ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2024.2.061

Geometric Algebra Modeling of Snake-like Robot Serpentine Locomotion: Preliminary study

Mohammed Ali Shehadeh[1,2] ✉, **Radomil Matousek**[1,3], **Tomas Hulka**[3], **Tomas Holoubek**[1]

[1]Brno University of Technology, FEEC, Brno, CZ
[2]Bosch Powertrain Ltd., Jihlava, CZ
[3]CEITEC - Central European Institute of Technology, Brno, CZ
{Mhd.Ali.Shehadeh, RMatousek}@vutbr.cz✉,Tomas.Hulka@vutbr.cz

Abstract

This study analyses and simulates a snake robot's locomotion in complex environments, exploring the impact of serpentine gait parameters like magnitude, shape, and curvature on its motion in linear and circular gaits. The research reveals how small errors in the approximation process can accumulate, significantly affecting the robot's path. Ninety-six scenarios are tested to validate the model and demonstrate that CoppeliaSim's simulations closely match Maple's numerical simulations, especially for longer snakes. The study highlights the minimal role of wheel positions in altering the trajectory and underscores the significance of design parameters and friction in the robot's motion. This comprehensive analysis enhances our understanding of snake robot locomotion and informs their development for diverse applications.

Keywords: *CoppeliaSim, Geometric-Algebra, Serpenoid Curve, Serpentine Gait, Snake-like Robot.*

Received: 18 February 2024
Accepted: 30 August 2024
Online: 20 December 2024
Published: 20 December 2024

1 Introduction

Bio-inspired robotics has been a viral topic in the field of robotics for the past few decades, drawing inspiration from the remarkable world of biological systems. The pioneering work of zoologist J. Gray in 1946 (13), who studied the locomotion mechanisms of snakes, marked the early exploration of this domain. Subsequently, in 1971, Hirose initiated bio-mechanical studies involving real snakes and the development of snake-like robots (17). Since then, numerous snake-like robots have been designed and constructed, reflecting the enduring fascination with this bio-inspired approach (36; 9; 11; 25; 21).

The fascination in bio-inspired robotics is justifiable, given that animals have undergone millennia of evolution to optimize their daily activities. The remarkable abilities exhibited by many of these animals serve as a rich source of inspiration for advancements in robotics. Notably, snake-like robots have emerged as a distinguished class of bio-inspired robots, renowned for their unique locomotion capabilities.

Unlike traditional robots that predominantly rely on wheels and legs for mobility, snake-like robots present an intriguing alternative. These robots are designed to mimic snakes locomotion, a promising area of research that offers several distinct advantages, including superior stability and terrain adaptability when compared to traditional wheeled and legged robots.

Snakes, due to their flexible gaits, exhibit remarkable ability to navigate diverse environments. These creatures can gracefully traverse not only solid surfaces but also maneuver through mud, water, grassy terrains, sandy landscapes, and various other environmental irregularities. Such unique capabilities have fueled active research aimed at emulating the biological locomotion of snakes in the development of robots (1).

Snake-like robots can be used in numerous applications, for instance, to explore challenging terrains or find missing people in disastrous situations such as earthquakes, fire sites, pipe inspections (5; 22; 38). Additionally, they have proven valuable in airplane fuel tank inspections (26). They have also been employed in the medical field for minimally invasive surgeries (28).

This paper addresses the challenge of controlling a snake-like robot, focusing on its motion along straight lines and curvatures with a constant radius. Essentially, any trajectory can be regarded as a synthesis of straight lines and curves with varying radii.

The mathematical analysis of the snake robot's serpentine motion on level ground with no-slip conditions is presented, based on geometrical algebra. The resulting gaits are then compared with simulations in CoppeliaSim, providing insights into the influence of parameters such as magnitude, shape, and curvature on its motion in linear and circular paths. Moreover, specific attention is given to understanding the impact of these parameters on the initial slip that occurs at the very beginning of the robot motion leaving the singular position q_0.

2 Biology of Snakes

The most common locomotion strategy employed by biological snakes relies mostly on interactions with obstacles present in the surrounding environment. As the snake's body undergoes lateral bends along its backbone, these bends can leverage obstacles to propel the snake forward (12). However, when obstacles are absent in the environment, the efficient propulsion through lateral undulation depends on the interactions between the snake's scales and the surface irregularities of the terrain's surface (19). Therefore, the frictional anisotropy created by the directional arrangement of scales on snake skin becomes crucial for propelling snakes across flat, hard surfaces. Branyan et al. (5) have demonstrated methods to enhance a robot's locomotion capabilities when implementing a lateral undulation gait by tuning the robot's skin design and fabrication.

While many robot models incorporate side wheels to provide essential lateral resistance for effective undulation (30; 27; 16; 21), this approach limits the traversable terrain to hard and flat surfaces. Our interest lies in analysing the movement rather than testing it across diverse terrains; hence, we opt for a model dependent on side wheels as well to simplify calculations. This choice allows us to focus on understanding and simulating snake-like motion with an emphasis on the impact of different parameters on lateral undulation.

3 Spectrum of Snake-Like Robot Locomotion

Several gaits of snake-like robots, such as the serpentine gait, side-winding gait, sinus-lift gait, rolling gait, rectilinear, etc., have been extensively studied (16; 6; 33; 23; 7; 35; 24). Among these gaits, the serpentine gait stands out as the most common form of locomotion, observed not only in terrestrial reptiles but also in numerous aquatic species, including almost all types of fish (18) and aquatic animals.

3.1 Serpentine Gait

The study of the serpentine gait can be broadly categorised into two classes: one based on the sinusoidal curve and the other on the central pattern generator (CPG). The sinusoidal curve-based approach, pioneered by Hirose in his study on biologically inspired robots (16), involves approximating the serpentine locomotion curvature of snakes using a sinusoidal curve. Hirose introduced the concept of the "serpenoid" curve, drawing inspiration from bionics. Various mathematical functions have been proposed to define the serpenoid curve (34; 31; 17). According to Prautsch et al. (31), the continuous serpenoid curve can be described as follows:

$$\begin{aligned} x(s) &= \int_0^s \cos(a\cos(b\tau) + c\tau)\, d\tau, \\ y(s) &= \int_0^s \sin(a\cos(b\tau) + c\tau)\, d\tau. \end{aligned} \tag{1}$$

The serpenoid parameters a, b, and c, intrinsic to the properties of snake robot locomotion, define the wave shape and can be manipulated to control the snake's motion. Specifically, parameter a represents undulation, b represents the number of periods, and c influences the curl of the curve.

Alternatively, the serpentine gait can be generated using the CPG method. Ijspeert and Crespi applied the CPG to control snake robots and optimized CPG parameters (10). A limitation of the sinusoidal gait is that the amplitude and the frequency are fixed and cannot correspond to the changing environment. In contrast, the CPG-based gait allows for oscillation modification through sensory feedback, enabling adaptation to varying environments (37).

Specifically, the CPG can couple the dynamics of robots, making the feedback controller more effective and robust (39; 8).

In the work by Guo, Ma, Li, and Fang (14), a novel serpentine gait generation method for snake-like robots based on geometric mechanics has been introduced. Notably, this proposed method offers greater flexibility by not restricting the gait to a sinusoidal signal.

3.2 Gait Pattern Constraints

In essence, both the sinusoidal curve-based gait and the CPG-based gait assume that the joint angles of snake-like robots follow signals from the sinusoidal family, thereby restricting the gait pattern (14). However, the most efficient serpentine gaits may not necessarily adhere to sinusoidal signals. Intuitively, effective serpentine gaits should manifest as different periodic signals tailored to diverse net locomotion requirements, rather than strictly conforming to sinusoidal family signals (14).

A snake-like robot can be viewed as a multibody system with non-holonomic constraints, which can be controlled by feeding sinusoidal signals to the joint angles, though it is not exclusively limited to this method. The geometric mechanics introduced by Poincaré, Arnold, and Marsden offer a descriptive framework for the geometric structure of locomotion (8). From a geometric perspective, the configuration of the snake-like robot can be seen as a principal fiber bundle, in which the base manifold is the joint angles space, and the fiber manifold is the net locomotion. Building upon this insight, Boyer et al. developed a suite of generic tools for studying bioinspired locomotion in robotics (4; 2; 3). In this paper, we follow a similar approach, where the configuration of the snake-like robot has the only permitted locomotion manifold restricted by Pfaffian constrains acting on all wheels.

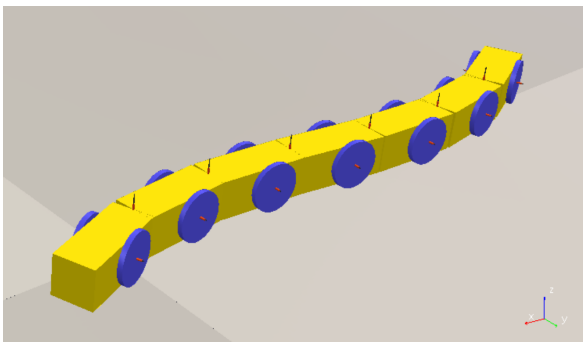

Figure 1: Illustration of the 7-link Snake-like Robot in CoppeliaSim.

4 Mathematical Modeling of a Multi-Link Snake-Like Robot

The mathematical model of our 7-link kinematic snake comprises seven solid blocks, each actuated by servo-motors. This representation, depicted in Fig. 1, serves as an approximation to real-life snakes.

All links share the same parameters, that is, the same servo motors, same dimensions, and same positioning of the wheels in each block. To explore the effects of changing the wheel position, we conduct all experiments with two settings: one where the wheel is placed in the middle to ensure static balance of the blocks, and another setting where it is positioned at an arbitrary point offset from the midpoint. For instance, we choose for the latter configuration to be positioned at the golden ratio, where the distance from the front side to the wheel, denoted as L_1, measures 0.062 m, while the distance from the wheel to the other end, L_2, is 0.038 m.

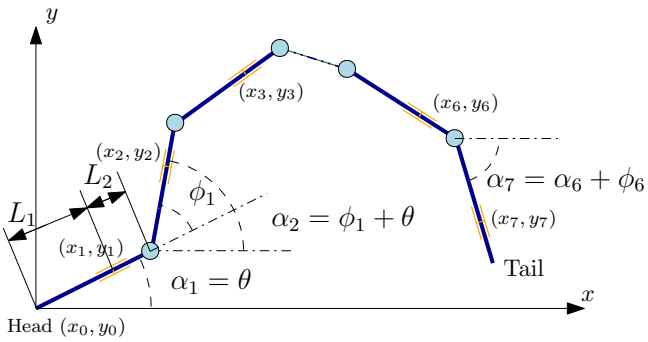

Figure 2: The schematic representation of the 7-link snake, illustrating the nine general coordinates.

The snake is represented by nine general coordinates, namely $q = x_0, y_0, \theta, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6$. Where the first three coordinates form the configuration vector $\mathbf{w} := (x_0, y_0, \theta)^T$, determine the snake-head's position. And the remaining six coordinates form the shape variables $\phi := (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)^T$, which describe the relative angular displacement between each two successive links. Given that the global angular displacement for each link, designated by α_i, accumulates through the previous links, it can be described as $\alpha_{i+1} = \alpha_i + \phi_i$. Notably, the initial global displacement α_1 corresponds solely to the head's orientation, θ.

5 Geometric Algebra Approach

To derive the kinematic description of the 7-link Snake-Like robot, let $(x_i, y_i) : i = 1, 2, ..., 7$ denote the coordinates of the wheels, respectively. Considering that the length of each link is much greater than its width, the latter can be neglected. The positions of the wheels can then be expressed as:

$$x_i = x_{i-1} + L_1.\cos(\alpha_i) + L_2.\cos(\alpha_i) \quad : i = 1, 2, ..., 6, \tag{2}$$

$$y_i = y_{i-1} + L_1.\sin(\alpha_i) + L_2.\sin(\alpha_i) \quad : i = 1, 2, ..., 6. \tag{3}$$

The velocity vector of each wheel, $v_i = (\dot{x}_i, \dot{y}_i)$, is derived by differentiating the position of the wheel with respect to time. The normal vector at each wheel may be described by global displacements as:

$$n_i := (-\sin(\alpha_i), \cos(\alpha_i)). \tag{4}$$

For simplicity, we do not impose any limitations on the friction forces acting on the snake's wheels, which is not guaranteed in reality since side-way slipping may occur when contact surfaces can not endure the applied friction force.
The nonholonomic constraint of the nonslip assumption asserts that the only admissible direction of the velocity vector is parallel to the corresponding link wheels, due to the assumption that the wheels do not slip nor rotate. Thus, the dot product of n_i, v_i will be zero:

$$-\sin(\alpha_i) \cdot \dot{x}_i + \cos(\alpha_i) \cdot \dot{y}_i = 0. \tag{5}$$

This results in a system of seven differential equations for nine unknowns. From these equations (5), we can derive the constraint matrix (Pfaffian matrix) $A(q)$, and the kinematic system is described as:

$$A(q) \cdot \dot{q} = 0. \tag{6}$$

Further splitting the matrix A with respect to the first three columns, denoted as $B(q)$, and the remaining six columns as $C(q)$, the dynamical system can be expressed as:

$$B(q) \cdot \begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{\theta} \end{pmatrix} + C(q) \cdot \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \\ \dot{\phi}_5 \\ \dot{\phi}_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{7}$$

Since we are interested in controlling the configuration vector $\mathbf{w}$, we write the system in the following way:

$$\begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{\theta} \end{pmatrix} = B^{-1}(q) \cdot -C(q) \cdot \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \\ \dot{\phi}_5 \\ \dot{\phi}_6 \end{pmatrix}. \tag{8}$$

Provided that the generalised inverse, pseudo-inverse, of B is well defined, which holds true since B can be considered as a linear operator between two Hilbert spaces $B : \dot{\mathbf{w}} \rightarrow -C\dot{\phi}$ (15). Therefore, $B^{-1}(q) \cdot -C(q)$ is the controlling matrix for $\mathbf{w} = (x, y, \theta)$. By extending this matrix with the unitary matrix $I(6 \times 6)$, we get the control matrix G:

$$G = \begin{pmatrix} B^{-1}(q) \cdot -C(q) \\ I_{6\cdot 6} \end{pmatrix}, \tag{9}$$

and the whole system can be concisely written as:

$$\dot{q} = G \cdot \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \\ \dot{\phi}_5 \\ \dot{\phi}_6 \end{pmatrix}. \tag{10}$$

6 Numerical Simulation

In the context of the system described by eq. (10), the gait is achieved by multiplying the control matrix G by the controlling angular velocities $\dot{\phi}$. The matrix G defines the tangent space in which the snake is allowed to move depending on the current position of all nine global coordinates q. The control input depends on $\dot{\phi}_1$, $\dot{\phi}_2$, and so forth.

To simulate the snake's movement, we perform recurrent numerical iterations using G and Q. The simulation initiates with an initial position q_0, and q can take any value from the reachable set within the control limits. For a direct comparison between the numerical results and the simulation environment, we model the snake to exhibit a serpentine motion pattern.

Let l denote the length of each link, and define $\beta = l \cdot b$, $\gamma = -l \cdot c$, and $\mu = 2a \sin\left(\frac{\beta}{2}\right)$. We derive from eq. (1), a more elegant expression for relative angles, as presented by (20).

$$\phi_i(t) = \mu \cdot \sin(\omega \cdot t + (i-1) \cdot \beta) + \gamma, \tag{11}$$

where ω is a constant angular frequency, and the controlling angular velocities are thus given by:

$$\begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \\ \dot{\phi}_5 \\ \dot{\phi}_6 \end{pmatrix} = \begin{pmatrix} \mu \cdot \omega \cdot cos(\omega \cdot t + 0 \cdot \beta) \\ \mu \cdot \omega \cdot cos(\omega \cdot t + 1 \cdot \beta) \\ \mu \cdot \omega \cdot cos(\omega \cdot t + 2 \cdot \beta) \\ \mu \cdot \omega \cdot cos(\omega \cdot t + 3 \cdot \beta) \\ \mu \cdot \omega \cdot cos(\omega \cdot t + 4 \cdot \beta) \\ \mu \cdot \omega \cdot cos(\omega \cdot t + 5 \cdot \beta) \end{pmatrix}. \tag{12}$$

By exploring various serpentine gait parameter settings, we compare twelve sets, each a combination of the following values: the amplitude $a = \{0.5, 1\}$ of a unitary or half unitary value, the shape coefficient $b = \{-\frac{10\times\pi}{n}, -\frac{5\times\pi}{n}\}$ in which the snake takes the shape of a full or half a sinusoidal wave, the curvature $c = \{0, 0.125, 0.5\}$ stands for a linear movement when $c = 0$, and circular otherwise, the chosen time-step is equal to 50 ms, and the angular frequency ω is set to be 1 $rad \cdot s^{-1}$.

7 Starting Angle

The snake robot is initially configured with $q_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0)$, aligning all links along the x-axis. To transition from this configuration and initiate any form of serpentine gait, the wheels experience high impulse forces, leading to an unavoidable initial slip. While the wheels are designed to avoid slipping sideways, a degree of slip during the first few iterations is inevitable. resulting in a rotated coordinate system.

Consider the control objective of steering the snake robot into a straight path while maintaining a heading parallel to the path. To achieve this, a global coordinate system is defined (see Fig. 2), with the x-axis aligned with the desired straight path and the y-axis representing the cross-track error (29). The control objective is formulated as $\lim_{i\rightarrow\infty} y_i = 0$.

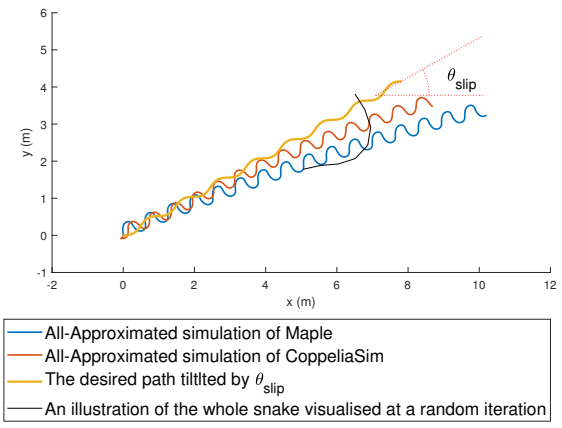

Figure 3: Visualization of the snake's head position during a 2000-step simulation with seven links applying All-Approximated Algorithm. The blue curve illustrates the Maple simulation, while the orange curve corresponds to CoppeliaSim. Additionally, a 5-times magnified representation of the entire snake is plotted to enhance readability.

Due to the aforementioned rotated coordinate system, the resulting path appears as a line tilted with an angle θ_{slip} away from the x-axis Fig. 3.

In all of our simulations, observations indicate that the winding angle μ, which represents the joint oscillation's amplitude, plays a pivotal role in determining how the snake departs from its initial configuration

and commences its motion in a sinusoidal curve θ_{slip}. We noticed that setting ϕ_i to start from a zero angle causes the snake to commence its motion from a different point on the sinusoidal curve. To address this, we suggested introducing an additional parameter, θ_{comp}, to regulate the starting angle, as follows:

$$\phi_i(t) = \mu \cdot sin(\omega \cdot t + (i-1) \cdot \beta + \theta_{copm}) + \gamma. \quad (13)$$

However, the derivation of a definitive formula for computing θ_{comp} remains an ongoing challenge. Interested readers are encouraged to refer to the work by Shi et al. (32), where the Line-of-Sight (LOS) guidance law is adeptly employed to address the compensation for slipping in snake-like robot locomotion. Upon determining θ_{comp}, and applying it as a pre-adjustment to the initial configuration q_0 of the snake, the system exhibits convergence towards the desired x-axis direction. In light of the diverse factors influencing the value of θ_{comp}, we have chosen to keep the snake's motion within the rotated coordinate frame. Once the definitive formula is established, its application as a pre-adjustment suffices to facilitate the desired motion.

8 Simulation Methodology

The simulation methodology is detailed through the following pseudo-code, implemented in Maple:

Algorithm 1 All-Approximated Algorithm

1: **Input:** Initial position of the snake $\mathbf{q}_0$
2: **Input:** θ_{comp}
3: **for** Every time step t **do**
4: Evaluate matrices $\mathbf{B}, \mathbf{C}$ using eq. (7)
5: Calculate pseudo-inverse matrix $\mathbf{B}^{-1}$
6: Evaluate matrix $\mathbf{G}$ using eq. (9)
7: Calculate current $\dot{\mathbf{q}}$ (derivative of the configuration space) using eq. (9)
8: **for** i from 1 to 9 **do**
9: $\mathbf{q}_{t+1}(i) = \dot{\mathbf{q}}_t(i) \times \text{step} + \mathbf{q}_t(i)$
10: Save $\mathbf{q}_{t+1}$ and iterate

In the mentioned algorithm, at lines 8-10, a first-order approximation is applied to compute all the general coordinates at each iteration.

Executing the above algorithm with the initial position q_0 and $\theta_{\text{comp}} = 0$, the numerical simulation in Maple showed a curve with some curvature, becoming more apparent with an increased number of iterations, see Fig. 3 and Fig. 4.

To address non-linearity, an attempt was made to compensate for the error only in the last six coordinates through constant, first and second order polynomials. When unsuccessful, a correction of the supposed values of serpentine gait was fed to all relative joint angles at each iteration step, and the remaining three coordinates were the only ones approximated:

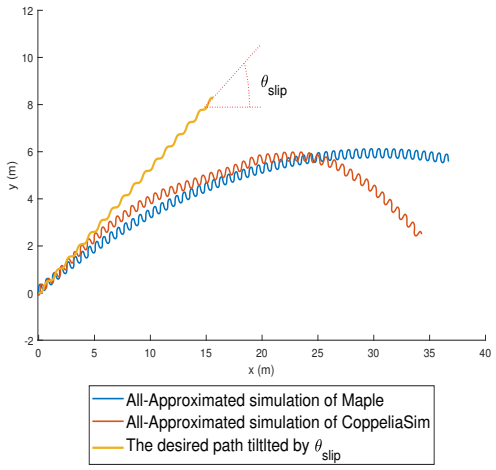

Figure 4: Curvature of the snake's head position path during a 8000-step simulation with seven links applying All-Approximated algorithm.

Algorithm 2 Partially Approximated Algorithm

1: Input: Initial position of the snake $\mathbf{q}_0$
2: Input: Time delay
3: **for** Every time step t **do**
4: Evaluate matrices $\mathbf{B}, \mathbf{C}$ using eq. (7)
5: Calculate pseudo-inverse matrix $\mathbf{B}^{-1}$
6: Evaluate matrix $\mathbf{G}$ using eq. (9)
7: Calculate current $\dot{\mathbf{q}}$ (derivative of the configuration space) using eq. (9)
8: **for** i from 1 to 3 **do**
9: $\mathbf{q}_{t+1}(i) = \dot{\mathbf{q}}_t(i) \times \text{step} + \mathbf{q}_t(i)$
10: **for** i from 4 to 9 **do**
11: $\mathbf{q}_{t+1}(i) = \phi_{t+1}(i)$ referring to eq. (13)
12: Save $\mathbf{q}_{t+1}$ and iterate

The corrected approach resulted in a purely linear trajectory. The difference between the two algorithms was found to be a sinusoidal function, as depicted in Fig. 5. This error, a second-order differential of eq. (13), and higher orders, cannot be calculated in our model due to its first-order approximation.

To address the question of whether the approximation error applies to all nine global coordinates on the manifold M (namely x, y, θ, and ϕ_1, ϕ_2, etc.), the accumulated error, with a maximum value of 0.03 in a 200-meter run, proves negligible on the x, y axis. However, for each joint, it results in a relative rotation of almost 1.7 degrees. Considering the composition of the six joints, the summed angle from head to tail is around 10 degrees, leading to a deviation from the linear path of the snake. Consequently, in light of these findings, we have chosen to adopt the Partially Approximated Algorithm for all subsequent simulations.

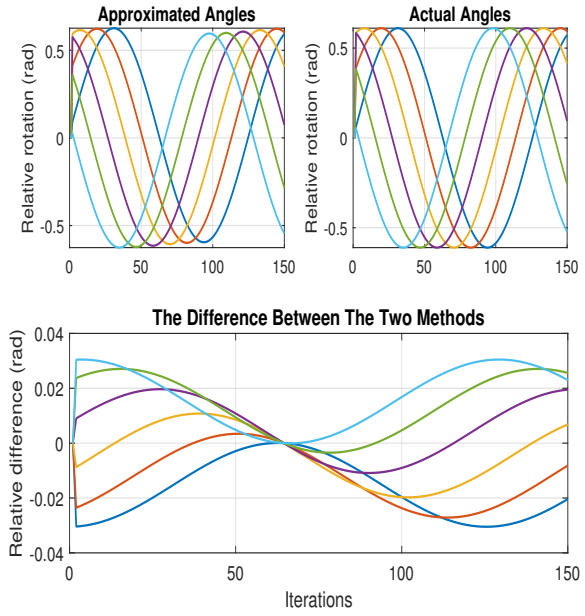

Figure 5: Propagation of the first-order approximation error.

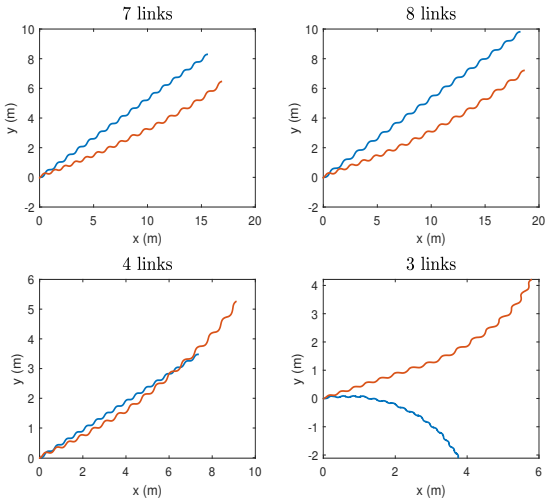

Figure 6: Comparison of the snake's head path at the first linear settings (simulations X01) with a different number of links. The blue curve represents the simulation in Maple, while the orange curve is from CoppeliaSim

Table 1: Naming Convention of the Ninety-six Scenarios

	a (Magnitude)	b (Shape)	c (Curvature)
X01			0
X02	0.5		0.125
X03		$-\frac{10\times\pi}{n}$	0.5
X04			0
X05	1		0.125
X06			0.5
X07			0
X08	0.5		0.125
X09		$-\frac{5\times\pi}{n}$	0.5
X10			0
X11	1		0.125
X12			0.5

9 Simulation Results

In order to validate our model and explore the impact of parameters a, b, and c on the snake robot locomotion, we conducted ninety-six different scenarios in CoppeliaSim and compared the results with those we have from Maple. For detailed information about the simulation environment, refer to Appendix A.These scenarios cover various combinations of amplitude (a), shape coefficient (b), and curvature (c). The labeling convention is structured as in Table 1, where X resembles the number of links and the relative position of the wheels, as shown in Table 2.

Table 2: Naming Convention of the Ninety-six Scenarios (continued)

	$L_1 = L_2 = 0.05$	$L_1 = 0.062, L_2 = 0.038$
7 links	$X = 1$	$X = 5$
8 links	$X = 2$	$X = 6$
4 links	$X = 3$	$X = 7$
3 links	$X = 4$	$X = 8$

The comparison of the eight settings reveals that the simulated gait in CoppeliaSim closely aligns with the numerical simulation conducted in Maple in general. This correlation becomes more evident with a greater number of snake links, indicating a more consistent behavior. One notable difference is that CoppeliaSim simulations consistently exhibit a smaller θ_{slip} compared to Maple across all scenarios, except for the three-links scheme. The three-links setting was a complete chaos in Maple and CoppeliaSim Fig. 6, 8, 9. Therefore we show the result here and acknowledge that our model is only suitable for snakes with four links or more.

Furthermore, it's worth noting that the relative position of the wheels exerts minimal influence on altering the snake's trajectory, as detailed in Table 3.

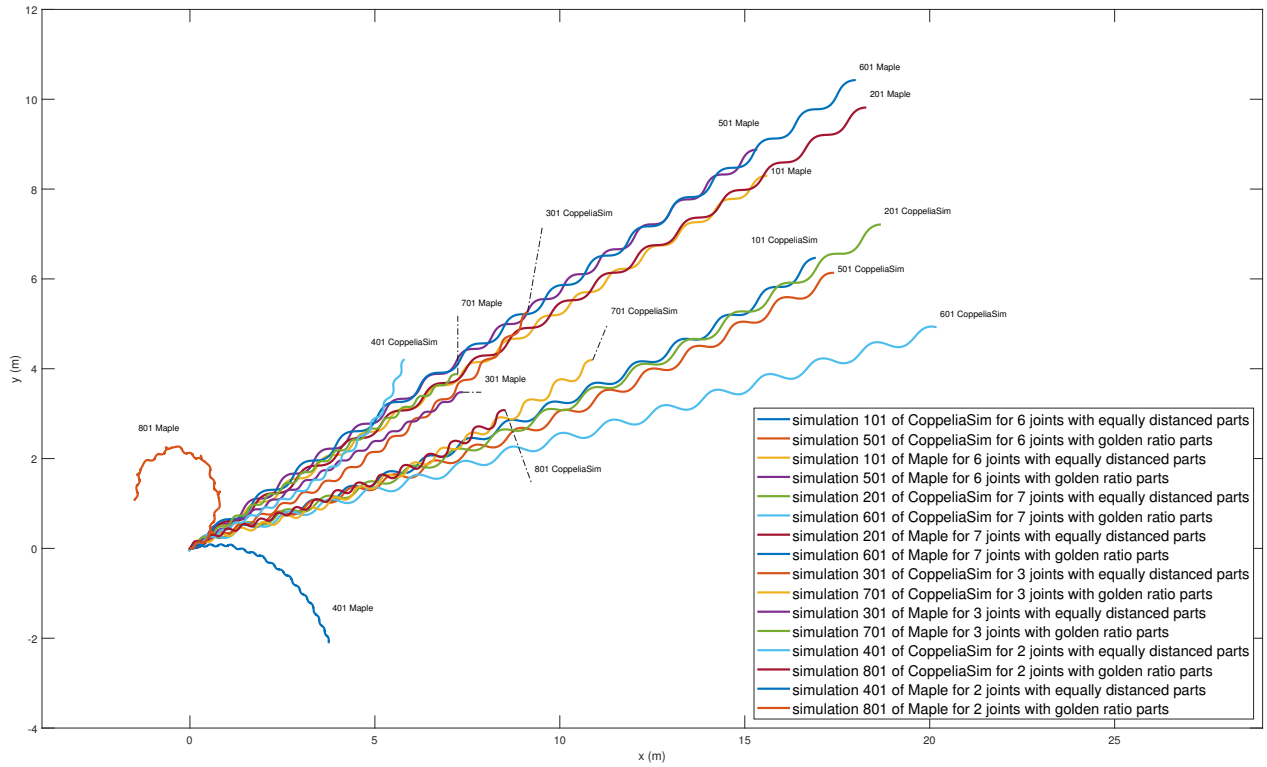

Figure 7: Comparison of the snake's head straight path with varying links and wheel positions under setting X01.

Linear Gait Comparison

We initiated the comparison by examining linear gaits (e.g., simualtions X01) with different numbers of links (Fig. 6). The consistent similarity in trajectory, especially with longer snakes, reinforces the accuracy of our model.

In Fig. 7, we examine the gait under identical parameters ($a = 0.5, b = \frac{-10\pi}{n}, c = 0$), identical time of travel, 2000 iteration steps, while varying the number of links and wheel positions. Notably, we observe slight variations in θ_{slip} based on the number of joints, except in the case of two joints, where the behavior appears chaotic.

Circular Gait Comparison

Similarly, we compared circular gaits, such as simulations X03 and X05, with varying numbers of links (Fig. 8, Fig. 9). Despite minor variations, the agreement between the two environments is evident.

Impact of Parameters on θ_{slip}

The primary factor influencing θ_{slip} is the magnitude parameter (a), as it is directly related to the winding angle μ. Larger a values result in a greater θ_{slip} but a shorter trajectory. In contrast, the shape coefficient (b) has minimal impact on θ_{slip}. In Maple, the trajectory nearly doubles when b is halved. CoppeliaSim has the same pattern with a different ratio Fig. 11.

We can also see that for the same values of a and b (and varying values of c), circular trajectories $c = \{0.125, 0.5\}$ are tangent to the linear ones $c = 0$, which suggests that c has no effect on θ_{slip}. In other words, the circles are rotated, with respect to the x-axis, by the exact same starting angle that the line exhibits. If c did affect the starting angle, we would expect to see a noticeable difference in the rotation, Fig. 10.

Distance Comparison Tables

Table 3 provides a quantitative analysis of the distance traveled along the curve and also measured from start to end in various simulations. The table reinforces the observation that, with a larger number of joints, the snake can achieve smoother and more extended motion.

In summary, the comparison of eight different gait settings reveals consistent behavior between CoppeliaSim and Maple simulations. While some discrepancies exist, particularly in θ_{slip}, these variations can be attributed to differences in simulation environments. Overall, the findings support the validity and robustness of our snake robot model.

			Length of the path				Distance between initial -final points			
			b=-10*pi/n		b=-5*pi/n		b=-10*pi/n		b=-5*pi/n	
Ratio	simulation SW	Nr. of links	a=0.5	a=1	a=0.5	a=1	a=0.5	a=1	a=0.5	a=1
L1=L2	Maple	7 links	18.67558	18.6427	38.40998	38.39869	17.67453	14.8009	36.2025	29.92019
		8 links	21.9258	21.91047	44.85918	44.85117	20.75458	17.41808	42.28393	34.9684
		4 links	8.633468	8.454522	18.94764	18.90575	8.157799	6.621743	17.8473	14.64467
		3 links	5.366687	4.806845	13.47052	13.36648	4.304117	2.228193	9.074914	2.759594
	CoppeliaSim	7 links	19.45722	17.5125	28.76702	26.69093	18.11472	13.29811	27.32677	21.62242
		8 links	21.48526	19.82498	28.50696	29.1087	20.02514	15.17946	27.35122	24.51912
		4 links	11.53532	8.576047	30.62347	19.968	10.53989	6.363339	25.08653	13.98837
		3 links	8.087829	4.434663	31.3276	19.18279	7.170279	3.204588	16.21488	8.014501
L1>L2	Maple	7 links	18.92688	19.54873	38.56868	38.82684	17.72259	14.83048	36.24324	29.93899
		8 links	22.17717	22.81799	45.01747	45.28084	20.80151	17.45199	42.3247	34.98979
		4 links	8.88114	9.329778	19.10811	19.31772	8.213851	6.621577	17.88777	14.64498
		3 links	7.18894	6.703535	13.01467	18.63773	1.844149	0.681359	10.5899	6.733784
	CoppeliaSim	7 links	20.04304	18.23823	28.9304	27.14333	18.46368	13.18776	27.39387	22.0158
		8 links	22.4551	20.86105	28.73391	31.37919	20.77788	15.26484	27.49043	26.00988
		4 links	13.07565	9.898847	28.78787	20.69712	11.6703	6.559985	22.16983	13.80086
		3 links	10.50022	5.883247	31.85803	20.37622	9.062806	3.471849	17.41794	8.438012

Table 3: The distance traveled along the entire curve vs The distance traveled from start to end

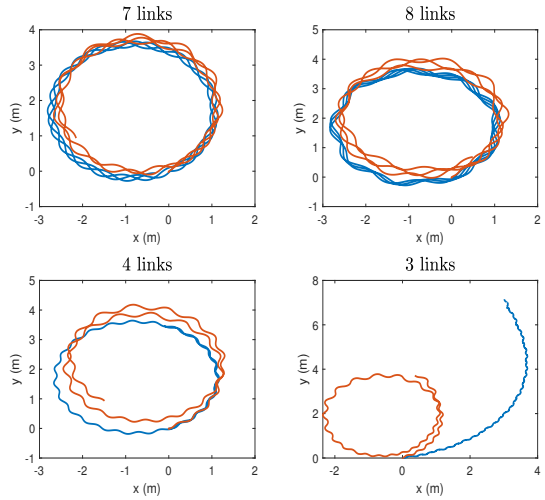

Figure 8: Comparison of the snake's head path at a circular setting (simulations X03, with curvature c=0.5) with a different number of links. The blue curve represents the simulation in Maple, while the orange curve is from CoppeliaSim

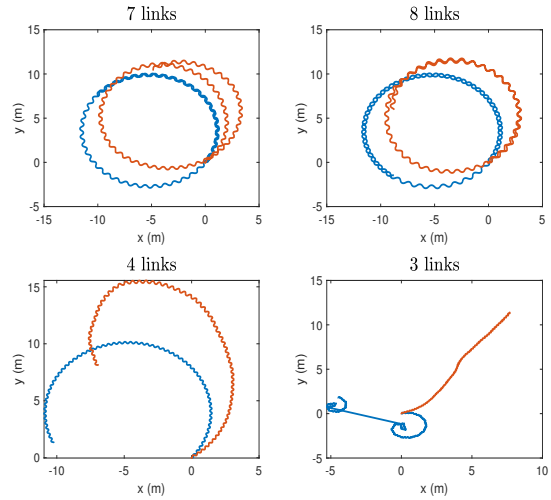

Figure 9: Comparison of the snake's head path at a circular setting (simulations X05, with curvature c=0.125) with a different number of links. The blue curve represents the simulation in Maple, while the orange curve is from CoppeliaSim

10 Conclusion

This paper focuses on the control and simulation of snake-like robots. We present a comprehensive analysis of serpentine gaits, investigating two distinct approaches based on sinusoidal curves and central pattern generators. Through meticulous mathematical modeling and numerical simulations, we gained valuable insights into the locomotion of snake-like robots.

Our findings suggest that effective serpentine gaits may take the form of different periodic signals, opening up new possibilities for robot control.

Taking inspiration from the works of Poincaré, Arnold, and Marsden, we embraced a geometric mechanics approach. By treating the snake robot as a multibody system with non-holonomic constraints, we presented an alternative perspective on locomotion, laying the foundation for innovative research in bioinspired robotics.

Despite shedding light on various aspects of snake-like robot control, we acknowledge the limitations inherent in our study. A notable constraint lies in our assumption of a perfect wheel grip on hard surfaces without any side-slips. Addressing this limitation, our research delved into the initial slip occurring during the

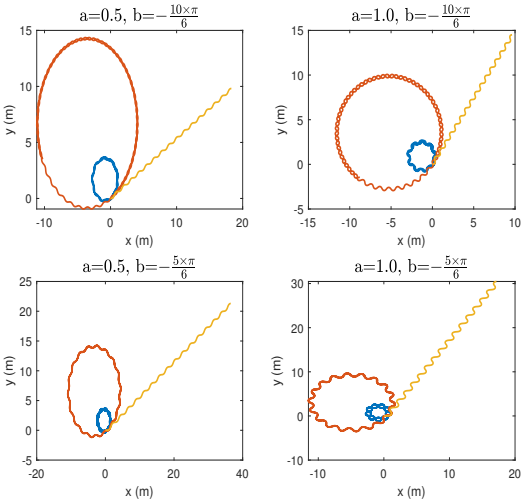

Figure 10: Comparison of different curvatures affect on θ_{slip} of an 8 links snake with equally distanced wheels in Maple.
Yellow colour depicts linear simulation c=0.
Orange colour depicts circular simulation with c= 0,125.
Blue colour depicts circular simulation with c= 0,5.

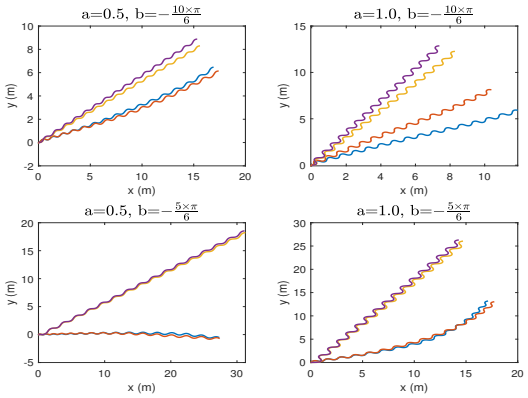

Figure 11: Comparison of wheel position effect on all linear simulations, c=0, for a 7 links snake.
Blue colour depicts simulation 104 of CoppeliaSim for seven links snake with equally distanced wheels.
Orange colour depicts simulation 504 of CoppeliaSim for a seven links snake with wheels aligned with the golden ratio.
Yellow colour depicts simulation 104 of Maple for seven links snake with equally distanced wheels.
Purple colour depicts simulation 504 of Maple for a seven links snake with wheels aligned with the golden ratio.

transition from a stacked configuration to a serpentine gait. Future endeavors may focus on formulating a precise slip angle formula or incorporating the effects of varying fractional factors into this formula.

Looking ahead, there are exciting opportunities for future work. Researchers can delve deeper into the application of our findings in fields such as disaster response, exploration, and even minimally invasive surgeries. The world of snake-like robots holds immense potential, and our research is but a stepping stone towards realising that potential.

In conclusion, our study has expanded the horizons of snake-like robot control and simulation. By exploring unconventional gaits and adopting geometric mechanics, we've paved the way for new possibilities in robotics. As we continue to push the boundaries of what snake-like robots can achieve, the future holds promise for remarkable advancements in technology and its applications.

Acknowledgement

This work was supported by the project IGA BUT No. FSI-S-23-8394 "Artificial intelligence methods in engineering tasks".

A CoppeliaSim Simulation Environment

To conduct our simulations, we employed the CoppeliaSim robotics simulation environment, utilising a detailed model of the snake robot. The model specifications include:

Snake Components

Wheels:

The mass of each wheel was set to 9.817 (g), with principal moments of inertia around the z-axis, perpendicular to the planar motion, being 3.125 $(cm)^2$, and principal moments around the x and y-axes being 1.583 $(cm)^2$. The wheels were configured with the following dynamics:

- Mass: 9.817 (g)
- Principal moments of inertia (z-axis): 3.125 (cm^2)
- Principal moments of inertia (x and y-axes): 1.583 (cm^2)
- Restitution: 0.5
- Linear damping: 0
- Angular damping: 0
- Sticky contact: false

Links:

Each link in the snake robot had a mass of 160 (g), with principal moments of inertia around the z-axis (perpendicular to the planar motion) being 9.667 (cm^2), and principal moments around the x-axis being 2.667 (cm^2). The links only rotate around the z-axis. The link dynamics were configured with the following properties:

- Mass: 160 (g)
- Principal moments of inertia (z-axis): 9.667 (cm^2)
- Principal moments of inertia (x-axis): 2.667 (cm^2)
- Principal moments of inertia (y-axis): 9.667 (cm^2)
- Sticky contact: false
- Restitution: 0
- Linear damping: 0
- Angular damping: 0
- Custom coil margin (enabled): false

Time Steps and Simulation Parameters

The simulation was executed with a time step of 10 (ms), while dynamics calculations utilised a time step of 5 (ms). The simulation passes per frame (ppf) were set to 1, maintaining real-time simulation. The dynamics engine employed in CoppeliaSim was Bullet 2.78, with specific properties configured as follows:

- Solver: Bullet 2.78 - Sequential Impulse (0)
- Iterations: 500
- Compute Inertias: false
- Internal Scaling: full, scaling factor: 10

References

[1] Bae,J., Kim, M., Song, B., Jin, M., and Yun, D. Snake robot with driving assistant mechanism. *Applied Sciences 10*, 21 (2020).

[2] Boyer,F., and Ali, S. Recursive inverse dynamics of mobile multibody systems with joints and wheels. *IEEE Transactions on Robotics 27*, 2 (2011), 215–228.

[3] Boyer,F., Ali, S., and Porez, M. Macro-continuous dynamics for hyperredundant robots: Application to kinematic locomotion bioinspired by elongated body animals. *IEEE Transactions on Robotics 28*, 2 (2012), 303–317.

[4] Boyer,F., and Porez, M. Multibody system dynamics for bio-inspired locomotion: from geometric structures to computational aspects. *Bioinspiration & Biomimetics 10*, 2 (mar 2015), 025007.

[5] Branyan,C., Hatton, R., and Menguc, Y. Snake-inspired kirigami skin for lateral undulation of a soft snake robot. *IEEE Robotics and Automation Letters PP* (01 2020), 1–1.

[6] Burdick, J., Radford, J., and Chirikjian, G. A 'sidewinding' locomotion gait for hyper-redundant robots. In *[1993] Proceedings IEEE International Conference on Robotics and Automation* (1993), pp. 101–106 vol.3.

[7] Chen,L., Wang, Y., Ma, S., and Li, B. Studies on lateral rolling locomotion of a snake robot. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004* (2004), vol. 5, pp. 5070–5074.

[8] Chen,Z., Iwasaki, T., and Zhu, L. Feedback control for natural oscillations of locomotion systems under continuous interactions with environment. *IEEE Transactions on Control Systems Technology 23*, 4 (2015), 1294–1306.

[9] Crespi,A., Badertscher, A., Guignard, A., and Ijspeert, A. J. Amphibot i: an amphibious snake-like robot. *Robotics and Autonomous Systems 50*, 4 (2005), 163–175. Biomimetic Robotics.

[10] Crespi,A., and Ijspeert, A. J. Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on Robotics 24*, 1 (2008), 75–87.

[11] Fukushima, H., Satomura, S., Kawai, T., Tanaka, M., Kamegawa, T., and Matsuno, F. Modeling and control of a snakelike robot using the screw-drive mechanism. *IEEE Transactions on Robotics - TRob 28* (06 2012), 541–554.

[12] Gans, C. Terrestrial locomotion without limbs. *American Zoologist 2*, 2 (1962), 167–182.

[13] Gray,J. The mechanism of locomotion in snakes. *The Journal of experimental biology 23, no. 2* (1946), 101–120.

[14] Guo,X., Ma, S., Li, B., and Fang, Y. A novel serpentine gait generation method for snakelike robots based on geometry mechanics. *IEEE/ASME Transactions on Mechatronics 23*, 3 (2018), 1249–1258.

[15] Hall,F. J. Generalized inverses of a bordered matrix of operators. *SIAM Journal on Applied Mathematics 29*, 1 (1975), 152–163.

[16] Hirose,S. *Biologically Inspired Robots (snakelike locomotor and manipulator)*. Oxford University Press, London, U.K., 1993.

[17] Hirose, S., and Yamada, H. Snake-like robots [tutorial]. *Robotics & Automation Magazine, IEEE 16* (04 2009), 88 – 98.

[18] Hrdina,J., Matousek, R., Navrat, A., and Vasik, P. Fisheye correction by cga non-linear transformation. *Mathematical Methods in the Applied Sciences 41*, 11 (2018), 4106–4116.

[19] Hu,D.,Nirody, J., Scott, T., and Shelley, M. The mechanics of slithering locomotion. *Proceedings of the National Academy of Sciences of the United States of America 106* (07 2009), 10081–5.

[20] H ˚ulka,T., Matou˘sek, R., Dobrovsk´y, L., Dosoudilov´a, M., and Nolle, L. Optimization of snake-like robot locomotion using ga: Serpenoid design. *MENDEL 26*, 1 (Aug. 2020), 1–6.

[21] Ishikawa,M.Trident snake robot: Locomotion analysis and control. *IFAC Proceedings Volumes 37*, 13 (2004), 895–900. 6th IFAC Symposium on Nonlinear Control Systems 2004 (NOLCOS 2004), Stuttgart, Germany, 1-3 September, 2004.

[22] Li,W.,Shen, M., Gao, A., Yang, G.-Z., and Lo, B. Towards a snake-like flexible robot for endoscopic submucosal dissection. *IEEE Transactions on Medical Robotics and Bionics 3*, 1 (2021), 257–260.

[23] Ma,S.,Ohmameuda, Y., and Inoue, K. Dynamic analysis of 3-dimensional snake robots. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)* (2004), vol. 1, pp. 767–772 vol.1.

[24] Manzoor,S.,Khan, U., and Ullah, I. Serpentine and rectilinear motion generation in snake robot using central pattern generator with gait transition. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering 44* (12 2019), 1–11.

[25] Mohammed, I. H., Gallardo, N., Benavidez, P., Jamshidi, M., and Champion, B. Design and control architecture of a 3d printed modular snake robot. In *2016 World Automation Congress (WAC)* (2016), pp. 1–6.

[26] Niu,G.,Zheng, Z., Gao, Q., Wang, W., and Wang, L. A novel design of aircraft fuel tank inspection robot. *TELKOMNIKA Indonesian Journal of Electrical Engineering 11* (07 2013).

[27] Onal,C.D., and Rus, D. Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot. *Bioinspiration & Biomimetics 8*, 2 (mar 2013), 026003.

[28] Orekhov, A. L., Abah, C., and Simaan, N. Snake-like robots for minimally invasive, single port, and intraluminal surgeries. *CoRR abs/1906.04852* (2019).

[29] Pettersen,K.,Liljeback, P., Stavdahl, Ø., and Gravdahl, J. Snake robots from biology to nonlinear control. *IFAC Proceedings Volumes 46*, 23 (2013), 110–115. 9th IFAC Symposium on Nonlinear Control Systems.

[30] Pettersen, K. Y. Snake robots. *Annual Reviews in Control 44* (2017), 19–44.

[31] Prautsch,P.,Mita, T., and Iwasaki, T. Analysis and control of a gait of snake robot. *IEEJ Transactions on Industry Applications 120* (01 2000), 372–381.

[32] Shi,P.,Shao, Q., and Liang, D. Design and improved serpentine curve locomotion control of a planar modular snake robot. In *2016 IEEE International Conference on Information and Automation (ICIA)* (2016), pp. 1398–1402.

[33] Tanev,I., Ray, T., and Buller, A. Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot. *IEEE Transactions on Robotics 21*, 4 (2005), 632–645.

[34] Tang,L., Zhu, L.-M., Zhu, X., and Gu, G. A serpentine curve based motion planning method for cable-driven snake robots. In *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)* (2018), pp. 1–6.

[35] Wang, X., Zhang, Q., Shen, D., and Chen, J. A novel rescue robot: Hybrid soft and rigid structures for narrow space searching. pp. 2207–2213.

[36] Wright, C., Buchan, A., Brown, B., Geist, J., Schwerin, M., Rollinson, D., Tesch, M., and Choset, H. Design and architecture of the unified modular snake robot. In *2012 IEEE International Conference on Robotics and Automation* (2012), pp. 4347–4354.

[37] Yu,X.,Nguyen, B., and Friesen, W. O. Sensory feedback can coordinate the swimming activity of the leech. *Journal of Neuroscience 19*, 11 (1999), 4634–4643.

[38] Yu,X.,Wang, X., Meng, D., and Liang, B. Differential kinematics for a tendon-driven snake-like robot. *2018 Chinese Control And Decision Conference (CCDC)* (2018), 5768–5773.

[39] Zhu,L.,Chen, Z., and Iwasaki, T. Synthesis of controllers for exact entrainment to natural oscillation. In *2010 8th World Congress on Intelligent Control and Automation* (2010), pp. 709–713.