An iterative path-following method for hyper-redundant snake-like manipulator with joint limits

Cheng Wang, Haibo Xie and Huayong Yang
State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou, China

Abstract

Purpose – This paper aims to present an iterative path-following method with joint limits to solve the problem of large computation cost, movement exceeding joint limits and poor path-following accuracy for the path planning of hyper-redundant snake-like manipulator.

Design/methodology/approach – When a desired path is given, new configuration of the snake-like manipulator is obtained through a geometrical approach, then the joints are repositioned through iterations until all the rotation angles satisfy the imposed joint limits. Finally, a new arrangement is obtained through the analytic solution of the inverse kinematics of hyper-redundant manipulator. Finally, simulations and experiments are carried out to analyze the performance of the proposed path-following method.

Findings – Simulation results show that the average computation time is 0.1 ms per step for a hyper-redundant manipulator with 12 degrees of freedom, and the deviation in tip position can be kept below 0.02 mm. Experiments show that all the rotation angles are within joint limits.

Research limitations/implications – Currently , the manipulator is working in open-loop, the elasticity of the driving cable will cause positioning error. In future, close-loop control based on real-time attitude detection will be used in in combination with the path-following method to achieve high-precision trajectory tracking.

Originality/value – Through a series of iterative processes, the proposed method can make the manipulator approach the desired path as much as possible within the joint constraints with high precision and less computation time.

Keywords Path-following, Snake-like manipulator, Hyper-redundant, Joint limits, Path planning

Paper type Research paper

1. Introduction

The hyper-redundant snake-like manipulator is intended for use in confined spaces, hazardous environments and complex tasks, it solves the problem that traditional industrial robots are difficult to carry out high efficiency operation in such condition due to size, drive configuration and freedom constraints. The term hyper-redundant was first proposed in 1990 by Chirikjian and Burdick (Chirikjian and Burdick, 1990), it refers to a robot which has a large or infinite number of DOFs (degrees of freedom), and a shape similar to a snake. This kind of manipulator usually consists of several joints in series, it can be divided into two categories, rigid structure hyper-redundant manipulator (Walker and Hannan, 1999; Buckingham, 2002; Hannan and Walker, 2002; Hannan and Walker, 2003; Zheng *et al.*, 2021) and continuum manipulator (Robinson and Davies, 2002; McMahan *et al.*, 2006; Corporate, 2010; Webster Iii and Jones, 2010; Liu *et al.*, 2021). Compared with the continuous manipulator, the hyper-redundant manipulator has better positioning accuracy and load capacity, thus it is suitable for using as a special industrial robot, which can enter the enclosed space through small holes or cracks, then carries out operation as nuclear plant maintenance, aircraft assembly or rescue work (Kamegawa *et al.*, 2004; OC, 2008; Buckingham and Graham, 2013). Recently, Qin and Wu, proposed an under-actuated snake robot, it synthesizes the respective advantages of hyper-redundant robots and continuous robots (Qin *et al.*, 2021; Qin *et al.*, 2022).

As the hyper-redundant manipulator relies on its redundancy, the path planning algorithm when avoiding obstacles is very important. Many path planning methods for snake-like manipulator have already been proposed. Williams *et al.* (Williams, 1995; William, 1997) proposed the follow-the-leader (FTL) heuristic algorithm by analyzing snake and rope movements in 1995, that is, the head guides the direction, tail

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/0143-991X.htm**

Industrial Robot: the international journal of robotics research and application
50/3 (2023) 505–519
Emerald Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-04-2022-0106]

Funding: This work was supported in part by National Key R&D Program of China (2018YFB1306702), The Fundamental Research Funds for the Central Universities (226-2022-00016) and State Key Laboratory of Fluid Power and Mechatronic Systems Independent Project (SKLoFP_ZZ_2106).

Received 16 April 2022
Revised 6 August 2022
4 November 2022
27 November 2022
6 December 2022
Accepted 12 December 2022

following the trajectory of the head. A FTL approach based on Generalized Voronoi Graphs was proposed by Choset and Henning (1999). David and Salvador (Palmer *et al.*, 2014) proposed a tip-following method using the sequential quadratic programming optimization approach to navigate the continuum manipulator. Aristidou *et al.* (Aristidou and Lasenby, 2011) proposed a Forward And Backward Reaching Inverse Kinematics algorithm for inverse kinematics of human joints. Sreenivasan *et al.* (2010) proposed an algorithm for simulation of hyper-redundant manipulators based on tractrix curve. Yuan *et al.* (2022) propose an iterative algorithm to solve the inverse displacement for the hyper-redundant elephant's trunk robot.

However, these path planning methods have certain limitations. The FTL and tractrix curves are unable to solve the problem that the joints exceed the limit. Quadratic programming optimization method is being performed using a non-linear constrained optimization by the means of the sequential quadratic programming (SQP) algorithm; it is able to incorporate joints constraints. However, the average computation time of SQP algorithm is 0.4 s (Palmer *et al.*, 2014), it becomes the main constraint for the control performance of the manipulator and its end positioning accuracy is relatively poor.

In this paper, a novel heuristic method is presented to realize path-following motion. Then, an iterative method for incorporating constraints is used within the path-following method, it gradually adjusts the attitude of the manipulator using geometric approach to make manipulator follow the desired path as much as possible without exceeding the joint limit.

This paper's structure is as follows. Section 2 introduces the mechanical configuration of a hyper-redundant manipulator. In Section 3, the proposed path-following method and an iterative algorithm for incorporating constraints are presented. Simulations and experiments under different situations are carried out to analyze the performance of the path-following method with joint limits in Section 4. Finally, the conclusion is summarized in Section 5.

2. Mechanical design of hyper-redundant snake-like manipulator

A typical design of a hyper-redundant snake-like manipulator is shown in Figure 1. The hyper-redundant manipulator system consists of actuation module, manipulator module and feed-in module. The manipulator module has several sections, each section has two DOFs. As cable can only transfer pulling force, one section should be driven by at least three cables. In this design, each section is driven by three steel wire ropes evenly spaced along the circumference (120° apart). The positions of cables driving each section are separated from each other with fixed angle; therefore, the driving cables will go through the hole in end disk of the former sections and eventually connected to the driving module. The actuation module is set in a pack at the base of the manipulator, which consists of transmission mechanisms and servo motors. The base of the manipulator is mounted on a feed-in mechanism, which makes the whole manipulator has a horizontal linear DOF.

3. Path-following method

The meaning of path-following is defined that all joints of a snake-like manipulator are kept on the path with minimal error when following a desired path. Path-following can be used not only for path planning under predetermined environment and obstacles but also for real-time navigation based on paths generated online in unknown environment.

The path-following method proposed is based on the discretized path curve. As shown in Figure 2, the desired path is discretized into target points, and the discretization of path curve can be achieved by different methods, as equal parameters, equal chord length, arc length, etc. Different sampling intervals may affect the path following accuracy of the manipulator. To improve the accuracy of path tracking, the distance between the selected target points should be far less than the length L of the manipulator link.

For rigid structure hyper-redundant robot consists of discrete links, if the joint of each section can be seen as a point, the whole manipulator can be simplified as a polygonal line and the center point of each joint can be called joint point, denoted as a_1 to a_n in Figure 2, which describes the macro geometric characteristics of the manipulator. For each step of the path-following method, the base of the manipulator advances a small step and new configuration within the joint limits of the manipulator will be calculated. New arrangement can be determined by calculating new bending angles and displacement of the base through inverse kinematics (Xie *et al.*, 2019). Finally, the manipulator can use the new arrangements to perform path-following motion based on the kinematics model. The kinematics modeling is not the focus of this paper, it can be found in our previous research.

3.1 Fitting the manipulator with path curve

As the joint points describe the configuration of the manipulator and the desired path is discretized into target

Figure 1 Illustration of design of a hyper-redundant snake-like manipulator

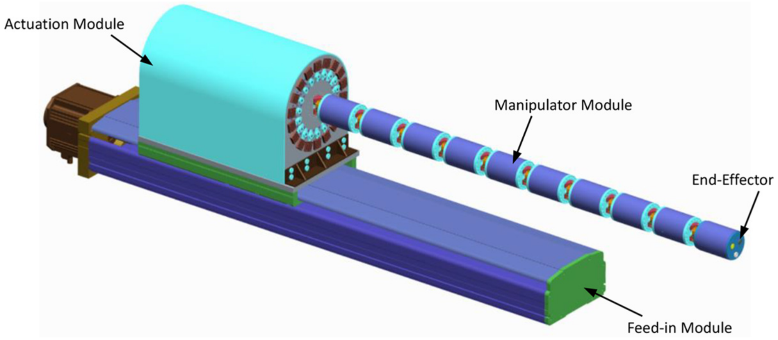
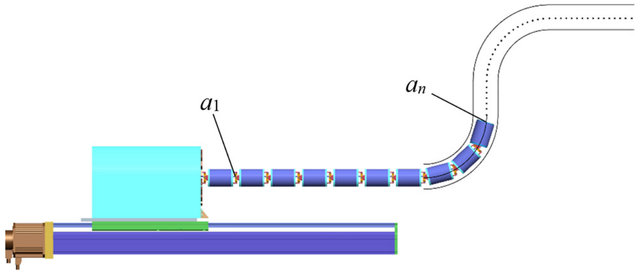


Figure 2 Illustration of path-following and discretized path curve



points, the key of path-following method is that the distance between the joint points of the manipulator and the targets point of the path curve should be kept minimum for each step. Thus, the problem turns into how to find each joint point closest to the target points quickly in turn.

The path-following method uses the previously calculated position of the joints to find the position of the next joint. The method may start from the root joint to the last joint of the manipulator, finding the position of each joint in turn. When the manipulator follows the desired path, a_1 is the first joint attached to the base, thus a_1 can only move along the horizontal guide, when a_1 is settled, a_2 to a_n can be solved in sequence.

The searching algorithm for a_n when a_{n-1} is known is shown in Figure 3, G_m is the m-th target point, as the coordinates of a_{n-1} is already known, to find a_n is to find a point with a distance of L_n from a_n in G_{m-4}-G_{m-3}-G_{m-2}-G_{m-1}-G_m-G_{m+1}. Start from the nearest target point G_{m-3}, the Euler distance between a_{n-1} and target point is calculated in turn and compared with L_n, until the distance is larger than L_n. If $|a_{n-1}G_m| < L_n$ and $|a_{n-1}G_{m+1}| > L_n$, a_n must be on G_m-G_{m+1}, as the path information between G_m and G_{m+1} is lost, the coordinates of a_n should be found by interpolation. There are two interpolation methods to find a_n, method by trigonometry or proportional approximation.

3.2 Find joint points by trigonometry

As shown in Figure 4, the path curve between G_m and G_{m+1} can be regarded as a straight line segment, then a_n can be found by solving triangles. The distance from G_m to the previous joint point a_n is recorded as p_m. Correspondingly, the distance from G_{m+1} to the previous joint point a_n is recorded as p_{m+1}. $|a_na_{n-1}|$ is the length of link n of the manipulator, denoted as L_n. In Section 3, it is mentioned that $|G_mG_{m+1}|$ is far less than $|a_na_{n-1}|$, assuming $10\cdot|G_mG_{m+1}| < |a_na_{n-1}|$. In addition, it is assumed that the chosen path curve is a path with appropriate curvature and can be tracked by the manipulator, so $\angle a_{n-1}G_{m+1}a_n$, denoted as $\angle\varphi_n$, should be a small angle, thus $\angle\varphi_n < 90°$. When $\angle\varphi_n \geq 90°$, it indicates that the curvature of the selected path curve is too large for the manipulator to follow. Therefore, the extreme case when $\angle\varphi_n \geq 90°$ is not considered in this paper.

Figure 3 Illustration of the position of joint points on the discretized path curve

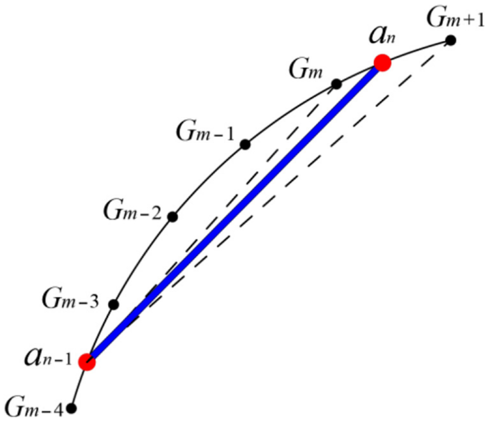
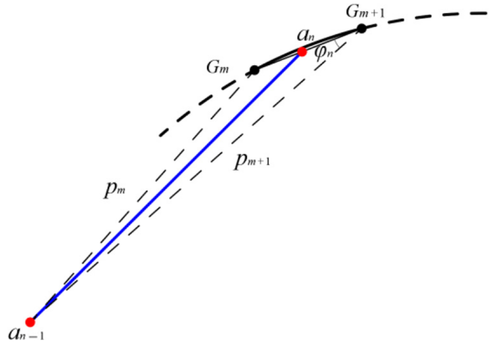


Figure 4 Illustration of finding joint points by trigonometry



In $\triangle a_{n-1}G_mG_{m+1}$, using cosine theorem, equation (1) is acquired:

$$\cos\varphi_n = \frac{|p_{m+1}|^2 + |G_mG_{m+1}|^2 - |p_m|^2}{2\cdot|p_{m+1}|\cdot|G_mG_{m+1}|} \qquad (1)$$

Then in $\triangle a_na_{n-1}G_{m+1}$, equation (2) can be obtained using cosine theorem:

$$L_n{}^2 = |a_nG_{m+1}|^2 + p_{m+1}{}^2 - 2\cdot|a_nG_{m+1}|\cdot p_{m+1}\cdot\cos\varphi_n \qquad (2)$$

Equation (2) is a quadric equation, it has multiple solutions as: $p_{m+1}\cdot\cos\varphi_n \pm\sqrt{p_{m+1}{}^2\cdot\cos^2\varphi_n - p_{m+1}{}^2 + L_n{}^2}$. As it is mentioned before that $\angle\varphi_n < 90°$ and $10\cdot|G_mG_{m+1}| < |a_na_{n-1}|$, considering $|a_{n-1}a_n| < |a_{n-1}G_{m+1}|$ and $|a_nG_{m+1}| < |G_mG_{m+1}|$, according to the geometric relationship in Figure 4, it is easy to know that $|a_nG_{m+1}|$ should be the smaller solution as equation (3):

$$|a_nG_{m+1}| = p_{m+1}\cdot\cos\varphi_n - \sqrt{p_{m+1}{}^2\cdot\cos^2\varphi_n - p_{m+1}{}^2 + L_n{}^2} \qquad (3)$$

Finally, the coordinate of a_n can be acquired as equation (4) through the slope of G_m-G_{m+1} and coordinate of G_{m+1}:

$$a_n = G_{m+1} + \frac{G_m - G_{m+1}}{|G_mG_{m+1}|}\cdot|a_nG_{m+1}| \qquad (4)$$

The remaining joint points of the manipulator can be calculated as the same method in equations (1)–(4). The algorithm for finding new paths is a process of constantly comparing, solving triangles and finding new points (The corresponding MATLAB code is included in the supplementary file).

3.3 Find joint points by proportional approximation

Another interpolation method for finding a_n is shown in Figure 5, take a_{n-1} as the center and L_n as the radius to make an arc, the intersection with G_mG_{m+1} is a_n and the intersection with $a_{n-1}G_{m+1}$ is C. Correspondingly, take a_{n-1} as the center and p_m as the radius to make an arc, the intersection with $a_{n-1}a_n$ is A and the intersection with $a_{n-1}G_{m+1}$ is B.

Based on the principles of space analytical geometry, the coordinates of B can be calculated as equations (5) and (6):

$$\lambda_1 = \frac{p_m}{p_{m+1}} \tag{5}$$

$$B = a_{n-1} + \lambda_1(G_{m+1} - a_{n-1}) \tag{6}$$

Assuming that $\triangle a_nAG_m$ is similar to $\triangle G_{m+1}BG_m$, equation (7) can be obtained according to the similar principle of triangle. Then, the coordinates of A can be calculated as equation (8), according to space analytic geometry:

$$\lambda_2 = \frac{G_mA}{G_mB} = \frac{Aa_n}{BG_{m+1}} = \frac{L_n - p_m}{p_{m+1} - p_m} \tag{7}$$

$$A = G_m + \lambda_2(B - G_m) \tag{8}$$

After the coordinates of A is obtained, and for the length of $|a_{n-1}a_n|$ is L_n, a_n can be calculated finally as equations (9) and (10):

$$\lambda_3 = \frac{L_n}{p_m} \tag{9}$$

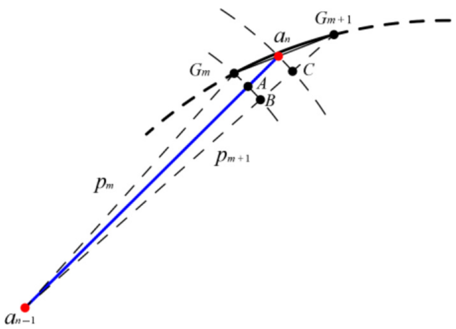
$$a_n = a_{n-1} + \lambda_3(A - a_{n-1}) \tag{10}$$

Taking equations (7)–(9) into (10), a_n can be obtained as equation (11):

$$a_n = a_{n-1} + \lambda_3 \cdot (G_m - a_{n-1} + \lambda_2 \cdot (a_{n-1} - G_m + \lambda_1 \cdot (G_{m-1} - a_{n-1}))) \tag{11}$$

Comparison of these two interpolation methods is carried out in Section 4.

Figure 5 Illustration of finding joint points by proportional approximation



3.4 Applying joint limits to path-following method

Although hyper-redundant manipulator can carry out complex movements, it has certain limitations. As shown in Figure 6, the rotation angle between every two sections of the manipulator shall not exceed the joint limit, for each section should not collide with other parts. In addition, hyper-redundant manipulators may have different joint limits due to their designs, increasing allowed range of motion may reduce the space inside the links of the manipulator, because the parameters of the manipulator restrict each other. In this paper, a new methodology is proposed to incorporate joint limit into path-following method (The corresponding MATLAB code is included in the supplementary file).

The main idea of the joint restriction method is to re-position the joint point within the joint limit. After obtaining the position of each joint using the method in Section 3.2 or 3.3, the rotation angle of each joint is checked, if it exceeds the joint limit, the corresponding joint points should be moved to satisfy joint limits with minimum deviation from the pre-defined path. In this method, the joint points are re-positioned through solving the triangle formed by three adjacent joint points. In this way, the complex three-dimensional problems are simplified to two-dimensional problems.

As shown in Figure 7, two sections of a manipulator are following a desired path. Assuming a_{n-1}, a_n, a_{n+1} are the joint points of the manipulator, and their position are figured out by the method in Section 3.2 or 3.3. The rotation angle θ_{n+1} can be calculated by equation (12):

$$\theta_{n+1} = \cos^{-1}\left(\frac{\overrightarrow{a_na_{n-1}} \cdot \overrightarrow{a_{n+1}a_n}}{|a_na_{n-1}| \cdot |a_{n+1}a_n|}\right) \tag{12}$$

When θ_{n+1} exceeds the joint limit, a_n and a_{n+1} are moved to a'_n and $a'_{n+1}$to satisfy joint limits, In $\triangle a_{n-1}a'_na'_{n+1}$, a'_{n+1} is on the predefined path, the angle between $\overrightarrow{a_{n-1}a_n'}$ and $\overrightarrow{a_n'a_{n+1}'}$ is the maximum setting angle, denoted as Φ. The maximum setting angle Φ is maximum rotation angle Φ_m minus a small quantity δ, it will be explained in next section. And a'_n and a'_{n+1} is on the plane formed by a_{n-1}, a_n and a_{n+1}. Point P is on the line segment $a_{n-1}a'_{n+1}$, and the line segment a'_nP is perpendicular to the line segment $a_{n-1}a'_{n+1}$. The total length of section n is denoted as L_n, thus $|a_na_{n-1}|= L_n$.

In $\triangle a_{n-1}a'_na'_{n+1}$, equation (13) can be easily obtained:

$$|a_{n-1}a_{n+1}^{'}| = \sqrt{L_n{}^2 + L_{n+1}{}^2 + 2L_n \cdot L_{n+1} \cdot \cos(\Phi)} \tag{13}$$

Figure 6 Illustration of joint limits of hyper-redundant manipulator

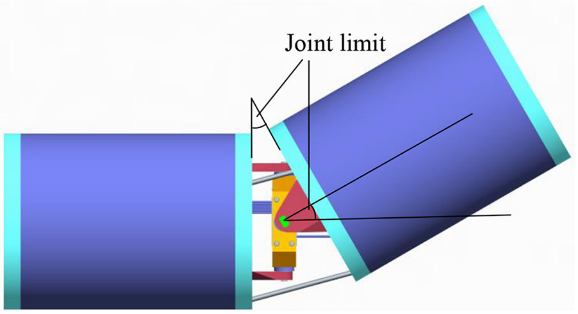
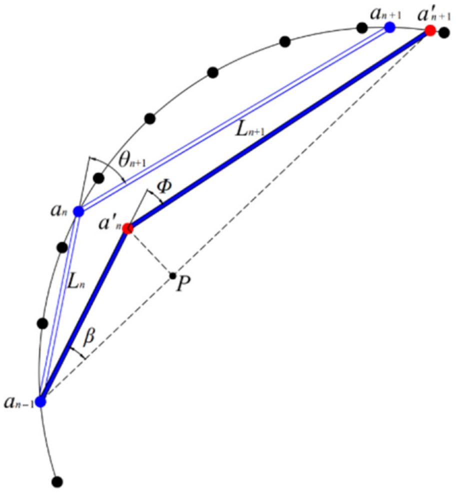


Figure 7 Incorporating joint constraints within path-following method



Thus, the position of a'_{n+1} can be calculated by the same method in Section 3.2 or 3.3 with the chord length of $|a_{n-1}a'_{n+1}|$. According to space analytic geometry, the position of point P can be obtained as equations (14) (15) (16):

$$\sin\beta = \frac{\sin(\Phi)\cdot L_{n+1}}{|a_{n-1}a'_{n+1}|} \tag{14}$$

$$\lambda = \frac{|a_{n-1}P|}{|a_{n-1}a'_{n+1}|} = \frac{\sqrt{1-\sin^2\beta}\cdot L_n}{|a_{n-1}a'_{n+1}|} \tag{15}$$

$$P = (1-\lambda)\cdot a_{n-1} + \lambda \cdot a'_{n+1} \tag{16}$$

As the line segment a'_nP is perpendicular to the line segment $a_{n-1}a'_{n+1}$, the direction vector of line a'_nP denoted as $\overrightarrow{k_n}$ can be carried out by calculating vector product as equation (17), and the position of a'_n can be calculated as equation (18), according to space analytic geometry:

$$\overrightarrow{k_n} = \overrightarrow{a_{n-1}a_n} \times \overrightarrow{a_{n-1}a'_{n+1}} \times \overrightarrow{a_{n-1}a'_{n+1}} \tag{17}$$

$$a'_n = P + \frac{\overrightarrow{k_n}}{|\overrightarrow{k_n}|}\cdot L_n \cdot \sin(\beta) \tag{18}$$

In particular, equation (19) can be used to simplify the calculation when all sections have the same length:

$$a'_n = \frac{a_{n-1} + a'_{n+1}}{2} + \frac{\overrightarrow{k_n}}{|\overrightarrow{k_n}|}\cdot L \cdot \sin\left(\frac{\Phi}{2}\right) \tag{19}$$

3.5 Iterative approach for finding new configuration with joint limits

Re-position of a joint point may cause the previous joint exceeding joint limit, thus an iterative approach is used to ensure the rotation angle of each joint is within the joint limits. Figure 8 is the process of gradual adjustments of joint points.

In Figure 8(a), the first two section of the manipulator is following the predefined path using the path-following method in Section 3.2. The angle between $\overrightarrow{a_1a_2}$ and $\overrightarrow{a_2a_3}$ is larger than the maximum rotation angle Φ_m, then using the method in Section 3.4, the joint a_1, a_2 and a_3 are re-positioned to satisfy joint limits in Figure 8(b), and the position of the third joint is determined. As angle between $\overrightarrow{a_2a_3}$ and $\overrightarrow{a_3a_4}$ is larger than maximum rotation angle Φ_m, the position of joint a_3 and a_4 are also moved to satisfy joint limits in Figure 8(c). However, the re-position of joint a_3 and a_4 lead to the previous section exceeding joint limits again. Thus, after one complete calculation of joints position, the rotation angle of each section is recalculated, if any section exceeding joint limit, the current joint and the remaining target point will be regarded as the new path in Figure 8(c), and the previous procedure is then repeated, until all the sections is within the joint limits. After several iterations, the rotation angle of each section is less than maximum rotation angle Φ_m as shown in Figure 8(d), the iteration process is completed. In the end, the actual maximum rotation angle is close to the setting angle Φ, but larger than Φ. If the setting angle is set to the maximum rotation angle Φ_m, the actual rotation angles will never equal to or less than Φ_m after infinite iterations. Therefore, the setting angle Φ is set to Φ_m minus a small tolerance δ, the smaller the δ, the closer the actual angle will be to the maximum rotation angle Φ_m, but more iterations and calculation time will be needed. Figure 9 shows the above solution process in pseudo-code.

4. Simulation and analysis of iterative path-following method with joint limits

As it is important to figure out how different path-following methods affect the computational time and path deviation, the simulations and analysis for the iterative path-following were performed on a desktop with an INTEL© i7-8700 processor. In the simulation, the length of a single section is set to 185 mm. The path to be passed by the manipulator can be selected according to the demand, and usually the path could

Figure 8 An example of an iteration process of path-following with joint limits

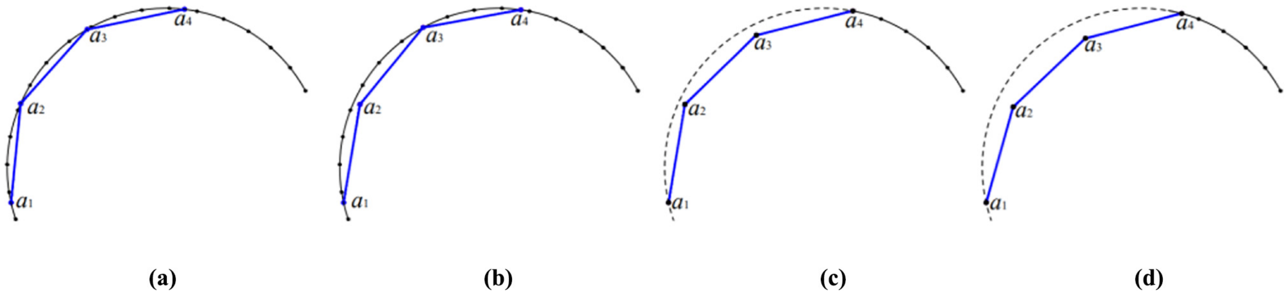


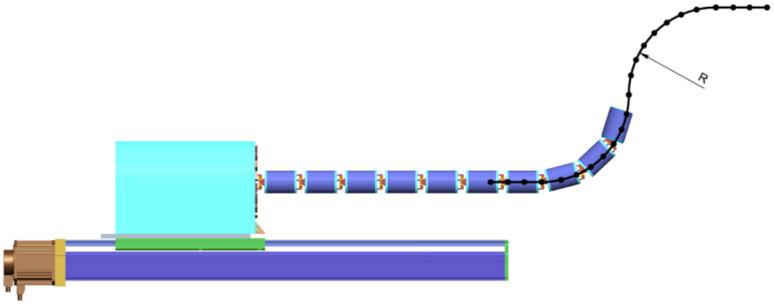
Figure 9 Pseudo-code for solving new arrangement

```
Initialize matrix $M$ to store calculated arrangement of the manipulator
Determine the advance distance of the base as $BaseStep$ in each step according
to the distance between the target points
While(the last target point is not reached yet)
Initialize matrix $a$ to store the position of joint point of the manipulator
% The initial displacement of the base is 0, the base moves forward by $BaseStep$
for each step
$BaseDisplacement = BaseDisplacement+BaseStep$
  % Starting from $a_1$, determine coordinates in turn
  For $i=1{:}n$
  % Find the next point $a_i$ in path $G$ according to $a_{i-1}$ and section length $L_i$
    If $i = 1$, $a_{i-1}$ should be replaced by x-axis
 $a_i$ = findPoint($a_{i-1}$, $L_i$, $G$)
     % If rotation angle of joint $i$ exceeds joint limit $\Phi_m$
     If $<\overrightarrow{a_{i-2}a_{i-1}},\ \overrightarrow{a_{i-1}a_i},> \ >\Phi_m$
     Reposition $a_{i-1}$ and $a_i$ using joint limit $\Phi_m$ minus iteration tolerance $\delta$
    ($a_{i-1}$, $a_i$) = reposition($a_{i-2}$, $L_{i-1}$, $L_i$, $\Phi_m$-$\delta$, $G$)
     End
  End
  While(AnyJointAngle $>\Phi_m$)
  Combine the current joint points $a$ with the remaining target points to form
  a new path $G_a$
    For $i=1{:}n$
     % Find the next point $a_i$ in path $G_a$ according to $a_{i-1}$ and section length $L_i$
       If $i = 1$,$a_{i-1}$ should be replaced by x-axis
    $a_i$ = findPoint($a_{i-1}$, $L_i$, $G_a$)
        % If rotation angle of joint $i$ exceeds joint limit $\Phi_m$
        If $<\overrightarrow{a_{i-2}a_{i-1}},\ \overrightarrow{a_{i-1}a_i},> \ >\Phi_m$
        Reposition $a_{i-1}$ and $a_i$ using joint limit $\Phi_m$ minus iteration tolerance $\delta$
        ($a_{i-1}$, $a_i$)= reposition($a_{i-2}$, $L_{i-1}$, $L_i$, $\Phi_m$-$\delta$, $G_a$)
        End
     End
  End
% Using the position of each joint to get the rotation angle of each section by
inverse kinematics
$R$ = getRotAngle($a$)
Save new arrangement $R$ to $M$
```

be some spatial curves, and the target points are obtained by equal chord length method with different sampling intervals.

In the simulations, S-bends are chosen as one path of the manipulator shown in Figure 10, which consist of two bends with a radius R of 300 mm in the plane.

Figure 10 Illustration of following S-bends path



And a helix path is also chosen for evaluating of the movement in three-dimension space. As shown in Figure 11, the target points of the path curve are obtained based on the conic helix, the expression is equation (20):

$$\begin{cases} x = 60t \cdot \sin(\pi \cdot t/15) \\ y = 24t \\ z = 60t \cdot \cos(\pi \cdot t/15) \end{cases} \tag{20}$$

Four different methods are used in the simulation, for each trial the total computation time is measured and the deviation from the desired path and position of each joint are recorded, all data and figures are obtained by MATLAB software.

4.1 Path deviation

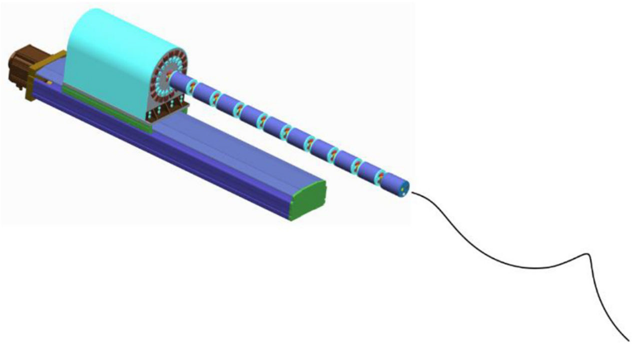
After the path is discretized, some path information is lost, which will cause the joint points of the manipulator to deviate from the desired path. And different path planning methods may have different positioning accuracy, it is necessary to analyze the deviation from the path and the tip positioning accuracy. During the simulation, the arrangements obtained by different path-following methods are substituted into the kinematic model, after that the position of each joint of the manipulator is obtained through the forward kinematic, finally the path deviation of different methods can be obtained.

Figure 12(a) shows the maximum deviation in the tip position of four path-following methods under different sampling intervals for S-bends path, the tolerance of the iteration is set to 0.01°. When the number of sampling points of the whole path is 20, the sampling interval is 50 mm, and when there are 200 sampling points on the path, the sampling interval is 5 mm. It can be seen from Figure 12(a) that the positioning accuracy of the three heuristic path-following methods is close. When the sampling interval is less than 25 mm, the positioning accuracy is higher than 0.1 mm. When there are 200 sampling points on the path, the positioning accuracy is about 0.02 mm.

When it comes to helix path in Figure 12(b), the trend of maximum deviation with sampling points is similar to that of S-curve path. Because of the motion in three-dimensional space, the overall error is larger than that in plane motion.

Numerical method uses sequence quadratic programming optimization approach to carry out the displacement of each joint that let the manipulator follow the desired path with minimal error. And it is performed using non-linear constrained optimization by MATLAB software. As the maximun distance between the joints of the manipulator and

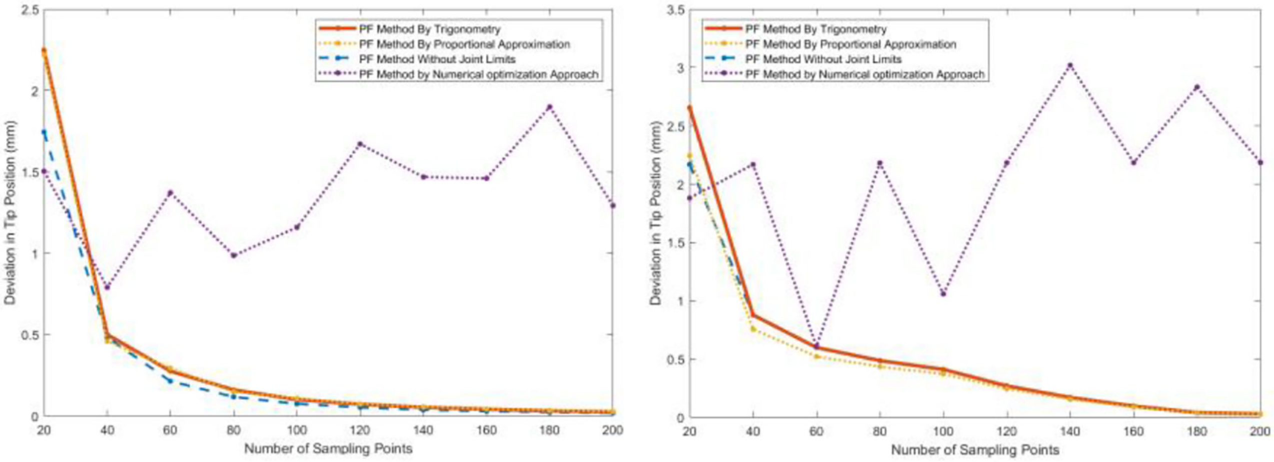
Figure 11 Illustration of following helix path



ideal path is used to build the objective function, it is hard to ensure the tip of the manipulator is on the ideal path. Although the distance between the new tip position and the ideal path is weighted by a factor to reduce the end positioning error, the tip devition of the numerical method is still unstable. The positioning accuracy is between 0.5 mm and 2 mm, and it is independent of the sampling interval.

As the hyper-redundant manipulator consists of discrete links, its new arrangements cannot meet the ideal path completely when carrying out FTL motion. The deviation from the ideal path is also analyzed. The pose of the manipulator in each step is recorded and overlaid in Figure 13, it shows the envelope of the path-following method. And this visualization of the deviations help to see where the maximum error occurs.

Figure 12 Maximum deviation in the tip position under different sampling intervals (PF is short for path-following)

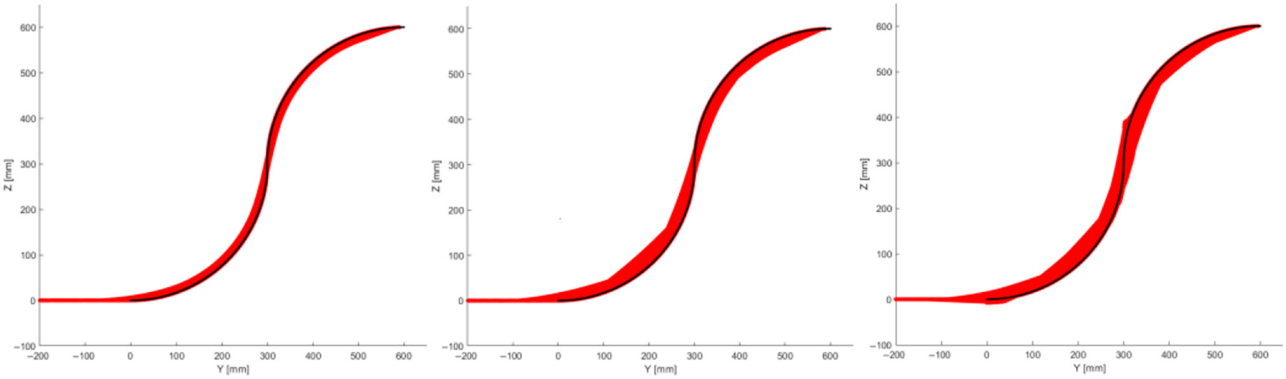


(a)

(b)

Notes: (a) Maximum deviation of pathfollowing methods for S-bends path; (b) maximum deviation of path-following methods for helix path

Figure 13 Envelope of three path-following method for S-bends path



(a)

(b)

(c)

Notes: (a) Envelop of path-following method without joint limits; (b) envelope of pathfollowing method with joint limits; (c) envelope of numerical method

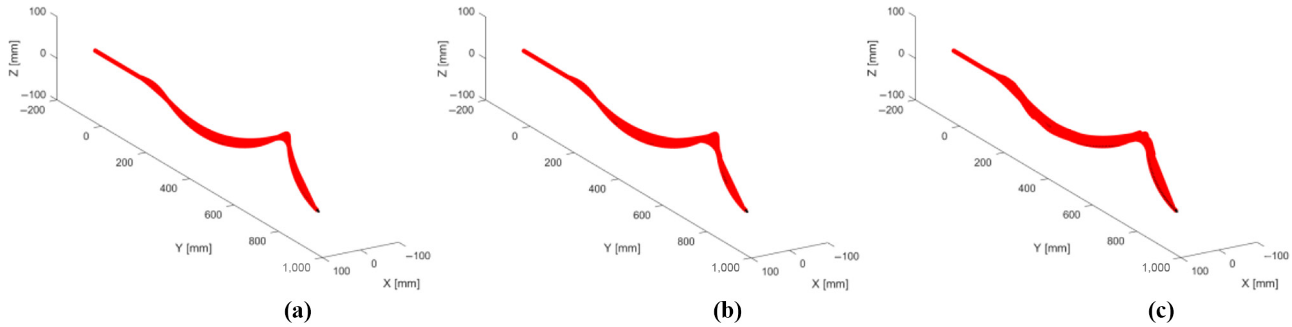
Figure 13(a) is the envelope of path-following method without joint limits, the edge of envelope of is very smooth, the maxiumu width of envelope is about 15.4 mm. Figure 13(b) is the envelope of path-following method with joint limits, the edge of envelope of is smooth, as the joint limits is set to 30°, the maxiumu width of envelope is about 32.18 mm. Figure 13(c) is the envelope of numerical method, the some part of the envelope of is rugged, the maxiumu width of envelope is about 39.86 mm.

Table 1 shows the simulation results of different path-following method when there are totally 200 sampling points on the path and the sampling interval is 5 mm, for the path-following method without joint limits the maximum rotation angle reaches

Table 1 Comparisons of the path deviations and the maximum rotation angle of three path-following method for S-bends path

Method	End accuracy	Envelope width	Maximum rotation angle
PF method without joint limits	0.020 mm	15.4 mm	35.92°
PF method with joint limits	0.026 mm	32.18 mm	29.99°
Numerical method	1.293 mm	39.86 mm	30°

Figure 14 Envelope of three path-following method for helix path

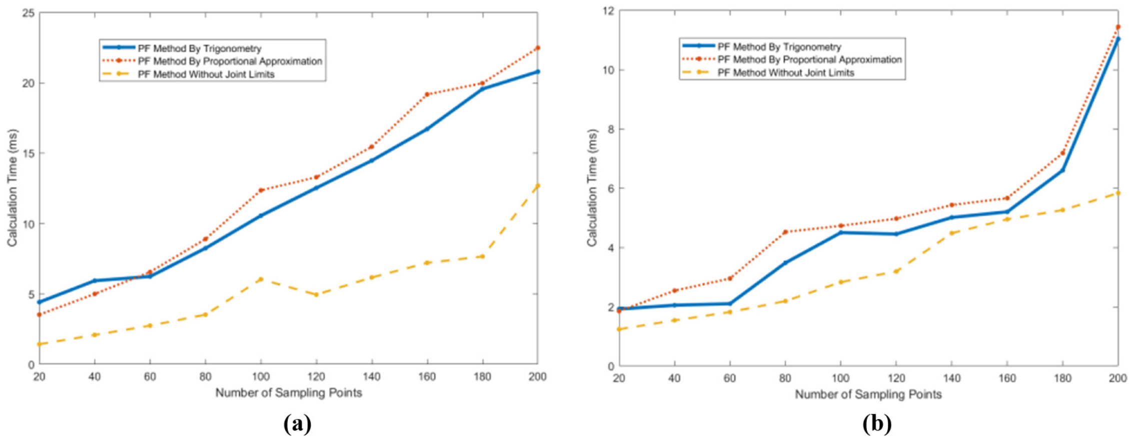


Notes: (a) Envelop of path-following method without joint limits; (b) envelope of pathfollowing method with joint limits; (c) envelope of numerical method

Table 2 Comparisons of the path deviations and the maximum rotation angle of three path-following method for helix path

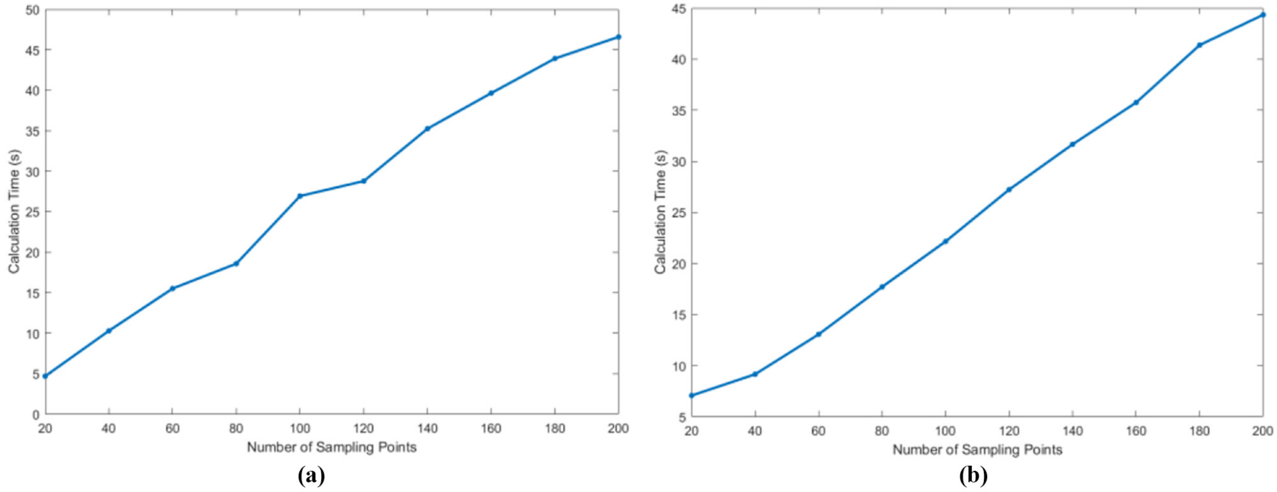
Method	End accuracy	Envelope width	Maximum rotation angle
PF method without joint limits	0.031 mm	27.14 mm	38.15°
PF method with joint limits	0.031 mm	35.97 mm	29.99°
Numerical method	2.185 mm	33.61 mm	30°

Figure 15 Calculation time of the three heuristic path-following method under different sampling interval



Notes: (a) Calculation time of path-following methods for S-bends path; (b) calculation time of path-following methods for helix path

Figure 16 Calculation time of path-following method by numerical optimization approach under different sampling interval



(a)

(b)

Notes: (a) Calculation time of numerical method for S-bends path; (b) calculation time of numerical method for helix path

Table 3 Comparisons of calculation time and number of iterations

Method	Calculation time of 200 sampling points (*ms*)	Calculation time per sampling point (*ms*)	No. of iterations
PF method by trigonometry	20.79	0.10	305
PF method by proportional approximation	22.48	0.11	357
PF method without joint limits	12.69	0.06	/
PF method by numerical optimization approach	46,590	230	/

35.92°, whereas maximum rotation angle the of path-following considering joint limits can be controlled within 30°.

Figure 14 and Table 2 show the envelope and simulation results of three path-following method for helix path. When the manipulator follows the three-dimensional path, the error in end position becomes larger and the envelope is wider. And the envelope width of numerical method becomes smaller than that of PF method with joint limits, although the difference is not big. And the maximum rotation angle can be controlled within 30°.

4.2 Calculation efficiency

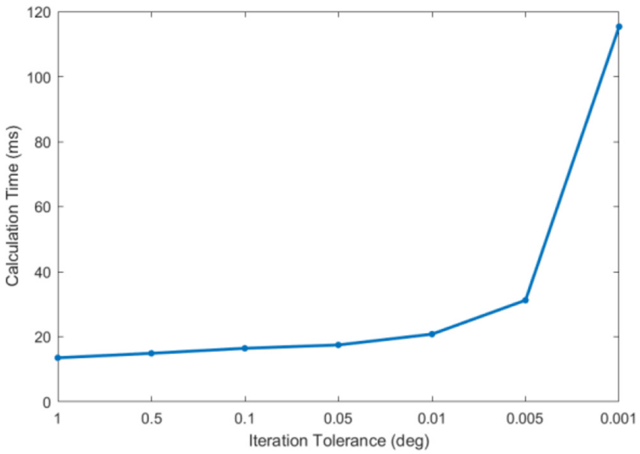
Figure 15 shows the calculation time of the three heuristic path-following method under different sampling interval, the tolerance of the iteration is set to 0.01°. In general, path-following method by trigonometry spend less calculation time than proportional approximation method, expect the numbers of sampling points are less than 60 for S-bends path, but the difference is small. In addition, the calculation time of path-following method without joints limits is the shortest among the three heuristic path-following method because no iterations are needed.

The calculation time of path-following method by numerical optimization approach under different sampling interval in

Table 4 Calculation time and number of iterations under different iteration tolerance

Iteration tolerance (*deg*)	Calculation time (*ms*)	Calculation time per step (*ms*)	No. of iterations
1	13.52	99	99
0.5	14.90	132	132
0.1	16.44	202	202
0.05	17.47	228	228
0.01	20.79	305	305
0.005	31.23	409	351
0.001	115.36	1,567	1,567

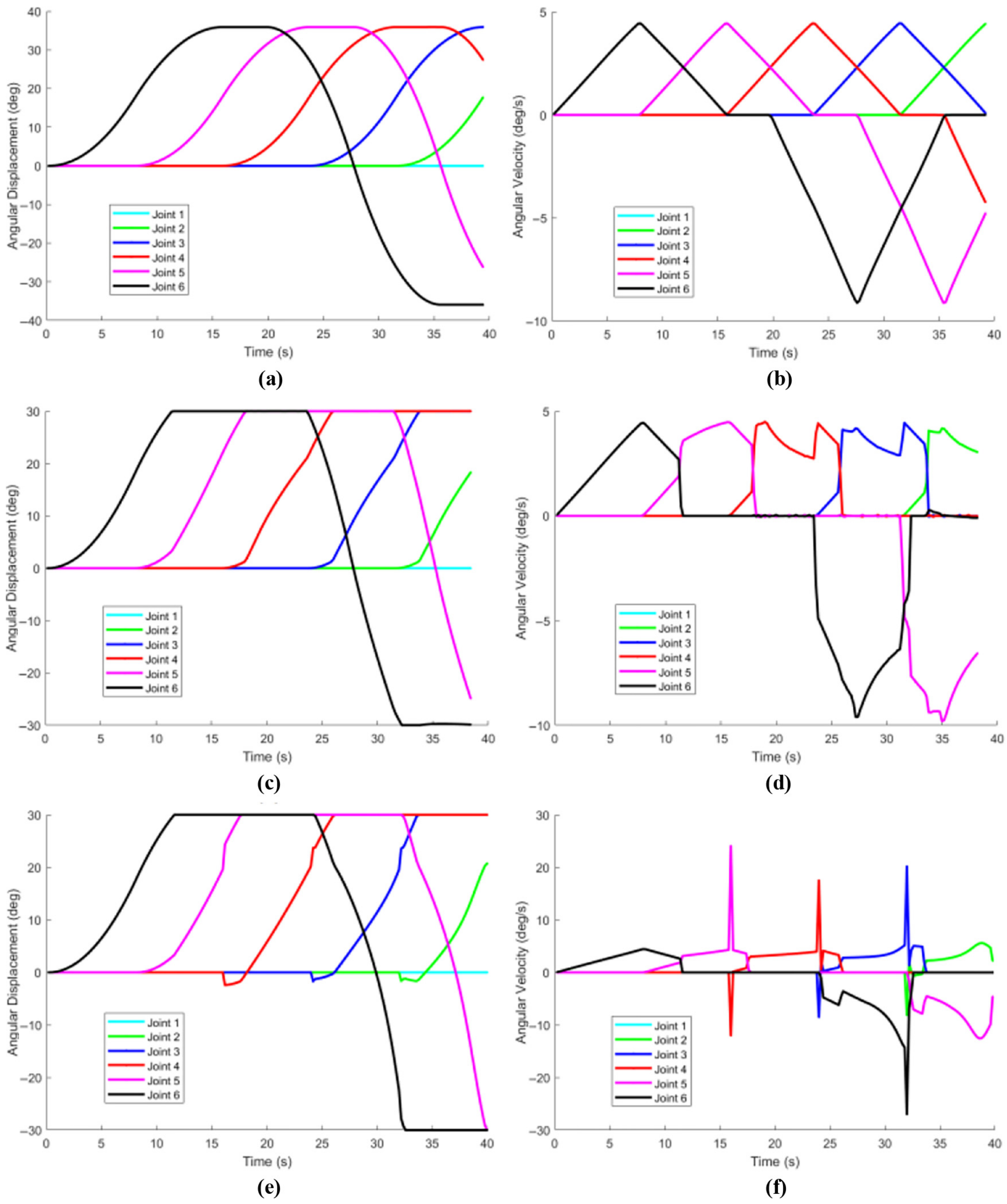
Figure 17 Calculation time for different iteration accuracy



shown in Figure 16. Because the non-linear constrained optimization requires a lot of calculation, the calculation time of the numerical method is much higher than the heuristic method.

The simulation results of the calculation efficiency for different path-following method when there are totally 200 sampling points on the S-bends path is shown in Table 3, for path-following method by trigonometry and path-following method by proportional approximation, the rotation angle tolerance of the iteration is set to 0.01°. Simulation results show that the calculation time of iterative path-following method with joint limits is much less than that of numerical

Figure 18 Movement smoothness analysis of path-following method for S-bend path



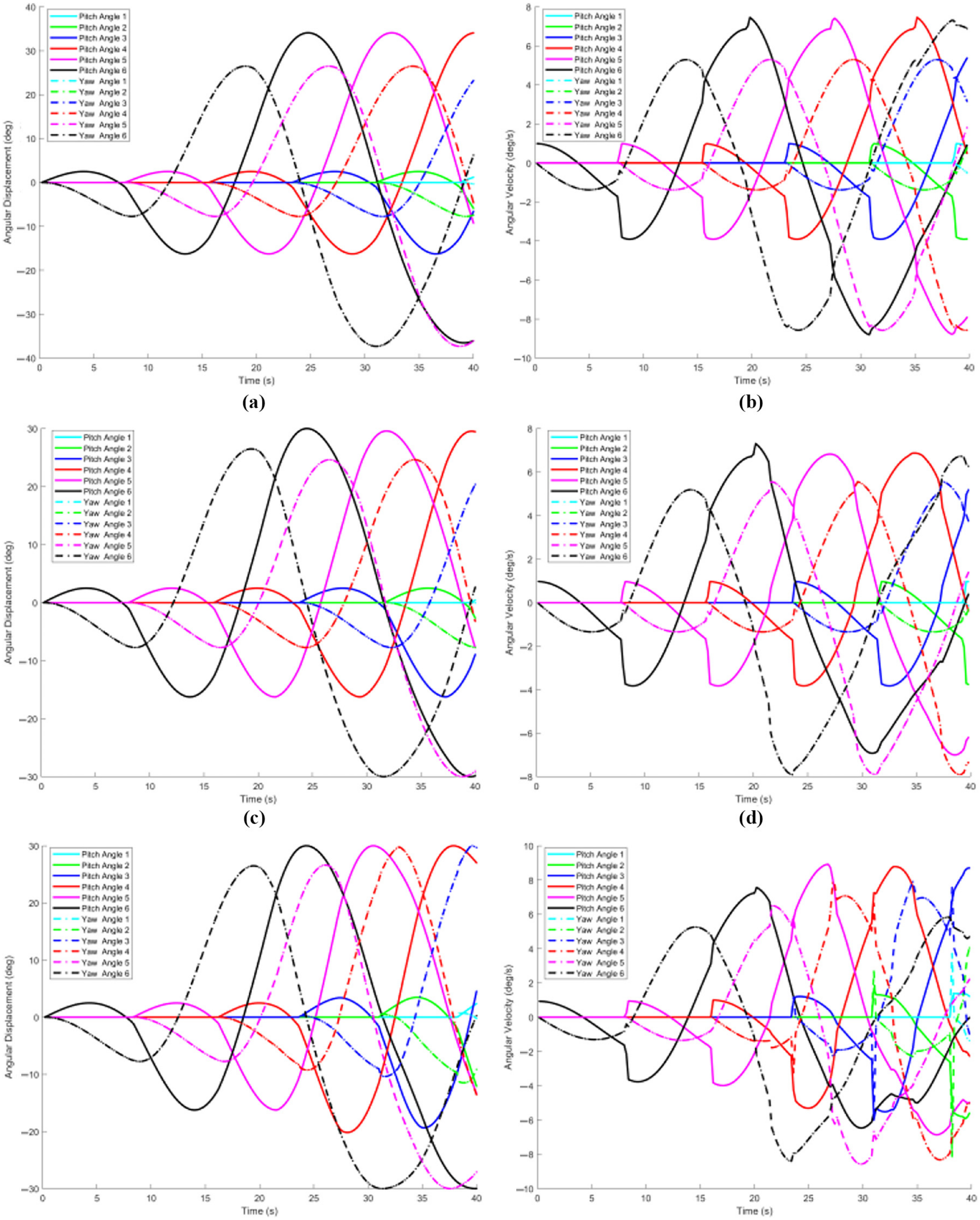
(a) (b)

(c) (d)

(e) (f)

Notes: (a) Angular displacement of path-following method without joint limits; (b) angular velocity of path-followin without joint limits; (c) angular displacement of path-following method with joint limits; (d) angular velocity of path method with joint limits; (e) angular displacement of numerical path-following method; (f) angular velocity of nume path-following method

Figure 19 Movement smoothness analysis of path-following method for helix path



(a)

(b)

(c)

(d)

(e)

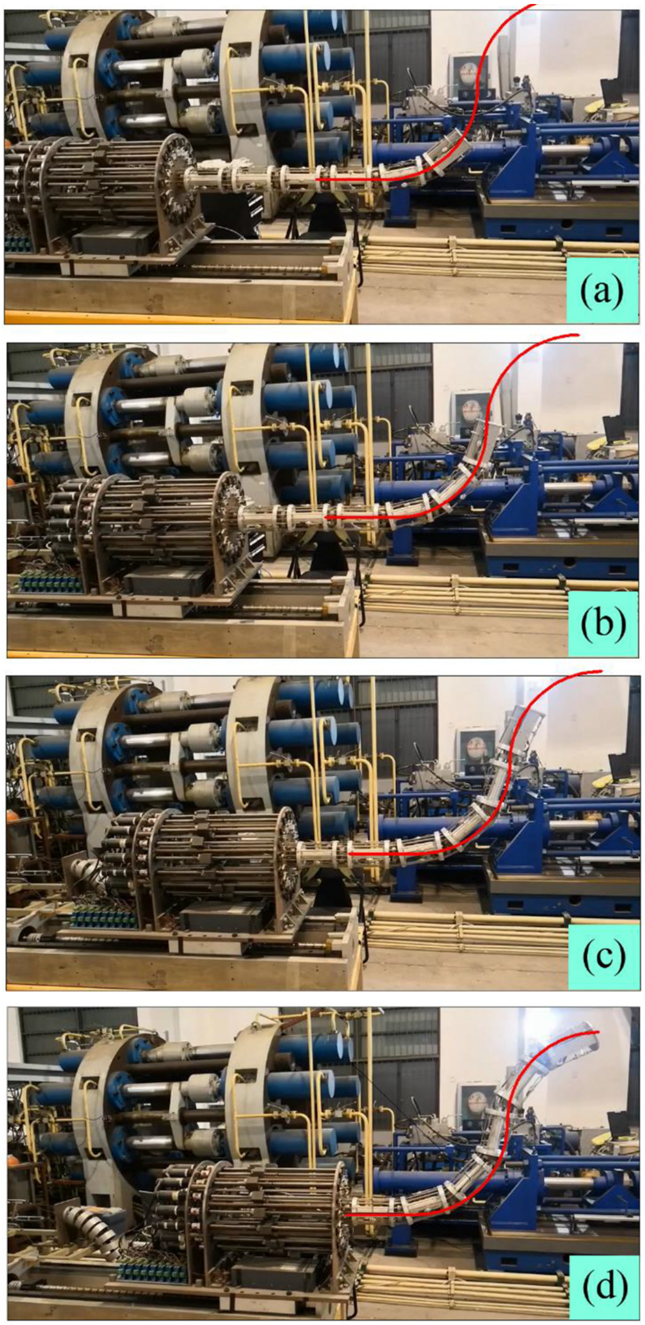
(f)

Notes: (a) Angular displacement of path-following method without joint limits; (b) angular velocity of path-following method without joint limits; (c) angular displacement of path-following method with joint limits; (d) angular velocity of path following method with joint limits; (e) angular displacement of numerical path-followingmethod; (f) angular velocity of numerical path-following method optimization method, about one in two thousands of it. Moreover, for the two heuristic path-following method, the calculation time of trigonometry approach is slightly less than that of proportional approximation approach, thus the triangular interpolation method is finally adopted.

The calculation efficiency of iterative path-following method with joint limits is also affected by the rotation angle tolerance of the iteration. The simulation of iterative path-following method with different iteration tolerance for S-bend path is carried out, and the number of iterations and its calculation time is recorded. Table 4 shows the number of iterations path-following method by trigonometry for S-bend path when the iteration tolerance varies from 1° to 0.001° under 200 sampling steps, and Figure 17 shows the corresponding calculation time.

Simulation results show that when iteration tolerance is less than 0.1°, the number of iterations increases exponentially. According to the data in Tables 1 and 2, when the iteration tolerance is 0.01°, the actual maximum rotation angle is 29.99°, it is close enough to the joint limit. Therefore, considering the calculation efficiency and joint limits, it is suitable to set the iteration tolerance at 0.01°.

Figure 20 Photo series of a hyper-redundant manipulator following S-bends path



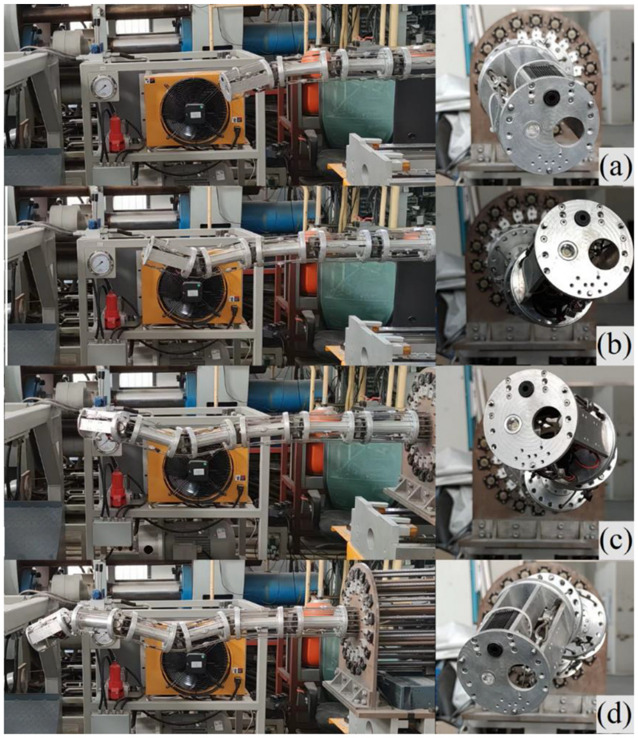
Notes: (a) 10 s; (b) 20 s; (c) 30 s; (d) 40 s

4.3 Movement smoothness

The movement smoothness is analyzed in terms of the velocity of the feed-in mechanism and the angular velocity of the joints.

Previous contents discuss the accuracy and efficiency of the path-following method, and it is necessary to analyze the movement smoothness of the manipulator when it is running directly according to the path-following method to evaluate the practical operating result of the method. The simulation is set to perform one step every 0.2 s, and the angular displacement and velocity of each joint of the manipulator are recorded to evaluate the smoothness of path-following movement. Figure 18 is the movement smoothness analysis of path-following method for S-bend path, as the manipulator only performs plane motion when following S-bend path, only pitch angle of each joint is shown in Figure 18.

Figure 21 Photo series of a hyper-redundant manipulator following helix path



Notes: (a) 10 s; (b) 20 s; (c) 30 s; and (d) 40 s

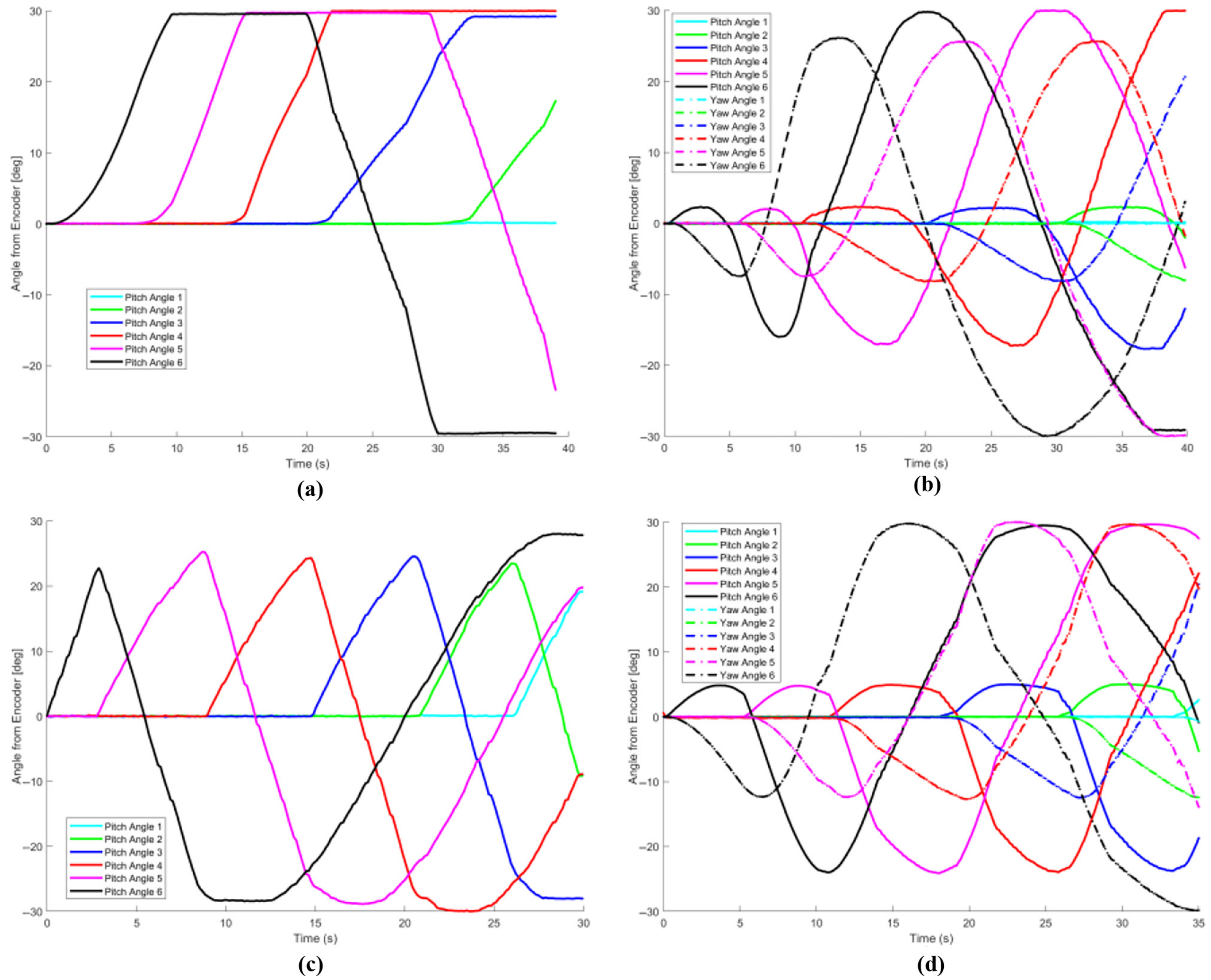
Figure 18(a), (b) show the angular displacement and velocity of each joint under path-following method without joint limits, angular displacement is very smooth, and the curve of angular velocity is continuous. Figure 18(c), (d) show the angular displacement and velocity of each joint under path-following method by trigonometry, the angular displacement is smooth, the maximum angular displacement does not exceed 30°, and there are small fluctuations in the curve of angular velocity. At this time, the movement is still smooth. Figure 18(e), (f) show the angular displacement and velocity of each joint under numerical path-following method, the maximum angular displacement is less than 30°, but the angular velocity curve shows a significant jump of more than 20 deg/s. In this case, it is difficult for most hyper-redundant manipulator to follow this path.

Figure 19 is the graph of angular displacement and angular velocity of three path-following methods for helix path. For the two heuristic path-following methods in Figure 19(a), 19(b) and 19(c), 19(d), angular displacement curve and velocity curve are similar. After the joint constraints are applied, the movement smoothness is not affected and the rotation angles can be limited within 30° in Figure 19(c). But for numerical path-following method in Figure 19(e), (f), there are some obvious peaks in the angular velocity curve. According to the simulation data, the angular velocity has spikes of more than 2 deg/s at 23.6 s, 31 s and 38.2 s. The largest spike appears at the yaw angle 2 of 38.2 s, reaching 7.5 deg/s. This will bring great difficulties to the control of the manipulator, resulting in reduced motion accuracy. It can be concluded from the simulation results of following S-bends and helix path that the smoothness of motion of the heuristic path planning method is better than that of the numerical method.

4.4 Demonstration on physical hardware

The proposed path-following method with joint limits is performed on a 12 DOFs hyper-redundant manipulator to

Figure 22 The joint angles collected from embedded encoders in the experiments



(a)

(b)

(c)

(d)

Notes: (a) The bending angles of joints in S-bends path; (b) the bending angles of joints in helix path; (c) the bending angles of joints in sinusoidal path; (d) the bending angles of joints in helix path prove its effectiveness. The experiment is performed based on the developed kinematics models and path-following method. The prototype of the manipulator has 6 sections, each section is 185 mm in length and the joint limit is 30°. As shown in Figure 20, the experiment (Demonstration video: https://youtu.be/Qt4ODORu4PM) uses the path-following method with joint limits based on trigonometry and the same S-Bends path as in the simulation. Performed at a rate of 0.2 s per step, the path-following motion was competed as expected in 40 s.

The experiment for path-following method in 3D space is also performed. In Figure 21, the manipulator is following the helix path thorough path-following method with joint limits based on trigonometry. The path-following motion of the manipulator is completed within 40 s just as the simulation result.

To verify the simulation results and evaluate the path following performance. Two magnetic rotary encoders are embedded into each universal joint, which can measure the rotation angles of each section. Using the 12-axis encoders, the 6 pitch angles and 6 yaw angles of the manipulator are collected every 50 ms in the experiments.

The rotation angles of S-bends path and helix path measured in the experiments are shown in Figure 22; the experiments in Figure 22(a) and (b) use the same paths as in the above simulation part. The experiments of a sinusoidal path and another helix path in Figure 22(c) and (d) was also performed to better verify the effectiveness of the proposed path-following method, the expressions of the paths are equation (21) and (22):

$$\begin{cases} y = 30t \\ z = 95\sin(\pi \cdot \mathrm{t}/15) \end{cases} \tag{21}$$

$$\begin{cases} x = 90t \cdot \sin(\pi \cdot t/20) \\ y = 24t \\ z = 90t \cdot \cos(\pi \cdot t/20) \end{cases} \tag{22}$$

The results prove that the kinematics model and path-following method with joint limit in the paper is effective, all the joint angles are within 30°. Compared with the simulation results in Figure 18(c) and 19(c), there are some errors in the experimental results of Figure 22. The reason for the error is that the proposed method is a path planning method, which is to find a series of target points to pass through, only joint position and joint angle are discussed here. Thus, the elasticity of the driving cable, friction, gravity and the clearance between the cable and the cable hole are not considered and the manipulator works in open-loop, close-loop control based on joint angle detection will be carried out in future work.

5. Conclusions

The main contribution of this paper is to propose a new heuristic iterative path-following method with joint limits to solve the problem of hyper-redundant manipulator exceeding joint limit in path planning process. The proposed method firstly finds the joint position quickly through a geometrical approach, and then an iterative algorithm is used to reposition the joints to satisfy the joint limits. Through a series of iterative processes, this method can make the manipulator approach the desired path as much as possible within the joint constraints with high precision and less computation time.

The proposed method is compared with some of the most popular existing methods regarding accuracy, computational cost and movement smoothness. Simulation results show that when the iteration tolerance of the imposed joint limit is set to 0.01°, the average computation time is 0.1 ms per step for hyper-redundant manipulator with 12 DOFs, iterative process has little effect on the calculation time, and the deviation of the tip can be kept below 0.02 mm. Compared with more sophisticated methods, the proposed path-following method can follow the desired path with very low computational cost and high accuracy.

Currently, the manipulator is working in open-loop, the elasticity of the driving cable will cause positioning error. In future, close-loop control based on real-time attitude detection will be used in in combination with the path-following method to achieve high-precision trajectory tracking.

References

Aristidou, A. and Lasenby, J. (2011), “FABRIK: a fast, iterative solver for the inverse kinematics problem”, *Graphical Models*, Vol. 73 No. 5, pp. 243-260.

Buckingham, R. (2002), “Snake arm robots”, *Industrial Robot: An International Journal*, Vol. 29 No. 3, pp. 242-245.

Buckingham, R. and Graham, A. (2013), “Snaking around in a nuclear jungle”, *Industrial Robot: An International Journal*, Vol. 32 No. 2, pp. 120-127.

Chirikjian, G.S. and Burdick, J.W. (1990), “An obstacle avoidance algorithm for hyper-redundant manipulators”, *IEEE International Conference on Robotics and Automation, 1990. Proceedings*, Vol. 1, pp. 625-631.

Choset, H. and Henning, W. (1999), “A follow-the-leader approach to serpentine robot motion planning”, *Journal of Aerospace Engineering*, Vol. 12 No. 2, pp. 65-73.

Corporate, F. (2010), “Bionic Handling assistant–flexible and compliant movement. from bionic learning network”,

Hannan, M.W. and Walker, I.D. (2003), “Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots”, *Journal of Robotic Systems*, Vol. 20 No. 2, pp. 45-63.

Hannan, M.W. and Walker, I.D. (2002), “The ’elephant trunk’ manipulator, design and implementation”, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2001. Proceedings*, Vol. 1, pp. 14-19.

Kamegawa, T., Yamasaki, T., Igarashi, H. and Matsuno, F. (2004), “Development of the snake-like rescue robot ‘KOHGA’”, *2004 IEEE International Conference on Robotics and Automation, Vols 1- 5, Proceedings*, 5081-5086.

Liu, B., Huang, L., Yin, L., Zhang, P. and Yi, K. (2021), “Design and analysis of a tendon-driven snake-arm robot based on a spherical magnets”, *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 46 No. 1, pp. 68-76.

Mcmahan, W., Chitrakaran, V., Csencsits, M., Dawson, D., Walker, I.D., Jones, B.A., Pritts, M., Dienno, D., Grissom, M. and Rahn, C.D. (2006), “Field trials and testing of the OctArm continuum manipulator”, *2006 Ieee International Conference on Robotics and Automation (Icra)*, Vol. s 1-10, pp. 2336. +.

Oc, R. (2008), “Snake-arm robots access the inaccessible”, *Nuclear Technology International*, Vol. 1, pp. 92-94.

Palmer, D., Cobos-Guzman, S. and Axinte, D. (2014), "Real-time method for tip following navigation of continuum snake arm robots", *Robotics and Autonomous Systems*, Vol. 62 No. 10, pp. 1478-1485.

Qin, G., Ji, A., Cheng, Y., Zhao, W., Pan, H., Shi, S. and Song, Y. (2021), "A Snake-Inspired Layer-Driven continuum robot", *Soft Robotics*, Vol. 9 No. 4.

Qin, G., Wu, H., Cheng, Y., Pan, H., Zhao, W., Shi, S., Song, Y. and Ji, A. (2022), "Adaptive trajectory control of an under-actuated snake robot", *Applied Mathematical Modelling*, Vol. 106, pp. 756-769.

Robinson, G. and Davies, J.B.C. (2002), "Continuum robots - a state of the art", *IEEE International Conference on Robotics and Automation, 1999. Proceedings*, Vol. 4, pp2849-2854.

Sreenivasan, S., Goel, P. and Ghosal, A. (2010), "A real-time algorithm for simulation of flexible objects and hyper-redundant manipulators", *Mechanism and Machine Theory*, Vol. 45 No. 3, pp. 454-466.

Walker, I.D. and Hannan, M.W. (1999), "A novel `elephant's trunk' robot", *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 1999. Proceedings*, pp. 410-415.

Webster, R.J. III. and Jones, B.A. (2010), "Design and kinematic modeling of constant curvature continuum robots: a review", *The International Journal of Robotics Research*, Vol. 29 No. 13, pp. 1661-1683.

Williams, R.L. (1995), "Follow-the-leader algorithm for the payload inspection and processing system", *The 1995 Research Reports: NASA/ASEE Summer Faculty Fellowship Program*, pp. 613-640, available at: https://ntrs.nasa.gov/citations/19960020735

William, R.L.M.I.B. II, (1997), "Obstacle-free control of the hyper-redundant NASA inspection manipulator", *In Proc. of the Fifth National Conf. on Applied Mechanics and Robotics*, pp. 12-15.

Xie, H.B., Wang, C., Li, S.S., Hu, L. and Yang, H.Y. (2019), "A geometric approach for follow-the-leader motion of serpentine manipulator", *International Journal of Advanced Robotic Systems*, Vol. 16 No. 5.

Yuan, F., Zhao, Y., Zhang, Y., Zhang, X. and Lu, X. (2022), "An iterative algorithm for inverse displacement analysis of hyper-redundant elephant's trunk robot", *Robotica*, pp. 1-18.

Zheng, Y., Wu, B., Chen, Y., Zeng, L., Gu, G., Zhu, X. and Xu, K. (2021), "Design and validation of cable-driven hyper-redundant manipulator with a closed-loop puller-follower controller", *Mechatronics*, Vol. 78, p. 102605.

Demonstration video of the path-following motion. [Online], available at: https://youtu.be/Qt4ODORu4PM

Corresponding author

Haibo Xie can be contacted at: 11425049@zju.edu.cn