Robotica: (2021) volume 39, pp. 503–523. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000533

Head-Raising Method of Snake Robots Based on the Bézier Curve

Yunhu Zhou, Yuanfei Zhang*, Fenglei Ni and Hong Liu

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, 150080, China

(Accepted May 28, 2020. First published online: June 30, 2020)

SUMMARY
For acquiring a broad view in an unknown environment, we proposed a control strategy based on the Bézier curve for the snake robot raising its head. Then, an improved discretization method was developed to accommodate the backbone curves with more complex shapes. Besides, in order to determine the condition of using the improved discretization method, energy of framed space curve is introduced originally to estimate the shape complexity of the backbone curve. At last, based on degree elevation of the Bézier curve, an obstacle avoidance strategy of the head-raising motion was proposed and validated through simulation.

KEYWORDS: Head-raising; Backbone curve; Bézier curve; Discretization; Obstacle avoidance.

1. Introduction

A snake robot is a mobile robot with hyper-redundant degrees of freedom (DOFs).[1–3] Therefore, it can move flexibly and adapt to the complex environment and it plays a prominent role in the search and rescue operation, inspection, and maintenance.[4,5] For completing a specific task in the complex environment, it is important that head-raising motion can help the snake robot obtain a broad visual space and act as a flexible manipulator.[4] However, the redundant DOFs of the robot increase the difficulty of control of the head-raising motion. So, the motion planning of the snake robot is a key problem.

Tanaka et al. proposed the head control method of the planar locomotion based on the non-side-slipping constraint for the wheeled snake robot.[6] Then, they proposed the head-raising model using the kinematic redundancy.[7] But these methods are only suitable for the wheeled snake robot because of non-side-slipping constraint. However, snake robots without wheels are widely studied for their better mobility and adaptability in the clutter environment.[8,9] In addition, Ye et al. proposed the head-raising method based on the modified serpentine curve.[10] However, the inflexibility of the serpentine curve limits the motion range of the head. Then, a predefined spiral curve method for the head-raising of the snake robot is proposed by Zhang et al.[4] But the head of the snake robot has a rather small workspace because of the inflexibility of the spiral curve. Besides, the head-raising motion is realized after the snake robot stops moving. Before the snake robot stops moving, the rolling gait or turning-in-place gait can be adopted to move forward or turn around.[8,11] The way of raising head after stopping is adopted in wheeless snake robots field.[4,12]

For realizing the head-raising motion, the backbone curve theory is adopted.[13] In order to overcome these shortcomings, we propose the head-raising method based on the Bézier curve. The Bézier curve has the rather high flexibility of its construction.[14] The robot is divided into two parts, the base part and the upper part. The backbone curve of the base part is designed as a planar arc to ensure the stability of the robot as much as possible. For the backbone curve of the upper part, it should be

* Corresponding author. E-mail: yuanfei.zhang@hit.edu.cn

constructed and modified flexibly so that the head can have a rather large workspace. Spline curves and Bézier curves are widely used in computer graphics.[15] The spline curves have been used in the kinematics of the hyper-redundant manipulators.[16,17] But, it is hard to ensure the smoothness of the connecting point and the local adjustment for the piecewise curve by the spline curve at the same time. Liljebäck et al. used the planar Bézier curve to construct the control frame for the snake robot locomotion.[18] In this paper, the space Bézier curve is adopted to design the backbone curve of the upper part. Then the local adjustment and piecewise construction for the Bézier curve can be easier to implement. In order to determine the degree of the Bézier curve, constraints including the arc length, second-order derivative continuity at the connection point, and the direction of the tangent line at the end point are used. And it is proved that the second-order derivative continuity at the connecting point of the backbone curve can avoid the sharp change of the phase angle. Once the design of the piecewise backbone curve is completed, the backbone curve can be discretized by the discretization method based on its curvature and torsion.[19,20] However, the discretization method is not applicable to the backbone curve with a large shape change. Then, a variable called energy of the framed space curve is introduced in computer graphics.[21] We originally applied it to estimate the shape change of the backbone curve. Hatton et al. proposed an annealed chain-fitting method,[22] but it is computationally expensive.[8] And it is not applicable to the backbone curve of the head-raising method. In order to decrease the discretization error of the backbone curve with a large shape change, a modified discretization method based on the optimization is proposed. Then the zero-moment point (ZMP) method and the support polygon method are adopted to analyze the stability of the motion. The two methods both have an assumption that the ground is a plane. However, the assumption can simplify the computation of the stability of snake robots though it has some limitations. Thus, the assumption is adopted by several researchers.[4,7,10] Besides, the ZMP method is also widely used in the humanoid robotics field.[23,24] Then the proposed head-raising method and the predefined spiral curve method are compared in terms of the size of the workspace obtained by them. At last, the strategy of obstacle avoidance of the head based on the degree elevation of the Bézier curve is proposed. The simulation verifies the effectiveness of the obstacle avoidance strategy.

The article is organized as follows. In the second section, the motion model of the head-raising is constructed. The Bézier curve is briefly introduced at first. Then the backbone curve of the base part is designed and optimized. The Bézier backbone curve for the upper part is determined, and the discretization method is modified for adapting to more complex backbone curve. What's more, the stability analysis based on the ZMP method is given. In the third section, the proposed head-raising method is compared with the predefined spiral curve method. And the relationship between the threshold and discretization error for the modified discretization method is studied. In the fourth section, the obstacle avoidance strategy is studied which is based on the degree elevation of the Bézier curve. The simulation result validates the effectiveness of the strategy. In the fifth section, the conclusions and future works are presented.

2. The Motion Model of Head-Raising

In this section, the head-raising method based on the Bézier curve and the improved discretization method are developed. The energy of the framed space curve is used to estimate the shape of backbone curve. In addition, the ZMP method and the support polygon method used for the stability analysis are presented.

2.1. An introduction to the Bézier curve

The Bézier curve is widely used in the computer graphics design.[15,25] It is a parametric curve based on the Bernstein polynomial. It is constructed on some control points and builds some smooth curves easily. The shape of the Bézier curve can be changed by moving these control points. The expression of the Bézier curve is the combination of the control points and Bernstein polynomial so it looks very intuitive, compact, and flexible. The control points of the Bézier curve are assumed as $\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2,...,$ $\boldsymbol{P}_{h-1}, \boldsymbol{P}_h$. The Bernstein polynomial is expressed as

$$B_{h,i}(t) = \frac{h!}{i!(h-i)!} t^i (1-t)^{h-i} \tag{1}$$

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

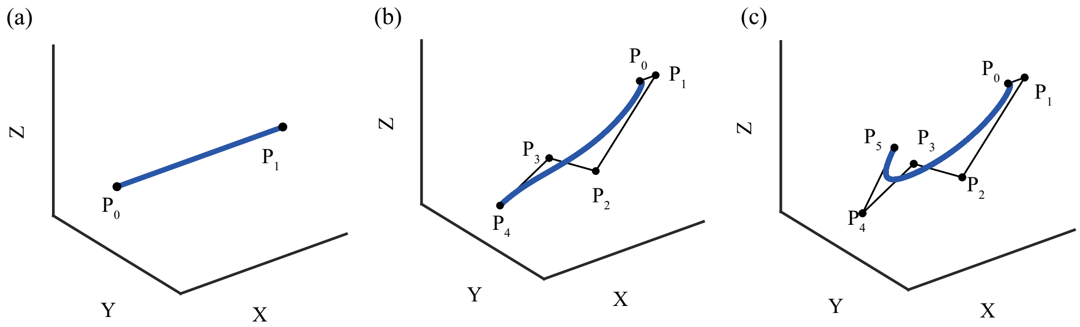

Fig. 1. (a) The linear Bézier curve. (b) The Bézier curve of degree 4. (c) The Bézier curve of degree 5.

where $t \in [0, 1]$ is a parameter for the curve. And the expression of the Bézier curve of degree h can be represented as

$$\boldsymbol{C}_h(t) = \sum_{i=0}^{n} B_{h,i}(t)\boldsymbol{P}_i \tag{2}$$

Then we can construct a linear Bézier curve firstly,

$$\boldsymbol{C}_1(t) = (1-t)\boldsymbol{P}_0 + t\boldsymbol{P}_1 \quad t \in [0, 1] \tag{3}$$

The equation represents a location moving from $\boldsymbol{P}_0$ to $\boldsymbol{P}_1$ along the curve when t increases. In this article, the Bézier curve of degree 4 and Bézier curve of degree 5 are used to design the backbone curve. So the Bézier curve of degree 4 is given as

$$\boldsymbol{C}_4(t) = \boldsymbol{P}_0(1-t)^4 + 4\boldsymbol{P}_1 t(1-t)^3 + 6\boldsymbol{P}_2 t^2(1-t)^2 + 4\boldsymbol{P}_3 t^3(1-t) + \boldsymbol{P}_4 t^4 \quad t \in [0, 1] \tag{4}$$

And the Bézier curve of degree 5 is given as

$$\begin{aligned}\boldsymbol{C}_5(t) = \boldsymbol{P}_0(1-t)^5 + 5\boldsymbol{P}_1 t(1-t)^4 + 10\boldsymbol{P}_2 t^2(1-t)^3 + 10\boldsymbol{P}_3 t^3(1-t)^2 + 5\boldsymbol{P}_4 t^4(1-t)\\ +\boldsymbol{P}_5 t^5 \quad t \in [0, 1]\end{aligned} \tag{5}$$

The linear Bézier curve, Bézier curve of degree 4, and Bézier curve of degree 5 are shown in Fig. 1. The Bézier curve of high degree can ensure the smoothness at the connection of the piecewise curve, but the Bézier curve of low degree can reduce the complexity of the computation.

2.2. The design of the backbone curve

Then the planar arc curve is adopted to design the backbone of the base part, and the quartic Bézier curve is used to design the backbone curve of the upper part of the snake robot. And the length of the entire backbone curve is equal to the length of the entire body of the snake robot. Though the proposed head-raising method can be applied to the snake robots with different kinematic chains,[8,20] the kinematic chain of the snake robot adopted in this article is shown in Fig. 2. The snake robot consists of joints with a single DOF and the axis of each joint is rotated 90 degrees relative to the previous joint axis, which is shown in Fig. 2(a). The configuration in Fig. 2 is widely adopted in the snake robot field.[8,26,27] And the snake robot in this article is divided into the base part and upper part, which is shown in Fig. 2(b).

2.2.1. The backbone curve of the base part. For the process of head-raising of the snake robot, the function of the base part is to ensure the stability of the body of the snake robot as much as possible. The support polygon method and ZMP method are usually used to determine the stability of the robot. And Ye et al. first introduced the ZMP method to determine the stability of the head-raising motion for the snake robot.[10] For the two methods, the projections of the center of mass of the body and the ZMP of the body on the ground inside the support polygon indicate that the motion of the body is stable. So the body will have a greater possibility of the stable standing when the support polygon constructed by the base part has a larger area. As we know, for a given perimeter of the

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

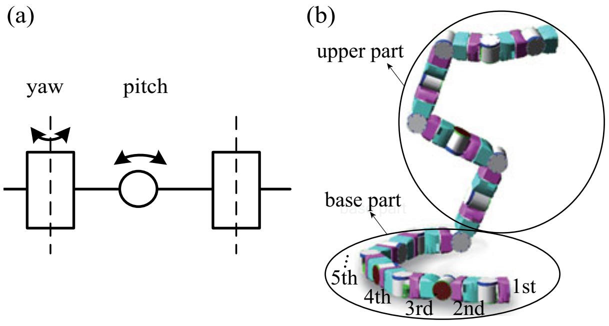

Fig. 2. (a) The axis of each joint is rotated 90 degrees relative to the previous joint axis; pitch joint generates rotation in the plane, while yaw joint generates rotation out of the plane in this illustration. (b) The structure of the snake robot.

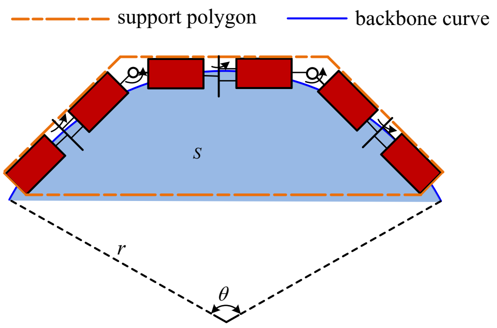

Fig. 3. The backbone curve of the base part and the support polygon constructed by the base part.

polygon, the circle has a largest coverage area in all the plane polygons. So we choose to design the backbone of the base part as a planar arc. Then we use the backbone curve theory to discretize the backbone curve to generate the joint angles of the base part. The support polygon and the backbone curve are shown in Fig. 3.

For the planar arc, the arc length, which equals the length of the base part of the snake robot, is assumed as *l*, and the radius of the arc is *r*. The central angle of the plane arc is θ. The equations are

$$r = \frac{l}{\theta} \tag{6}$$

$$S = \frac{1}{2} r^2 (\theta - \sin(\theta)) \quad \theta \in (0, 2\pi) \tag{7}$$

where *S* indicates the area enclosed by the backbone curve of the base part, which is shown in Fig. 3. Then the area of the support polygon has a maximum value. When θ equals π, *S* has the maximum value, $1/2\pi r^2$. So the central angle θ of the arc is determined to be π and the radius *r* is calculated as l/π. Then the backbone curve of the base part of the snake robot can be discretized to calculate the joint angles for the base part. The discretization method based on the curvature and torsion of the backbone curve is given as,[8,20,28]

$$\begin{cases} \kappa_y(s) = \kappa(s)\cos(\phi(s)) \\ \kappa_p(s) = \kappa(s)\sin(\phi(s)) \end{cases} \tag{8}$$

$$\phi(s) = \phi_0 + \int_0^s \tau(s)\mathrm{d}s \tag{9}$$

where *s* is the arc length of the backbone curve. $\kappa(s)$ and $\tau(s)$ are the curvature and torsion of the backbone curve respectively. $\kappa_y(s)$ and $\kappa_p(s)$ are the projections of the curvature in the yaw joint and pitch joint respectively, which are shown in Fig. 2. ϕ_0 is the initial phase angle of the discretization and $\phi(s)$ is the phase angle of the discretization.

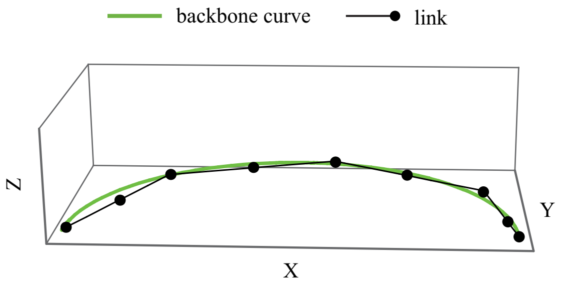

Fig. 4. The discretization of the planar arc backbone curve.

$$\begin{cases} \theta_y = \int_{s_{i-1}}^{s_{i+1}} \kappa_y(s)\mathrm{d}s & \text{yaw} \quad \text{joint} \quad i = 2, 4, 6, \ldots \\ \theta_p = \int_{s_{i-1}}^{s_{i+1}} \kappa_p(s)\mathrm{d}s & \text{pitch} \quad \text{joint} \quad i = 1, 3, 5, \ldots \end{cases} \tag{10}$$

where θ_y and θ_p are the joint angles of yaw joints and the joint angles of pitch joints for the snake robot, respectively. Here s_0 is 0. And s_i represents the sum of lengths of the first i links. The discretization result of the backbone curve of the base part is shown in Fig. 4.

2.2.2. The backbone curve of the upper part. For the backbone curve of the upper part of the snake robot, it should connect with the backbone curve of the base part smoothly. In general, it should be ensured that at least the first derivative of the piecewise backbone curve is continuous at the connecting point. But there will be a sharp variation of the phase angle of the discretization at the connecting point of the backbone curve if only the first derivative of the backbone curve is continuous at the connecting point, which is introduced in refs. [4, 29]. The problem will lead to sharp changes in motion trajectories.[4] In order to ensure the smoothness of the junction point for the backbone curve, at least first-order derivative continuity or at least second-order derivative continuity should be satisfied. Since the general parameter representation of the parametric curve is easier than the arc length parameter representation to calculate, we adopt the general parameter representation of the backbone curve. And we assume that the planar arc and Bézier curve are denoted as $\boldsymbol{C}_a(t), t \in [0, \pi]$ and $\boldsymbol{C}_b(t), t \in [0, 1]$. Then the conditions of the smoothness at the connecting point are given,

$$\boldsymbol{C}'_a(\pi) = \boldsymbol{C}'_b(0) \tag{11}$$

$$\boldsymbol{C}''_a(\pi) = \boldsymbol{C}''_b(0) \tag{12}$$

where symbol $'$ and symbol $''$ indicate the first-order derivative and the second-order derivative respect to the parameter t. The case of satisfying Eq. (11) is C^1 continuous. And the case of satisfying Eqs. (11) and (12) is C^2 continuous.

The Frenet–Serret frame can be calculated as follows,[30]

$$\begin{cases} \boldsymbol{T}(t) = \frac{\boldsymbol{C}'(t)}{\|\boldsymbol{C}'(t)\|} \\ \boldsymbol{N}_s(t) = \frac{\boldsymbol{C}''(t)}{\|\boldsymbol{C}''(t)\|} \\ \boldsymbol{B}(t) = \boldsymbol{T}(t) \times \boldsymbol{N}_s(t) \\ \boldsymbol{N}(t) = \boldsymbol{B}(t) \times \boldsymbol{T}(t) \end{cases} \tag{13}$$

where $\boldsymbol{C}(t)$ is a parametric curve. The $\boldsymbol{T}(t)$, $\boldsymbol{N}(t)$, and $\boldsymbol{B}(t)$ are the unit tangent vector, the unit normal vector, and the unit binormal vector at parameter t, respectively, which construct a Frenet–Serret frame. In Fig. 5(a), the backbone curves of the base part and the upper part are the planar arc and the Bézier curve of degree 2, respectively. The connection of them is C^1 continuous. The Frenet–Serret frames of them at the junction point $\boldsymbol{P}_0$ are $\{\boldsymbol{P}_0 : \boldsymbol{T}_a, \boldsymbol{N}_a, \boldsymbol{B}_a\}$ and $\{\boldsymbol{P}_0 : \boldsymbol{T}_b, \boldsymbol{N}_b, \boldsymbol{B}_b\}$, respectively. The frame $\{\boldsymbol{P}_0 : \boldsymbol{T}_a, \boldsymbol{N}_a, \boldsymbol{B}_a\}$ and the frame $\{\boldsymbol{P}_0 : \boldsymbol{T}_b, \boldsymbol{N}_b, \boldsymbol{B}_b\}$ have a phase angle variation ϕ_e along the axis $\boldsymbol{T}_a$ or $\boldsymbol{T}_b$,

$$\phi_e = ((\boldsymbol{B}_a \times \boldsymbol{B}_b) \bullet \boldsymbol{T}_a) \arccos(\boldsymbol{B}_a \bullet \boldsymbol{B}_b) \tag{14}$$

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

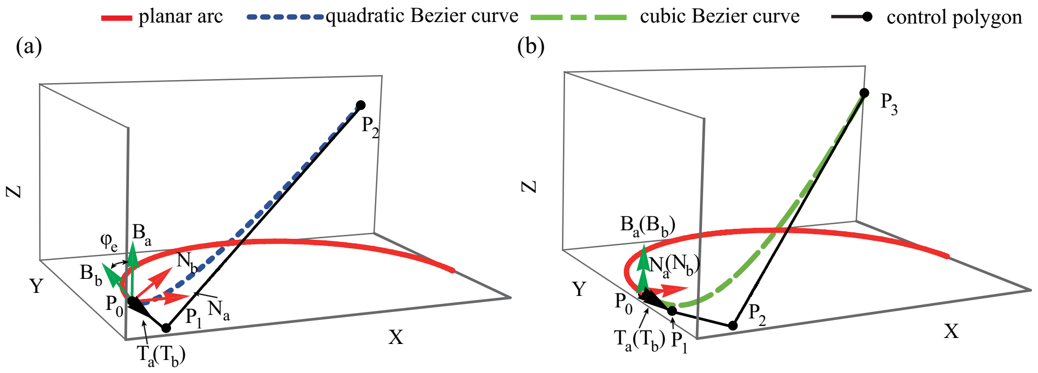

Fig. 5. (a) The backbone curve constructed by the planar arc and the Bézier curve of degree 2. (b) The backbone curve constructed by the planar arc and the cubic Bézier curve.

where $((\boldsymbol{B}_a \times \boldsymbol{B}_b) \bullet \boldsymbol{T}_a)$ is used to describe the signum of ϕ_e. The existence of the phase angle variation causes the non-smoothness of backbone curve discretization and the sharp changes in some motion trajectory for the snake robot. In Fig. 5(b), the backbone curves of the base part and the upper part are the planar arc and the cubic Bézier curve, respectively, whose connection is C^2 continuous. The frame $\{\boldsymbol{P}_0 : \boldsymbol{T}_a, \boldsymbol{N}_a, \boldsymbol{B}_a\}$ and the frame $\{\boldsymbol{P}_0 : \boldsymbol{T}_b, \boldsymbol{N}_b, \boldsymbol{B}_b\}$ have a phase angle variation ϕ_e along the axis $\boldsymbol{T}_a$ or $\boldsymbol{T}_b$ is 0. So the C^2 continuity can avoid these problems.

For the Bézier curve, the first-order and second-order derivative at $\boldsymbol{P}_0$ is controlled by the first two control points, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. Besides, there are also some restrictions that the backbone curve of the snake robot should satisfy. The restrictions are the arc length of the backbone curve, the desired tangent direction, and the desired position of end point of the backbone curve. Assuming that the backbone curve of upper part is the Bézier curve of degree h, the desired position of the end point is the known control point $\boldsymbol{P}_h$. The rest restrictions can be represented as,

$$\int_0^{\pi} \|\boldsymbol{C}'_a(t)\|dt + \int_0^1 \|\boldsymbol{C}'_b(t)\|dt = \sum_{i=1}^{n} l_i \tag{15}$$

$$\frac{\boldsymbol{C}'_b(1)}{\|\boldsymbol{C}'_b(1)\|} \times \boldsymbol{r}_a = \boldsymbol{0} \tag{16}$$

where n denotes the number of the links of the snake robot. l_i denotes the length of i-th link of the snake robot. And $\boldsymbol{r}_a$ is a desired unit tangent vector of the end point of the backbone curve. $\boldsymbol{0}$ is a 3×1 zero vector. The way in which Eq. (16) describes the end direction constraints greatly reduces the computational complexity. Then the backbone curve of the upper part needs five control points to construct a quartic Bézier curve considering these restrictions.

For determining the quartic Bézier curve, we assume that the control points are $\boldsymbol{P}_0(a_0, b_0, c_0)$, $\boldsymbol{P}_1(a_1, b_1, c_1)$, $\boldsymbol{P}_2(a_2, b_2, c_2)$, $\boldsymbol{P}_3(a_3, b_3, c_3)$, and $\boldsymbol{P}_4(a_4, b_4, c_4)$. $\boldsymbol{P}_0$ as the end point of the backbone curve of the base part and $\boldsymbol{P}_4$ as the desired position of the end point of the backbone curve for the upper part are both known. $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ can be determined by Eqs. (11) and (12), respectively. a_3 and b_3 can be represented by c_3 through Eq. (16). Then a_3 and b_3 are substituted into Eq. (15), so a_3, b_3, and c_3 can be solved. In order to avoid the tangent direction of the end point of the Bézier curve converging to the direction $-\boldsymbol{r}_a$ for Eq. (16), a predefined $\boldsymbol{P}_3$ can be set $\boldsymbol{P}_3 = \boldsymbol{P}_4 - k\boldsymbol{r}_a$. k is an adjustable parameter. The initial value of c_3 used for iteration of solving c_3 from Eq. (15) should be determined by the predefined $\boldsymbol{P}_3$. The desired backbone curve solved through these equations is shown in Fig. 6.

2.3. *The modified discretization method*

2.3.1. The discretization error. The discretization method based on the curvature and torsion of the backbone curve is proposed by Yamada et al.[19,20] The method is widely applied in the discretization of the backbone curve which has a small curvature and torsion change with the arc length parameter.[8,31] And the quality of the discretization can be measured by the discretization error,[20]

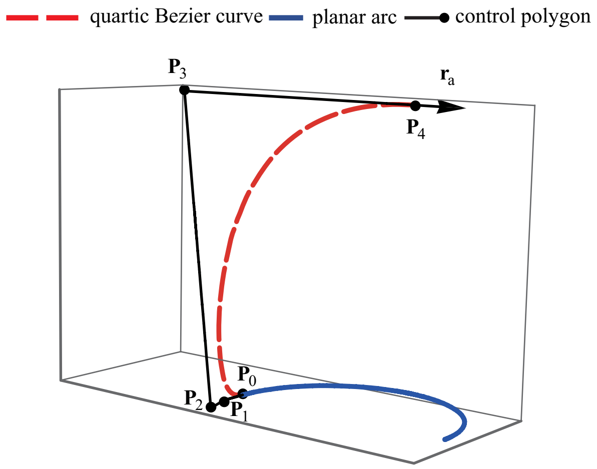

Fig. 6. The backbone curves of the head-raising method corresponding to the base part and upper part are a planar arc and a quartic Bézier curve.

$$E_{nm} = \frac{\int_0^{L_t} \|\boldsymbol{C}(s) - \boldsymbol{C}_d(s)\| ds}{L_t^2} \tag{17}$$

where L_t is the total length of the snake robot. $\boldsymbol{C}(s)$ indicates the backbone curve of the snake robot and $\boldsymbol{C}_d(s)$ is the vector of the position of the point on the link model for the snake robot. E_{nm} denotes the discretization error of the backbone curve. A greater discretization error indicates a worse discretization result. However, for the backbone curve with a large change of the curvature and the torsion, the error of the discretization based on the curvature and the torsion of the backbone curve is rather large.

The discretization of the backbone curve with a small change of the curvature and torsion is shown in Fig. 7(a). And the discretization of the backbone with a large change of the curvature and torsion is shown in Fig. 7(b). They have the same backbone curve of the base part and the same arc length of the entire backbone curve. The relationship between the curvature and the arc length and the relationship between the torsion and the arc length corresponding to the backbone curve in Fig. 7(a) are shown in Fig. 8(a) and Fig. 9(a), respectively. The relationship between the curvature and the arc length and the relationship between the torsion and the arc length corresponding to the backbone curve in Fig. 7(b) are shown in Figs. 8(b) and 9(b), respectively. We can find that the discretization result of the backbone curve with a large curvature and torsion change is worse. The discretization error E_{nm} of the backbone curve in Fig. 7(a) which is calculated by Eq. (17) is 0.00568705. The discretization error E_{nm} of the backbone curve in Fig. 7(b) is 0.0530048. The discretization error of the backbone curve in Fig. 7(b) is nearly 10 times that in Fig. 7(a). So the discretization method is not applicable for the curve with a large change of the curvature and torsion.

2.3.2. The modified discretization method. In order to improve the discretization of the backbone curve with a large curvature and torsion change, a modified discretization method is proposed. In the modified method, an optimization algorithm is adopted at the backbone curve segment where the changes of the curvature and the torsion are relatively large to minimize the discretization error. Firstly, we need an evaluation criterion of the shape change of the backbone curve whose change of the curvature and torsion is relatively large. Borrowing a concept from the computer graphics literature,[21,32] it is called energy of the framed space curve. Then the energy U can be calculated as,

$$U = \int_0^s \omega(s)^2 ds \tag{18}$$

$$\omega(s) = \sqrt{\kappa(s)^2 + \tau(s)^2} \tag{19}$$

$$ds = \|\boldsymbol{C}'(t)\| dt \tag{20}$$

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

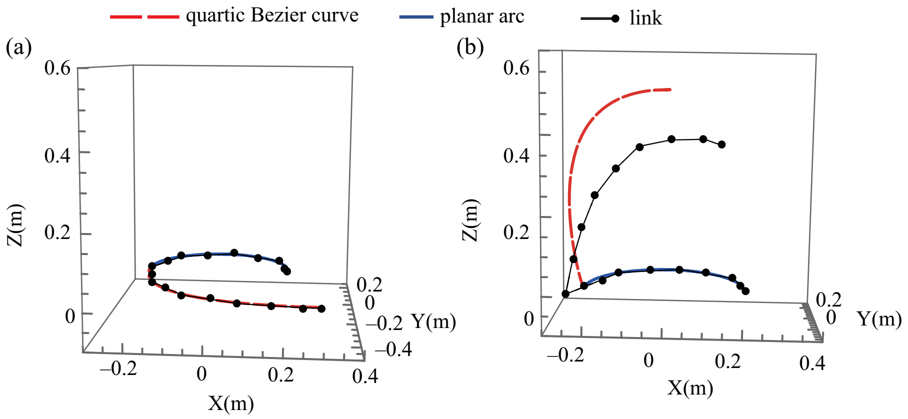

Fig. 7. (a) The discretization result of the backbone curve with a relatively small changes of the curvature and torsion. (b) The discretization result of the backbone curve with a relatively large changes of the curvature and torsion.

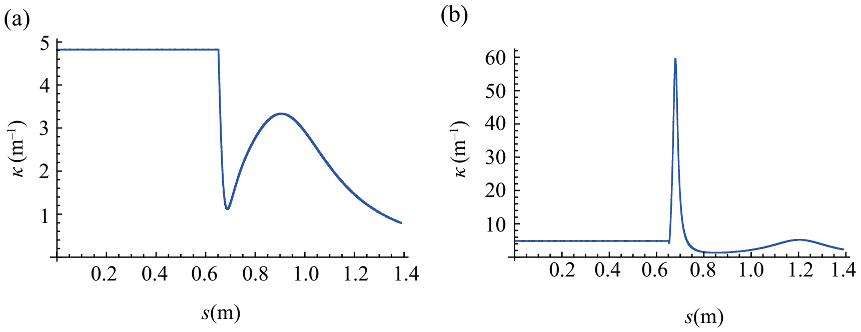

Fig. 8. (a) The relationship between the curvature κ and the arc length s of the backbone curve with a relatively small change of the curvature and torsion. (b) The relationship between the torsion κ and the arc length s of the backbone curve with a relatively large change of the curvature and torsion.

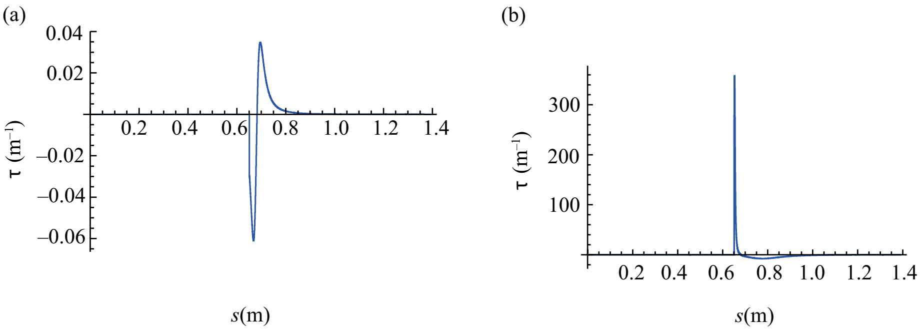

Fig. 9. (a) The relationship between the curvature τ and the arc length s of the backbone curve with a relative small change of the curvature and torsion. (b) The relationship between the torsion τ and the arc length s of the backbone curve with a relative large change of the curvature and torsion.

$$\kappa(t) = \frac{\|\boldsymbol{C}'(t) \times \boldsymbol{C}''(t)\|}{\|\boldsymbol{C}'(t)\|^3} \tag{21}$$

$$\tau(t) = \frac{(\boldsymbol{C}'(t) \times \boldsymbol{C}''(t)) \cdot \boldsymbol{C}'''(t)}{\|\boldsymbol{C}'(t) \times \boldsymbol{C}''(t)\|^2} \tag{22}$$

where U is the energy of the framed space curve based on the Frenet–Serret frame. And ω denotes the magnitude of the rate of the rotation for Darboux vector $\boldsymbol{d} = \tau\boldsymbol{T} + \kappa\boldsymbol{B}$ relative to the Frenet–Serret frame. Since the backbone curve is designed based on the general parameter t, the relationship

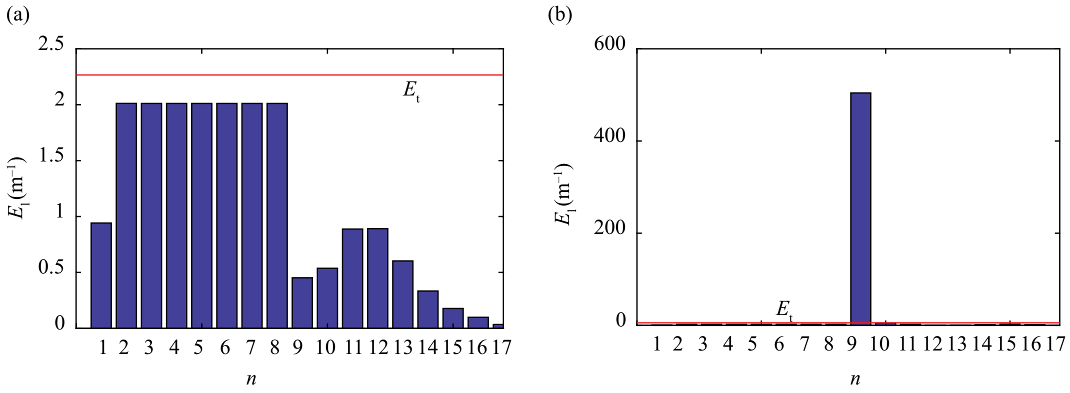

Fig. 10. (a) The local energy of the curve segments of the backbone curve in Fig. 7(a). (b) The local energy of the curve segments of the backbone curve in Fig. 7(b).

between the arc length s and the general parameter t is given in Eq. (20). $\boldsymbol{C}(t)$ is the backbone curve, and $\kappa(t)$ and $\tau(t)$ are the curvature and torsion represented based on the general parameter t.

Then the energy is used to estimate the shape change of the backbone curve and provide a measuring guide to the backbone curve segments that need an optimization discretization. Then the local energy of the i-th curve segment of the backbone curve E_l^i corresponding to the i-th link is defined as,

$$E_l^i = \int_{s_{i-1}}^{s_i} \omega(s)^2 ds \quad i = 1, 2, 3, \ldots, n \tag{23}$$

where a threshold of the local energy of the backbone curve E_t is set and the local optimization of the discretization method is adopted if $E_l^i > E_t$. The arc length of the local energy integration for i-th curve segment is equal to the length of the i-th link. The local energy of all links for the snake robot and the threshold E_t corresponding to the backbone curve in Fig. 7(a) is shown in Fig. 10(a). The local energy of all links for the snake robot and the threshold E_t corresponding to the backbone curve in Fig. 7(b) is shown in Fig. 10(b).

The modified discretization method with an optimization algorithm is adopted when the local energy E_l^i of a curve segment is larger than E_t. Here, we assume that the local energy of the n_k-th curve segment is larger than the threshold. Then the discretization of the curve segment with a serial number greater than or equal to n_k needs to be optimized. The serial number of the link for the snake robot is shown in Fig. 2(b). Since the transformation matrix ${}^0\boldsymbol{T}_{n_k-1}$ of the $(n_k - 1)$-th link relative to the world frame 0 is known, we have

$$\begin{gathered}\arg\min_{q_i \in \mathbb{R}^n} \Big(\sum_{i=n_k}^{n_k+m-1} \|\boldsymbol{p}_i(\boldsymbol{q}) - \boldsymbol{C}(t_{i-n_k+1})\|^2 \Big) \\ s.t. \begin{cases} 0 < t_1 < t_2 < \ldots < t_m \leq 1 \\ q_{dlim}^i \leq q_i \leq q_{ulim}^i \end{cases}\end{gathered} \tag{24}$$

where $\boldsymbol{p}_i(\boldsymbol{q})$ indicates the position vector of the transformation matrix ${}^0\boldsymbol{T}_i$ of the i-th link and m represents the number of the curve segments needed to be optimized. q_{dlim}^i and q_{ulim}^i are the lower bound and upper bound of the joint limit of i-th joint for the snake robot. $\boldsymbol{q}$ is the joint angle vector for the links corresponding these curve segments. In order to reduce the computational complexity of the optimization, $m_0(m_0 \leq m)$ curve segments are optimized at a time and then the next m_0 curve segment is optimized until all joint angles q_i of the robot are calculated. The discretization of links whose number is less than m_0 is calculated by Eq. (24) alone.

The process is given as follows. (a) Calculate the local energy of the curve segments of the backbone curve for the snake robot and determine the threshold E_t. (b) Determine if the local energy of the curve segment is greater than E_t. If not, the joint angles of the robot can be obtained by discretizing the backbone curve using the method in Eq. (10), which is based on the curvature and torsion of the

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

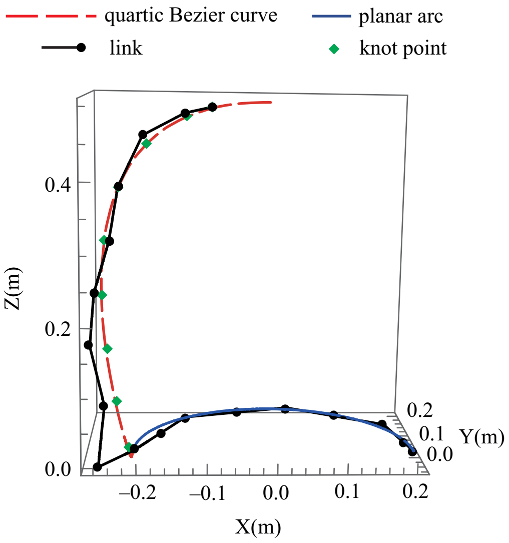

Fig. 11. The discretization result of the backbone curve in Fig. 7(b) using the modified discretization method.

backbone curve. (c) Otherwise, the discretization of the curve segment with a serial number greater than or equal to the serial number of the curve segment whose local energy is larger than E_t needs to be optimized. Then the other part of the backbone curve is discretized using the method in Eq. (10). The discretization result of the backbone curve in Fig. 7(b) using the modified discretization method is shown in Fig. 11. Here E_t is set $2.25\,\mathrm{m}^{-1}$ and m_0 is set 3. And the selection of E_t will be studied in Subsection 3.2. The point on the curve with the solved parameter t_i is the knot point. The optimization algorithm can ensure that the end point of the last link approaches the end point of the backbone curve as can as possible through the discretization result obtained by the modified discretization method, which is as shown in Fig. 11. So it is more reasonable to replace L_t to $s(t_m)$. $s(t_m)$ represents the arc length from the start point of the backbone curve to knot point $\boldsymbol{C}(t_m)$. Then the discretization error of the backbone curve in Fig. 7(b) using the modified discretization method is 0.0106609. The discretization error of the modified discretization method is one-fifth of the error of the non-modified method. The result is much better than before.

2.4. *Stability analysis of head-raising motion*

For the snake robot head-raising motion, the stability of the snake robot is the prerequisite for completing a specific task. In general, the support polygon method and ZMP method are used to analyze the stability of the mobile robot. The support polygon method has been introduced briefly in Section 2.2.1. The support polygon method here is used to determine the static stability of the snake robot. And the ZMP method can be used to analyze the dynamical stability of the robot. For the support polygon method,[23]

$$\boldsymbol{M} = \text{polygon}(\boldsymbol{p}_i) \quad i = 1, 2, ..., n_b + 1 \tag{25}$$

$$\boldsymbol{p}_c = \frac{\sum_{i=1}^{n} \boldsymbol{p}_c^i m_i}{\sum_{i=1}^{n} m_i} \tag{26}$$

where $\boldsymbol{p}_i$ represents the joint position for the base part of the snake robot. n_b denotes the number of the links of the base part. $\boldsymbol{M}$ is the planar region of the support polygon, and the function polygon output a polygon constructed by point $\boldsymbol{p}_i$. $\boldsymbol{p}_c$ indicates the center of mass of the robot. And the $\boldsymbol{p}_c^i$ is the center of mass of the i-th link. m_i is the mass of the i-th link. If the projection of $\boldsymbol{p}_c$ on the ground is inside the support polygon, the configuration of the robot is statically stable.

In the process of the motion for the robot, the ZMP method is used for dynamically stability determination for the snake robot. Since the ZMP is calculated by the condition of the torque equilibrium, the linear velocity and the angular velocity of the link ought to be computed.[23] The linear velocity and angular velocity of the i-th link are known firstly, then

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

$$^{0}\boldsymbol{v}_j = {}^{0}\boldsymbol{v}_i + {}^{0}\boldsymbol{\omega}_i \times {}^{0}\boldsymbol{R}_i{}^{i}\boldsymbol{b}_j \tag{27}$$

$$^{0}\boldsymbol{\omega}_j = {}^{0}\boldsymbol{\omega}_i + {}^{0}\boldsymbol{R}_i \times {}^{i}z_j\dot{q}_j \tag{28}$$

where $^{0}\boldsymbol{v}_j$ and $^{0}\boldsymbol{\omega}_j$ are the linear velocity and angular velocity of the j-th link represented in the world frame 0. The $^{i}z_j$ is the z-axis of frame j represented in frame i. $^{0}\boldsymbol{R}_i$ denotes the rotation matrix of the frame i relative to the frame 0. $^{i}\boldsymbol{b}_j$ is the joint position vector for j-th link represented in the i-th frame. $\dot{q}_j$ indicates the derivative of j-th joint angle relative to the time.

Next, the linear momentum and angular momentum of the robot are computed by the velocities of the links,

$$^{0}\dot{\boldsymbol{c}}_j = {}^{0}\boldsymbol{v}_j + {}^{0}\boldsymbol{\omega}_j \times ({}^{0}\boldsymbol{R}_j\bar{\boldsymbol{c}}_j) \tag{29}$$

$$^{0}\mathcal{P}_j = m_j{}^{0}\dot{\boldsymbol{c}}_j \tag{30}$$

$$^{0}\mathcal{P} = \sum_{j=1}^{N} {}^{0}\mathcal{P}_j \tag{31}$$

$$^{0}\mathcal{L}_j = {}^{0}\boldsymbol{c}_j \times {}^{0}\mathcal{P}_j + {}^{0}\boldsymbol{R}_j\bar{\boldsymbol{I}}_j{}^{0}\boldsymbol{R}_j^{\mathrm{T}0}\boldsymbol{\omega}_j \tag{32}$$

$$^{0}\mathcal{L} = \sum_{j=1}^{N} {}^{0}\mathcal{L}_j \tag{33}$$

where $\bar{c}_j$ represents the coordinate of the center of mass for the j-th link described in frame j. And $^{0}\dot{\boldsymbol{c}}_j$ denotes the derivative of the center of mass of the j-th respect to time represented in the world frame. $^{0}\mathcal{P}_j$ and $^{0}\mathcal{P}$ represent the linear momentum for i-th link and the robot, respectively. $^{0}\mathcal{L}_j$ and $^{0}\mathcal{L}$ denote the angular momentum for i-th link and the robot, respectively. $\bar{\boldsymbol{I}}_j$ is the inertia tensor matrix of the j-th link relative to the local frame.

Then the ZMP can be solved based on the dynamics algorithm,

$$\boldsymbol{\tau} = \boldsymbol{p} \times \boldsymbol{f} + \boldsymbol{\tau}_p \tag{34}$$

$$^{0}\dot{\mathcal{P}} = M\boldsymbol{g} + \boldsymbol{f} \tag{35}$$

$$^{0}\dot{\mathcal{L}} = \boldsymbol{c} \times M\boldsymbol{g} + \boldsymbol{\tau} \tag{36}$$

substitute Eqs. (35) and (36) into Eq. (34), we have

$$\boldsymbol{\tau}_p = {}^{0}\dot{\mathcal{L}} - \boldsymbol{c} \times M\boldsymbol{g} + ({}^{0}\dot{\mathcal{P}} - M\boldsymbol{g}) \times \boldsymbol{p} \tag{37}$$

where $\boldsymbol{f}$ denotes the reaction force of the ground to the link. And $\boldsymbol{p}$ denotes the ZMP, $M = \sum_{j=1}^{N} m_j$. $\boldsymbol{g}$ is the gravity acceleration vector. So, we can obtain

$$\tau_{px} = \dot{\mathcal{L}}_x + Mgy + \dot{\mathcal{P}}_y p_z - (\dot{\mathcal{P}}_z + Mg)p_y \tag{38}$$

$$\tau_{py} = \dot{\mathcal{L}}_y - Mgx - \dot{\mathcal{P}}_x p_z + (\dot{\mathcal{P}}_z + Mg)p_x \tag{39}$$

where

$$\begin{cases} {}^{0}\mathcal{P} = [\mathcal{P}_x \quad \mathcal{P}_y \quad \mathcal{P}_z]^{\mathrm{T}} \\ {}^{0}\mathcal{L} = [\mathcal{L}_x \quad \mathcal{L}_y \quad \mathcal{L}_z]^{\mathrm{T}} \\ \boldsymbol{c} = [x \quad y \quad z]^{\mathrm{T}} \\ \boldsymbol{g} = [0 \quad 0 \quad -g]^{\mathrm{T}} \\ \boldsymbol{p} = [p_x \quad p_y \quad p_z]^{\mathrm{T}} \end{cases} \tag{40}$$

Let the x component τ_{px} and y component τ_{py} of the moment $\boldsymbol{\tau}_p$ be 0. The p_x and p_y of the ZMP can be solved.

$$p_x = \frac{Mgx + p_z\dot{\mathcal{P}}_x - \dot{\mathcal{L}}_y}{Mg + \dot{\mathcal{P}}_z} \tag{41}$$

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

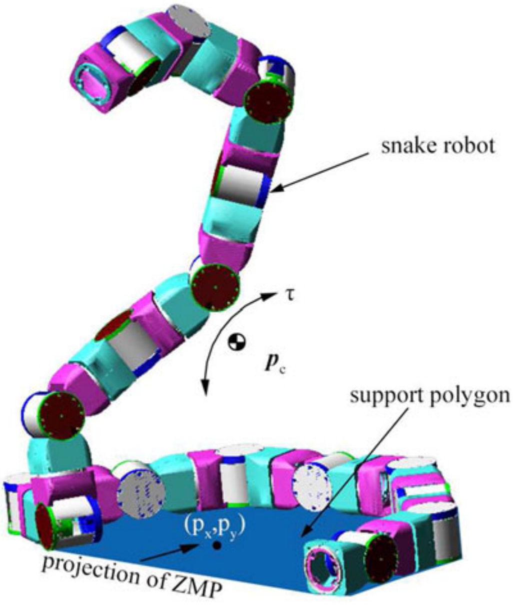

Fig. 12. The illustration of the ZMP and the support polygon for the snake robot.

$$p_y = \frac{Mgy + p_z\dot{\mathcal{P}}_y + \dot{\mathcal{L}}_x}{Mg + \dot{\mathcal{P}}_z} \tag{42}$$

Then, we have $p_z = 0$ considering that the snake robot locomotes on the ground. If the projection of the ZMP on the ground, (p_x, p_y) lies inside the support polygon, the snake robot is dynamically stable.[24] The illustration is shown in Fig. 12.

3. The Workspace Comparison with the Existing Method and Relationship Between the Threshold and the Discretization Error

3.1. The workspace comparison with the existing method

3.1.1. The workspace of the predefined spiral curve method. Zhang et al. proposed a predefined spiral curve method for the head-raising of the snake robot.[4] Then, a shape-fitting method based on the 3-DOF joint is proposed.[4] And the roll, pitch, and yaw angles of the joint can be solved. But for the snake robot with single-DOF joints, it is not suitable. Therefore, we will compare the backbone curve based on the Bézier curve with that of a predefined spiral curve. And the modified discretization method is also applied to the predefined spiral curve. Here, we assume that the planar spiral curve of the predefined spiral method and the planar arc of the proposed method have the same length, and the length equals $\sum_{i=1}^{8} l_i$. Then equations of the spiral curve of the predefined spiral curve method are given as,[4]

$$\begin{cases} x_l(t) = x_s(\hat{t}_1) & t \in (0, \hat{t}_1) \\ y_l(t) = y_s(\hat{t}_1) - t + \hat{t}_1 & t \in (0, \hat{t}_1) \\ z_l(t) = 0 & t \in (0, \hat{t}_1) \end{cases} \tag{43}$$

$$\begin{cases} x_s(t) = ab(\hat{t}_3 - t)\sin(t + \phi_0) & t \in (\hat{t}_1, \hat{t}_2) \\ y_s(t) = a(\hat{t}_3 - t)\cos(t + \phi_0) & t \in (\hat{t}_1, \hat{t}_2) \\ z_s(t) = 0 & t \in (\hat{t}_1, \hat{t}_2) \end{cases} \tag{44}$$

$$\begin{cases} x_s(t) = ab(\hat{t}_3 - t)\sin(t + \phi_0) & t \in (\hat{t}_2, \hat{t}_3) \\ y_s(t) = a(\hat{t}_3 - t)\cos(t + \phi_0) & t \in (\hat{t}_2, \hat{t}_3) \\ z_s(t) = ct - c\hat{t}_2 & t \in (\hat{t}_2, \hat{t}_3) \end{cases} \tag{45}$$

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

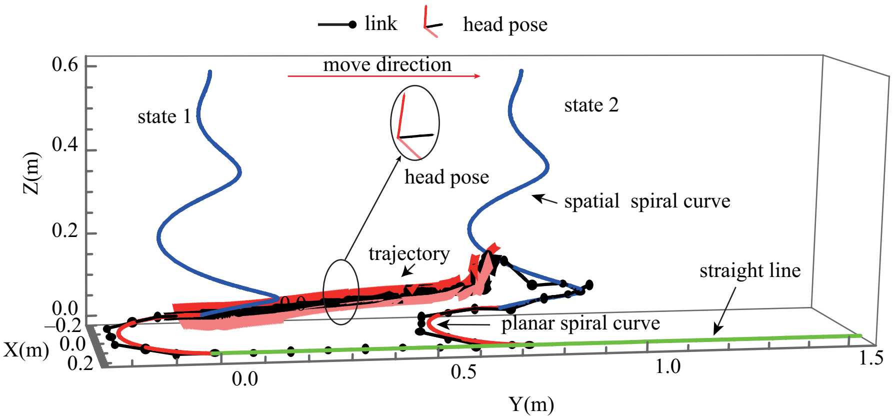

Fig. 13. The head-raising motion by the predefined spiral curve method.

$$\begin{cases} \hat{t}_1 = \sum_{i=1}^{n} l_i \\ \hat{t}_2 = \hat{t}_1 + \phi_{base} \\ \hat{t}_3 = \hat{t}_1 + 2n_c\pi \end{cases} \tag{46}$$

where a, b, and c are adjustable parameters. ϕ_{base} is the phase value of the second interval, and n_c is the cycle number of the spiral curve. And Eq. (43) represents the straight line segment, and Eq. (44) describes the planar spiral curve segment. And Eq. (45) represents the spatial spiral curve segment. The predefined spiral curve is shown in Fig. 13. Then the snake robot moves from state 1 to state 2, and the pose of the head forms a spatial trajectory. The parameters of the predefined curve are as follows: $a = 14.7$, $b = 1$, $c = 48.5$, $t_1 = 1552$, $t_2 = 1555.1$, $t_3 = 1567.7$, $\phi_0 = 1.6022$, $\phi_{base} = \pi$, and $n_c = 2.5$. The motion process of the head-raising for the snake robot is shown in Fig. 13. The starting state in which the snake robot raises its head is the state 1 as shown in Fig. 13. And the end state in which the snake robot raises its head is the state 2. The workspace of the method is the spatial trajectory formed by the pose of the head.

3.1.2. The workspace of the method based on the Bézier curve. In order to get the workspace of the head for the snake robot, we are inspired by the method in ref. [33]. Firstly, n_0 grid points are evenly distributed in the hemisphere space whose center point is $\boldsymbol{P}_0$ and radius r_0 equals the length of the upper part of robot, $\sum_{i=9}^{17} l_i$. $\boldsymbol{P}_0$ is the end point of the base part of the snake robot. These grid points are shown in Fig. 14. Every grid point is used as the end point of the backbone curve of the Bézier curve. Then a sphere, whose center is a grid point and radius is r_1, is constructed. The n_1 points are distributed on the sphere, and every unit vector from the center of the sphere to the point on the sphere form a desired tangent direction for the end point of the backbone curve of the Bézier curve, $\boldsymbol{C}'(1)/\|\boldsymbol{C}'(1)\|$. Then the point set with n_0 points and n_1 directions for each point is used to solve the valid Bézier curves by Eqs. (11), (12), (15), and (16). Then the backbone curve of the base part and these Bézier curve construct several full backbone curves. And these full backbone curves are discretized to generate some configurations for the snake robot. But these configurations of the snake robot should meet the following conditions: (a) avoiding self-collision; (b) the center of mass of the robot is inside the support polygon; (c) the upper part of the snake robot has no collision with the ground. The computation of avoiding self-collision is completed in Moveit! software of the Robot Operating System (ROS) Kinetic Kame through the Flexible Collision Library.[34] The condition (b) can be determined by Eqs. (25) and (26). It can be judged whether the condition (c) is satisfied by the relationship between the coordinates of the z-axis of the link centroids and the ground coordinate. Here $n_0 = 15,708$, $\boldsymbol{P}_0 = [-0.2051, 0.0066, 0]$, $r_0 = 0.738$ m, $r_1 = 0.01$ m, and $n_1 = 20$. These samples construct 314,160 backbone curves and the number of curves satisfying above 3 conditions is 75,160. The configurations of the snake robot obtained by these reasonable curves form the workspace of the head, which is shown in Fig. 15. $\boldsymbol{P}_{head}$ indicates the position of the head in Fig. 15.

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

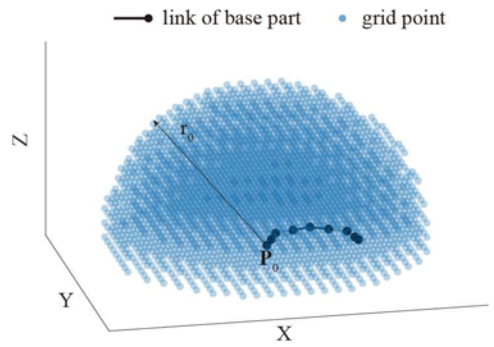

Fig. 14. The grid points are evenly distributed in the hemisphere.

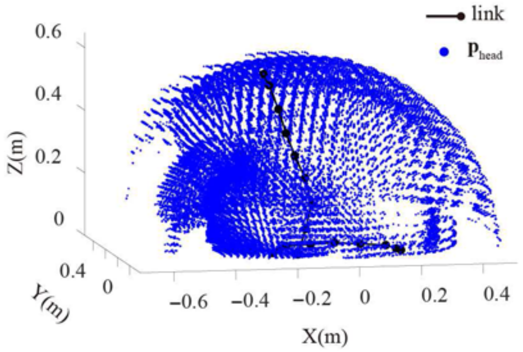

Fig. 15. The workspace of the head calculated by the method based on the Bézier curve method.

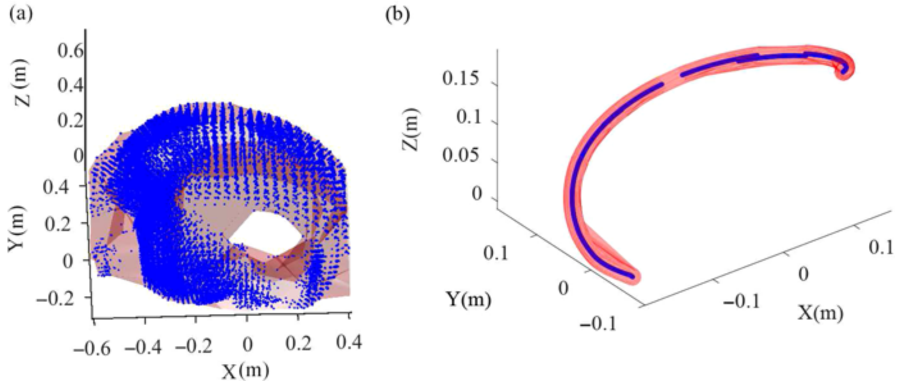

Fig. 16. (a) The polyhedrons enclosing the workspace of the head obtained by the proposed method. (b) The polyhedrons enclosing the workspace of the head obtained by the predefined spiral curve method.

3.1.3. Result of comparison. For comparing the workspaces of the head obtained by the proposed method and the predefined spiral curve method, the volumes of the smallest polyhedron enclosing the workspace are calculated. The polyhedrons enclosing the workspace of the head obtained by the proposed method and the predefined spiral curve method are shown in Figs. 16(a) and (b), respectively. The volumes of the polyhedrons in Figs. 16(a) and (b) are 0.1991 and 0.00023985 m^3, respectively. The workspace of the proposed method is much better than that of the predefined spiral curve method.

3.2. Relationship between the threshold and the discretization error

In order to give a guide to select a proper threshold for the modified discretization method, we studied the relationship between the threshold and the discretization error by the modified discretization method. In Subsection 3.1.2, 75,160 valid curves are obtained. Then, 32,580 curves are selected

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

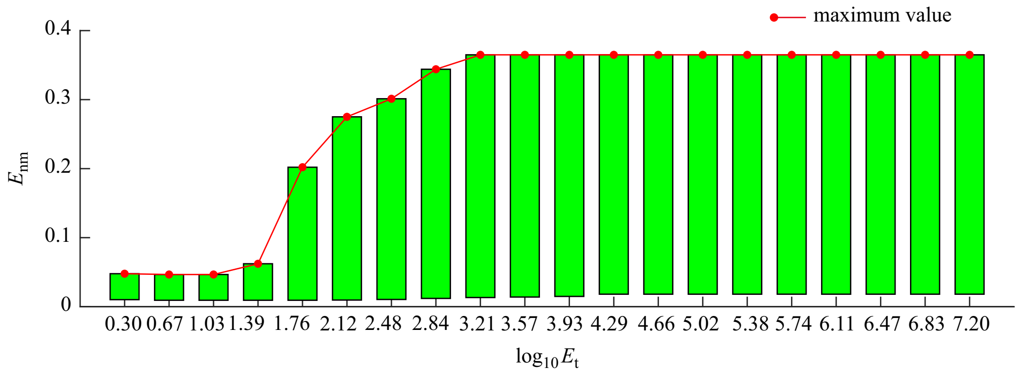

Fig. 17. The relationship between the discretization error and the logarithm of the threshold.

randomly from them, and the local energy of the curve segments for selected curves are calculated. Considering that the curvature and torsion of the arc backbone curve for the base part are both constants, the discretization of the base part by the method based on the curvature and torsion is convenient and accurate. Therefore, we set such an interval as the value range of the threshold E_t, whose upper bound is the maximum value of all local energy for the selected curves, and the lower bound is the maximum local energy of the curve segments of the backbone curve for the base part. Then, we obtain that $E_t \in [2.01563, 15, 666, 600]$. Aim to better study the relationship, $\log_{10} E_t$ is taken as the independent variable and the value range of $\log_{10} E_t$ is interpolated linearly to generate 20 values. Then the combination of the threshold and the selected backbone curves generates $20 \times 32, 580$ samples. When the local energy of a curve segment of an entire backbone curve is larger than the threshold, the optimization is adopted. Otherwise, the backbone curve is discretized by the method based on the curvature and torsion. The result is shown in Fig. 17. When $\log_{10} E_t \leq 1.03(E_t \leq 10.7086)$, the discretization error has a rather small variation. When the threshold $\log_{10} E_t \geq 3.21$, the maximum value of E_{nm} keeps 0.3649. The threshold can be selected according to the desired discretization error referring to Fig. 17.

4. The Obstacle Avoidance Strategy of the Head-Raising Method and Simulation

4.1. The obstacle avoidance of the head-raising

Another advantage of the proposed method over the predefined spiral curve method is that it can achieve obstacle avoidance. Since the Bézier backbone curve is constructed by control points, the Bézier backbone curve can be adjusted by moving these control points. The poses of the links will be changed once the backbone curve is changed. This property can be used in the obstacle avoidance of the snake robot. The obstacle avoidance of the snake robot based on the Bézier curve method is illustrated in Fig. 18(a). In Fig. 18(a), the links of snake robot are simplified by slim boxes, and the links of the robot with low opacity are in a state of collision with an obstacle. The links of the robot with high opacity are in the state of successfully avoiding an obstacle. For the process, the change of the Bézier backbone curve is shown in Fig. 18(b). $\{\boldsymbol{P}_0, \boldsymbol{P}_1, ...\boldsymbol{P}_4\}$ indicates the control points of the Bézier backbone curve for the snake robot when the snake robot collides with the obstacle. In order to locally adjust the backbone curve to avoid the obstacle, the new control points are needed. One of the excellent features of the Bézier curve is that new control points can be added by the degree elevation without changing the shape of the curve. For the process of the Bézier curve of degree h elevating to the Bézier curve of degree $h+1$, we assume that the control points of the Bézier curve of degree $h+1$ are $\boldsymbol{G}_0, \boldsymbol{G}_1, ..., \boldsymbol{G}_{h+1}$. The control points of the Bézier curve of degree h are $\boldsymbol{P}_0, \boldsymbol{P}_1, ..., \boldsymbol{P}_h$.

$$\begin{cases} \boldsymbol{G}_0 = \boldsymbol{P}_0 \\ \boldsymbol{G}_i = \frac{i}{h+1}\boldsymbol{P}_{i-1} + (1 - \frac{i}{h+1})\boldsymbol{P}_i \quad 1 \leq i \leq h \\ \boldsymbol{G}_{h+1} = \boldsymbol{P}_h \end{cases} \tag{47}$$

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

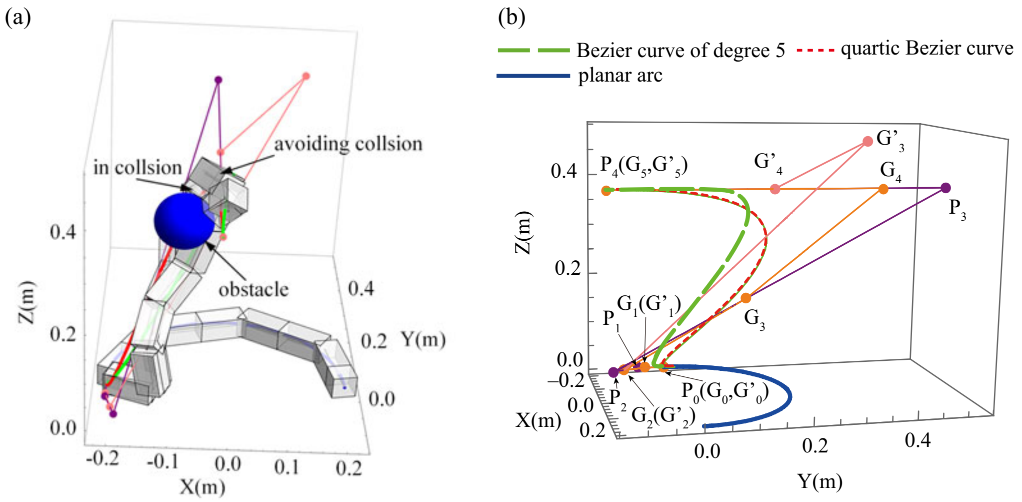

Fig. 18. (a) The obstacle avoidance of the snake robot based on the Bézier curve method. (b) The change of the Bézier backbone curve for the snake robot.

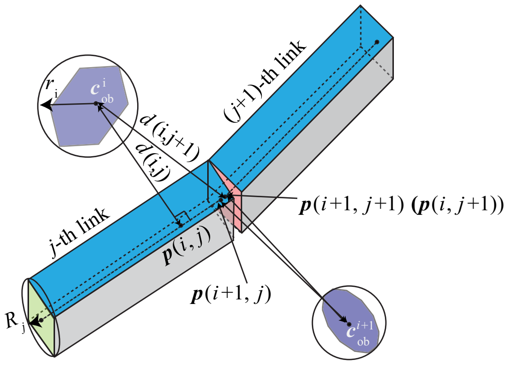

Fig. 19. The illustration of envelope of the obstacles and the links.

Calculating the control points by Eq. (47) can achieve one-degree elevation. Besides, Eq. (47) used for iteration can achieve a multi-degree elevation of the Bézier curve. However, computation of the Bézier curve will become expensive with the degree of the curve increasing. So we choose to elevate the Bézier curve of degree 4 to one of degree 5 as the backbone curve. The control points $\boldsymbol{G}_0$, $\boldsymbol{G}_1$, ..., $\boldsymbol{G}_5$ in Fig. 18(b) are computed according to $\{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots \boldsymbol{P}_4\}$ using Eq. (47). Then $\boldsymbol{G}_3$ is moved to $\boldsymbol{G}'_3$ for adjusting the Bézier curve locally. Then, $\boldsymbol{G}_0 = \boldsymbol{G}'_0$, $\boldsymbol{G}_1 = \boldsymbol{G}'_1$, $\boldsymbol{G}_2 = \boldsymbol{G}'_2$, and $\boldsymbol{G}_5 = \boldsymbol{G}'_5$. $\boldsymbol{G}'_4$ is solved by Eqs. (11), (12), (15), and (16).

For the steps of the avoiding obstacle, they can be expressed in Algorithm 1. These variables in Algorithm 1 are shown in Fig. 19. And the process of the avoiding obstacles is to loop the algorithm until all element of *ds* are not smaller than 0 or the loop time reaches the set maximum iteration time. η in Algorithm 1 is a step size of the point movement. As described in Eq. (47), the flexibility of the backbone curve of the snake robot can be increased by adding more control points to the Bézier curve. When the backbone curve has more flexibility, the head of snake robot can avoid more obstacles in the environment at the same time. However, adding more control points will enlarge the order of the Bézier curve and increase the computation burden of the discretization of the backbone curve largely since the discretization of the backbone curve needs to calculate the torsion and curvature of the backbone curve as shown in Eqs. (21) and (22).

4.2. *Simulation of the obstacle avoidance*

In order to validate the effectiveness of obstacle avoidance of the head-raising method based on the Bézier curve, we conducted a simulation of the upper part of snake robot passing through a

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

Algorithm 1 The strategy of obstacle avoidance based on the degree elevation of the Bézier backbone curve

```
Assume $n_0$ obstacles, $n$ links for the robot, $\boldsymbol{c}^i_{ob}$ for the center of mass as $i$-th obstacle and $\boldsymbol{p}(i,j)$ as the point on the center line of $j$-th link closest to the $i$-th obstacle. Define $\boldsymbol{Q}_j$ and $\boldsymbol{Q}_{j+1}$ as the two end point of the $j$-th link. Define $r_i$ as the radius of envelope sphere of the $i$-th irregular obstacle, $R_j$ as the radius of envelope cylinder of the $j$-th link and $l_{tl}$ as a tolerance length. The initial Bézier backbone curve is of degree 4.
for each $i \in [1, n_0]$ do
    for each $j \in [1, n]$ do
        $\boldsymbol{p}(i,j) = \boldsymbol{Q}_j + t(\boldsymbol{Q}_{j+1} - \boldsymbol{Q}_j)$
        solve $t$ by the condition that $(\boldsymbol{Q}_{j+1} - \boldsymbol{Q}_j) \cdot (\boldsymbol{p}(i,j) - \boldsymbol{c}^i_{ob}) = 0$
        if $t < 0$ then
            $t = 0$
        end if
        if $t > 1$ then
            $t = 1$
        end if
        $d(i,j) = \|\boldsymbol{c}^i_{ob} - \boldsymbol{p}(i,j)\|$
        $ds(i,j) = d(i,j) - r_i - R_j - l_{tl}$
        $k = 0$
        if $ds(i,j) < 0$ then
            $k = k + 1$
            elevate degree of the Bézier backbone curve using Eq. (47) and generate a new free control point $\boldsymbol{P}_{2+k}$
            $\boldsymbol{v}(k) = (\boldsymbol{p}(i,j) - \boldsymbol{c}^i_{ob}) / \|\boldsymbol{p}(i,j) - \boldsymbol{c}^i_{ob}\|$
            $\boldsymbol{P}_{2+k} = \boldsymbol{P}_{2+k} + \eta \boldsymbol{v}(k)$
        end if
    end for
end for
```

Table I. The overview of the snake robot.

Dimensions	Module: 4.3 cm × 4.3 cm × 8.65 cm
	Length (full 16 module robot): 138.4 cm
	Length (start link): 4.05 cm
	Length (head link): 4.6 cm
Mass	Module: 0.2 kg
	Full 16 module robot: 3.2 kg
Joint limit	±99 degree

clutter environment constructed by three spheres. The structure parameters of the snake robot are given in Table I. The simulation is completed on the dynamics simulation software. The simulation scenario is shown in Fig. 20. The proportional-derivative controller with gravity compensation is adopted to ensure the accuracy of the trajectory tracking in the joint space. For the obstacle avoidance simulation, the positions of the obstacle spheres are known in advance. The radii of the three obstacle spheres, r_1, r_2, and r_3, are all 5 cm. The radii of envelope cylinder of the links R are all 3.04 cm. The distances between sphere 1, sphere 2, and sphere 3 are 6.8, 7.2, and 7.2 cm respectively. The positions of the spheres are $[-0.281767, 0.035917, 0.319962]$, $[-0.116983, 0.00319943, 0.319914]$, and $[-0.199326, 0.0195819, 0.469938]$ m, respectively. The positions of the end point of the links for the base part of the snake robot are $[0, -0.0405, 0]$, $[0, 0, 0]$, $[0, 0.0865, 0]$, $[-0.06412, 0.1446, 0]$, $[-0.1282, 0.2026, 0]$, $[-0.2143, 0.194, 0]$, $[-0.3004, 0.1855, 0]$, $[-0.3518, 0.1159, 0]$, and $[-0.40322, 0.04637, 0]$ m. The head of the snake robot moves through the clutter environment along the direction like black arrows in Fig. 20.

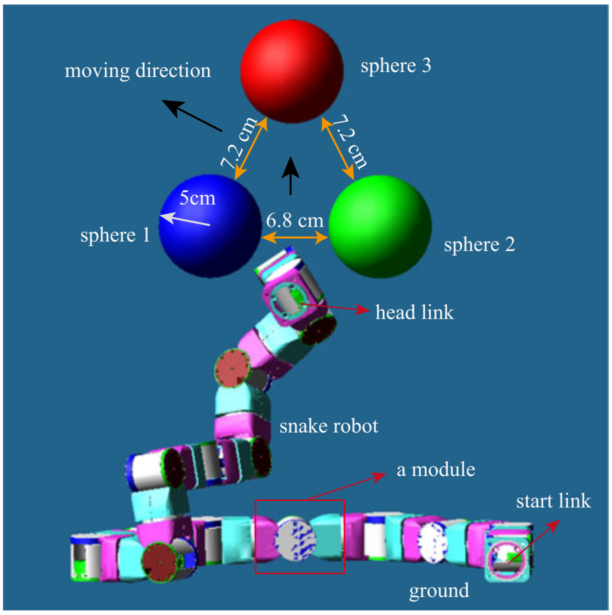

Fig. 20. The illustration of the simulation scenario.

The specific process of obstacle avoidance is shown in Fig. 21. The initial state of the snake robot is shown in Fig. 21(a). Then the snake robot head approached the spheres in Fig. 21(e). The three intermediate states for approaching the spheres are designed based on the Bézier backbone curve as shown in Figs. 21(b), (c), and (d). Then the head pass through the gap between the sphere 1 and the sphere 2 is shown in Fig. 21(f) and (g). In this process, the distances between the links and the obstacles are calculated to determine whether the robot has collisions with the obstacles. If so, the degree of Bézier backbone curve is elevated and the new control point is moved to adjust the poses of the links of the robot. In this simulation, the Bézier backbone curve of degree 4 is elevated to the one of degree 5 to achieve collision avoidance. The same method is applied in passing the gap between the sphere 1 and sphere 3, which is shown in Figs. 21(h), (i), (j), (k), and (l). For determining the collision between the snake robot and the obstacle, the tolerance length l_{tl} in Algorithm 1 is set 0.0001 m and the array ds is computed. Then the minimum value d_{min} of the array ds for every time interval is treated as a variable for determining the collision. If $d_{min} \leq 0$, the collision occurs. If not, the robot is treated to be safe. The changes of d_{min} with the obstacle avoidance and without the obstacle avoidance are shown in Fig. 22. The figure is clipped from the state shown in Fig. 21(e) (corresponding to 16 s) for a better show of the obstacle avoidance. The projections of ZMP of the robot on the ground in the obstacle avoidance process are shown in Fig. 23(a). The projection of ZMP for the planned trajectory and the projection of ZMP for the actual trajectory both inside the support polygon prove that the motion of the snake robot is stable. And the local magnification of the projections in Fig. 23(a) is shown in Fig. 23(b).

5. Conclusions

In this paper, the head-raising method of snake robots based on the Bézier curve is proposed. The backbone curve of the snake robot for head-raising motion is piecewise. The backbone curve of the base part of the robot is set to a planar arc for ensuring the stability of the snake robot. The backbone curve of the upper part is the Bézier curve. The continuity condition of the piecewise backbone curve is given for avoiding the sharp change of the phase angle. Besides, the discretization method is improved for making the links of the robot better approach the piecewise backbone curve. In order to determine the condition of adopting the modified discretization method, the energy of the framed space curve concept in the computer graphics filed is used to estimate the shape change of the backbone curve. When the curve energy of the backbone curve is larger than a threshold, the modified discretization method should be adopted to decrease the discretization error. For selecting a proper threshold, the relationship between the discretization error and the threshold is studied.

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

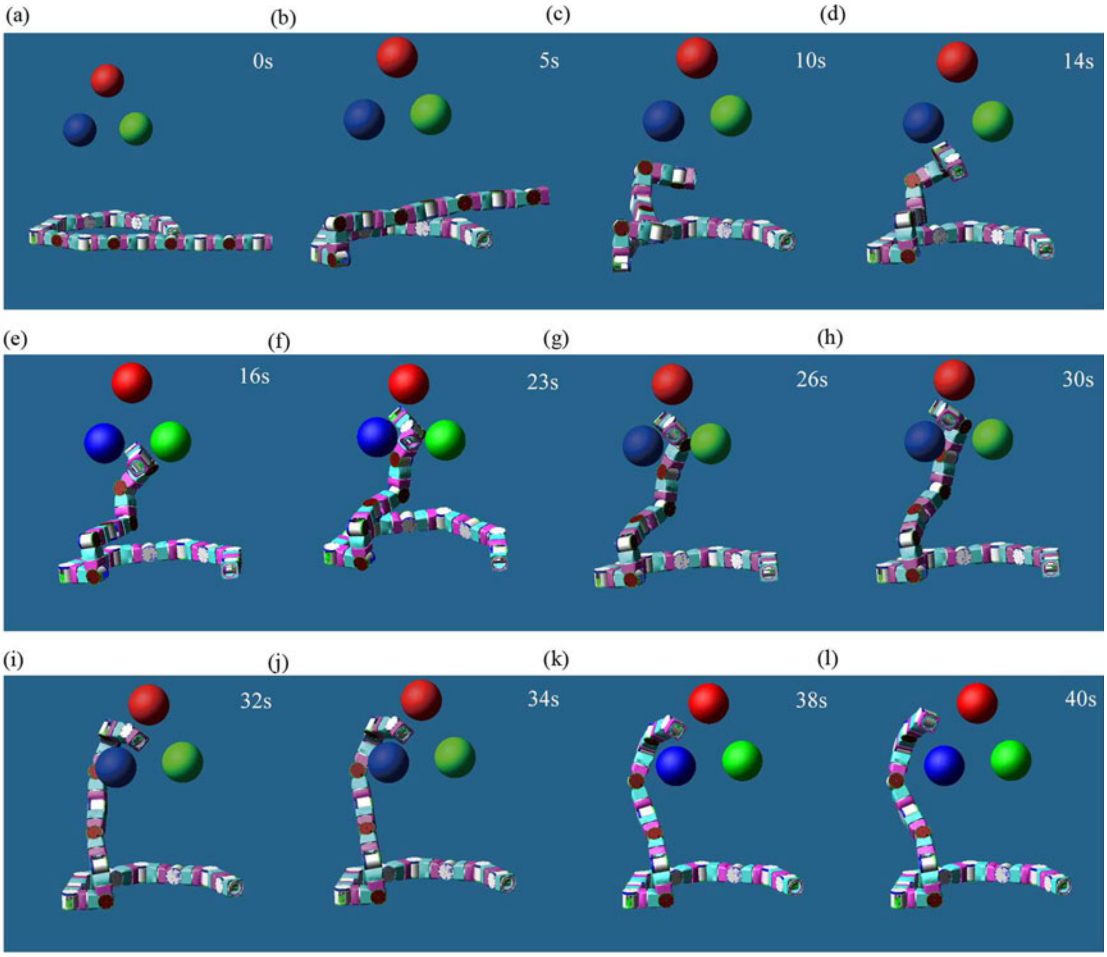

Fig. 21. The process of the obstacle avoidance for the snake robot.

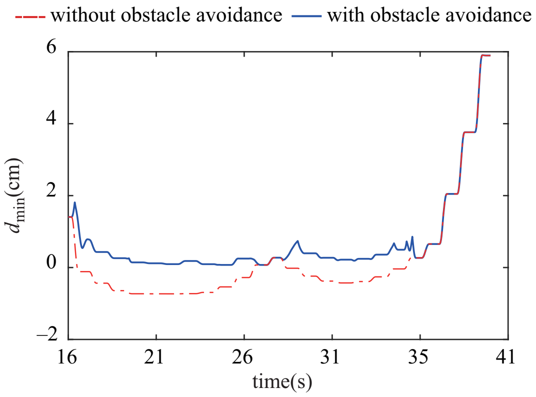

Fig. 22. The relationships between minimum distance d_{min} and the time.

Then the proposed method is compared with the existing head-raising method that uses a predefined spiral curve. The workspace of the proposed method is much larger than the existing method. In addition, the proposed method performs well in obstacle avoidance. The simulation result validates the effectiveness of the obstacle avoidance strategy. In the future, we will study the obstacle avoidance of the head of the snake robot in a more complex environment.

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press

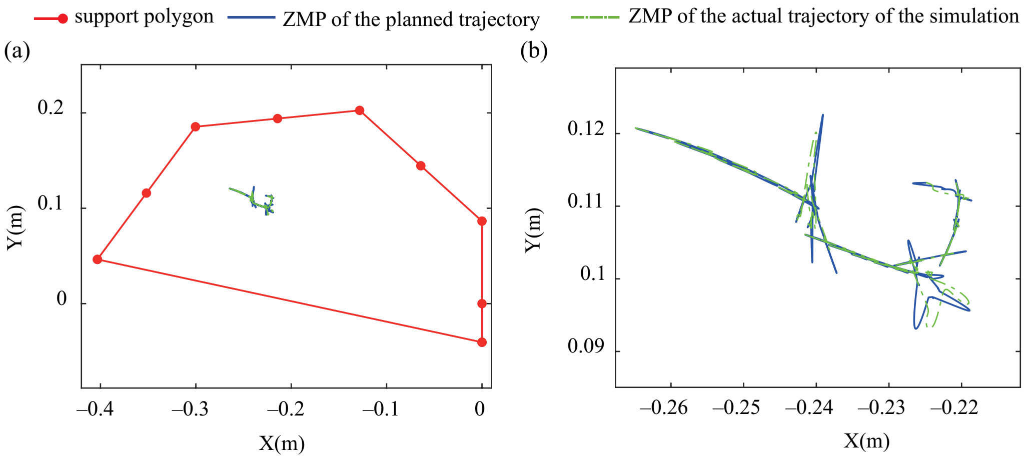

Fig. 23. (a) The projections of ZMPs of the robot on ground and the support polygon. (b) The local magnification of the projections of ZMPs.

Acknowledgment

The research leading to these results was supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 51521003) and the National Natural Science Foundation of China (No. 61503095, No. 91848101).

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574720000533.

References

1. L. Douadi, D. Spinello, W. Gueaieb and H. Sarfraz, “Planar kinematics analysis of a snake-like robot,” *Robotica.* **32**(5), 659–675 (2014).
2. N. Li, T. Zhao, Y. Zhao and Y. Lin, “Design and realization of a snake-like robot system based on a spatial linkage mechanism,” *Robotica.* **27**(5), 779–788 (2009).
3. Z. Mu, H. Wang, W. Xu, T. Liu and H. Wang, “Two types of snake-like robots for complex environment exploration: Design, development, and experiment,” *Adv. Mech. Eng.* **9**(9), 1–15 (2017).
4. X. Zhang, J. Liu, Z. Ju and C. Yang, “Head-raising of snake robots based on a predefined spiral curve method,” *Appl. Sci.* **8**(11), 2011 (2018).
5. T. Kano, R. Yoshizawa and A. Ishiguro, “Tegotae-based decentralised control scheme for autonomous gait transition of snake-like robots,” *Bioinspir. Biomim.* **12**(4), 046009 (2017).
6. R. Ariizumi, M. Tanaka and F. Matsuno, “Analysis and heading control of continuum planar snake robot based on kinematics and a general solution thereof,” *Adv. Robot.* **30**(5), 301–314 (2016).
7. M. Tanaka and F. Matsuno, “Modeling and control of head raising snake robots by using kinematic redundancy,” *J. Intell. Robot. Syst.* **75**(1), 53–69 (2014).
8. W. Zhen, C. Gong and H. Choset, “Modeling Rolling Gaits of a Snake Robot,” *IEEE International Conference on Robotics and Automation*, Seattle, WA, USA (2015) pp. 3741–3746.
9. M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz and H. Choset, “Parameterized and scripted gaits for modular snake robots,” *Adv. Robot.* **23**(9), 1131–1158 (2009).
10. C. Ye, S. Ma, B. Li and Y. Wang, “Head-Raising Motion of Snake-Like Robots,” *IEEE International Conference on Robotics and Biomimetics*, Shenyang, China (2004) pp. 595–600.
11. X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M.J. Travers and H. Choset, “Locomotive Reduction for Snake Robots,” *IEEE International Conference on Robotics and Automation*, Seattle, WA, USA (2015) pp. 3735–3740.
12. D. Rollinson and H. Choset, “Pipe network locomotion with a snake robot,” *J. Field Robot.* **33**(3), 322–336 (2016).
13. G. S. Chirikjian and J. W. Burdick, “A modal approach to hyper-redundant manipulator kinematics,” *IEEE Trans. Robot. Automat.* **10**(3), 343–354 (1994).
14. M. Kamermans, “A primer on bézier curves (2014),” https://pomax.github.io/bezierinfo, oneline. Accessed October 17, 2017.
15. D. Salomon, *Curves and Surfaces for Computer Graphics* (Springer Press, New York, USA, 2006).
16. K. E. Zanganeh and J. Angeles, “The Inverse Kinematics of Hyper-Redundant Manipulators Using splines,” *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan (1995) pp. 2797–2802.

17. A. Chibani, C. Mahfoudi, T. Chettibi, R. Merzouki and A. Zaatri, "Generating optimal reference kinematic configurations for hyper-redundant parallel robots," *P. I. Mech. Eng. I-J. Sys.* **229**(9), 867–882 (2015).
18. P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl, "A Control Framework for Snake Robot Locomotion Based on Shape Control Points Interconnected by Bézier Curves," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal (2012) pp. 3111–3118.
19. H. Yamada and S. Hirose, "Study of active cord mechanism-approximations to continuous curves of a multi-joint body," *J. Robot. Soc. Jpn.* **26**(1), 110 (2008).
20. H. Yamada and S. Hirose, "Approximations to Continuous Curves of Active Cord Mechanism Made of Arc-Shaped Joints or Double Joints," *IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA (2010) pp. 703–708.
21. H. Edelsbrunner, L. Kobbelt and K. Polthier, *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable* (Springer Press, Berlin Heidelberg, 2008).
22. R. L. Hatton and H. Choset, "Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction," *Auton. Robots* **28**(3), 271–281 (2010).
23. S. Kajita, H. Hirukawa, K. Harada and K. Yokoi, *Introduction to Humanoid Robotics*, vol. 101 (Springer Press, New York, USA, 2014).
24. M. Dekker, *Zero-Moment Point Method for Stable Biped Walking* (Eindhoven University of Technology, Eindhoven, 2009).
25. G. Farin, *Curves and Surfaces for CAGD, A Practical Guide*. (Morgan Kaufmann Publishers, San Francisco, CA, 2001)
26. T. Kamegawa, T. Harada and A. Gofuku, "Realization of Cylinder Climbing Locomotion with Helical form by a Snake Robot with Passive Wheels," *IEEE International Conference on Robotics and Automation*, Kobe, Japan (2009) pp. 3067–3072.
27. Z. Bing, L. Cheng, G. Chen, F. Röhrbein, K. Huang and A. Knoll, "Towards autonomous locomotion: Cpg-based control of smooth 3D slithering gait transition of a snake-like robot," *Bioinspir. Biomim.* **12**(3), 035001 (2017).
28. H. Yamada and S. Hirose, "Study on the 3D Shape of Active Cord Mechanism," *Proceedings IEEE International Conference on Robotics and Automation*, Orlando, FL, USA (2006) pp. 2890–2895.
29. T. Takemori, M. Tanaka and F. Matsuno, "Gait Design of a Snake Robot by Connecting Simple Shapes," *IEEE International Symposium on Safety, Security, and Rescue Robotics*, Lausanne, Switzerland (2016) pp. 189–194.
30. J. Oprea, *Differential Geometry and Its Applications* (Pearson Education Inc., Washington DC, USA, 2007).
31. T. Kamegawa, T. Baba and A. Gofuku, "V-shift Control for Snake Robot Moving the Inside of a Pipe with Helical Rolling Motion," *IEEE International Symposium on Safety, Security, and Rescue Robotics*, Kyoto, Japan (2011) pp. 1–6.
32. R. T. Farouki, M. Al Kandari and T. Sakkalis, "Hermite interpolation by rotation-invariant spatial pythagorean-hodograph curves," *Adv. Comput. Math.* **17**(4), 369–383 (2002).
33. F. Zacharias, C. Borst and G. Hirzinger, "Capturing Robot Workspace Structure: Representing Robot Capabilities," *IEEE/RSJ International Conference on Intelligent Robots & Systems*, San Diego, CA, USA (2007) pp. 3229–3236.
34. S. Chitta, I. Sucan and S. Cousins, "Moveit![ros topics]," *IEEE Robot. Autom. Mag.* **19**(1), 18–19 (2012).

https://doi.org/10.1017/S0263574720000533 Published online by Cambridge University Press