Motion Planning and Locomotive Reduction for Twistable Snake Robots

Qingling Fan[1], Zhongqiang Ren[1†]

***Abstract*— This paper studies motion planning and locomotive reduction of snake robots, a class of hyper-redundant mechanisms that can traverse complex environments with limbless locomotion. To coordinate the large number of degrees of freedom of snake robots, existing research has developed versatile gaits that are combined via locomotive reduction techniques to simplify the control of a snake robot to that of a simpler system such as a differential-drive car. While being effective in many situations, the terrain traversal speed of the robot is often limited due to its reliance on solely undulatory motion. We believe combining undulation and twisting motion has the potential to speed up terrain traversal and improve motion robustness on complex terrains. With that in mind, this paper develops a rolling vector approach for a twistable snake robot, which cannot only undulate its backbone curve as existing snake robots do, but also twist about its backbone curve infinitely to achieve wheel-like rolling when needed. Results show that, compared to the existing locomotive reduction method that relies solely on undulatory motion, ours is more robust across complex terrains and moves faster (up to a 66.5% improvement) due to twisting motion. We also verify the various motion modes of our rolling vector approach on a real robot.**

I. Introduction

Snake robots are a class of hyper-redundant mechanisms that can traverse complex environments [1]–[3] such as pipes and compact spaces with limbless locomotion. Despite such locomotion benefits, motion planning for hyper-redundant robots is difficult since one must coordinate all internal degrees of freedom (DoF) in robot's shape space to achieve a desired net displacement in the workspace. To address the challenge, on the one hand, existing research has developed versatile gaits, cyclic motions in the shape space, to move the snake robot in desired ways [4]–[6]. On the other hand, by combining these gaits with a state machine and switching between them as needed, locomotive reduction techniques simplify the control of a snake robot to that of a simpler system such as a differential-drive car [7], which can then navigate complex environments in a more intuitive way.

While such a combination of gaits and locomotive reduction is effective in many situations, the motion efficiency in terms of terrain traversal speeds of the robot is often limited. Most existing gaits for limbless locomotion rely on either pure undulatory motion or rolling motion of the snake robot, without an effective combination of both. On the one hand, the undulatory motion, such as sidewinding [8], [9], allows the robot to continuously shift the contact segments along the backbone curve of the robot, and thereby transport itself by the friction at the varying contact segments. While being efficient and robust for various terrains, such undulation-based gaits are still relatively slow to traverse even flat ground. On the other hand, relying solely on rolling [10], [11] can also be slow for terrain traversal due to the small cross section of the robot and the torque limits of the joints.

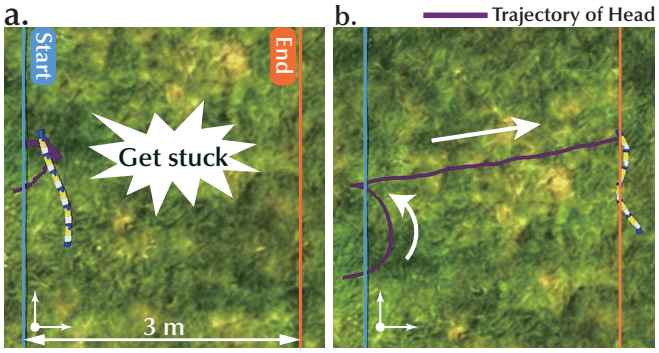

Fig. 1. Comparison of traversal speed and robustness between the baseline method and our rolling vector approach on uneven terrain. (a) Top-down view of the baseline method that achieves locomotive reduction via conical sidewinding, which gets stuck and fails to complete the 3 m translation on uneven terrain. (b) Top-down view of our rolling vector approach, which successfully completes the 180° turn and 3 m translation on the same uneven terrain without getting stuck.

We believe that relying solely on undulatory or rolling motion limits the locomotive capability of snake robots, and combining the two can enhance the speed of snake robot motion, without losing various gaits of snakes to locomote in complex environments. With that in mind, this paper considers a mechanism which we refer to as a twistable snake robot (T-Snake), and focuses on its motion planning and locomotive reduction. T-Snake cannot only undulate its backbone curve as existing snake robots do, but also roll about its backbone curve infinitely to achieve wheel-like rolling when needed. Furthermore, different parts of T-Snake can roll with different angular velocities, resulting in twisting motion (and hence the name twistable snake).

To plan motion for T-Snake, we introduce a rolling vector approach, which uses one or multiple rolling vectors to control the twist motion of the T-Snake, so that the robot can roll like wheels during undulation-based gaits such as sidewinding [8], [9]. Intuitively, when T-Snake is divided into two segments along its body, and two rolling vectors are employed, one for each segment, T-Snake can be locomotively reduced to a differential-drive car that can move forwards and turn in place.

For verification, we run both simulation and real-robot experiments. Results show that, compared to a previous locomotive reduction method [7] that relies solely on undulatory

[1]Global College, Shanghai Jiao Tong University, China (qingling.fan@sjtu.edu.cn, zhongqiang.ren@sjtu.edu.cn). This work was supported by the Natural Science Foundation of China under Grant 62403313.
† Corresponding author

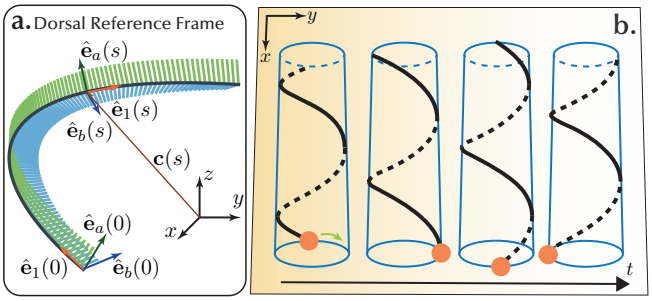

Fig. 2. Overview of the dorsal reference curve and the backbone curve of sidewinding. (a) The dorsal reference frames along the curve $\mathbf{c}(s)$. (b) The backbone curve of sidewinding is a time-varying elliptical helix.

motion such as conical sidewinding [12], ours is more robust and moves faster due to the integration of twisting motion. We find cases where the existing approach [7] gets stuck on the uneven terrain while ours does not (Fig. 1). Besides, our simulation and real-robot experiments also verify the various motion modes provided by our rolling vector approach.

II. Related Works

Designing gaits for snake robots is challenging. Existing approaches for gait design include parameterized gait equations [9], [13], [14], central pattern generators [6], [15], [16], reinforcement learning [17]–[19] and the backbone curve method [20]–[22]. This paper follows the backbone curve approach to design and analyze gaits as it allows for intuitive design of the snake robot's shape and motion.

Existing methods for locomotive reduction can be roughly grouped into two classes: bio-inspired undulatory motion and hybrid locomotion modes. Based on the omnidirectional mobility of conical sidewinding [12], the snake robot can be simplified as a differential-drive car for navigation [7]. This approach relies solely on the undulatory motion of the robot and provides relatively slow traversal speeds. Another approach involves fixing the robot's pose and relying on active wheels [23] or screw-drive units [2], [24] on the robot's skin for propulsion. Recent work [25] also proposed a barrel-like tumbling mode combined with undulatory motions to navigate steep terrain.

III. Method

We first review technical background from the literature and then introduce our rolling vector approach.

A. Description of Backbone Curves

The snake robot can be described by its backbone curve [20], including both the geometric shape of a spatial curve and the torsion along the curve. The commonly used Frenet-Serret frames can describe the geometric shape of the curve, yet lack the ability to capture an arbitrary torsion distribution along the curve [26]. Instead, the dorsal reference curve [26] can describe not only the geometric shape, but also the torsion along the curve. This work involves torsion along the curve, and we thus use the dorsal reference curve.

1) Dorsal Reference Curve: Specifically, let $\mathcal{W} : \{x^{\mathcal{W}}, y^{\mathcal{W}}, z^{\mathcal{W}}\}$ denote the world frame of a 3D workspace, and let $\mathbf{c}(s) = [x(s), y(s), z(s)]^{\mathrm{T}} \in \mathbb{R}^3$ denote a spatial curve represented in this world frame, which is parameterized by its arc-length $s \in [0, L]$. The dorsal reference frames $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_a, \hat{\mathbf{e}}_b\}$ are defined along the curve $\mathbf{c}(s)$ (Fig. 2(a)) with three orthonormal vectors, where $\hat{\mathbf{e}}_1(s)$ is the unit tangent vector along the curve $\mathbf{c}(s)$, and $\hat{\mathbf{e}}_a(s), \hat{\mathbf{e}}_b(s)$ are the unit normal and binormal vectors along the curve, respectively. The dorsal reference frames along the curve satisfy the following differential equations [26]:

$$\begin{cases} \dfrac{d\mathbf{c}}{ds} &= \hat{\mathbf{e}}_1 \\ \dfrac{d\hat{\mathbf{e}}_1}{ds} &= \kappa_b(s)\hat{\mathbf{e}}_a \quad -\kappa_a(s)\hat{\mathbf{e}}_b \\ \dfrac{d\hat{\mathbf{e}}_a}{ds} &= -\kappa_b(s)\hat{\mathbf{e}}_1 \quad +\tau(s)\hat{\mathbf{e}}_b \\ \dfrac{d\hat{\mathbf{e}}_b}{ds} &= \kappa_a(s)\hat{\mathbf{e}}_1 \quad -\tau(s)\hat{\mathbf{e}}_a \end{cases} \tag{1}$$

Equations (1) are called the *shape equations* of the dorsal reference curve, where $\kappa_a(s), \kappa_b(s)$ and $\tau(s)$ are termed *shape functions*. Here, κ_a and κ_b represent the curvatures along two axes perpendicular to the tangent axis of the curve, while τ represents the torsion along the curve.

2) Torsion Description: Based on whether κ_a, κ_b and τ are zero or not, four special cases of the dorsal reference curve were discussed in [26]. Among them, this paper uses the *complete model*, where all three shape functions $\kappa_a^{\mathcal{C}}, \kappa_b^{\mathcal{C}}$ and $\tau^{\mathcal{C}}$ in Eq. (1) can be non-zero along the curve. We refer to the dorsal reference frame defined by this complete model as the *complete frame* $\mathcal{C} : \{\hat{\mathbf{e}}_1^{\mathcal{C}}, \hat{\mathbf{e}}_a^{\mathcal{C}}, \hat{\mathbf{e}}_b^{\mathcal{C}}\}$. A curve defined by a set of complete frames $\mathcal{C}$ can exhibit both bending and twisting about its tangent vectors along the curve.

3) Backbone Curve of Sidewinding: To model the backbone curve $\mathbf{c}(s)$ of sidewinding gaits, we utilize the elliptical helix model (Fig. 2(b)) analyzed in [9]. This model is defined by time-varying parametric equations of the elliptical helix curve $\mathbf{c}^{\mathrm{eh}}$. The curve $\mathbf{c}^{\mathrm{eh}}$ is parametrized by an auxiliary variable θ to simplify the expression, as opposed to using the arc-length s for parametrization:

$$\mathbf{c}^{\mathrm{eh}}(\theta, t) = \begin{bmatrix} k_\theta \theta \\ k_b \sin(\theta + \omega_c t) \\ k_a \cos(\theta + \omega_c t) \end{bmatrix} \tag{2}$$

Here, ω_c is the temporal frequency of wave propagation along the curve; k_θ measures the pitch size; k_b and k_a stand for the horizontal and vertical gait amplitudes respectively.

To design the dorsal reference curve with an elliptical helix, we need to reparameterize Eq. (2) by arc-length s. Due to the lack of a closed-form arc-length parameterization, we use numerical integration to obtain the inverse map from θ to s based on the arc-length formula [27]:

$$s(\theta, t) = \int_0^{\theta} \| \frac{\partial \mathbf{c}^{\mathrm{eh}}(\alpha, t)}{\partial \alpha} \| \, d\alpha, \tag{3}$$

where $\theta(s, t)$ can be derived by taking the inverse of $s(\theta, t)$, i.e., $\theta(s, t) = s^{-1}(s, t)$. Substituting $\theta(s, t)$ into Eq. (2)

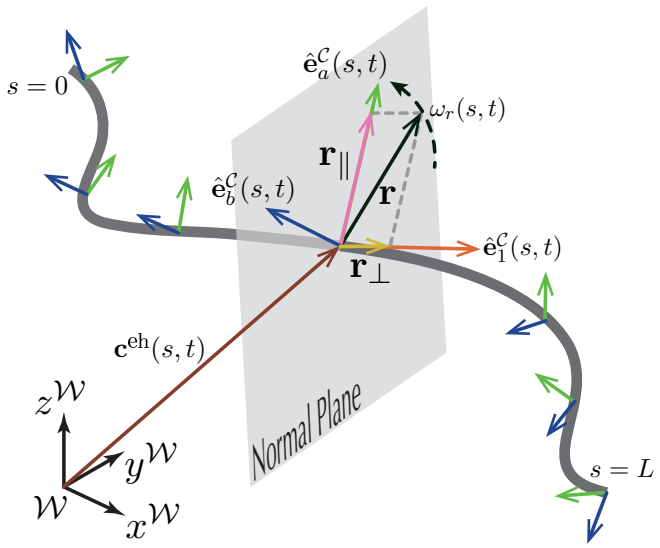

Fig. 3. Constructing complete frames via rolling vector approach. The vector $\mathbf{r}_{\parallel}$ is obtained by finding the component of $\mathbf{r}$ within the normal plane. The normal vector of complete frames, $\hat{\mathbf{e}}_a^{\mathcal{C}}$, is determined by normalizing $\mathbf{r}_{\parallel}$. The rolling vector $\mathbf{r}$ rotates about the x-axis of the world frame at an angular speed of ω_r.

yields the arc-length parametric equations $\mathbf{c}^{\text{eh}}(s,t)$ of the backbone curve of sidewinding (elliptical helix).

B. Rolling Vector Approach

This subsection constructs the complete frames $\mathcal{C}$ along the backbone curve $\mathbf{c}^{\text{eh}}$ using our rolling vector approach, and derives the shape functions within frame $\mathcal{C}$.

1) Construction of the Complete Frames: Let $\mathcal{C} : \{\hat{\mathbf{e}}_1^{\mathcal{C}}(s,t), \hat{\mathbf{e}}_a^{\mathcal{C}}(s,t), \hat{\mathbf{e}}_b^{\mathcal{C}}(s,t)\}$ denote a set of frames along the time-varying backbone curve $\mathbf{c}^{\text{eh}}(s,t)$. The tangent vector $\hat{\mathbf{e}}_1^{\mathcal{C}}$ can first be obtained using Eq. (1): $\hat{\mathbf{e}}_1^{\mathcal{C}}(s,t) = \partial \mathbf{c}^{\text{eh}}/\partial s$. To define the normal and binormal vectors of frame $\mathcal{C}$, as shown in Fig. 3, we define the normal vector $\hat{\mathbf{e}}_a^{\mathcal{C}}$ based on an intermediate vector, which we call the *rolling vector*. Let a rolling vector in the world frame $\mathcal{W}$ be $\mathbf{r}(s,t) = [r_x, r_y, r_z]^T \in \mathbb{R}^3$, which is both time-varying and arc-length-varying. We define $\mathbf{r}$ explicitly below. We project $\mathbf{r}$ onto the normal plane at any point (specified by the arc-length s) along the curve, and then obtain the component of $\mathbf{r}(s,t)$ within this plane, denoted as $\mathbf{r}_{\parallel}(s,t)$:

$$\mathbf{r}_{\parallel} = \mathbf{r} - \mathbf{r}_{\perp}, \quad \mathbf{r}_{\perp} = \left[\mathbf{r} \cdot \hat{\mathbf{e}}_1^{\mathcal{C}}\right] \hat{\mathbf{e}}_1^{\mathcal{C}} \tag{4}$$

where $\mathbf{r}_{\perp}$ is the vector generated by projecting $\mathbf{r}$ onto the tangent vector $\hat{\mathbf{e}}_1^{\mathcal{C}}$. Subsequently, we derive the normal vector of frame $\mathcal{C}$ by normalizing the vector $\mathbf{r}_{\parallel}$.[1] The binormal vector can be determined according to the right-hand rule:

$$\hat{\mathbf{e}}_a^{\mathcal{C}}(s,t) = \frac{\mathbf{r}_{\parallel}}{\|\mathbf{r}_{\parallel}\|}, \quad \hat{\mathbf{e}}_b^{\mathcal{C}}(s,t) = \hat{\mathbf{e}}_1^{\mathcal{C}} \times \hat{\mathbf{e}}_a^{\mathcal{C}} \tag{5}$$

where '$\times$' denotes the cross product of two vectors.

[1]This work considers the elliptical helix curve, along which, $\mathbf{r}_{\parallel}$ cannot be zero. For other curves, it is possible that $\mathbf{r}_{\parallel}$ is zero, which is out of the scope of this work.

2) From Frames to Shape Functions: Once the complete frames $\mathcal{C}$ along the curve have been constructed, we can apply Eq. (1) to compute the shape functions, $\kappa_a^{\mathcal{C}}, \kappa_b^{\mathcal{C}}$ and $\tau^{\mathcal{C}}$, of the curve within frame $\mathcal{C}$:

$$\kappa_a^{\mathcal{C}}(s,t) = -\frac{\partial \hat{\mathbf{e}}_1^{\mathcal{C}}}{\partial s} \cdot \hat{\mathbf{e}}_b^{\mathcal{C}} \tag{6a}$$

$$\kappa_b^{\mathcal{C}}(s,t) = \frac{\partial \hat{\mathbf{e}}_1^{\mathcal{C}}}{\partial s} \cdot \hat{\mathbf{e}}_a^{\mathcal{C}} \tag{6b}$$

$$\tau^{\mathcal{C}}(s,t) = \frac{\partial \hat{\mathbf{e}}_a^{\mathcal{C}}}{\partial s} \cdot \hat{\mathbf{e}}_b^{\mathcal{C}} \tag{6c}$$

Later sections show these shape functions bridge the backbone curve and the discrete joint angles of T-Snake.

C. Locomotive Reduction

This subsection introduces our locomotive reduction by defining the rolling vector as a piecewise function, which unifies the rolling and twisting motions of T-Snake. Furthermore, we review the kinematics of the differential-drive car, and demonstrate how this unified representation simplifies the high-dimensional snake locomotion into simpler differential-drive kinematics.

1) Piecewise Rolling Vector: We define the locus of the rolling vector $\mathbf{r}$ as a unit circle on the YoZ plane of the world frame $\mathcal{W}$. Specifically,

$$\mathbf{r}(s,t) = [0, \sin(\omega_r t), \cos(\omega_r t)]^T \tag{7}$$

where ω_r is the angular speed of the rolling vector. Recall that in the dorsal reference frame, the orientations of the dorsal and lateral sides of T-Snake can be represented by the normal and binormal vectors of the complete frames, respectively. In our rolling vector approach, the normal vector is mapped from the rolling vector. By rotating the rolling vector about the x-axis of the world frame, the normal vector along the backbone curve rotates accordingly, which induces a rolling motion of the T-Snake's body. We design the angular speed of such rolling motion, ω_r, as a piecewise linear function with respect to the arc-length s:

$$\omega_r(s) = \begin{cases} \omega_h, & s \in [0, L_h) \\ (1-\lambda)\omega_h + \lambda\omega_t, & s \in [L_h, L_t) \\ \omega_t, & s \in [L_t, L] \end{cases} \tag{8}$$

where $\lambda = \frac{s - L_h}{L_t - L_h}$. In Eq. (8), the curve is divided into three segments, the *head segment* $[0, L_h)$, the *buffer segment* $[L_h, L_t)$, and the *tail segment* $[L_t, L]$. The angular speeds of the head and tail segments are constants ω_h and ω_t, respectively. The angular speed of the buffer segment is linearly interpolated, bridging the angular speeds of the head and tail segments. By designing such piecewise linear angular speed of the rolling vector, the head and tail segments of T-Snake are transformed into two independent motion units with controllable rolling speeds. When the angular speeds of these two segments are equal, ω_r becomes a constant, generating a uniform rolling motion of the entire robot. When the speeds differ, the torsional variation between the two segments is separated by the buffer segment, which induces twisting motion in T-Snake.

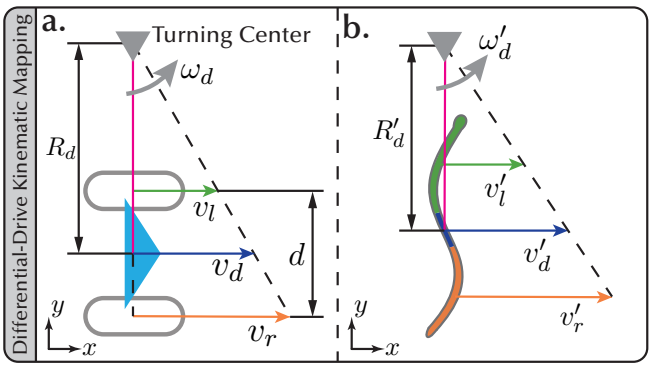

Fig. 4. Locomotive reduction via differential-drive kinematics. (a) Kinematics of a differential-drive robot. (b) The equivalent kinematics of T-Snake after locomotive reduction. With our rolling vector approach, the head (orange) and tail (green) segments function as the left and right wheels, which reduces the high-dimensional joint-space control of the snake robot into the control of a differential-drive car.

2) Differential-Drive Kinematic Mapping: Differential-drive kinematics has been extensively studied. As shown in Fig. 4(a), let v_l and v_r denote the linear velocities of the left and right wheels of a differential-drive robot at a given instant, respectively, and d be the distance between the two wheels. The linear velocity v_d of the robot is $v_d = (v_l + v_r)/2$, and its angular velocity ω_d is $\omega_d = (v_r - v_l)/d$. The turning radius R_d is $R_d = v_d/\omega_d$.

In the context of the sidewinding gait, T-Snake translates laterally relative to its backbone curve. Consequently, the spatially separated head and tail segments act like the left and right wheels of a differential-drive robot. Let k denote the velocity transmission coefficient that linearly maps the rolling angular speed of a segment to its linear velocity on the XoY plane of the world frame. The equivalent wheel velocities v'_l and v'_r can then be derived as follows:

$$v'_l = k\omega_h, \quad v'_r = k\omega_t \tag{9}$$

Furthermore, the equivalent distance between the two wheels d' corresponds to the effective distance between the centers of the head and tail segments. The centers of the head and tail segments are at $s_h = L_h/2$ and $s_t = (L_t + L)/2$, respectively. The distance d' can be approximated by the arc-length between these two points along the backbone curve: $d' \approx s_t - s_h$.

By substituting Eqs. (9) and d' into the differential-drive kinematics, we obtain the kinematics of T-Snake in the world frame parameterized solely by our piecewise rolling vector:

$$v'_d = \frac{k}{2}(\omega_h + \omega_t), \quad \omega'_d = \frac{k}{d'}(\omega_t - \omega_h), \quad R'_d = \frac{v'_d}{\omega'_d} \tag{10}$$

Through this locomotive reduction, the high-dimensional joint space control of T-Snake is reduced to a 2D control problem (ω_h, ω_t). By modulating the ratio between ω_h and ω_t, the locomotion of T-Snake can transition among translating ($R'_d \to \infty$), regular turning ($0 < R'_d < \infty$), and in-place turning ($R'_d = 0$).

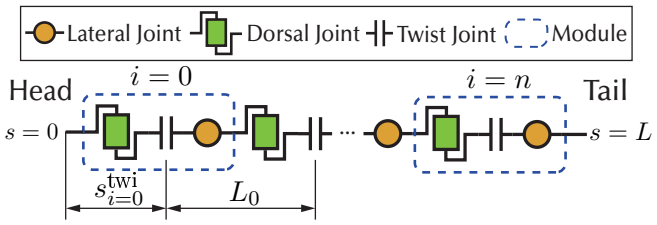

Fig. 5. Illustration of the discrete mechanism of T-Snake.

D. Mechanism and Numerical Integration

The shape functions can be numerically integrated to obtain joint angles from the shape functions [4], [10], [21]. However, for the Twistable Snake (T-Snake) considered in this paper, this integration approach needs additional modification to remain numerically stable.

1) Discrete Mechanism of T-Snake: As shown in Fig. 5, the T-Snake mechanism consists of modules, each of which incorporates three distinct 1 DoF joints: the *dorsal joint*, the *lateral joint*, and the *twist joint*. The rotation axes of the dorsal and lateral joints lie in the normal plane of the snake robot's backbone curve and pass through the geometric center of the robot's cross-section. In the robot's zero-configuration (straight pose), these two radial axes are mutually orthogonal. The axis of the twist joint aligns with the tangent axis of the snake robot's backbone curve, making it naturally orthogonal to both radial axes.

The mechanism is modular, consisting of a sequential *dorsal-twist-lateral* joint arrangement. Let the length of each module be L_0. A complete mechanism composed of n such modules has $3n$ DoFs and a total length of $L = nL_0$. This arrangement of orthogonal joints corresponds to the three shape functions defined in the complete model, where the lateral and dorsal joints are responsible for realizing the spatial accumulation of curvatures κ_a and κ_b along the curve, respectively. Meanwhile, the twist joints correspond to the accumulation of the torsion τ.

2) Naive Integration Method for T-Snake: The integration approach from [4] can be directly applied to the T-Snake by adding an integration term for the torsion.

Let θ_i and s_i be the joint angle and position of the i-th joint of the same type (lateral, dorsal or twist) along the robot. Let L_0 be the distance between two adjacent joints of the same type. Then, by integrating the shape functions (i.e., curvatures and torsion) along the curve between two adjacent dorsal (lateral, or twist) joints, the corresponding joint angles can be obtained, which can then be used to control the snake:

$$\theta^{\text{lat}}_i(t) = \int_{s^{\text{lat}}_i - L_0/2}^{s^{\text{lat}}_i + L_0/2} \kappa^{\mathcal{C}}_a(s,t)\, ds \tag{11a}$$

$$\theta^{\text{dor}}_i(t) = \int_{s^{\text{dor}}_i - L_0/2}^{s^{\text{dor}}_i + L_0/2} \kappa^{\mathcal{C}}_b(s,t)\, ds \tag{11b}$$

$$\theta^{\text{twi}}_i(t) = \int_{s^{\text{twi}}_i - L_0/2}^{s^{\text{twi}}_i + L_0/2} \tau^{\mathcal{C}}(s,t)\, ds \tag{11c}$$

Algorithm 1 Motion Generation with Phase Resetting

Input: Curve $\mathbf{c}^{\text{eh}}(s,t)$; Angular speeds ω_h, ω_t; Time step Δt; Boundaries L_h, L_t; Integration segments for joints.
Output: Joint angles $\theta_i^{\text{dor}}, \theta_i^{\text{lat}}, \theta_i^{\text{twi}}$.

Initialize phases $\phi_h \leftarrow 0, \phi_t \leftarrow 0$, and cycle counter $C_{\text{twist}} \leftarrow 0$
for each control step t **do**
`STEP-1: Phase Update & Resetting`
  $\phi_h \leftarrow \phi_h + \omega_h \Delta t, \quad \phi_t \leftarrow \phi_t + \omega_t \Delta t$
  if $|\phi_h - \phi_t| \geq 2\pi$ **then**
    $sgn \leftarrow \text{sign}(\phi_h - \phi_t)$ ▷ $sgn \in \{-1, 1\}$
    $\phi_h \leftarrow \phi_h - 2\pi \cdot sgn, \quad C_{\text{twist}} \leftarrow C_{\text{twist}} + sgn$
`STEP-2: Build Frames via Rolling Vector`
  $\hat{\mathbf{e}}_1^{\mathcal{C}} \leftarrow \partial \mathbf{c}^{\text{eh}} / \partial s, \quad \mathbf{r} \leftarrow [0, \sin(\omega_r t), \cos(\omega_r t)]^T$
  $\hat{\mathbf{e}}_a^{\mathcal{C}} \leftarrow \text{normalize}(\mathbf{r} - (\mathbf{r} \cdot \hat{\mathbf{e}}_1^{\mathcal{C}})\hat{\mathbf{e}}_1^{\mathcal{C}}), \quad \hat{\mathbf{e}}_b^{\mathcal{C}} \leftarrow \hat{\mathbf{e}}_1^{\mathcal{C}} \times \hat{\mathbf{e}}_a^{\mathcal{C}}$
`STEP-3: Shape Functions & Joint Angles`
  Compute $\kappa_a^{\mathcal{C}}, \kappa_b^{\mathcal{C}}, \tau^{\mathcal{C}}$ using Eqs. (6)
  $\theta_i^{\text{dor}} \leftarrow \int \kappa_b^{\mathcal{C}} ds, \quad \theta_i^{\text{lat}} \leftarrow \int \kappa_a^{\mathcal{C}} ds, \quad \theta_i^{\text{twi}} \leftarrow \int \tau^{\mathcal{C}} ds$
`STEP-4: Twist-Joint Angle Compensation`
  $\theta_i^{\text{twi}} \leftarrow \theta_i^{\text{twi}} - C_{\text{twist}} \cdot \pi$ **for** each joint $s_i^{\text{twi}} \in [L_h, L_t)$
return $\theta_i^{\text{dor}}, \theta_i^{\text{lat}}, \theta_i^{\text{twi}}$

3) Numerical Problem: As mentioned in Sec. III-C, when the angular speeds of the head and tail segments differ, the torsional deformation accumulates within the buffer segment of T-Snake, inducing turning motions. The extent of such torsional accumulation can be quantified by the phase difference between the rolling vectors of the two segments: $\Delta\phi = (\omega_t - \omega_h)t$. By applying the existing integration method (Eqs. (11)) for the turning motions of T-Snake, we observe that as $|\Delta\phi|$ increases, the resulting joint angles of the dorsal and lateral joints within the buffer segment become numerically unstable, which further leads to distorted motion of T-Snake, resulting in unintended motion trajectories. The reason for this numerical problem is left for future work.

4) Phase Resetting: To overcome this numerical problem, we introduce a trick which we call it *phase resetting*. The main idea is that the torsional deformation of the backbone curve is periodic. A relative phase accumulation of 2π between the head and tail rolling vectors can be physically "absorbed" by the twist joints of T-Snake. Specifically, we monitor the phase difference $\Delta\phi$ during the motion generation. When $|\Delta\phi|$ reaches or exceeds 2π, a phase reset is triggered. We subtract 2π from the leading phase to bound the relative phase difference $\Delta\phi$ strictly within $(-2\pi, 2\pi)$. Concurrently, to prevent discontinuous jumps in the physical robot's motion due to this mathematical reset, we introduce a discrete compensation term to the twist joints within the buffer segment. By maintaining an integer cycle counter C_{twist}, an angular compensation of $-C_{\text{twist}} \cdot \pi$ is directly added to these twist joints. The entire approach with rolling vector and phase resetting is summarized in Alg. 1.

IV. Experimental Results

This section shows the results of both simulation and real-robot experiments on T-Snake. We first conduct multiple simulations to validate our phase resetting approach in computing stable joint angles (Sec. IV-A.1). Then, we illustrate various motion modes generated by different combinations of rolling angular speeds (Sec. IV-A.2). Additionally, compared with the existing locomotive reduction based on conical sidewinding [7], [12], we show that our method provides better robustness on complex terrains (Sec. IV-A.3). Finally, we run real-robot experiments to validate the translating and turning capabilities on the T-Snake hardware (Sec. IV-B).

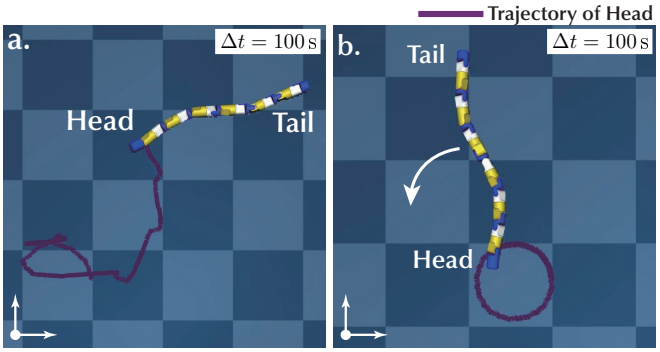

Fig. 6. Snapshots from the simulation of turning motion, where the joint angles are computed using the existing numerical integration method (a) without and (b) with phase resetting, respectively. (a) suffers from the numerical problem and the trajectory of the snake head distorts as time increases, while (b) avoids the problem.

In our experiments, for the geometric shape of $\mathbf{c}^{\text{eh}}$ (Eq. (2)), k_a, k_b and k_θ are 0.02, 0.06 and $(1/2\pi)$ m, respectively. The T-Snake used for experiments consists of 6 modules, and the length L_0 of each module is 0.2055 m. For the buffer segment (Eq. (8)), the start and end points, L_h and L_t, are set at 0.5430 and 0.7485 m, respectively. The time step Δt in Algorithm 1 is 0.02 s. Other parameters for generating distinct motion modes are provided later. Our simulations are conducted in MuJoCo [28].

A. Simulation Experiments

1) Validation of Phase Resetting Approach: This experiment simulates the turning motions ($\omega_h \neq \omega_t$) of T-Snake. We set the head and tail angular speeds, ω_h and ω_t, as 1 and 3 rad/s to generate turning motion. Both our method and the baseline utilize the existing integration method (Eqs. (11)) to compute joint angles; the only difference lies in whether the phase resetting approach is enabled. We simulate the motions for 100 seconds. As shown in Fig. 6, the resulting motion of the robot head without phase resetting distorts as time increases, which is supposed to be a circle (Fig. 6(a)). In contrast, with our method enabled, the resulting head trajectory maintains an obvious circular shape and continuously rotates around the turning center as time evolves (Fig. 6(b)). This experiment verifies the existence of the numerical problem and the effectiveness of our approach.

2) Different Motion Modes: By varying the ratio between ω_h and ω_t, we show multiple motion modes of T-Snake. As shown in Fig. 7(a), for translation, we set these two angular speeds to be equal (2 rad/s) and the temporal frequency of the backbone curve, ω_c, to a non-zero constant (1 rad/s) to generate the rolling and undulatory motions on T-Snake simultaneously. To achieve in-place turning (Fig. 7(b)), we

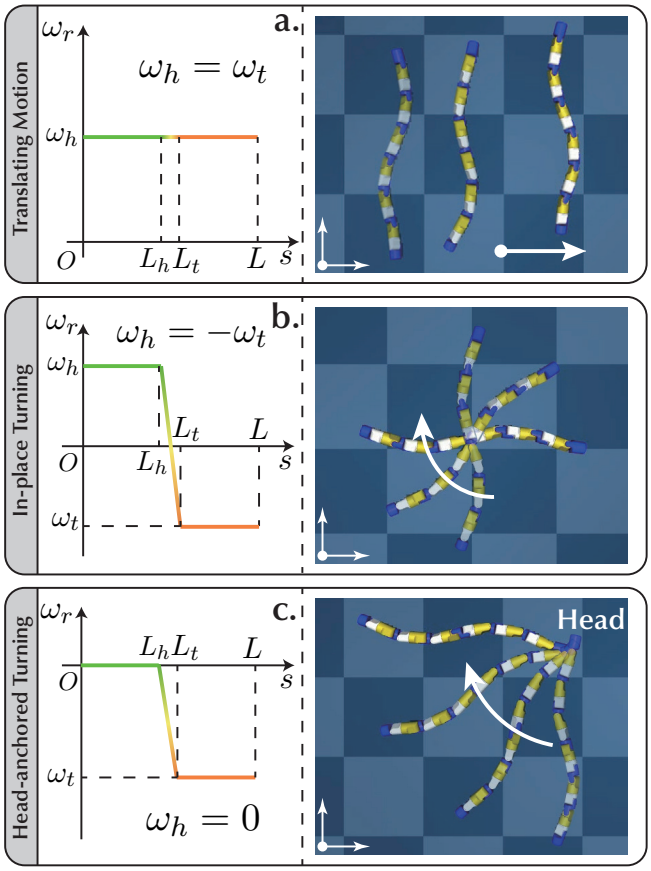

Fig. 7. Experimental validation of the proposed motion generating method on the T-Snake. The left column illustrates the piecewise linear angular speed ω_r along the backbone curve for different motion modes. The right column shows the corresponding snapshots of T-Snake's locomotion, including (a) translating motion ($\omega_h = \omega_t$), (b) in-place turning ($\omega_h = -\omega_t$), and (c) head-anchored turning ($\omega_h = 0$).

set the two angular speeds as opposites (i.e., $\omega_h = -\omega_t = 2$ rad/s). By setting $\omega_h = 0$, we can also realize the head-anchored turning (Fig. 7(c)). These results validate our method's ability to generate translation and turning motions for T-Snake.

3) Robustness across Complex Terrains: This experiment is conducted on randomly generated uneven terrain, with a maximum height variation of 0.2 m and a standard deviation of 0.0593 m. We compare our method against the existing locomotive reduction method based on conical sidewinding [7], [12], [29], which enables snake robots to turn by transforming the backbone curve from the elliptical helix of regular sidewinding into a conical helix, and relies on undulatory motion only.

The task is to execute a 180° in-place turn first, then traverse a 3 m track (Fig. 1). Additionally, the experiments are conducted on both flat ground and the aforementioned complex terrain. We record both the turning and translation times for comparison, and each test is repeated 5 times.

As shown in Fig.8, our approach achieves faster traversal speeds. On flat ground, ours reduces the turning and translating times by 42.3% and 66.5%, respectively. Moreover, our method exhibits better robustness on complex terrains. During the uneven terrain trials, the baseline method suffers a performance drop in turning (58.78 ± 7.87 s) and gets stuck during translation (Fig. 1(a)), while ours does not. Ours completes the turning on uneven terrain in 20.86 ± 1.66 s (a 64.5% improvement) and successfully navigates the 3 m translation in 37.90 ± 11.0 s without getting stuck (Fig.1(b)). These results confirm that our rolling vector approach not only accelerates T-Snake locomotion, but also improves its robustness in the navigation of unstructured environments.

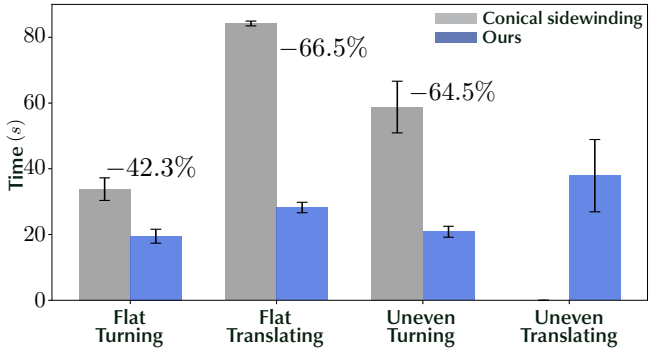

Fig. 8. Bar chart comparing the mean times for turning and translating tasks on flat and uneven terrain for the baseline locomotive reduction method and our new method. Snapshots from these simulations are shown in Fig. 1.

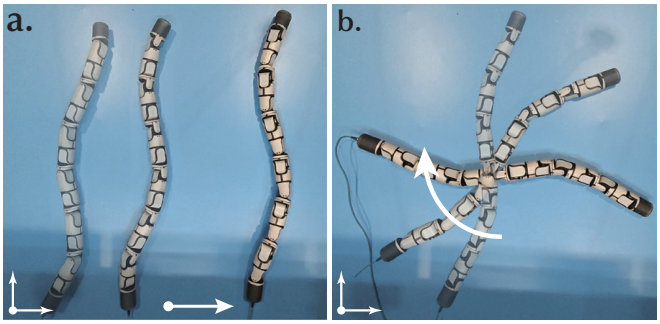

Fig. 9. Real-robot validation of the proposed motion-generating method on the T-Snake hardware. (a) Translating motion ($\omega_h = \omega_t$). (b) In-place turning ($\omega_h = -\omega_t$).

B. Real Robot Experiments

This experiment validates the translating and turning capabilities of our method on a physical T-Snake.[2] The hardware consists of six modules with a total of $3 \times 6 = 18$ DoFs. We conduct this experiment in a flat indoor environment.

1) Translating: We set $\omega_c = \omega_h = \omega_t = 0.6$ rad/s to realize this motion. Under this condition, the angular speed ω_r (Eq. (8)) degrades to a constant. The resulting translation is illustrated in Fig. 9(a).

[2]Our physical T-Snake implements the mechanism in Fig. 5 as described in Sec. III-D, and the design of this robot is neither the focus nor claimed as a contribution of this paper.

2) In-place Turning: We then set $\omega_c = 0$ and $\omega_h = -\omega_t = 0.4$ rad/s to generate the in-place turning motion. Fig. 9(b) shows the resulting turning motion.

These results verify the effectiveness of the proposed motion generating method on real robots.

V. Conclusion

In this paper, we present a novel locomotive reduction method for a twistable snake robot (T-Snake) based on a piecewise rolling vector approach. By decoupling the rolling angular speeds of the head and tail segments, we map the high-dimensional shape space control problem into an intuitive 2D differential-drive kinematic model. Furthermore, we introduce a phase resetting method to eliminate the torsional accumulation of the buffer segment and ensure motion consistency. Experimental results on both flat and uneven terrains confirm that, compared to the conventional locomotive-reduction method based on conical sidewinding, our approach speeds up terrain traversal and improves robustness when traversing uneven terrains.

Future work includes investigating the fundamental reasons and principles behind the numerical problem why the existing integration method fails to produce correct motion when the backbone curve twists significantly.

References

[1] K. Y. Pettersen, "Snake robots," *Annual Reviews in Control*, vol. 44, pp. 19–44, 2017.

[2] F. Richter, P. V. Gavrilov, H. M. Lam, A. Degani, and M. C. Yip, "Arcsnake: Reconfigurable snakelike robot with archimedean screw propulsion for multidomain mobility," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 797–809, 2021.

[3] T. S. Vaquero, G. Daddi, R. Thakker, M. Paton, A. Jasour, M. P. Strub, R. M. Swan, R. Royce, M. Gildner, P. Tosi *et al.*, "Eels: Autonomous snake-like robot with task and motion planning capabilities for ice world exploration," *Science robotics*, vol. 9, no. 88, p. eadh8332, 2024.

[4] H. Yamada and S. Hirose, "Approximations to continuous curves of active cord mechanism made of arc-shaped joints or double joints," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 703–708.

[5] R. L. Hatton and H. Choset, "Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction," *Autonomous Robots*, vol. 28, no. 3, pp. 271–281, 2010.

[6] Z. Wang, Q. Gao, and H. Zhao, "Cpg-inspired locomotion control for a snake robot basing on nonlinear oscillators," *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 209–227, 2017.

[7] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset, "Locomotive reduction for snake robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3735–3740.

[8] J. W. Burdick, J. Radford, and G. S. Chirikjian, "A'sidewinding'locomotion gait for hyper-redundant robots," *Advanced Robotics*, vol. 9, no. 3, pp. 195–216, 1994.

[9] R. L. Hatton and H. Choset, "Sidewinding on slopes," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 691–696.

[10] W. Zhen, C. Gong, and H. Choset, "Modeling rolling gaits of a snake robot," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 3741–3746.

[11] D. Rollinson and H. Choset, "Pipe network locomotion with a snake robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 322–336, 2016.

[12] C. Gong, M. J. Travers, X. Fu, and H. Choset, "Extended gait equation for sidewinding," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5162–5167.

[13] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and scripted gaits for modular snake robots," *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, 2009.

[14] B. Chong, T. Wang, K. Diaz, C. J. Pierce, E. Erickson, J. Whitman, Y. Deng, E. Flores, R. Fu, J. He *et al.*, "The omega turn: A general turning template for elongate robots," *arXiv preprint arXiv:2510.12970*, 2025.

[15] X. Yu, B. Nguyen, and W. O. Friesen, "Sensory feedback can coordinate the swimming activity of the leech," *Journal of Neuroscience*, vol. 19, no. 11, pp. 4634–4643, 1999.

[16] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[17] Z. Bing, L. Cheng, K. Huang, and A. Knoll, "Simulation to real: learning energy-efficient slithering gaits for a snake-like robot," *IEEE Robotics & Automation Magazine*, vol. 29, no. 4, pp. 92–103, 2022.

[18] S. Jiang, A. Salagame, A. Ramezani, and L. L. Wong, "Hierarchical rl-guided large-scale navigation of a snake robot," in *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2024, pp. 1347–1352.

[19] F. Sheng, F. Wan, C. Tang, and X. Guo, "A maneuverable winding gait for snake robots based on a delay-aware swing and grasp framework combining rules and learning methods (ra-l)," *IEEE Robotics and Automation Letters*, 2024.

[20] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on robotics and automation*, vol. 10, no. 3, pp. 343–354, 2002.

[21] T. Takemori, M. Tanaka, and F. Matsuno, "Adaptive helical rolling of a snake robot to a straight pipe with irregular cross-sectional shape," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 437–451, 2022.

[22] M. Tanaka, M. Nakajima, Y. Suzuki, and K. Tanaka, "Development and control of articulated mobile robot for climbing steep stairs," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 531–541, 2018.

[23] Y. Takagi, M. Ishikawa, and K. Osuka, "A kinematic-dual snake robot: Undulatory mobile robot driven by controllable side-thrust links," *Mechatronics*, vol. 90, p. 102944, 2023.

[24] H. Fukushima, S. Satomura, T. Kawai, M. Tanaka, T. Kamegawa, and F. Matsuno, "Modeling and control of a snake-like robot using the screw-drive mechanism," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 541–554, 2012.

[25] A. Salagame, H. Noyes, E. Sihite, A. Kalantari, and A. Ramezani, "Crater observing bioinspired rolling articulator (cobra)," *Advanced Intelligent Systems*, vol. 8, no. 3, p. 2500352, 2026.

[26] H. Yamada and S. Hirose, "Study on the 3d shape of active cord mechanism," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 2890–2895.

[27] M. P. Do Carmo, *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[28] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[29] I. Tanev, T. Ray, and A. Buller, "Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot," *IEEE Transactions on robotics*, vol. 21, no. 4, pp. 632–645, 2005.