Decentralized Control Strategy for Path-Following in Snake Robots

Arpit Dwivedi[1], Hemanta Hazarika[2,3], Dwaipayan Mukherjee[3], and Debasattam Pal[3]

Abstract— **Path following and control research done so far for the modular snake robots is known to have the foundational assumption that friction acts exactly at the link's center of mass. However, inspired by the biological snakes, we observe that friction acts as an uneven distribution over the snake's body. The paper contributes towards a general modeling of the snake robot that incorporates an eccentric point where the friction force acts on each module. Moreover, we have proposed a decentralized path following control algorithm for our general model. The proposed algorithm exploits local link-level autonomy and helps the robot maneuver smoothly for any general path curve. We apply our algorithm to four different types of path curves and find promising results in the path following.**

I. Introduction

Biological creatures have long been a great source of inspiration for robotics. One of which has been a snake for the last two decades. Snakes possess a unique ability to maneuver into confined and compact spaces like pipes, cracks, craters, and collapsed building sites [1]. This capability is deeply connected to their remarkable locomotion techniques. Through serpentine, concertina, rectilinear (caterpillar-like), and sidewinding movements, these limbless creatures skillfully exploit their environment, leveraging their flexible body structure to navigate through even the most challenging terrains [2]. Therefore, fabricated snake robots can have invaluable applications like navigating inside pipes that are beyond human reach, extraterrestrial exploration, and search and rescue operations.

The actual physics behind snake locomotion has been challenging to decipher. Early attempts to explain it suggested that snakes generate forward thrust by pushing against obstacles in their path [3]. However, smooth movements of snakes have been observed in environments like sand and slippery roads, where such obstacles are minimal or absent. Recent studies have revealed two key factors behind this movement. First, the snake redistributes its weight by selectively lifting parts of its body. Second, it utilizes frictional anisotropy of the contact surface to aid in locomotion [4]. The anisotropy arises from the unique arrangement of a snake's body scales [5]. These scales are structured in a way that reduces friction for longitudinal (along the body) movement while increasing friction during lateral (perpendicular to body) movement. Numerous studies have explored nonlinear systems analysis techniques to investigate the fundamental dynamics of snake robots, providing insights into their underlying properties [5]–[8].

The dynamics of planar snake robots have been previously formulated using fundamental principles. There are several reported works on wheel-based snake robots [7], [9], [10]. In [11], the authors present a model of a planar snake robot that operates without the use of wheels. The mathematical modeling in this paper is developed using the same approach as in [8]. The feedback linearized model in [8] assumes that the ground friction force on each link acts solely at the center of mass (CM) of the link. However, the dynamic model that has been structured in this paper accounts for the eccentric point where friction forces act on each module. This introduces an additional term in the final model equation, resulting in an offset that distinguishes it from conventional models discussed in [8].

During locomotion, biological snakes predominantly employ a movement strategy known as *lateral undulation*, wherein traveling waves propagate from head to tail along the body. Hirose [12], through empirical studies, proposed the *serpenoid curve* as a mathematical model that effectively characterizes the body shape of a snake undergoing such motion. As shown in [3], a planar snake robot can replicate this undulatory gait by actuating its joints according to the reference trajectory: $\phi_{i,\text{ref}} = A\sin(\omega t + (i-1)\delta) + \phi_0, \quad i \in \{1, \dots, N-1\}$.

In this parameterization, A and ω denote the amplitude and angular frequency of the sinusoidal input applied to each joint, respectively, while δ specifies the inter-joint phase difference that governs the spatial wavelength of the traveling wave. The term ϕ_0 introduces a uniform angular offset across all joints, effectively modulating the curvature bias of the robot's body. Notably, ϕ_0 serves as a critical control parameter for steering and directional modulation, as elaborated in Chapter 4 of [8].

Prior research suggests that determining ϕ_0 is a low-level control function, typically executed at the motor level. Since a snake robot comprises a series of interconnected links, centrally computing ϕ_0 can increase computational complexity and result in potential data loss during communication from the head to the tail. Consequently, this may lead to higher control effort at the low-level control stage. Considering that the inner control loop operates at high speed, ensuring precise computation with minimal control effort becomes crucial.

In this context, a decentralized control approach, which utilizes localized information for control, is particularly well-

[1] Department of Aeronautics & Astronautics, Stanford University, Stanford, CA 94305, {dwivedi7@stanford.edu}

[2] Department of Electrical and Instrumentation Engineering, Assam Engineering College {hh.ins@aec.ac.in}

[3] Department of Electrical Engineering, IIT Bombay-400076 {dm@ee.iitb.ac.in, debasattam@ee.iitb.ac.in}

The work was supported in part by a grant of the Indian Space Research Organization (ISRO) bearing code RD/0122-ISROC00-006.

suited. To our knowledge, existing literature has yet to explore decentralized control strategies for this low-level task. In this paper, we investigate the possibility of designing ϕ_0 in a decentralized manner to facilitate effective tracking of the snake's center of mass with respect to (i) a specified target point, (ii) discrete waypoints, and (iii) an arbitrary continuous path.

Our contributions are threefold: (1) we develop a generalized, partially feedback-linearized dynamic model of the robot that accommodates both actuated and unactuated dynamics; (2) we introduce a novel decentralized path-following control strategy that enables the robot to accurately track arbitrary, potentially nonlinear—reference trajectories. While the controller in [3] ensures convergence to a straight-line path using a Poincaré map-based stability analysis, our approach significantly extends this tracking capability to general nonlinear paths, representing a key advancement in decentralized snake robot control.

II. Notation and Preliminaries

We introduce the following vectors and matrices, which will be used throughout the analysis:

$$\boldsymbol{A} = \begin{bmatrix} (1-\alpha)l & \alpha l & \cdots \\ \vdots & \ddots & \ddots \\ \cdots & (1-\alpha)l & \alpha l \end{bmatrix} \in \mathbb{R}^{(n-1)\times n};$$

$$\boldsymbol{D} = \begin{bmatrix} -1 & 1 & \cdots \\ \vdots & \ddots & \ddots \\ \cdots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n};$$

$$\mathcal{S}_\theta = \operatorname{diag}(\sin\theta) \in \mathbb{R}^{n\times n}; \mathcal{C}_\theta = \operatorname{diag}(\cos\theta) \in \mathbb{R}^{n\times n};$$

$$\theta = \begin{bmatrix} \theta_1, & \cdots, & \theta_N \end{bmatrix}^T \in \mathbb{R}^n; e = \begin{bmatrix} 1, & \cdots, & 1 \end{bmatrix}^T \in \mathbb{R}^n;$$

$$\sin\theta = \begin{bmatrix} \sin\theta_1, & \cdots, & \sin\theta_N \end{bmatrix}^T \& \cos\theta = \begin{bmatrix} \cos\theta_1, & \cdots, & \cos\theta_N \end{bmatrix}^T \in \mathbb{R}^n$$

Table I & II compiles all the mathematical symbols used in the analysis.

TABLE I: Defined Symbols and their Descriptions

Symbol	Meaning	Associated Vector
N	Number of links in the system.	
l	Link length.	
m	Link mass.	
J	Rotational inertia of a link.	
θ_i	Orientation of link i relative to x-axis.	$\theta \in \mathbb{R}^n$
ϕ_i	Relative angle at joint i.	$\phi \in \mathbb{R}^{n-1}$
(x_i, y_i)	Center of mass position of link i in world frame.	$x, y \in \mathbb{R}^n$
(p_x, p_y)	Center of mass of the full robot in global frame.	$p \in \mathbb{R}^2$
u_i	Torque on link i due to link $i+1$.	$u \in \mathbb{R}^{n-1}$
u_{i-1}	Torque on link i due to link $i-1$.	$u \in \mathbb{R}^{n-1}$
$f_{R,x,i}$	Friction on link i in x-direction.	$f_{R,x} \in \mathbb{R}^n$
$f_{R,y,i}$	Friction on link i in y-direction.	$f_{R,y} \in \mathbb{R}^n$
$f_{R,i}$	Total friction acting on link i.	$f_{R,i} \in \mathbb{R}$
$h_{x,i}$	x-force from link $i+1$ at joint.	$h_x \in \mathbb{R}^{n-1}$
$h_{y,i}$	y-force from link $i+1$ at joint.	$h_y \in \mathbb{R}^{n-1}$
$h_{x,i-1}$	x-force from link $i-1$ at joint.	$h_x \in \mathbb{R}^{n-1}$
$h_{y,i-1}$	y-force from link $i-1$ at joint.	$h_y \in \mathbb{R}^{n-1}$

TABLE II: Defined Parameters

Symbol	Description
α	Fraction for position of CM
β	Fraction for the position of the point of application of friction force
k_p	Proportional gain
k_d	Derivative gain
k_θ	Proportional gain for turning control

III. Snake Robot Dynamics

The planar modular snake robot is modeled mathematically in this section. The architecture of development is similar to the model discussed in [8] except for the point of application of friction force. The schematic diagram for a snake robot for kinematic and dynamic analysis is illustrated in Fig. 1 and Fig. 2.

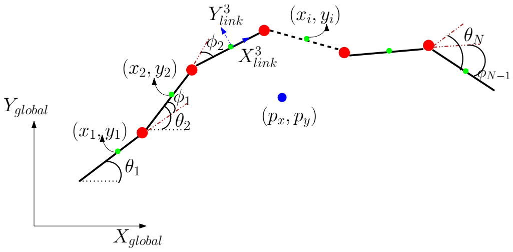

Fig. 1: Snake robot geometry and kinematics

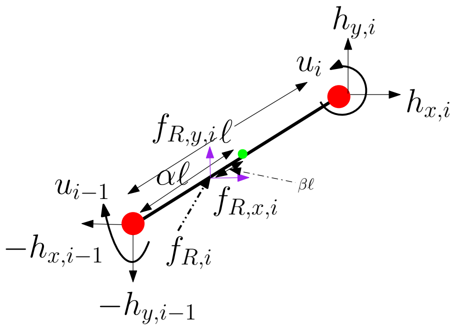

Fig. 2: Link-wise forces and torques in the snake robot

A. Kinematics

Two holonomic constraints must be satisfied at the connection between link i and link $i+1$ at joint $i \in \{1, ..., N-1\}$

$$x_{i+1} - x_i = (1-\alpha)l\cos\theta_{i+1} + \alpha l\cos\theta_i \tag{1}$$

$$y_{i+1} - y_i = (1-\alpha)l\sin\theta_{i+1} + \alpha l\sin\theta_i \tag{2}$$

For all the links it can be written as:

$$\boldsymbol{D}X + l\boldsymbol{A}\cos\theta = 0 \tag{3}$$

$$\boldsymbol{D}Y + l\boldsymbol{A}\sin\theta = 0 \tag{4}$$

Define $\boldsymbol{K} = A^T(DD^T)^{-1}D \in \mathbb{R}^{N\times N}$, and now on differentiating (3) and (4) with respect to time once and twice yields the following set of equations

$$\dot{\mathbf{X}} = l\mathbf{K}^T\mathcal{S}_\theta\dot{\theta} + e\dot{p_x} \tag{5}$$

$$\dot{\mathbf{Y}} = -l\mathbf{K}^T\mathcal{C}_\theta\dot{\theta} + e\dot{p_y} \tag{6}$$

$$D\ddot{X} = A\mathcal{C}_\theta \underbrace{\text{diag}(\dot{\theta})\dot{\theta}}_{\dot{\theta}^2} + A\mathcal{S}_\theta\ddot{\theta} \tag{7}$$

$$D\ddot{Y} = A\mathcal{S}_\theta \underbrace{\text{diag}(\dot{\theta})\dot{\theta}}_{\theta^2} - A\mathcal{C}_\theta\ddot{\theta} \tag{8}$$

B. Dynamic Analysis

The equations of motion are now expressed with respect to the angular accelerations of the links, $\ddot{\theta}$, and the center-of-mass accelerations, $\ddot{p}$. The corresponding force balance for the i^{th} link is given by

$$m\ddot{x}_i - f_{R,x,i} - h_{x,i} + h_{x,i-1} = 0 \tag{9}$$

$$m\ddot{y}_i - f_{R,y,i} - h_{y,i} + h_{y,i-1} = 0 \tag{10}$$

The force balance for the entire set of links can be represented compactly in matrix form as:

$$m\ddot{X} - f_{R,x} - D^T h_x = 0 \tag{11}$$

$$m\ddot{Y} - f_{R,y} - D^T h_y = 0 \tag{12}$$

The torque equilibrium condition for link i can be written as:

$$\begin{aligned} J\ddot{\theta}_i = {} & u_i - u_{i-1} - (h_{x,i} + h_{x,i-1})\alpha l \sin\theta_i \\ & + (h_{y,i} + h_{y,i-1})(1-\alpha) l \cos\theta_i \\ & + \beta l[\sin(\theta) f_{R,i,x} - \cos(\theta) f_{R,i,y}] \end{aligned} \tag{13}$$

The torque balance equation for all the links

$$\begin{aligned} \mathbf{J}\ddot{\theta} = {} & \mathbf{D}^T\mathbf{u} - l\mathcal{S}_\theta\mathbf{A}^T\mathbf{h}_x + l\mathcal{C}_\theta\mathbf{A}^T\mathbf{h}_y \\ & + \beta l[\mathcal{S}_\theta f_{R,x} - \mathcal{C}_\theta f_{R,y}] \end{aligned} \tag{14}$$

Substituting the joint constraint force yields the final model of the snake robot as:

$$\begin{aligned} \boldsymbol{D}^T\boldsymbol{u} = {} & \boldsymbol{M}_\theta\ddot{\boldsymbol{\theta}} + \boldsymbol{W} diag(\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} \\ & - l\beta\mathcal{S}_\theta(I_N + \frac{1}{\beta}\boldsymbol{K})\boldsymbol{f}_{R,x} + l\mathcal{C}_\theta(I_N + \frac{1}{\beta}\boldsymbol{K})\boldsymbol{f}_{R,y} \end{aligned} \tag{15}$$

$$Nm\ddot{\boldsymbol{p}} = Nm\begin{bmatrix}\ddot{p_x} \\ \ddot{p_y}\end{bmatrix} = \begin{bmatrix}\boldsymbol{e}^T\boldsymbol{f}_{R,x} \\ \boldsymbol{e}^T\boldsymbol{f}_{R,y}\end{bmatrix} \tag{16}$$

where f_R can be the Coulomb friction force or the viscous friction force and,

$$\begin{aligned} \boldsymbol{M}_\theta &= \boldsymbol{J}IN + ml^2\mathcal{S}_\theta\boldsymbol{V}\mathcal{S}_\theta + ml^2\mathcal{C}_\theta\boldsymbol{V}\mathcal{C}_\theta \\ \boldsymbol{W} &= ml^2\mathcal{S}_\theta\boldsymbol{V}\mathcal{C}_\theta - ml^2\mathcal{C}_\theta\boldsymbol{V}\mathcal{S}_\theta \\ \boldsymbol{V} &= A^T(DD^T)^{-1}A \end{aligned}$$

To express the snake robot's dynamics in a compact form, we define the state vector as $x = [\theta^T, p^T, \dot{\theta}^T, \dot{p}^T]^T \in \mathbb{R}^{2n+4}$. Using this representation, the system dynamics can be written in state-space form as:

$$\frac{d}{dt}x = \begin{bmatrix}\dot{\theta} \\ \dot{p} \\ \ddot{\theta} \\ \ddot{p}\end{bmatrix} = F(x, u) \tag{17}$$

IV. Main Results

We leverage partial feedback linearization to reformulate the snake robot's dynamics into a more tractable structure, as discussed in [13]. This approach specifically linearizes the dynamics associated with the system's actuated degrees of freedom. By adopting this method, we effectively decouple the dynamics of the underactuated states from the actuated states of the robot. To this end, we redefine the configuration variables by expressing the system in terms of relative joint angles rather than absolute link angles.

A. Separation of Dynamics

The center of mass (CM) acceleration of the snake robot, denoted by $\ddot{p}$, belongs to the unactuated portion of the system, as it is not directly controlled by the input u. In contrast, the angular accelerations of the links, $\ddot{\theta}$, include both actuated and unactuated components. Specifically, although there are N angular degrees of freedom ($\theta \in \mathbb{R}^N$), only $N-1$ are actuated due to the limited number of control inputs ($u \in \mathbb{R}^{N-1}$). This mismatch creates a natural partitioning of the dynamics into actuated and unactuated subsystems, enabling more structured control design.

To formalize this, we introduce the transformed configuration vector:

$$q_\phi = \begin{bmatrix}\bar{\phi} \\ p\end{bmatrix} \in \mathbb{R}^{N+2},$$

where

$$\bar{\boldsymbol{\phi}} = [\phi_1, \ldots, \phi_{N-1}, \theta_N]^T \in \mathbb{R}^N.$$

Following is the introduced coordinate transformation to transition from absolute link angles to relative joint angles, which simplifies the analysis of the snake robot dynamic:

$$\boldsymbol{\theta} = \boldsymbol{\mathcal{Z}}\bar{\boldsymbol{\phi}}, \text{where } \boldsymbol{\mathcal{Z}} = \begin{bmatrix}1 & 1 & \cdots & 1 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1\end{bmatrix} \in \mathbb{R}^{N\times N}$$

Consequently, (15) & (16) transforms into:

$$\begin{aligned} & \boldsymbol{M}_\theta\boldsymbol{\mathcal{Z}}\ddot{\bar{\boldsymbol{\phi}}} + \boldsymbol{W}\text{diag}(\boldsymbol{\mathcal{Z}}\dot{\bar{\phi}})\boldsymbol{\mathcal{Z}}\dot{\bar{\phi}} - \ell\beta\mathcal{S}_\theta(I_N + \frac{1}{\beta}K)f_{R,x} \\ & \qquad + \ell\beta\mathcal{C}_\theta(I_N + \frac{1}{\beta}K)f_{R,y} = D^T u \end{aligned} \tag{18}$$

$$Nm\ddot{\boldsymbol{p}} = \begin{bmatrix}\boldsymbol{e}^T\boldsymbol{f}_{R,x} \\ \boldsymbol{e}^T\boldsymbol{f}_{R,y}\end{bmatrix} \tag{19}$$

The terms $(I_N + \frac{1}{\beta}K)$ differ fundamentally from the conventional model found in the literature, such as in [8]. The first $N-1$ equations from (18) and (19) collectively capture the dynamics of the relative joint angles, corresponding to the actuated degrees of freedom of the snake robot. In contrast, the final three equations govern the evolution of the robot's absolute orientation and position, which represent the unactuated degrees of freedom. This naturally motivates a partitioning of the system dynamics and associated matrices into actuated and unactuated components.

$$\boldsymbol{M}_{11}\ddot{q}_a + \boldsymbol{M}_{12}\ddot{q}_u + \boldsymbol{W}_1 + G_1 f_R = u \tag{20}$$

$$\boldsymbol{M}_{21}\ddot{q}_a + \boldsymbol{M}_{22}\ddot{q}_u + \boldsymbol{W}_2 + G_2 f_R = \mathbf{0}_{3\times 1} \tag{21}$$

where $q_a = [\phi_1, \ldots, \phi_{N-1}]^T \in \mathbb{R}^{N-1}$ represents the actuated degrees of freedom .The matrices $M_{11}, M_{12}, M_{21}, M_{22}$ are block components of a matrix $M \in \mathbb{R}^{(N+2)\times(N+2)}$, partitioned according to actuated and unactuated coordinates. Similarly, $\boldsymbol{W}_1, \boldsymbol{W}_2$ and G_1, G_2 are chunks of matrices $\boldsymbol{W} \in \mathbb{R}^{N+2}$ and $G \in \mathbb{R}^{(N+2)\times 2N}$.

B. Partial Feedback Linearization

Now equipped with the partitioned model described in equations (20) & (21), we are poised to streamline the model into a more simplified form through the method discussed above. This approach, as outlined in [3] for snake robots, involves introduces an input transformation to linearize the actuated dynamics described below in the equation. To this end, we have

$$\begin{aligned} u = {} & (M_{11} - M_{12}M_{22}^{-1}M_{21})\bar{u} + \boldsymbol{W}_1 + G_1 f_R \\ & - M_{12}M_{22}^{-1}(\boldsymbol{W}_2 + G_2 f_R) \end{aligned} \tag{22}$$

where $\bar{u}$ are new set of control inputs:

$$\bar{u} = \begin{bmatrix} \bar{u_1} & \ldots & \bar{u_{N-1}} \end{bmatrix} \in \mathbb{R}^{N-1} \tag{23}$$

The final concise expression of the robot's model is as follows:

$$\ddot{q_a} = \bar{u}, \tag{24}$$

$$\ddot{q_u} = \mathcal{A}(q_{\phi,\dot{\phi}}) + \mathcal{B}(q_a)\bar{u}, \tag{25}$$

where,

$$\mathcal{A}(q_{\phi,\dot{\phi}}) = -\bar{\boldsymbol{M}}_{\mathbf{22}}{}^{-1}(\bar{\boldsymbol{W}}_2 + \bar{\boldsymbol{G}}_2 \boldsymbol{f_R});\ \mathcal{B}(q_a) = -\bar{\boldsymbol{M}}_{\mathbf{22}}{}^{-1}\bar{\boldsymbol{M}}_{\mathbf{21}}$$

Now we have the control-affine system with $x_1 = q_a, x_2 = q_u, x_3 = \dot{q}_a, x_4 = \dot{q}_u,$ and $x = [x_1^T, x_2^T, x_3^T, x_4^T]^T \in \mathbb{R}^{2N+4}$.

$$\frac{dx}{dt} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} \dot{x_3} \\ \dot{x_4} \\ \bar{u} \\ \mathcal{A}(x) + \mathcal{B}(x_1)\bar{u} \end{bmatrix} = \boldsymbol{f}(x) + \sum_{j=1}^{N-1} \boldsymbol{g}_j(\boldsymbol{x}_1)\bar{u}_j \tag{26}$$

where,

$$\boldsymbol{f(x)} = \begin{bmatrix} \dot{x_3} \\ \dot{x_4} \\ \mathbf{0}_{N-1} \\ \mathcal{A}(x) \end{bmatrix}, \boldsymbol{g_j(x_1)} = \begin{bmatrix} \mathbf{0}_{N-1} \\ \mathbf{0}_3 \\ \boldsymbol{e_j} \\ \mathcal{B}_j(\boldsymbol{x_1}) \end{bmatrix}$$

For each $j \in \{1, \ldots, N-1\}$, let e_j denote the j-th standard basis vector in $\mathbb{R}^{N-1}$, corresponding to the j-th column of the identity matrix I_{N-1}. Similarly, $B_j(x_1)$ refers to the j-th column of the matrix $B(x_1)$. Within the framework of control-affine systems, the term $f(x)$ is typically identified as the *drift vector field*, while each $g_j(x_1)$ represents a distinct *control vector field.*

C. Control Law for Joint Angles

A planar snake robot with N discrete links can approximate a serpenoid curve by modulating its joint angles to follow sinusoidal patterns with a uniform phase offset between adjacent links. Realizing such a serpenoid configuration requires specifying $N-1$ joint angles. As demonstrated in [3], lateral undulatory motion can be generated by synchronizing the joint angles to follow the reference trajectory defined below:

$$\phi_{i,\text{ref}} = A\sin(\omega t + (i-1)\delta) + \phi_0, \tag{27}$$

where $i \in \{1, 2, \cdots, N-1\}$. In this expression, A and ω are the amplitude and angular frequency of the reference sinusoidal motion. The parameter δ is the phase shift between consecutive joints, and ϕ_0 denotes the joint offset. The joint offset's physical significance in turning the snake robot in a particular direction has been discussed in [8]. Now, we extend the theory by designing the joint offset in a decentralized manner to achieve the desired specifications while leveraging autonomy at the local level. We propose decentralized gait patterns as follows:

$$\phi_{i,\text{ref}} = A\sin(\omega t + (i-1)\delta) + \phi_{0i}, \tag{28}$$

where ϕ_{0i}, each joint offset, is calculated as per the proposed **Algorithm 1**. Furthermore, the computation of ϕ_{0i} can be visualized using the schematic diagram provided in Fig. 3. The control objective is to make $\boldsymbol{\phi} = [\phi_1, ..., \phi_{N-1}] \in \mathbb{R}^{N-1}$ closely follow the prescribed signal denoted by $\boldsymbol{\phi_{ref}} = [\phi_{1,ref}, ..., \phi_{N-1,ref}] \in \mathbb{R}^{N-1}$. We have utilized a *simple joint controller* discussed in [8]. The control law is as follows

$$\bar{u} = k_p(\phi_{ref} - \phi) - k_d\dot{\phi} \tag{29}$$

where controller gainsk_p and k_d are positive scalars. The control law in (29) remains stable for positive k_p and k_d, provided that ϕ_{ref} is small. Therefore, the proposed control law is stable.

Algorithm 1: Computation of ϕ_0

```
Input: Target Point $(x_{ref}, y_{ref}) \in (\mathbb{R} \times \mathbb{R})$
Output: $\boldsymbol{\phi_0} \in \mathbb{R}^{N-1}$
  Initialisation :
  $\boldsymbol{\phi_0} = \mathbf{0}_{N-1\times 1}$
  LOOP Process
  for $i \leftarrow 1$ to $N-1$ do
      $\mathcal{N}(i)$ : set of neighbouring links
      $X_{l_i} = \frac{1}{|\mathcal{N}(i)|+1}(x_i + \sum_{j\in\mathcal{N}(i)} x_j)$
      $Y_{l_i} = \frac{1}{|\mathcal{N}(i)|+1}(y_i + \sum_{j\in\mathcal{N}(i)} y_j)$
      $\theta_{l_i,\text{ref}} = \begin{cases} \tan^{-1}\left(\frac{y_{ref}-Y_{l_i}}{x_{ref}-X_{l_i}}\right), & \text{if } x_{ref} - X_{l_i} > 0 \\ \pi + \tan^{-1}\left(\frac{y_{ref}-Y_{l_i}}{x_{ref}-X_{l_i}}\right), & \text{if } x_{ref} - X_{l_i} < 0 \\ & \text{and } y_{ref} - Y_{l_i} \geq 0 \\ -\pi + \tan^{-1}\left(\frac{y_{ref}-Y_{l_i}}{x_{ref}-X_{l_i}}\right), & \text{if } x_{ref} - X_{l_i} < 0 \\ & \text{and } y_{ref} - Y_{l_i} < 0 \end{cases}$
      $\theta_{l_i,H}$ : Defining local heading angle for link i
      $\theta_{l_i,H} = \frac{1}{|\mathcal{N}(i)|+1}(\theta_i + \sum_{j\in\mathcal{N}(i)} \theta_j)$
```

```
    $\phi_{0,i} \leftarrow k_\theta(\theta_{l_i,H} - \theta_{l_i,ref})$
end for
return $\phi_0$
```

The reference trajectory in (27) drives the snake robot forward. To steer the robot toward the desired waypoint, we define the heading reference angle ($\theta_{l_i,ref}$) for each joint based on the Line-of-Sight (LOS), as illustrated in the schematic diagram in Fig. 3. In Fig. 3, the computation of ϕ_{0i} for any i^{th} link is explained. For any ϕ_{0i}, except for the head or tail links, we determine the local center of mass for an intermediate segment of the snake robot by considering the $(i-1)^{\text{th}}$, i^{th} and $(i+1)^{\text{th}}$ links denoted as (X_{li}, Y_{li}). Similarly, the angles $\theta_{l_i,H}$ and $\theta_{l_i,ref}$ are computed as locally shown in the figure 3. Finally, the joint offset angle ϕ_{0i} is computed in a decentralized manner as follows:

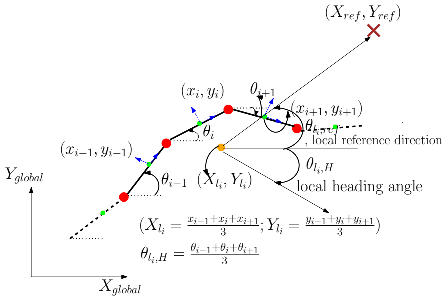

Fig. 3: Line of Sight guidance law

$$\phi_{0,i} = k_\theta(\theta_{l_i,H} - \theta_{l_i,ref}) \tag{30}$$

where $k_\theta > 0$ is a controller gain. Substituting (30) in (29), for joint angle i,

$$\begin{aligned} \bar{u}_i = k_p[&A\sin(\omega t + (i-1)\delta) \\ &+ k_\theta(\theta_{l_i,H} - \theta_{l_i,ref}) - \phi_i] - k_d\dot{\phi}_i; \end{aligned} \tag{31}$$

where $k_p > 0$, $k_\theta > 0$ and $k_d > 0$ are controller gains. Since the control law in (29) is stable, a similar argument can be used to demonstrate the stability of the decentralized control law given in (31). Thus, the decentralized control law in (31) is stable for all $i \in \{1, 2, \cdots, N-1\}$. The results obtained in this section are validated through simulations in the Section V.

V. Simulation Results

To evaluate the effectiveness of the proposed control strategy, we simulate its application on a snake robot composed of $N = 10$ uniform links. Each segment possesses a mass of $m = 0.1$ kg, length $\ell = 0.1$ m, and a rotational inertia $J = 0.0006$ kgm^2 about its center of mass. The interaction between the robot and its environment is modeled using direction-dependent friction, with coefficients $c_t = 1$ (tangential) and $c_n = 4$ (normal), capturing the anisotropy typical of undulatory locomotion. The reference joint trajectories are synthesized using a sinusoidal function characterized by an amplitude of $A = \frac{30\pi}{180}$ rad, a spatial frequency of $\omega = \frac{50\pi}{180}$ rad, and a phase shift of $\delta = \frac{40\pi}{180}$ rad.

The initial conditions for the simulation are defined as follows: The joint angles are set to $\theta(0) = \begin{bmatrix} 0 & 0 & 0.33 & 0.85 & 1.30 & 1.48 & 1.30 & 0.852 & 0.336 & 0 \end{bmatrix}^T$, with the joint angular velocities initialized to $\dot{\theta}(0) = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}^T$. The initial position of the robot is $p(0) = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}^T$, and the initial velocity is $\dot{p}(0) = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}^T$.

The control parameters used in the simulations are as follows: the proportional gain $k_p = 4$, the derivative gain $k_d = 1$, and the angular gain $k_\theta = 0.26$.

A. Particular target point

The snake robot is desired to reach the point $(18, 3.25)$ in the XY-plane. The simulation results show that the snake robot headed towards the target point. Fig. 4 depicts the movement of the center of mass of the robot. Fig. 5 depicts the heading angle of the robot.

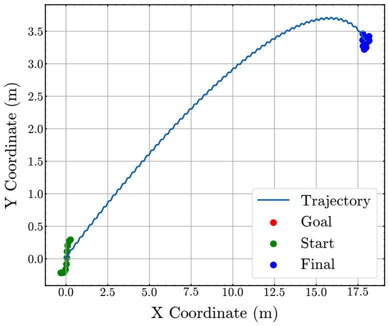

Fig. 4: Path followed by Center of mass

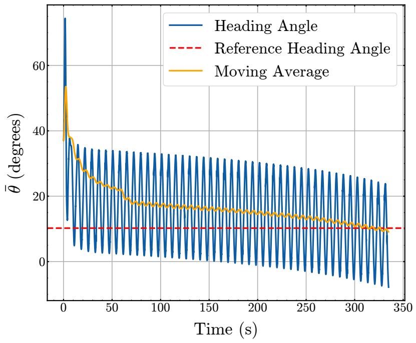

Fig. 5: Heading Angle

B. Set of discrete path point

The objective is to guide the snake robot through a sequence of target waypoints in the XY-plane: $(7.18, 3.23)$, $(10.92, 6.45)$, $(14.64, 4.87)$, and $(20.95, 5.99)$. Simulation results demonstrate that the proposed control strategy enables the robot to accurately follow the specified trajectory while ensuring that the system states remain uniformly and ultimately bounded. The evolution of the robot's center of mass and its trajectory are illustrated in Fig. 6, whereas Fig. 7 presents the temporal profile of the robot's heading angle.

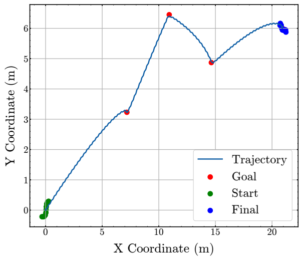

Fig. 6: Path followed by Center of mass

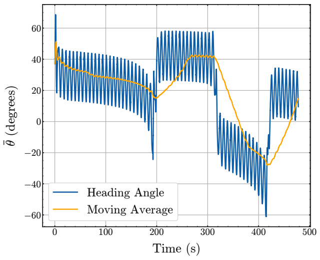

Fig. 7: Heading Angle

C. General path equation

The snake robot is tasked with tracking the trajectory $y = -4\sin(0.5x)$ in the XY-plane. Fig. 8 illustrates the robot along with its center of mass trajectory, while Fig. 9 presents the robot's heading angle. A full simulation of the experiment can be viewed in the accompanying video. In steady-state conditions, the root mean square (RMS) error of the path-following performance is 0.09759 m. The controller has been tested on various trajectory shapes, and the corresponding RMS errors are summarized in Table III.

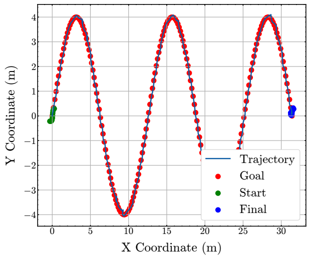

Fig. 8: Path followed by Center of mass

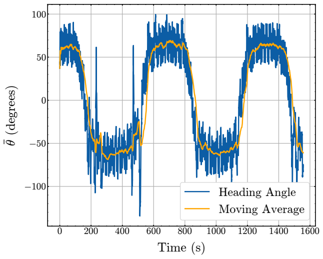

Fig. 9: Heading Angle

TABLE III: Numerical error for different path equations

Reference Path Equation	**Numerical error in simulation(m)**
$y = -0.0001x(x+1)(x-40)(x-28)$	0.05746
$y = -4sin(0.5x)$	0.09759
$y = 10 - 10e^{-x}$	0.39635
$y = \sqrt{(225 - (x-15)^2)}$	0.36333

VI. Conclusion

In this paper, we have derived a more general mathematical model for planar snake robots. The model has been partitioned and formulated into actuated and unactuated dynamics in terms of relative joint angles. Furthermore, an input transformation has been introduced to partially linearize the dynamics. Additionally, we have addressed the problem of designing joint offset in a decentralized manner to leverage local link-level autonomy, enabling snake robots to track arbitrary path equations.

We simulated the control algorithm across four distinct path equations, polynomial, sinusoidal, exponential, and circular, and achieved minimal RMS errors in path tracking. These results demonstrate precise and consistent path-following performance. The theoretical guarantees established in this paper are currently being implemented on a hardware model, and the outcomes will be reported in future work.

References

[1] J. Liu, Y. Tong, and J. Liu, "Review of snake robots in constrained environments," *Robotics and Autonomous Systems*, vol. 141, p. 103785, 2021.

[2] M. Nakajima and M. Tanaka, "Motion control of a snake robot on multiple inclined planes," *Advanced Robotics*, vol. 38, no. 11, pp. 784–800, 2024.

[3] P. Liljeback, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Controllability and stability analysis of planar snake robot locomotion," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1365–1380, 2010.

[4] J. Ha, "Robotic snake locomotion exploiting body compliance and uniform body tensions," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3875–3887, 2023.

[5] G. Seeja, A. S. A. Doss, and V. B. Hency, "A survey on snake robot locomotion," *IEEE Access*, vol. 10, pp. 112 100–112 116, 2022.

[6] Shugen, "Analysis of creeping locomotion of a snake-like robot," *Advanced Robotics*, vol. 15, no. 2, pp. 205–224, 2001.

[7] J. P. Ostrowski, *The mechanics and control of undulatory robotic locomotion*. California Institute of Technology, 1996.

[8] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control*. Springer, 2013.

[9] P. Prautsch and T. Mita, "Control and analysis of the gait of snake robots," in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*, vol. 1. IEEE, 1999, pp. 502–507.

[10] H. Date, M. Sampei, and S. Nakaura, "Control of a snake robot in consideration of constraint force," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01)(Cat. No. 01CH37204)*. IEEE, 2001, pp. 966–971.

[11] S. Ma and N. Tadokoro, "Analysis of creeping locomotion of a snake-like robot on a slope," *Autonomous Robots*, vol. 20, pp. 15–23, 2006.

[12] S. Hirose, "Biologically inspired robots," *Snake-Like Locomotors and Manipulators*, 1993.

[13] Y.-L. Gu and Y. Xu, "A normal form augmentation approach to adaptive control of space robot systems," *Dynamics and Control*, vol. 5, no. 3, pp. 275–294, 1995.