of the HRETR's central link without constraint conditions.

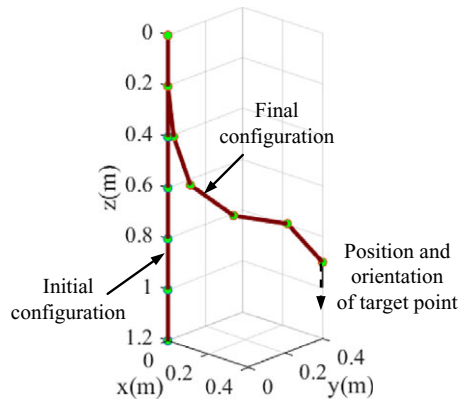


Figure 10. Configuration of the HRETR's central link with constraint conditions.

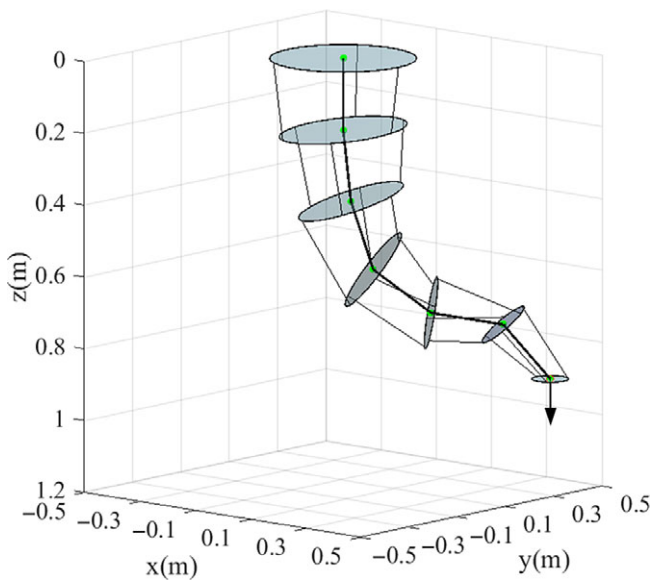


Figure 11. HRETR configuration with constraint conditions.

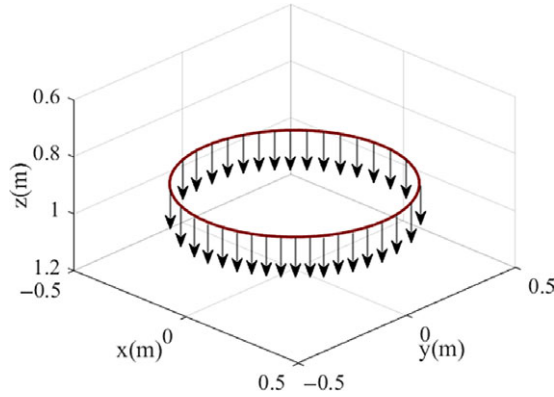


Figure 12. Trajectory 1: The normal vector at the end remains vertical and downward.

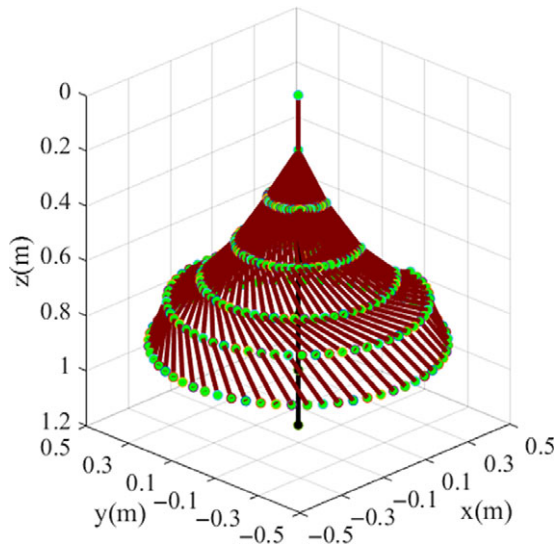


Figure 13. HRETR configuration along trajectory 1.

(3) Angle of the universal joint of the UPS links

At the center of rotation of the universal joint of the $i - th$ UPS link of the $k - th$ parallel module, let the $z - axis$ point to the moving platform. The two vertical axes of the universal joint are the $x - axis$ and $y - axis$, respectively, where the $x - axis$ points to the center of the base platform. ${}^{k-1}\mathbf{w}_{ki}$ is the vector expression of the i -th kinematics chain of the k -th parallel module in the k -th reference coordinate system and can be described by:

$${}^{k-1}\mathbf{w}_{ki} = {}^{k-1}\mathbf{R}_{ki}({}^{k-1}\mathbf{r}_k/d) \tag{20}$$

where ${}^{k-1}\mathbf{R}_{ki}$ denotes the transformation relationship between the direction vector of the i -th UPS link of the k -th parallel module and the central link of the same parallel module and it can be described by:

$${}^{k-1}\mathbf{R}_{ki} = \text{Rot}(z, \pi/3)\text{Rot}(y', \varphi_{kiy})\text{Rot}(x'', \varphi_{kix}) \tag{21}$$

Due to the hyper-redundant mechanical structure, each two adjacent parallel modules are arranged at an interval of angle along the direction of the z -axis and the rotation angle with the z -axis is $\pi/3$. Then

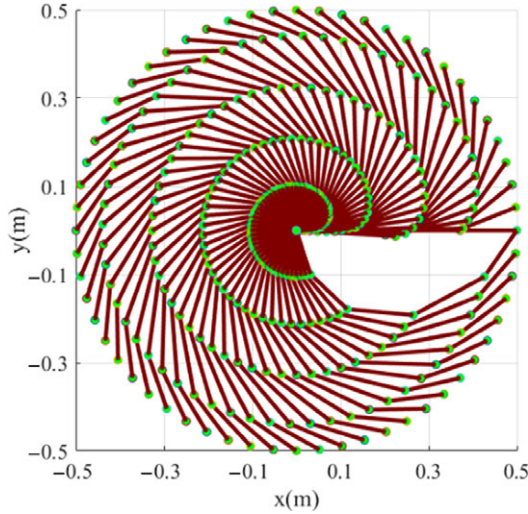


Figure 14. Top view of HRETR configurations along trajectory 1.

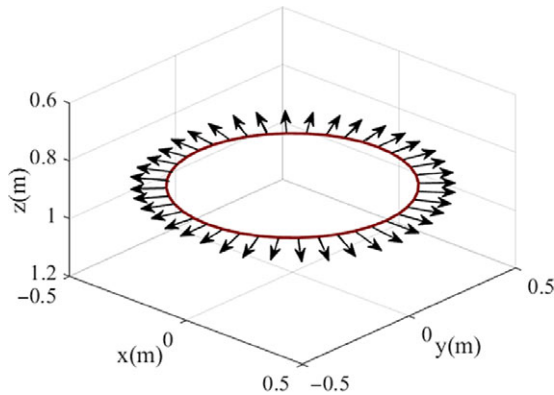


Figure 15. Trajectory 2: The normal vector at the end is divergent all around.

the rotation angle of the universal joint of the $k - th$ parallel module's $i - th$ UPS can be obtained by:

$$\begin{cases} \varphi_{kix} = \arcsin \left(\frac{\sqrt{3}^{k-1} w_{kix} - {}^{k-1} w_{kiy}}{2} \right) \\ \varphi_{kiy} = \arctan \left(\frac{{}^{k-1} w_{kix} + \sqrt{3}^{k-1} w_{kiy}}{2^{i-1} w_{kiz}} \right) \end{cases} \quad (22)$$

The rotation range of the universal joint of the $i - th$ UPS link of the $k - th$ parallel module should meet the following constraints:

$$\begin{cases} \varphi_{kix} \leq \varphi_{kix \max} \\ \varphi_{kiy} \leq \varphi_{kiy \max} \end{cases} \quad (23)$$

when the universal joint's rotation angle does not satisfy the rotation range, let $\theta_k = \theta_k - \Delta\theta$ and recalculate φ_{kix} and φ_{kiy} until the angle constraint is satisfied.

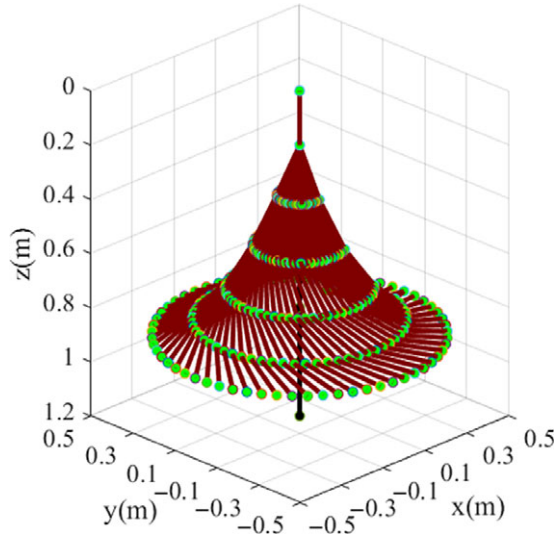


Figure 16. HRETR configuration along trajectory 2.

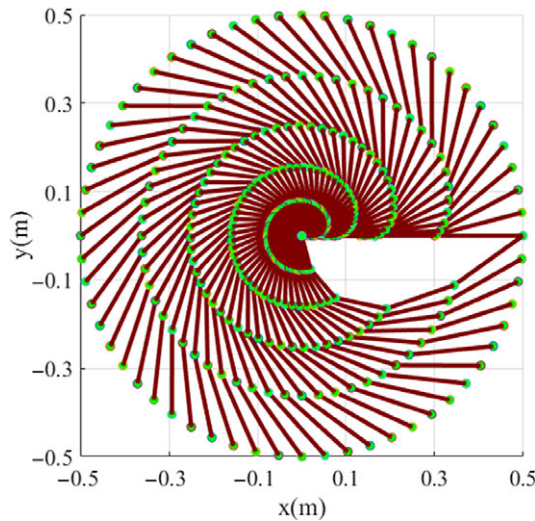


Figure 17. Top view of HRETR configurations along trajectory 2.

(4) Angle of the spherical joint of the UPS links

Assuming the initial installation angle of the spherical joint of the UPS links is γ_0 , the normal vector of the spherical joint installation plane can be obtained from the normal vector of the moving platform through the $y - x - y$ Euler angle rotation transformation which can be described by:

$$\mathbf{R}_m = \text{Rot}\left(y, -\frac{\pi}{2}\right) \text{Rot}\left(x', -\beta_i\right) \text{Rot}\left(y'', \frac{\pi}{2} - \gamma_0\right) \tag{24}$$

where $\beta = [0 \quad 2\pi/3 \quad 4\pi/3]$ denotes the angle between the two adjacent spherical joints on the moving platform. The rotation angle of the spherical joint of the i -th UPS link of the k -th parallel module is then given by:

$$\eta_{ki} = \arccos\left(\left({}^{Ok-1}\mathbf{R}_{Ok}\mathbf{R}_{ki}\mathbf{r}_k\right)^T \mathbf{w}_{ki}\right) \tag{25}$$

Table IV. Position error, iteration times, and operating time with trajectory 1

Position error ε/mm	1×10^{-1}	1×10^{-3}	1×10^{-6}	1×10^{-10}
Iteration times p	391	675	1118	1708
Operating time e/s	86.999	129.897	207.499	320.316

where ${}^{k-1}\mathbf{w}_{ki}$ is the direction vector of the central link vector of the k -th parallel module in its own base platform coordinate system. The angle constraint of the spherical joint should meet $\eta_{ki} \leq \eta_{ki,\max}$. Otherwise, let $\theta_k = \theta_k - \Delta\theta$ and recalculate η_{ki} until the angle constraint is satisfied.

The complete flow chart for the HRETR inverse displacement solution algorithm is shown in Fig. 8.

4. Numerical example

The parameters of the HRETR's mechanical structure are given in the following tables. Tables I and II present the radius of each platform of the HRETR and the lengths of the sleeves and push rods in each module's UPS link, respectively. The target point's position and orientation are given as $[0.4 \ 0.4 \ 0.9 \ 0 \ 0]^T$, where the first three values describe the position of the target and the last two values denote its orientation. Inverse displacement solutions for different values of position error ε (mm) are shown in Table III, where, p is the iterative times and e (s) is the running time. φ_x and φ_y denote the rotation angle of the moving platform concerning its own base platform. The numerical example was implemented using an Intel i7-7700 (3.6 GHz) processor, 8 GB RAM, and MATLAB® software.

The HRETR's central link configurations are shown without and with constraint conditions in Figs. 9 and 10, respectively. The solid red line denotes each module's central link, and the green dot represents the each platform mass center. To verify the HRETR's inverse displacement, its configuration was obtained based on the iterative algorithm and inverse displacement model proposed in this paper, as shown in Fig. 11. The simulation result obtained demonstrates the accuracy of the algorithm and model.

To verify the inverse displacement algorithm's effectiveness, two trajectories with different constraint conditions were provided to simulate the moving configuration of an HRETR composed of six parallel modules in series. The HRETR's configuration, when its end effector reached the given path points, was calculated and the result were analyzed.

The two circular trajectories are shown in Figs. 12 and 15, the radius was 0.5 m. Taking 60 path points on the two circular trajectories, respectively, the normal vector of the robot's end point must be vertical and downward at each point in Fig. 12 and the normal vector is required to be divergent to all around in Fig. 15. The HRETR's initial configuration is vertical and down before the simulation. Because of the difference in the position and orientation of the HRETR's end effector in two circular trajectories and the joint constraint conditions are existed in this algorithm, the configurations of the HRETR are different. As shown in Figs. 13 and 16, the end of the HRETR initially reaches the first target point on the trajectory, and then the HRETR's configuration at the remaining target points is solved by the iterative algorithm. Considering the end moving platform's normal vector to be the virtual central link and calculating the angle between the two adjacent central links and the UPS link's length continuously ensures the constraints are met. Due to the joint constraint conditions of the HRETR, six joints need to be verified if the joint constraint conditions are satisfied or not. If the angle between the two adjacent central links and the UPS link's length satisfies the constraint conditions, the end effector of the HRETR will reach the next target point. If not, the joints need to adjust the pose and position to meet the constraint conditions. Until all the constraints conditions are met, the end effector of the HRETR will reach to the next target point. When the end of the HRETR returns to the first target point again, the simulation ends. Figs. 13, 14, 16 and 17 show the HRETR's configuration change with and without constraints.

To illustrate the effectiveness of the algorithm, a numerical example was implemented on an Intel i7-7700 (3.6 GHz) processor with 8 GB RAM. The algorithm was run 100 times for the trajectory 1, as shown in Fig. 13 and calculated the average running time. The results are shown in Table IV.

5. Conclusions

This paper introduces an iterative algorithm for solving the inverse displacement of an HRETR. The HRETR which is composed of six parallel modules is presented in the paper. The work can be summarized as follows:

Each of the HRETR's parallel modules is abstracted as a geometric line segment and point model, and then the inverse displacement of the HRETR composed of six parallel modules is implemented. The algorithm process can be divided into forward iteration and backward iteration according to the principle of forward approximation and inverse pose adjustment. The inverse displacement of the HRETR is obtained while considering physical joint constraints. The inverse displacement problem of the HRETR is considered to solve the transformation matrices of each parallel module firstly, and then the inverse displacement of the parallel module is obtained. In this way, the complex inverse displacement problem of the HRETR is transformed into the inverse displacement of the parallel module.

The algorithm considering the joints constraints was used to simulate the HRETR composed of six parallel modules in series. The simulation results demonstrate that motion and configuration of the HRETR are guaranteed to be smooth. The algorithm uses simple geometric elements instead of complex structure to describe the joint position relationships of the HRETR, which can be generalized for solving the inverse displacement of an HRETR.

Acknowledgments. The research is jointly supported by the National Natural Science Foundation of China (Grant No. 51375288), the Science and Technology Program of Guangdong Province (Grant No. 2020ST004), and the Special Project for Science and Technology Innovation Team of Foshan City (Grant No. 2018IT100052). The authors would also like to thank the anonymous reviewers for their very useful comments.

Author contributions. Feifei Yuan and Yongjie Zhao conceived and designed the study. Yongxing Zhang and Xingwei Zhang conducted data gathering. Xinjian Lu performed statistical analyses. Feifei Yuan and Yongjie Zhao wrote the article.

Competing interests declaration. No competing financial interests exist.

References

- [1] E. Kelasidi, P. Liljebäck, K. Y. Pettersen and J. T. Gravdahl, "Integral line-of-sight guidance for path following control of underwater snake robots: theory and experiments," *IEEE Trans. Robot.* **33**(3), 610–628 (2017).
- [2] H. Yamada and S. Hirose, "Study of a 2-DOF Joint for the Small Active Cord Mechanism," **In: Proceedings of the IEEE International Conference on Robotics and Automation**, Kobe, Japan (2009) pp. 3827–3832.
- [3] Q. J. Gao, W. J. Wang, G. C. Niu, L. Wang and Z. C. Zheng, "Study of bionic structure and kinematics of robot for aircraft fuel tank inspection," *Acta Aeronaut. ET Astronaut. Sinica* **34**(7), 1748–1756 (2013).
- [4] R. Buckingham and A. Graham, "Snaking around in a nuclear jungle," *Ind Robot: An Int. J.* **32**(2), 120–127 (2005).
- [5] R. Bogue, "Robots in the nuclear industry: A review of technologies and applications," *Ind Robot An Int. J.* **38**(2), 113–118 (2011).
- [6] H. Y. Hu, P. F. Wang, B. Zhao, M. T. Li and L. N. Sun, "Design of a Novel Snake-Like Robotic Colonoscope," **In: Proceedings of the IEEE International Conference on Robotics and Biomimetics**, Guilin, China (2009) pp. 1957–1961.
- [7] Y. J. Zhao, "Dimensional synthesis of a three translational degrees of freedom parallel robot while considering kinematic anisotropic property," *Rob. Comput.-Integr. Manuf.* **29**(1), 169–179 (2013).
- [8] X. J. Liu, Z. L. Jin and F. Gao, "Optimum design of 3-DOF spherical parallel manipulators with respect to the conditioning and stiffness indices," *Mech. Mach. Theory* **35**(9), 1257–1267 (2000).
- [9] S. P. Bai, "Optimum design of spherical parallel manipulators for a prescribed workspace," *Mech Mach. Theory* **45**(2), 200–211 (2010).
- [10] K. T. Zhang, C. Qiu and J. S. Dai, "An extensible continuum robot with integrated origami parallel modules," *ASME J. Mech. Robot.* **8**(3), 031010–1–031010–35 (2015).
- [11] D. Trivedi, C. D. Rahn, W. M. Kier and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Appl. Bionics Biomech.* **5**(3), 99–117 (2008).
- [12] A. Hamidi, Y. Almubarak and Y. Tadesse, "Multidirectional 3D-printed functionally graded modular joint actuated by TCPFL muscles for soft robots," *Bio.-Des. Manuf.* **2**(4), 256–268 (2019).
- [13] O. Salomon and A. Wolf, "Inclined links hyper-redundant elephant trunk-like robot," *ASME J. Mech. Robot.* **4**(4), 045001–1–045001–6 (2012).
- [14] O. Xie, Q. X. Zhu, L. Shen and K. Ren, "Kinematic study on a self-propelled bionic underwater robot with undulation and jet propulsion modes," *Robotica* **36**(11), 1613–1626 (2018).

- [15] H. Komura, H. Yamada and S. Hirose, “Development of snake-like robot ACM-R8 with large and mono-tread wheel,” *Adv. Robot.* **29**(17), 1081–1094 (2015).
- [16] A. Aristidou and J. Lasenby, *Inverse Kinematics: A Review of Existing Techniques and Introduction of a New Fast Iterative Solver*, Cambridge University Department of Engineering, CUED. *Tech. Rep.* F-INFENG/TR. Vol. 632 (2009).
- [17] K. Tchoń and M. Janiak, “Repeatable approximation of the Jacobian pseudo-inverse,” *Syst. Control Lett.* **58**(12), 849–856 (2009).
- [18] S. A. A. Moosavian and E. Papadopoulos, “Modified transpose Jacobian control of robotic systems,” *Automatica* **43**(7), 1226–1233 (2007).
- [19] A. Colomé and C. Torras, “Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements,” *IEEE/ASME Trans. Mechatron.* **20**(2), 944–955 (2015).
- [20] W. F. Xu, J. T. Zhang, B. Liang and B. Li, “Singularity analysis and avoidance for robot manipulators with nonspherical wrists,” *IEEE Trans. Ind. Electron.* **63**(1), 277–290 (2016).
- [21] C. W. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Trans. Syst., Man Cybern. Syst.* **16**(1), 93–101 (1986).
- [22] G. S. Chirikjian and J. W. Burdick, “A geometric approach to hyper-redundant manipulator obstacle avoidance,” *ASME J. Mech. Des.* **114**(4), 580–585 (1992).
- [23] G. S. Chirikjian and J. W. Burdick, “A modal approach to hyper-redundant manipulator kinematics,” *IEEE Trans. Robot. Automat.* **10**(3), 343–354 (1994).
- [24] J. S. Kim and G. S. Chirikjian, “Inverse kinematic solutions of 6-D.O.F. biopolymer segments,” *Robotica* **34**(8), 1734–1753 (2016).
- [25] D. Kohli and M. Osvatic, “Inverse kinematics of general 6R and 5R, P serial manipulators,” *ASME J. Mech. Des.* **115**(4), 922–931 (1993).
- [26] Y. J. Zhao, L. Jin, P. Zhang and J. Y. Li, “Inverse displacement analysis of a hyper-redundant elephant’s trunk robot,” *J. Bionic Eng.* **15**(2), 397–407 (2018).
- [27] L.-C. T. Wang and C. C. Chen, “A combined optimization method for solving the inverse kinematics problems of mechanical manipulators,” *IEEE Trans. Robot. Automat.* **7**(4), 489–499 (1991).
- [28] Z. L. Xiong, J. F. Tao and C. L. Liu, “Inverse kinematics of hyper-redundant snake-arm robots with improved tip following movement,” *Jiqiren/Robot* **40**(1), 37–45 (2018).
- [29] A. Aristidou and J. Lasenby, “FABRIK: A fast, iterative solver for the inverse kinematics problem,” *Graph Models* **73**(5), 243–260 (2011).
- [30] A. Aristidou, Y. Chrysanthou and J. Lasenby, “Extending FABRIK with model constraints,” *Comput. Animat. Virtual Worlds* **27**(1), 35–57 (2016).
- [31] H. Ananthanarayanan and R. Ordóñez, “Real-time inverse kinematics of $(2n+1)$ DOF hyper-redundant manipulator arm via a combined numerical and analytical approach,” *Mech. Mach. Theory* **91**(4), 209–226 (2015).
- [32] S. Neppalli, M. A. Csencsits, B. A. Jones and I. D. Walker, “Closed-form inverse kinematics for continuum manipulators,” *Adv. Robot.* **23**(15), 2077–2091 (2009).
- [33] S. Neppalli, M. A. Csencsits, B. A. Jones and I. D. Walker, “A Geometrical Approach to Inverse Kinematics for Continuum Manipulators,” **In: IEEE/RISJ International Conference on Intelligent Robots and Systems**, Nice, France (2008) pp. 3565–3570.
- [34] Y. T. Kaganov and A. P. Karpenko, “Multiagent Approach to Control a Multisection Trunk-Type Manipulator,” **In: Smart Electromechanical Systems: The central nervous system, Series Studies in Systems, Decision and Control** (Springer International Publishing AG, 2017) pp. 203–212.
- [35] M. Husty, I. Birllescu, P. Tucan, C. Vaida and D. Pislă, “An algebraic parameterization approach for parallel robots analysis,” *Mech. Mach. Theory* **140**(2), 245–257 (2019).
- [36] J. A. Saglia, J. S. Dai and D. G. Caldwell, “Geometry and kinematic analysis of a redundantly actuated parallel mechanism that eliminates singularities and improves dexterity,” *ASME J. Mech. Des.* **130**(12), 124501–1–124501–5 (2008).
- [37] D. Gan, J. S. Dai, J. Dias and L. Seneviratne, “Forward kinematics solution distribution and analytic singularity-free workspace of linear-actuated symmetrical spherical parallel manipulators,” *ASME J. Mech. Robot.* **7**(4), 041007–1–041007–8 (2015).
- [38] Y. J. Zhao, X. G. Song, X. W. Zhang and X. J. Lu, “A hyper-redundant elephant’s trunk robot with an open structure: Design, kinematics, control and prototype,” *Chin J. Mech. Eng.* **33**(1), 96 (2020).