Modelling and control of obstacle-aided snake robot locomotion based on jam resolution

Pål Liljebäck, Kristin Y. Pettersen, and Øyvind Stavdahl

***Abstract*— A snake robot can traverse cluttered and irregular environments by using irregularities around its body as push-points to aid the propulsion. This characteristic feature of snake locomotion, denoted *obstacle-aided locomotion*, has received limited focus in previous literature. This paper presents a model of this phenomenon and a control strategy employing measured contact forces to maintain propulsion while simultaneously preventing the snake robot from being jammed between obstacles in its path. The simulation results validate the contact modelling approach and the effectiveness of the proposed control strategy.**

I. Introduction

Inspired by biological snake locomotion, snake robots carry the potential of meeting the growing need for robotic mobility in unknown and challenging environments. These mechanisms typically consist of serially connected modules capable of bending in one or more planes. The many degrees of freedom of snake robots make them difficult to control, but provides traversability in irregular environments that surpasses the mobility of the more conventional wheeled, tracked and legged forms of robotic mobility.

The unique feature of snake robot locomotion compared to other forms of robotic mobility is that irregularities on the ground are actually beneficial for the propulsion since they provide push-points for the snake robot. While *obstacle avoidance* is an important topic for wheeled, tracked and legged robots, the goal of snake locomotion is rather *obstacle exploitation*. The term *obstacle-aided locomotion* was introduced by Transeth *et al.* [1] and captures the essence of this concept.

The majority of previous snake robot research has focused on understanding and demonstrating snake locomotion on flat surfaces with preprogrammed motion patterns that resemble gaits displayed by biological snakes. Gray [2] conducted empirical and analytical studies of snake locomotion as far back as the middle of the last century. Hirose [3] studied biological snakes and developed mathematical relationships characterizing their motion, such as the *serpenoid curve*. Several mathematical models of the kinematics and dynamics of snake locomotion have been developed. Some models [4]–[6] assume that the links of the snake cannot move sideways (no-slip conditions achieved by e.g. mounting passive wheels along the snake body), while others assume isotropic or anisotropic friction conditions [1], [7]–[12]. The works in [1], [11], [12] present, to the authors' best knowledge, the only known models of snake robot dynamics that also include obstacle contact forces. These models are, however, pure simulation models and not suited for synthesis of model-based control strategies. The works in [11], [13], [14] present, to the authors' best knowledge, the only known control strategies related to obstacle-aided snake locomotion.

This paper provides two contributions. The first contribution is the development of a 2D model of snake robot dynamics that incorporates obstacle contact forces based on the introduction of nonholonomic constraints for contacted links. The model has been developed to facilitate synthesis of model-based control strategies for obstacle-aided locomotion. The second contribution is a control strategy for obstacle-aided locomotion aimed at resolving situations where the snake robot is jammed between obstacles. The concept of detecting and resolving snake robot jams has, to the authors' best knowledge, not been treated in previous literature, but is a genuine challenge often encountered during snake robot motion in cluttered environments. To our knowledge, this is the first published control strategy for a snake robot involving feedback and explicit use of measured contact forces. The paper presents simulation results that validate the contact modelling approach and the effectiveness of the proposed control strategy.

The paper is organized as follows. Section II presents a 2D model of snake robot kinematics and dynamics without obstacle contact forces. Section III extends the 2D model with an obstacle contact force model. Section IV presents the control strategy for the snake robot. Section V presents simulation results, and Section VI presents concluding remarks.

II. Snake robot model without obstacles

This section presents a 2D model of the kinematics and dynamics of a snake robot. The model is based on [7], but is presented in a somewhat different format. The model is extended with an obstacle contact model in Section III.

A. Notations and defined identities

The snake robot consists of n links of length $2l$ interconnected by $n-1$ joints. The mathematical identities defined in order to describe the kinematics and dynamics of the snake robot are described in Table I and illustrated in Fig. 1 and Fig. 2. All n links have the same length, mass and moment of inertia. The total mass of the snake robot is therefore nm. The mass of each link is uniformly distributed so that the link CM (center of mass) is located at its center point (at length l from the joint at each side).

Vectors are either expressed in the global coordinate system or in the local link coordinate system of link i. This is indicated by superscript *global* or *link,i*, respectively. If

Affiliation of Pål Liljebäck is shared between the Department of Engineering Cybernetics at the Norwegian University of Science and Technology, NO-7491 Trondheim, Norway, and SINTEF ICT, Dept. of Applied Cybernetics, N-7465 Trondheim, Norway. E-mail: Pal.Liljeback@sintef.no

K. Y. Pettersen and Øyvind Stavdahl are with the Department of Engineering Cybernetics at the Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. E-mail: {Kristin.Y.Pettersen, Oyvind.Stavdahl}@itk.ntnu.no

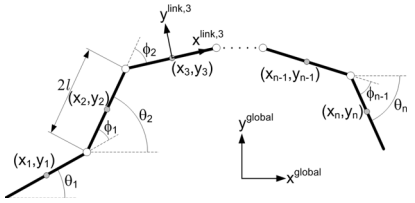

Fig. 1. Kinematic parameters for the snake robot.

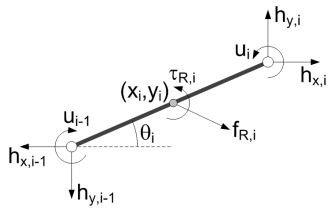

Fig. 2. Forces and torques acting on each link of the snake robot.

otherwise is not specified, a vector with no superscript is expressed in the global coordinate system.

The following vectors and matrices are used in the subsequent sections:

$$A := \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix}, D := \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}$$

where $A \in \mathbb{R}^{(n-1)\times n}$ and $D \in \mathbb{R}^{(n-1)\times n}$. Furthermore,

$e := \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^n$

$\sin\theta := \begin{bmatrix} \sin\theta_1 \,..\, \sin\theta_n \end{bmatrix}^T \in \mathbb{R}^n \quad \cos\theta := \begin{bmatrix} \cos\theta_1 \,..\, \cos\theta_n \end{bmatrix}^T \in \mathbb{R}^n$

$S_\theta := \text{diag}(\sin\theta_1, .., \sin\theta_n) \quad C_\theta := \text{diag}(\cos\theta_1, .., \cos\theta_n)$

B. Kinematics

The snake robot is capable of 2D motion in the horizontal plane and has a total of $n+2$ degrees of freedom. The absolute angle, θ_i, of link i is expressed with respect to the global x axis with counterclockwise positive direction. As seen in Fig. 1, the relative angle between link i and link $i+1$ is given by

$$\phi_i = \theta_i - \theta_{i+1} \tag{1}$$

The model of the snake robot will be derived using absolute angles since this simplifies the mathematical expressions. The local coordinate system of each link is fixed in the CM (center of mass) of the link with x (tangential) and y (normal) axis oriented such that they are oriented in the directions of the global x and y axis, respectively, when the link angle is zero. The rotation matrix from the global frame to the frame of link i is given by

$$R^{\text{global}}_{\text{link},i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \tag{2}$$

The position of the snake robot, p, is described through the coordinates of its CM (center of mass) and is given by

$$p := \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{nm}\sum_{i=1}^{n} m x_i \\ \frac{1}{nm}\sum_{i=1}^{n} m y_i \end{bmatrix} = \frac{1}{n}\begin{bmatrix} e^T x \\ e^T y \end{bmatrix} \tag{3}$$

Symbol	Description	Associated vector
n	Number of links.	
l	Half the length of a link.	
m	Mass of a link.	
J	Moment of inertia of a link. $J = \frac{1}{3}ml^2$.	
g	Gravitational acceleration constant.	
θ_i	Angle between link i and global x axis.	$\theta \in \mathbb{R}^n$
(x_i, y_i)	Global coordinates of CM of link i.	$x,\, y \in \mathbb{R}^n$
(p_x, p_y)	Global coordinates of CM of snake robot.	$p \in \mathbb{R}^2$
u_i	Actuator torque exerted on link i from link $i+1$.	$u \in \mathbb{R}^{n-1}$
u_{i-1}	Actuator torque exerted on link i from link $i-1$.	$u \in \mathbb{R}^{n-1}$
$f_{R,x,i}$	Friction force on link i in x direction.	$f_{R,x} \in \mathbb{R}^n$
$f_{R,y,i}$	Friction force on link i in y direction.	$f_{R,y} \in \mathbb{R}^n$
τ_i	Friction torque about CM of link i.	$\tau \in \mathbb{R}^n$
$h_{x,i}$	Joint constraint force in x direction on link i from link $i+1$.	$h_x \in \mathbb{R}^n$
$h_{y,i}$	Joint constraint force in y direction on link i from link $i+1$.	$h_y \in \mathbb{R}^n$
$h_{x,i-1}$	Joint constraint force in x direction on link i from link $i-1$.	$h_x \in \mathbb{R}^n$
$h_{y,i-1}$	Joint constraint force in y direction on link i from link $i-1$.	$h_y \in \mathbb{R}^n$

TABLE I
DEFINED MATHEMATICAL IDENTITIES.

The following derives an expression for the link positions as a function of the CM position of the snake robot. The links are constrained by the joints according to

$$\begin{aligned} Dx + lA\cos\theta &= 0 \\ Dy + lA\sin\theta &= 0 \end{aligned} \tag{4}$$

We may combine (3) and (4) into

$$Tx = \begin{bmatrix} -lA\cos\theta \\ p_x \end{bmatrix} \quad , \quad Ty = \begin{bmatrix} -lA\sin\theta \\ p_y \end{bmatrix} \tag{5}$$

where

$$T := \begin{bmatrix} D \\ \frac{1}{n}e^T \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{6}$$

We may now solve (5) for x and y to get

$$\begin{aligned} x &= T^{-1}\begin{bmatrix} -lA\cos\theta \\ p_x \end{bmatrix} = -lN^T\cos\theta + ep_x \\ y &= T^{-1}\begin{bmatrix} -lA\sin\theta \\ p_y \end{bmatrix} = -lN^T\sin\theta + ep_y \end{aligned} \tag{7}$$

where

$$N := A^T\left(DD^T\right)^{-1}D \in \mathbb{R}^{n\times n} \tag{8}$$

and DD^T is nonsingular and thereby invertible.

C. Differential kinematics

The linear velocities of the links are derived by differentiating (7). This gives

$$\begin{aligned} \dot{x} &= lN^T S_\theta \dot{\theta} + e\dot{p}_x \\ \dot{y} &= -lN^T C_\theta \dot{\theta} + e\dot{p}_y \end{aligned} \tag{9}$$

The above equations give the linear velocities of all the links. However, the obstacle contact model derived in Section III requires an expression for the velocity of a single link. This

expression may be found by investigating the structure of each row in (9). The derivation is not included here due to space restrictions, but it may be verified that the linear velocity of link i is given by

$$\begin{aligned}\dot{x}_i = \dot{p}_x - \textstyle\sum_{j=1}^{i-1} a_j \left(\sin\theta_j\dot{\theta}_j + \sin\theta_{j+1}\dot{\theta}_{j+1}\right) \\ - \textstyle\sum_{j=i}^{n-1} b_j \left(\sin\theta_j\dot{\theta}_j + \sin\theta_{j+1}\dot{\theta}_{j+1}\right)\end{aligned} \tag{10}$$

$$\begin{aligned}\dot{y}_i = \dot{p}_y + \textstyle\sum_{j=1}^{i-1} a_j \left(\cos\theta_j\dot{\theta}_j + \cos\theta_{j+1}\dot{\theta}_{j+1}\right) \\ + \textstyle\sum_{j=i}^{n-1} b_j \left(\cos\theta_j\dot{\theta}_j + \cos\theta_{j+1}\dot{\theta}_{j+1}\right)\end{aligned} \tag{11}$$

where the two coefficients, a_j and b_j, are given by

$$a_j := \tfrac{l}{n} j \quad , \quad b_j := -\tfrac{l(n-j)}{n} \tag{12}$$

D. Coulomb friction model

Each link is subjected to a friction force from the ground acting on the CM of the link and also a friction torque acting about the CM. A Coulomb friction model is employed in a form which allows for anisotropic friction on each link. This means that a link has two Coulomb friction coefficients, μ_t and μ_n, describing the friction force in the tangential (along link x axis) and normal (along link y axis) direction of the link, respectively. The friction force on link i in the local link frame may be expressed as

$$f_{R,i}^{\text{link},i} = -mg \begin{bmatrix}\mu_t & 0 \\ 0 & \mu_n\end{bmatrix} \text{sgn}\left(v_i^{\text{link},i}\right) \tag{13}$$

where g is the gravitational acceleration constant, $v_i^{\text{link},i}$ is the link velocity expressed in the local link frame, and sgn(.) is the signum function. Expressing $f_{R,i}^{\text{link},i}$ and $v_i^{\text{link},i}$ in the global frame by use of (2), the friction force on all links, $f_R = \left[f_{R,x}^T, f_{R,y}^T\right]^T \in \mathbb{R}^{2n}$, may be expressed in matrix form as

$$f_R = -mg \begin{bmatrix}\mu_t C_\theta & -\mu_n S_\theta \\ \mu_t S_\theta & \mu_n C_\theta\end{bmatrix} \text{sgn}\left(\begin{bmatrix} C_\theta & S_\theta \\ -S_\theta & C_\theta\end{bmatrix}\begin{bmatrix}\dot{x} \\ \dot{y}\end{bmatrix}\right) \tag{14}$$

The friction torque about the CM of link i is a result of friction forces acting normal to the link during link rotation and is characterized by the normal friction coefficient, μ_n. The direction of velocity of a point along the link with respect to the ground is actually also dependent on the translational velocity of the link. However, in order to simplify the friction model, the friction torque is modelled based on the link rotation only. The friction force, df, on an infinitesimal mass element, dm, of link i due to the link rotation, $\dot{\theta}_i$, produces a friction torque about the CM of the link given by

$$d\tau_i = s df_i = s\left(-\mu_n g \cdot \text{sgn}\left(s\dot{\theta}_i\right) \cdot dm\right) \tag{15}$$

where s is the distance from the CM of link i to the mass element, dm. Using the relation $dm = \frac{m}{2l} ds$, we may write the total friction torque on link i as

$$\tau_{R,i} = \int_{-l}^{l} d\tau_i = -\frac{1}{2}\mu_n mgl \cdot \text{sgn}\left(\dot{\theta}_i\right) \tag{16}$$

The global frame friction torque on all links, $\tau_R \in \mathbb{R}^n$, may be expressed in matrix form as

$$\tau_R = -\frac{1}{2}\mu_n mgl \cdot \text{sgn}\left(\dot{\theta}\right) \tag{17}$$

E. Equations of motion without contact forces

This section derives the equations of motion for the snake robot in terms of the acceleration of the link angles, $\ddot{\theta}$, and the acceleration of the CM of the snake robot, $\ddot{p}$. These coordinates describe all $N+2$ DOFs of the snake robot.

The forces and torques acting on link i are visualized in Fig. 2. The force balance for link i in global frame coordinates is given by

$$\begin{aligned} m\ddot{x}_i &= f_{R,x,i} + h_{x,i} - h_{x,i-1} \\ m\ddot{y}_i &= f_{R,y,i} + h_{y,i} - h_{y,i-1}\end{aligned} \tag{18}$$

The force balance equations for all links may be expressed in matrix form as

$$\begin{aligned} m\ddot{x} &= f_{R,x} + D^T h_x \\ m\ddot{y} &= f_{R,y} + D^T h_y\end{aligned} \tag{19}$$

Note that the link accelerations may also be expressed by differentiating (4) twice with respect to time. This gives

$$\begin{aligned} D\ddot{x} &= lA\left(C_\theta\dot{\theta}^2 + S_\theta\ddot{\theta}\right) \\ D\ddot{y} &= lA\left(S_\theta\dot{\theta}^2 - C_\theta\ddot{\theta}\right)\end{aligned} \tag{20}$$

We obtain the acceleration of the CM by differentiating (3) twice with respect to time, inserting (19), and noting that the constraint forces, h_x and h_y, are cancelled out when the link accelerations are summed. This gives

$$\begin{bmatrix}\ddot{p}_x \\ \ddot{p}_y\end{bmatrix} = \frac{1}{n}\begin{bmatrix}e^T\ddot{x} \\ e^T\ddot{y}\end{bmatrix} = \frac{1}{nm}\begin{bmatrix}e^T & 0_{1\times n} \\ 0_{1\times n} & e^T\end{bmatrix} f_R \tag{21}$$

This equation simply states, as would be expected, that the acceleration of the CM of a snake robot equals the sum of the external forces acting on the robot divided by its mass.

The torque balance for link i is given by

$$\begin{aligned} J\ddot{\theta}_i &= u_i - u_{i-1} + \tau_{R,i} \\ -l\sin\theta_i(h_{x,i} + h_{x,i-1}) &+ l\cos\theta_i(h_{y,i} + h_{y,i-1})\end{aligned} \tag{22}$$

The torque balance equations for all links may be expressed in matrix form as

$$J\ddot{\theta} = D^T u + \tau_R - lS_\theta A^T h_x + lC_\theta A^T h_y \tag{23}$$

What now remains is to remove the constraint forces from (23). Premultiplying (19) by D, solving for h_x and h_y, and also inserting (20), gives

$$\begin{aligned} h_x &= \left(DD^T\right)^{-1}\left(mlA\left(C_\theta\dot{\theta}^2 + S_\theta\ddot{\theta}\right) - Df_{R,x}\right) \\ h_y &= \left(DD^T\right)^{-1}\left(mlA\left(S_\theta\dot{\theta}^2 - C_\theta\ddot{\theta}\right) - Df_{R,y}\right)\end{aligned} \tag{24}$$

Inserting into (23) and solving for $\ddot{\theta}$ finally gives

$$M\ddot{\theta} = W\dot{\theta}^2 + lS_\theta N f_{R,x} - lC_\theta N f_{R,y} + \tau_R + D^T u \tag{25}$$

where

$$\begin{aligned} M &:= JI_{n\times n} + ml^2 S_\theta V S_\theta + ml^2 C_\theta V C_\theta \\ W &:= ml^2 C_\theta V S_\theta - ml^2 S_\theta V C_\theta \\ N &:= A^T\left(DD^T\right)^{-1} D \\ V &:= A^T\left(DD^T\right)^{-1} A\end{aligned} \tag{26}$$

The equations of motion for the snake robot are in other words given by (21) and (25). These equations may be combined into the following single differential equation:

$$Q\left(q\right)\ddot{q} = F\left(q,\dot{q},u\right) \tag{27}$$

where

$$\begin{aligned} q &= \left[\theta^T, p_x, p_y\right]^T \in \mathbb{R}^{n+2} \\ Q\left(q\right) &= \begin{bmatrix} M & 0_{n\times 1} & 0_{n\times 1} \\ 0_{1\times n} & nm & 0 \\ 0_{1\times n} & 0 & nm \end{bmatrix} \in \mathbb{R}^{n+2\times n+2} \\ F(q,\dot{q},u) &= \begin{bmatrix} lS_\theta N & -lC_\theta N \\ e^T & 0_{1\times n} \\ 0_{1\times n} & e^T \end{bmatrix} f_R + \begin{bmatrix} W\dot{\theta}^2 + \tau_R + D^T u \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{28}$$

III. Modelling of obstacle contact forces

This section presents an obstacle contact model for the snake robot as an extension to the model in (27).

A. Description of the approach

The interactions between the snake robot links and external obstacles are modelled by introducing a unilateral nonholonomic constraint for each link in contact with an obstacle. This constraint will prevent a link from moving sideways towards (and thereby into) the obstacle. The constraint is unilateral (acts in one lateral direction only) since the constraint force should not prevent the link from moving away from the obstacle.

The approach is basically to directly calculate the forces needed to satisfy the nonholonomic constraints on the links and then add these constraint forces to the equations of motion in (27) in order to cancel out the applied forces acting against the constraints. The constraint forces are in other words added to the equations of motion in order to convert the accelerations of the links into admissible accelerations that are consistent with the constraints. The main advantage of this approach over more conventional approaches, such as modelling interactions as mass-spring-damper systems, is that it produces an explicit and analytical expression for the contact forces on the snake robot. This facilitates model-based controller design for obstacle-aided locomotion. More information about modelling of systems with nonholonomic constraints may be found in [15], [16].

Note that the proposed approach will not prevent a link from continuing 'into' an obstacle with constant velocity after the acceleration has been removed. A viscous damping term is therefore added to the equations of motion in order to damp out the normal direction velocity of a contacted link. The friction forces between the links and the obstacles are also modelled.

We will in the following extend the model in (27) with obstacle contact forces, f_C, velocity damping forces, f_D, and obstacle friction forces, f_μ. The complete model of the snake robot will have the form

$$Q(q)\ddot{q} = F(q,\dot{q},u) + \begin{bmatrix} lS_\theta N & -lC_\theta N \\ e^T & 0_{1\times n} \\ 0_{1\times n} & e^T \end{bmatrix} (f_C + f_D + f_\mu) \tag{29}$$

B. Nonholonomic constraint equations

This section derives a matrix equation containing the nonholonomic constraint equations for all the links of the snake robot on the form

$$C(q)\dot{q} = 0 \tag{30}$$

where $C(q) \in \mathbb{R}^{n\times n+2}$ and $q = \left[\theta^T, p_x, p_y\right]^T \in \mathbb{R}^{n+2}$. In the contact model, row i of this matrix equation will only be valid if the link is in contact with an obstacle. A nonholonomic constraint on link i means that the normal direction velocity, here denoted $v_{n,i}$, of the link is constrained to zero, which will be the case when the link contacts an obstacle. This constraint may be expressed as

$$v_{n,i} = -\dot{x}_i \sin\theta_i + \dot{y}_i \cos\theta_i = 0 \tag{31}$$

Inserting (10) and (11) into (31), and also using the trigonometrical relation $\sin\theta_i \sin\theta_j + \cos\theta_i \cos\theta_j = \cos\left(\theta_i - \theta_j\right)$ and the notation $c_{ij} = \cos\left(\theta_i - \theta_j\right)$, the constraint equation may be written as

$$\begin{aligned} -\dot{p}_x \sin\theta_i + \dot{p}_y \cos\theta_i + \textstyle\sum_{j=1}^{i-1} a_j \left(c_{ij}\dot{\theta}_j + c_{i,j+1}\dot{\theta}_{j+1}\right) \\ + \textstyle\sum_{j=i}^{n-1} b_j \left(c_{ij}\dot{\theta}_j + c_{i,j+1}\dot{\theta}_{j+1}\right) = 0 \end{aligned} \tag{32}$$

By inspecting (32), it is seen that this equation can be expressed for all links in the following matrix form:

$$\begin{bmatrix} b_1c_{11} & (b_1+b_2)c_{12} & (b_2+b_3)c_{13} & .. & b_{n-1}c_{1n} & -s\theta_1 & c\theta_1 \\ a_1c_{21} & (a_1+b_2)c_{22} & (b_2+b_3)c_{23} & .. & b_{n-1}c_{2n} & -s\theta_2 & c\theta_2 \\ a_1c_{31} & (a_1+a_2)c_{32} & (a_2+b_3)c_{33} & .. & b_{n-1}c_{3n} & -s\theta_3 & c\theta_3 \\ \vdots & & & & & & \vdots \\ a_1c_{n1} & (a_1+a_2)c_{n2} & (a_2+a_3)c_{n3} & .. & a_{n-1}c_{nn} & -s\theta_n & c\theta_n \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{p}_x \\ \dot{p}_y \end{bmatrix} = 0 \tag{33}$$

where $s\theta_i = \sin\theta_i$ and $c\theta_i = \cos\theta_i$. We have now derived (30). From (33) it is easily seen that the matrix $C\left(q\right) \in \mathbb{R}^{n\times n+2}$ and that element ij of this matrix is given by

$$C_{ij}\left(q\right) = \begin{cases} \left(b_{j-1} + b_j\right) c_{ij} & \forall \quad i < j \leqslant n \\ \left(a_{j-1} + b_j\right) c_{ij} & \forall \quad j = i \\ \left(a_{j-1} + a_j\right) c_{ij} & \forall \quad j < i \\ -\sin\theta_i & \forall \quad j = n+1 \\ \cos\theta_i & \forall \quad j = n+2 \end{cases} \tag{34}$$

As will be seen in the next section, the calculation of contact forces requires the time derivative of the matrix $C\left(q\right)$. Differentiating each component in (34) and using the notation $s_{ij} = \sin\left(\theta_i - \theta_j\right)$ gives

$$\dot{C}_{ij}(q) = \begin{cases} \left(b_{j-1}+b_j\right)\left(\dot{\theta}_j - \dot{\theta}_i\right) s_{ij} & \forall \;\; i < j \leqslant n \\ \left(a_{j-1}+b_j\right)\left(\dot{\theta}_j - \dot{\theta}_i\right) s_{ij} & \forall \;\; j = i \\ \left(a_{j-1}+a_j\right)\left(\dot{\theta}_j - \dot{\theta}_i\right) s_{ij} & \forall \;\; j < i \\ -\cos\theta_i\dot{\theta}_i & \forall \;\; j = n+1 \\ -\sin\theta_i\dot{\theta}_i & \forall \;\; j = n+2 \end{cases} \tag{35}$$

C. Calculation of the contact forces

With nonholonomic constraints, a term must be added to the equations of motion in (27) that represents the forces needed to satisfy these constraints [16]. The constrained equations of motion therefore take on the form

$$Q\left(q\right)\ddot{q} = F\left(q,\dot{q},u\right) + \begin{bmatrix} \widehat{u} \\ \widehat{f} \end{bmatrix} = F + \widehat{F} \tag{36}$$

where $\widehat{u} \in \mathbb{R}^n$ is a vector containing the torques about the CM of each link resulting from all the constraint forces, $\widehat{f} \in \mathbb{R}^2$ contains the sum of the constraint forces in the

global x and y direction, respectively, and $\widehat{F} = \left[\widehat{u}^T, \widehat{f}^T\right]^T \in \mathbb{R}^{n+2}$. Note that the term *constraint forces* denotes both the constraint forces and constraint torques of the system.

Assume that m links of the snake robot are in contact with obstacles. We express the m constraint equations for the contacted links as

$$C(q)\dot{q} = 0 \quad , \quad C(q) \in \mathbb{R}^{m \times n+2} \tag{37}$$

where all rows from (33) corresponding to links that are not in contact with an obstacle have been excluded. Differentiating (37) gives

$$\dot{C}(q)\dot{q} + C(q)\ddot{q} = 0 \tag{38}$$

The matrix $\dot{C}(q)$ is given by (35). Insertion of (36) gives

$$\dot{C}\dot{q} + CQ^{-1}\left(F + \widehat{F}\right) = 0 \tag{39}$$

which are m equations with the $n + 2 > m$ components in $\widehat{F}$ as unknowns. We need an additional condition in order to solve for $\widehat{F}$. We get that condition by employing the *principle of virtual work* [16], which states that the constraint forces, $\widehat{F}$, should not change the energy of the system, i.e. they should not do any work. This gives us the additional condition

$$\widehat{F}^T\dot{q} = 0 \quad \forall \dot{q} \mid C(q)\dot{q} = 0 \tag{40}$$

where $\widehat{F}^T\dot{q}$ is the work done by the constraint forces. Comparing (30) and (40), we see that vectors $\widehat{F}$ that satisfy this requirement can be expressed as

$$\widehat{F} = C^T\lambda \tag{41}$$

where $\lambda \in \mathbb{R}^m$ is a vector of scalars known as *Lagrange multipliers* [16]. This vector is found by inserting (41) into (39) and then solving for λ to get

$$\lambda = -\left(CQ^{-1}C^T\right)^{-1}\left(CQ^{-1}F + \dot{C}\dot{q}\right) \tag{42}$$

By substituting (42) into (41), the constraint forces are found.

The term *contact force* denotes the translational force exerted on a link from an obstacle. We now derive the obstacle contact forces from the constraint forces in (41) as these are needed in order to compute the obstacle friction forces. The obstacle contact forces are assumed to only act on the CM of the links and are denoted by

$$f_C = \begin{bmatrix} f_{C,x} \\ f_{C,y} \end{bmatrix} = \begin{bmatrix} -S_\theta \\ C_\theta \end{bmatrix} f_{C,n} \tag{43}$$

where $f_{C,x} \in \mathbb{R}^n$ and $f_{C,y} \in \mathbb{R}^n$ contain the contact forces on the links in the global x and y direction, respectively, and $f_{C,n} \in \mathbb{R}^n$ contains the contact force along the local y axis of each link. A contact force always acts normal to a link and thereby along the local y axis of the link. The constraint forces $\widehat{F} = \left[\widehat{u}^T, \widehat{f}^T\right]^T$ derived in (41) are simply the contact forces on each link, f_C, mapped into the resulting torques, $\widehat{u}$, and forces, $\widehat{f}$. The mapping between f_C and $\widehat{u}$ must necessarily be equal to the mapping between the ground friction forces and the resulting link torques, which is seen from (25). This gives the relationship

$$\widehat{u} = lS_\theta N f_{C,x} - lC_\theta N f_{C,y} \tag{44}$$

Using (43) and solving for $f_{C,n}$ gives

$$f_{C,n} = -\left(lS_\theta N S_\theta + lC_\theta N C_\theta\right)^{-1}\widehat{u} \tag{45}$$

The global frame contact forces are finally found by inserting (45) into (43). Note that the contact forces should only act on a link in the direction away from the obstacle and should not prevent the link from moving away from the obstacle.

D. Velocity damping for contacted links

The constraint forces remove the normal direction acceleration, but not the normal direction velocity, $v_{n,i}$, that a link has at the moment it contacts an obstacle. A viscous damping force, f_D, is therefore calculated to damp out the normal direction velocity of each contacted link. Note that velocity damping would not have been necessary if the nonholonomic constraints had been present all the time (which is the case for more conventional nonholonomic systems) since the permanent acceleration constraints then would have prevented the system from ever reaching nonzero velocities that conflict with the constraints. The obstacle constraints are, however, only present when a snake robot link makes contact with an obstacle.

We assume that the damping forces act on the CM of the links and write the damping force, $f_{D,i}^{\text{link},i}$, on link i in the local link frame as

$$f_{D,i}^{\text{link},i} = -d\begin{bmatrix} 0 \\ v_{n,i} \end{bmatrix} = -d\begin{bmatrix} 0 \\ -\dot{x}_i \sin\theta_i + \dot{y}_i \cos\theta_i \end{bmatrix} \tag{46}$$

where d is the damping coefficient. Expressing this force in the global frame using (2) and assembling the damping forces for all the links into matrix form gives

$$f_D = \begin{bmatrix} f_{D,x} \\ f_{D,y} \end{bmatrix} = -d\begin{bmatrix} (S_\theta)^2 & -S_\theta C_\theta \\ -S_\theta C_\theta & (C_\theta)^2 \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{47}$$

where $f_{D,x} \in \mathbb{R}^n$ and $f_{D,y} \in \mathbb{R}^n$ contain the damping forces on all the links in the global x and y direction, respectively. Note that the damping force is set to zero for any link that is not in contact with an obstacle and also for contacted links with normal direction velocity pointing away from the obstacle.

E. Obstacle friction forces

A Coulomb friction model is employed in order to describe the gliding friction forces, here denoted f_μ, between the links and the obstacles. The local link frame friction force, $f_{\mu,i}^{\text{link},i}$, on link i acts in the tangential direction of this link (along local x axis) and is given by

$$f_{\mu,i}^{\text{link},i} = \begin{bmatrix} -\mu_o \cdot \text{sgn}\left(v_{t,i}\right) \cdot \left|f_{C,n,i}\right| \\ 0 \end{bmatrix} \tag{48}$$

where μ_o is the Coulomb friction coefficient of the obstacles, $f_{C,n,i}$ is given by (45), and $v_{t,i}$ is the tangential link velocity. Expressing this force and the tangential link velocity in the global frame using (2), and also assembling the obstacle friction forces for all the links into matrix form gives

$$f_\mu = \begin{bmatrix} f_{\mu,x} \\ f_{\mu,y} \end{bmatrix} = -\mu_o \begin{bmatrix} C_\theta \\ S_\theta \end{bmatrix} \text{diag}\left(\left|f_{C,n}\right|\right) \text{sgn}\left(C_\theta\dot{x} + S_\theta\dot{y}\right) \tag{49}$$

where $f_{\mu,x} \in \mathbb{R}^n$ and $f_{\mu,y} \in \mathbb{R}^n$ contain the obstacle friction forces on all the links in the global x and y direction, respectively. Note that the obstacle friction force is set to zero for any link that is not in contact with an obstacle.

IV. CONTROL

A major challenge during obstacle-aided locomotion is to prevent the snake robot from being jammed between the obstacles. In a jammed situation, the propulsive components of the contact forces from the obstacles are too small to overcome the friction forces from the ground and the obstacles, at the same time as the contact forces prevent a number of the snake robot joints from moving to their commanded angle. The control strategy proposed in the following has two states consisting of a *leader-follower* algorithm with automatic *jam detection and resolution*. A leader-follower algorithm is carried out as long as the snake robot is able to move without being jammed between the obstacles. If a jam situation occurs, a jam resolution algorithm is carried out in order to effectively 'unlock' the jammed joints. The concept of detecting and resolving snake robot jams has, to the authors' best knowledge, not been treated in previous literature.

A. The leader-follower algorithm

The leader-follower algorithm is inspired by coordinated and cooperative control theory. We view the snake robot as a constellation of n links that should cooperate in order to achieve their common goal of locomotion. Choosing between a coordinated and cooperative control approach, we choose the coordinated leader-follower scheme. This choice is motivated by the fact that each part of a biological snake conducting lateral undulation follows the path traced out by the head [2]. We therefore choose the head joint angle (the foremost joint), ϕ_{n-1}, as the reference angle for all subsequent joints. The preferred direction of motion for the snake is defined to be along the global positive x axis.

The reference trajectory of the leader, i.e. the head link, is given as follows. In order to achieve the sinusoidal motion characteristic of lateral undulation, we alternate between moving the head in the leftward and rightward direction with respect to the global positive x axis (the direction in which the snake should move). This may be achieved by choosing the reference angle for the head link, $\theta_{n,\text{ref}}$, equal to a suitable positive constant α when the head should move leftward and a negative constant $-\alpha$ when the head should move rightward. The criterion for switching between these two reference directions is defined to be the instant when the distance, $\triangle y$, between the position of the head along the global y axis, y_n, (perpendicular to the direction of locomotion) and the y axis coordinate of the CM of the snake robot, p_y, becomes greater than some defined amplitude, $\triangle y_{\max}$. This criterion is illustrated in Fig. 3. The distance, $\triangle y$, may easily be calculated as a function of the link angles by following the same logical steps that led to the expression in (11). The derivation is not included here due to space restrictions, but leads to the expression

$$\triangle y := y_n - p_y = \sum_{j=1}^{n-1} \frac{l}{n} j \left(\sin\theta_j + \sin\theta_{j+1}\right) \tag{50}$$

The reference angle of the head link is in other words set according to the rule

$$\begin{array}{c} \text{Leftward motion: } \theta_{n,\text{ref}} = \alpha \text{ until } \triangle y > \triangle y_{\max} \\ \text{Rightward motion: } \theta_{n,\text{ref}} = -\alpha \text{ until } \triangle y < -\triangle y_{\max} \end{array} \tag{51}$$

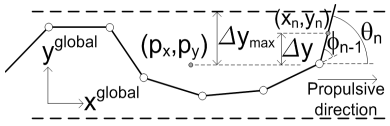

Fig. 3. Control of the head of the snake robot.

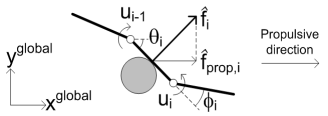

Fig. 4. The contact force on a link.

To obtain this link angle, the head joint angle is controlled according to the reference

$$\phi_{n-1,\text{ref}} = \theta_{n-1} - \theta_{n,\text{ref}} \tag{52}$$

In order to generate a leader-follower based control reference to the remaining links, the head joint angle, ϕ_{n-1}, is propagated backwards along the snake body at a constant propagation velocity, v_{ref}, and used as the reference angle for all subsequent joints. For a given choice of v_{ref}, the time offset, $\triangle t$, between two consecutive joints with intermediate distance $2l$ is found as $\triangle t = \frac{2l}{v_{\text{ref}}}$.

To summarize, the reference angles for all the joints of the snake robot in this leader-follower scheme are

$$\begin{array}{c} \phi_{n-1,\text{ref}}(t) = \theta_{n-1}(t) - \theta_{n,\text{ref}} \\ \phi_{i,\text{ref}}(t) = \phi_{n-1}(t-(n-i-1)\triangle t) \;\; \forall \;\; i = 1..n-2 \end{array} \tag{53}$$

B. The jam resolution algorithm

A joint of the snake robot is defined to be *jammed* if the deviation between the joint angle and its reference angle exceeds a certain limit, $\triangle\phi_{\max}$. It is reasonable to assume that a jam of a single joint will resolve by itself. However, two jammed joints could be caused by a situation where the contact forces cause the jammed joints to act 'against' each other. This situation may not always resolve by itself. If two joints are in a jammed state for a period longer than $t_{\text{jam,max}}$, the snake robot is defined to be in a jammed state. This causes the leader-follower algorithm to be stopped in order to carry out a jam resolution algorithm. The snake robot stays in the jam resolution state for a predefined amount of time, $t_{\text{resolution,max}}$, since it is difficult to come up with a specific criteria for when a jam has been resolved.

The idea behind the jam resolution algorithm is to *rotate the links affected by contact forces so that the propulsive component of each contact force increases*. In a jammed situation, the propulsive components of the contact forces from the obstacles are too small to overcome the friction forces from the ground and the obstacles. Rotating the contacted links (and thereby the direction of the contact forces) to increase the total propulsive contact force should therefore resolve the jammed situation.

The contact force on link i, illustrated in Fig. 4, is assumed to be measured. This measurement is denoted $\hat{f}_i$. The propulsive component, $\hat{f}_{\text{prop},i}$, of a contact force on link

i is defined as the force component in the desired direction of motion and is given by

$$\hat{f}_{\text{prop},i} = -\hat{f}_i \sin\theta_i \tag{54}$$

The change of the propulsive component due to a change of the link angle is found by differentiating (54) with respect to θ_i:

$$\frac{\partial \hat{f}_{\text{prop},i}}{\partial \theta_i} = -\hat{f}_i \cos\theta_i \tag{55}$$

This expression shows that a change of a contact force near perpendicular to the direction of motion has greater effect on the propulsive component than a similar change of a contact force near parallel to the direction of motion. The jam resolution algorithm should therefore prioritize to rotate links with a high propulsive force gradient with respect to the link angle. This suggests the following expression for the desired change of the link angle during jam resolution:

$$\triangle\theta_{i,\text{ref}} = k_\theta \frac{\partial \hat{f}_{\text{prop},i}}{\partial \theta_i} = -k_\theta \hat{f}_i \cos\theta_i \tag{56}$$

where k_θ is the gain of the jam resolution controller. We would only like to change the angle of link i while leaving the angle of link $i-1$ and $i+1$ unchanged. This means that $\triangle\theta_{i-1,\text{ref}} = \triangle\theta_{i+1,\text{ref}} = 0$ and that the resulting desired change of the joint angles is given by

$$\begin{aligned} \triangle\phi_{i-1,\text{ref}} &= \triangle\theta_{i-1,\text{ref}} - \triangle\theta_{i,\text{ref}} = k_\theta \hat{f}_i \cos\theta_i \\ \triangle\phi_{i,\text{ref}} &= \triangle\theta_{i,\text{ref}} - \triangle\theta_{i+1,\text{ref}} = -k_\theta \hat{f}_i \cos\theta_i \end{aligned} \tag{57}$$

During jam resolution, we leave the head joint angle unchanged to maintain a smooth head angle. We also use the measured contact forces at the instant the jam was detected as feedback and do not update the force vector in the jam resolution state. This ensures a steady rotation of the contacted links in accordance with the contact forces that produced the jam. The reference angles for all the joints of the snake robot in the jam resolution state may now be summarized as

$$\begin{aligned} \phi_{n-1,\text{ref}} &= \phi_{n-1} \\ \phi_{i,\text{ref}} &= \phi_i + k_\theta\left(-\hat{f}_i \cos\theta_i + \hat{f}_{i+1}\cos\theta_{i+1}\right) \;\forall\, i{=}1..n{-}2 \end{aligned} \tag{58}$$

C. Low-level joint controller

A standard PD-controller is used to calculate the joint actuator torques from the joint reference angles according to

$$u_i = k_P\left(\phi_{i,\text{ref}} - \phi_i\right) - k_D\dot{\phi}_i \tag{59}$$

where k_P and k_D are the gains of the controller. A velocity reference is not included in (59) since the transitions between the two states of the proposed control strategy produce steps in the reference angles, which would lead to large velocity references. The joint reference angles are saturated to ensure that they stay within the range $[-\phi_{\max}, \phi_{\max}]$ corresponding to the maximum joint angle of the snake robot joints. The actual joint angles are also forced to stay within this range by adding a large restoring joint torque towards the zero angle if a joint is outside its legal range.

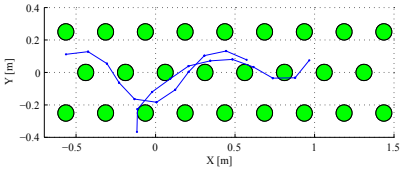

Fig. 5. Path taken by the snake robot without jam resolution. The motion is jammed after about 0.4 m.

V. Simulation results

This section presents results from simulations of the model in (29) and the control strategy in (53) and (58). The model was simulated using *Matlab R2008a* on a laptop running *Windows XP*. The *ode45* solver in Matlab was used with a relative and absolute error tolerance of 10^{-3}.

A. Simulation parameters

The parameters characterizing the simulated snake robot were n = 10, l = 0.07 m, m = 1 kg, and J = 0.0016 kgm^2. These parameters characterize a physical snake robot currently under development in our lab.

The various controller parameters were set to k_P = 20, k_D = 5, v_{ref} = 0.2 $^m/_s$, $\triangle y_{\max}$ = 0.14 m, α = 50°, k_θ = 0.05, $\phi_{\max}$ = 50°, $\triangle\phi_{\max}$ = 20°, $t_{\text{jam,max}}$ = 0.5 s, $t_{\text{resolution,max}}$ = 0.5 s. These were found based on both physical insight and through trial and error.

The ground and obstacle friction coefficients were set to $\mu_t = \mu_n = \mu_o = 0.3$, with obstacle damping coefficient d = 1000. The obstacles were chosen to be three rows (parallel to the x axis) of cylindrical objects with a diameter of 10 cm each. The center distance between two obstacles in a row and the distance between two rows were 25 cm. The middle row was displaced with respect to the other two rows by half this distance (12.5 cm) along the x axis.

The initial link angles and CM (center of mass) position of the snake robot were set to $\theta_0 = [7, -32, -57, -46, -8, 33, 53, 45, 12, -23]^T$ [deg] and $p_0 = [0, 0]^T$, respectively. The initial shape was more or less randomly chosen in order to give the snake an initial curl around the obstacles.

B. Simulation without jam resolution

In order to demonstrate the problem of snake robot jams during obstacle-aided locomotion, this section first presents simulation results where the leader-follower algorithm was employed, but where jam detection and resolution was disabled. The initial (t = 0 s) and final (t = 7 s) shape and position of the snake robot is shown in Fig. 5. The snake robot managed to crawl about 0.4 m along the global positive x axis before the motion stopped due to a jam between the obstacles. The velocity of the CM is plotted in Fig. 6, which clearly shows how the speed gained during the first 3.5 s is reduced to zero after about 4.5 s. The snake robot was jammed because the propulsive components of the contact forces from the obstacles were too small to overcome the friction forces from the ground and the obstacles.

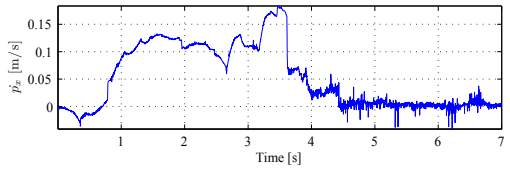

Fig. 6. Speed of the snake robot without jam resolution. The motion is jammed after about 4.5 s.

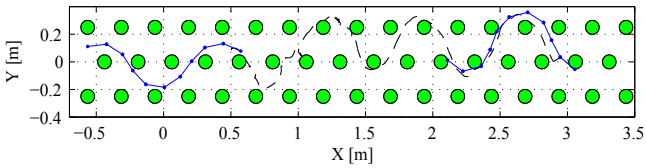

Fig. 7. Path taken by the snake robot with jam resolution.

C. Simulation with jam resolution

The next simulation results show the effect of the jam resolution algorithm. The initial ($t = 0$ s) and final ($t = 30$ s) shape and position of the snake robot is shown in Fig. 7. The trace of the head link is plotted with a dashed line in this figure. After 30 s, the snake robot has managed to crawl more than 2.5 m along the global positive x axis. The velocity of the CM is plotted in Fig. 8, which shows that the speed varies around 10 $^{\text{cm}}/_{\text{s}}$. The triangles in the figure indicate when the snake robot is jammed (triangle pointing downwards), which initiates jam resolution, and when the jam resolution algorithm has finished (triangle pointing upwards). Each jam resolution phase is characterized by a decrease in the speed of the snake due to the reorientation of the links, but a subsequent increase of the speed to a similar or higher value compared to the speed the snake had at the instant it was jammed. All the jams that occurred during a run of 30 s were resolved by the proposed algorithm.

To clearly illustrate the effect of the jam resolution algorithm, a plot of the snake robot before (dashed) and after (solid) a jam resolution is shown in Fig. 9. The figure shows the jam occurring at time $t = 5.96$ s and ending at time $t = 6.46$ s. The jam is caused by obstacle contact forces acting on links 7 and 9, and is resolved by rotating link 7 clockwise and link 9 counterclockwise during the jam resolution, thereby increasing the propulsive components of the two contact forces enough to overcome the friction forces from the ground and the obstacles.

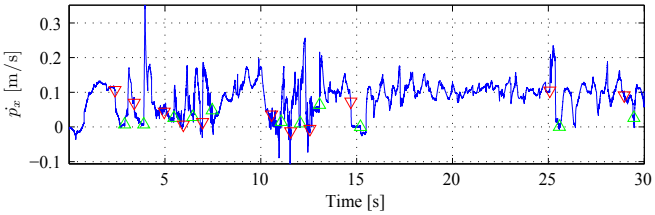

Fig. 8. Speed of the snake robot with jam resolution.

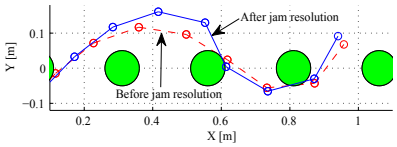

Fig. 9. Snake robot before (dashed) and after (solid) jam resolution.

VI. Conclusion

This paper has presented a model of the dynamics of a snake robot where interactions between the snake robot and external obstacles are modelled as nonholonomic constraints. A control strategy was derived for enabling the snake robot to propel itself forward by using obstacles along its path as push-points (denoted *obstacle-aided locomotion*) while simultaneously preventing the obstacles from jamming the motion. The simulation results validated the contact modelling approach and illustrated how the proposed jam detection and resolution algorithm can help to maintain the propulsion of a snake robot in a cluttered environment.

Future work will validate the presented simulation results through experiments with a snake robot currently under development at NTNU/SINTEF in Norway.

References

[1] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle aided locomotion: Modeling, simulations, and experiments," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 88–104, February 2008.

[2] J. Gray, "The mechanism of locomotion in snakes," *J. Exp. Biol.*, vol. 23, no. 2, pp. 101–120, 1946.

[3] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford: Oxford University Press, 1993.

[4] H. Date, M. Sampei, and S. Nakaura, "Control of a snake robot in consideration of constraint force," in *Proc. IEEE Int. Conf. on Control Applications*, 2001, pp. 966–971.

[5] S. Ma, "Analysis of creeping locomotion of a snake-like robot," *Adv. Robotics*, vol. 15, no. 2, pp. 205–224, 2001.

[6] P. Prautsch and T. Mita, "Control and analysis of the gait of snake robots," in *Proc. IEEE Int. Conf. Control Applications*, Kohala Coast, HI USA, 1999, pp. 502–507.

[7] M. Saito, M. Fukaya, and T. Iwasaki, "Serpentine locomotion with robotic snakes," *IEEE Contr. Syst. Mag.*, vol. 22, no. 1, pp. 64–81, February 2002.

[8] I. Grabec, "Control of a creeping snake-like robot," in *Proc. 7th Int. Workshop on Advanced Motion Control*, July 2002, pp. 526–513.

[9] G. P. Hicks, "Modeling and control of a snake-like serial-link structure," Ph.D. dissertation, North Carolina State University, 2003.

[10] P. Liljebäck, Ø. Stavdahl, and K. Y. Pettersen, "Modular pneumatic snake robot: 3D modelling, implementation and control," in *Proc. 16th IFAC World Congress*, July 2005.

[11] Z. Bayraktaroglu and P. Blazevic, "Understanding snakelike locomotion through a novel push-point approach," *J. Dyn. Syst. - Trans. ASME*, vol. 127, no. 1, pp. 146–152, March 2005.

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

[13] Y. Shan and Y. Koren, "Design and motion planning of a mechanical snake," *IEEE Trans. Syst. Man Cyb.*, vol. 23, no. 4, pp. 1091–1100, July-August 1993.

[14] Z. Y. Bayraktaroglu, A. Kilicarslan, A. Kuzucu, V. Hugel, and P. Blazevic, "Design and control of biologically inspired wheel-less snake-like robot," in *Proc. IEEE/RAS-EMBS Int. Conf. Biomedical Robotics and Biomechatronics*, February 2006, pp. 1001–1006.

[15] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1746–1757, Nov 1992.

[16] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics - Third Edition*. Addision Wesley, 2002.