# Path following control of snake robots in unstructured environments

Pål Liljebäck, Kristin Y. Pettersen, Øyvind Stavdahl, and Jan Tommy Gravdahl

*Abstract*— As a step towards enabling snake robots to move in unstructured environments, this paper considers control strategies where environment adaptation is combined with directional control of snake robot locomotion. The first contribution of the paper is a general framework for motion control of snake robots, which allows the motion to be specified in terms of a *body wave* component, an *environment adaptation* component, and a *heading control* component. As a second contribution, we employ the controller framework to propose a control law for straight line path following control of snake robots in environments with obstacles. The paper presents simulation results where the path following controller is combined with a waypoint guidance strategy in order to steer the snake robot between waypoints in an obstacle environment.

## I. Introduction

Inspired by biological snakes, snake robots carry the potential of meeting the growing need for robotic mobility in challenging environments. Snake robots 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 provide traversability in irregular environments that surpasses the mobility of the more conventional wheeled, tracked and legged types of robots.

A unique feature of snake robot locomotion compared to other forms of robotic mobility is that ground irregularities are beneficial for the propulsion since they provide push-points for the robot. These ideas are in line with early work by Gray [1], who concluded that forward motion of a planar snake requires the existence of external forces acting in the normal direction of the snake body, and also the work in [2], which studies the motion of biological snakes as they interact with pegs in order to push themselves forward. While *obstacle avoidance* is important 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.* [3] and captures the essence of this concept.

The majority of previous research on control of snake robots has focused on open-loop strategies for flat surface motion aimed at resembling gaits displayed by biological snakes. Only a few works present control strategies where the surface is no longer assumed to be flat. Hirose [4] proposed a strategy for *lateral inhibition* that modifies the shape of a snake robot based on contact force sensing in order to *avoid* obstacles. The work in [5] proposes an inverse dynamics approach by formulating and numerically solving an optimization problem in order to, for a given set of

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

Kristin Y. Pettersen, Øyvind Stavdahl, and Jan Tommy Gravdahl are with the Dept. of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway. E-mail: {Kristin.Y.Pettersen, Oyvind.Stavdahl, Tommy.Gravdahl}@itk.ntnu.no.

obstacle contacts, calculate the contact forces required to propel the snake in a desired direction. A kinematic approach is proposed in [6], where a curve fitting procedure is used to determine the body shape of the robot with respect to the obstacles. The work in [7] presents a control strategy that uses motor current measurements to adjust the shape of a snake robot moving through an elastically deformable channel. Along with these works, we should also mention the work in [8], which analyses how obstacles around a snake robot affect its degrees of freedom, and the work in [9], which proposes a strategy for adapting the shape of a snake robot to its environment based on measured joint angles.

This paper has two contributions which extend previous work by the authors in [10]. The first contribution is the formulation of a general framework for motion control of snake robots. The framework allows the motion of the snake robot to be specified in terms of three separate components, namely a *body wave* component, an *environment adaptation* component, and a *heading control* component.

As a second contribution, we employ the controller framework to propose a control law that enables a snake robot to track a straight path while simultaneously adapting the body shape to its environment. The *body wave* component of the control law is based on a *predecessor-follower* scheme, where each joint follows the angle of the preceding joint ahead of itself. This approach is an improvement over previous work by the authors in [10], where each joint is controlled according to the angle of the head link. The drawback of the previous approach is that it relies on the assumption that the robot moves forward with the same speed as the head angle propagates backward. The *environment adaptation* component is based on the jam resolution principle from [10]. However, whereas a complex hybrid formulation is employed in [10], the jam resolution motion in this paper is specified in terms of simple continuous equations. The *heading control* component is similar to a guidance law employed by the authors in [11]. The paper presents simulation results where the path following controller is combined with a waypoint guidance strategy proposed in [12] in order to steer the snake robot between waypoints in an environment with obstacles.

The paper is organized as follows. Section II summarizes a mathematical model of a snake robot in an environment with obstacles. The general controller framework is presented in Section III, while the path following controller is specified within this framework in Section IV. The waypoint guidance strategy is summarized in Section V, and simulation results are presented in Section VI. Finally, Section VII presents some concluding remarks.

## II. The model of the snake robot

This section summarizes a hybrid model of the dynamics of a snake robot interacting with obstacles. The notation

in this section will be employed in the presentation of the controller in Section IV. For a more detailed presentation of the model, the reader is referred to [10].

The snake robot is illustrated in Fig. 1(a) and consists of $n$ links of length $2l$ interconnected by $n-1$ joints. All $n$ links have the same mass $m$ and moment of inertia $J = \frac{1}{3}ml^2$. The snake robot moves in the horizontal plane and has $n + 2$ degrees of freedom. The position of the CM (center of mass) of the robot is denoted by $\boldsymbol{p} = (p_x, p_y)$. The absolute angle $\theta_i$ of link $i \in \{1, \cdots, n\}$ (i.e. the orientation of link $i$) is expressed with respect to the global $x$ axis with counterclockwise positive direction. The local coordinate system of each link is fixed in the CM of the link with $x$ (tangential) and $y$ (normal) axis oriented such that they are aligned with the global $x$ and $y$ axis, respectively, when the link angle is zero. As seen in Fig. 1(a), the angle of joint $i$ is given by
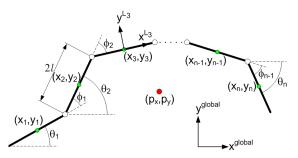
$$\phi_i = \theta_i - \theta_{i+1} \qquad (1)$$

The forces and torques acting on a link that is not in contact with an obstacle are shown in Fig. 1(b). The forces $h_{x,i}$ and $h_{y,i}$ are constraint forces that hold joint $i$ together. The actuator torque at joint $i \in \{1, \cdots, n-1\}$ is given by $u_i$. The isotropic Coulomb ground friction force on link $i$, denoted by $\boldsymbol{f}_{R,i}$, acts on the CM of the link and the friction torque, $\tau_{R,i}$, acts about the link CM. The ground friction coefficient is denoted by $\mu$.

The planar environment of the snake robot consists of an arbitrary number of external obstacles with circular shape. This requirement is not very restrictive since most objects can locally be approximated by circular shapes. The interaction between a snake robot link and an obstacle is modelled by introducing a *unilateral velocity constraint* for the link when it comes into contact with an obstacle. The constraint is *unilateral* (acts in one lateral direction only) since the constraint shall allow sideways motion of the link *away* from the obstacle, but prevent any sideways motion *towards* (and thereby into) the obstacle. As illustrated in Fig. 1(c), the obstacle contact force on link $i$ is assumed to act on the CM of the link and consists of two orthogonal components. The first component is the *constraint force*, $\boldsymbol{f}_{c,i}$, acting in the normal direction of link $i$ and away from the obstacle. The second component is the obstacle *friction force*, $\boldsymbol{f}_{\mu,i}$, acting in the tangential direction of link $i$ and in the opposite direction of the tangential link velocity. The friction coefficient between the snake robot and any obstacle is denoted by $\mu_o$.
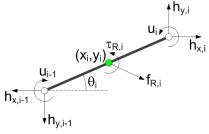
The model of the snake robot is developed within the framework of a *hybrid dynamical system* to handle the discontinuous nature of the model, i.e. to ensure that obstacle contact forces only occur when a link comes into contact with an obstacle. It is shown in [10] that the complete model can be written as
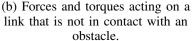
$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \quad \text{for all} \quad \boldsymbol{x} \in \boldsymbol{C} \\ \boldsymbol{x}^+ &= \boldsymbol{G}(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \boldsymbol{D} \end{aligned} \qquad (2)$$
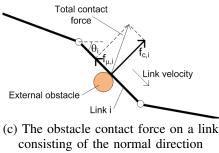
where $\boldsymbol{x}$ is the state vector and $\boldsymbol{u}$ is the vector of joint actuator torques. The state vector $\boldsymbol{x}$ flows continuously according to the *flow map* $\boldsymbol{F}$ as long as the state belongs to the *flow set* $\boldsymbol{C}$, which is the set of all states where the set of links that are in contact with obstacles remains fixed. The state vector enters the *jump set* $\boldsymbol{D}$ when a link impacts



(a) The kinematic parameters of the snake robot.



(b) Forces and torques acting on a link that is not in contact with an obstacle.



(c) The obstacle contact force on a link consisting of the normal direction constraint force and the tangential direction friction force.

Fig. 1. Parameters that define the kinematics and dynamics of the snake robot.

or detaches from an obstacle. In this case, the state will experience a jump according to the *jump map* $\boldsymbol{G}$. During an impact, the jump map basically cancels the normal direction velocity of the impacted link to prevent it from continuing into the obstacle.

## III. A CONTROLLER FRAMEWORK FOR SNAKE ROBOT LOCOMOTION

Motivated by previous research results by the authors, we propose in this section a general framework for motion control of snake robots. The controller proposed in Section IV will be specified within this framework.

In [10], [13], the authors propose and experimentally investigate a hybrid controller for obstacle-aided locomotion. The controller switches between a *leader-follower* scheme, where all joints in turn follow the angle of the foremost (head) joint, and a *jam resolution* scheme, where each link is rotated to increase the propulsive component of its contact force. Motivated by the results from [10], [13], we now

state a set of claims and subsequently propose a controller framework based on these claims.

**The controller should produce body wave motion:** An analysis of snake robot locomotion presented by the authors in [14] shows that propulsion of a snake robot is produced by moving the links transversal to the forward direction of motion and that the joint angles should be out of phase with each other. A wave motion is a natural consequence of these two requirements. The claim that the motion of a snake robot should include body wave motion is also supported by the motion of biological snakes in nature.

**The controller should continuously perform environment adaptation:** During the experiments of the hybrid controller in [13], the motion of the snake robot was jammed quite frequently. These results indicate that conducting wave motion in open-loop *will* eventually jam the motion of the robot, which strongly suggests that the wave motion should *not* be conducted in open-loop, but rather adjusted continuously according to the environment of the robot. We therefore claim that environment adaptation should be conducted continuously in parallel with the cyclic wave motion of the snake robot.

**The controller should steer the heading:** This requirement is obvious in order to be able to steer the snake robot to a desired location. The controller in [10] was aimed at propelling the robot forward, but did not explicitly control the direction of the motion.

Based on the above claims, we propose the following general controller framework for snake robots:

*Conjecture 1:* **The controller framework.**
Efficient and intelligent snake robot locomotion in unknown and unstructured environments can be achieved by specifying the reference angles, $\phi_{\mathrm{ref}} = \begin{bmatrix} \phi_{1,\mathrm{ref}} & \cdots & \phi_{n-1,\mathrm{ref}} \end{bmatrix}^T \in \mathbb{R}^{n-1}$, of the robot as the sum of three individual motion components, namely as

$$\phi_{\mathrm{ref}} = \phi_{\mathrm{wave}} + \phi_{\mathrm{adapt}} + \phi_{\mathrm{heading}} \tag{3}$$

where $\phi_{\mathrm{wave}}$ is a *body wave* component that induces propulsive forces on the robot from the environment, $\phi_{\mathrm{adapt}}$ is an *environment adaptation* component that adjusts the body shape to the environment, and $\phi_{\mathrm{heading}}$ is a *heading control* component that steers the robot according to a specified reference direction.

*Remark 2:* The *serpenoid curve* motion proposed by Hirose [4], which is the gait pattern employed in the majority of the literature on snake robot locomotion, fits nicely within the framework proposed in (3). This gait pattern is achieved by controlling joint $i$ of the snake robot according to

$$\phi_{i,\mathrm{ref}} = \underbrace{\alpha \sin\left(\omega t + (i-1)\,\delta\right)}_{\phi_{\mathrm{wave}}} + \underbrace{\phi_o}_{\phi_{\mathrm{heading}}} \tag{4}$$

where the sinus term constitutes the body wave component, $\phi_{\mathrm{wave}}$, and $\phi_o$, which is an angular offset used to control the direction of the motion, constitutes the heading component, $\phi_{\mathrm{heading}}$. The gait pattern does not involve adaptation of the body shape to the environment, which means that $\phi_{\mathrm{adapt}} = \mathbf{0}$.

## IV. Straight Line Path Following Control Based on Obstacle-Aided Locomotion

In this section, we employ the controller framework presented in Section III to propose a straight line path following
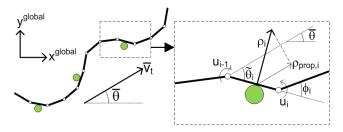


Fig. 2. Illustration of the heading of the robot, $\overline{\theta}$, the heading-adjusted link angle, $\widetilde{\theta}_i$, and the propulsive component, $\rho_{\mathrm{prop},i}$, of the measured contact force on link $i$.

controller for snake robots in environments with obstacles.

### A. Notation

The following notation is illustrated in Fig. 2 and will be employed to specify the control law.

*Definition 3:* **The heading of the robot.**
The *heading* (or *orientation*) of the snake robot is denoted by $\overline{\theta}$ and is defined as the average of the link angles, i.e. as

$$\overline{\theta} = \frac{1}{n} \sum_{i=1}^{n} \theta_i \tag{5}$$

*Definition 4:* **The heading-adjusted link angle.**
The *heading-adjusted* angle of link $i \in \{1, \cdots, n\}$ is denoted by $\widetilde{\theta}_i$ and is given as the angle of link $i$ with respect to the current heading $\overline{\theta}$, i.e. as

$$\widetilde{\theta}_i = \theta_i - \overline{\theta} \tag{6}$$

*Definition 5:* **Forward velocity.**
The *forward velocity* (tangential velocity) of the snake robot is denoted by $\overline{v}_t$ and is defined as the component of the CM velocity $\dot{p}$ along the current heading $\overline{\theta}$, i.e. as

$$\overline{v}_t = \dot{p}_x \cos\overline{\theta} + \dot{p}_y \sin\overline{\theta} \tag{7}$$

*Definition 6:* **Measured contact force.**
The *measured* contact force on link $i \in \{1, \cdots, n\}$ is denoted by $\rho_i \in \mathbb{R}$ and is given as the component of the constraint force vector $\boldsymbol{f}_{c,i}$ along the $y$ axis of the local frame of link $i$ (see illustration of the local link frame in Fig. 1(a)), i.e. as

$$\rho_i = \begin{bmatrix} -\sin\theta_i & \cos\theta_i \end{bmatrix} \boldsymbol{f}_{c,i} \tag{8}$$

*Definition 7:* **Propulsive component.**
The *propulsive component* of the contact force on link $i \in \{1, \cdots, n\}$ is denoted by $\rho_{\mathrm{prop},i} \in \mathbb{R}$ and is given as the component of the constraint force vector $\boldsymbol{f}_{c,i}$ along the current heading $\overline{\theta}$, i.e. as

$$\rho_{\mathrm{prop},i} = -\rho_i \sin\widetilde{\theta}_i \tag{9}$$

The joint angles and the measured contact forces will be assembled in the vectors $\phi = \begin{bmatrix} \phi_1 & \cdots & \phi_{n-1} \end{bmatrix}^T \in \mathbb{R}^{n-1}$ and $\boldsymbol{\rho} = \begin{bmatrix} \rho_1 \cdots & \rho_n \end{bmatrix}^T \in \mathbb{R}^n$, respectively.

## B. Control objective

The control objective is to steer the snake robot so that it converges to and subsequently tracks a straight path while maintaining a heading which is parallell to the path. To this end, we define the global coordinate system so that the global $x$ axis is aligned with the desired straight path. The position of the snake robot along the global $y$ axis, $p_y$, is then the shortest distance from the robot to the desired path (i.e. the cross-track error) and the heading of the snake robot, $\bar{\theta}$, is the angle that the robot forms with the desired path. Since the obstacle environment is unknown and potentially challenging, it makes sense to focus all the control efforts on converging to the path and subsequently progressing along the path at some nonzero forward velocity, $\bar{v}_t > 0$. The authors consider it less important to accurately control the forward velocity of the robot.

From the above discussion, the control problem is to design a feedback control law for the joint torques $\boldsymbol{u} \in \mathbb{R}^{n-1}$ such that the following control objectives are reached:

$$\lim_{t\to\infty} p_y(t) = 0 \tag{10}$$

$$\lim_{t\to\infty} \bar{\theta}(t) = 0 \tag{11}$$

$$\bar{v}_t(t) > 0 \tag{12}$$

The idea behind the controller is to use the body wave component $\phi_{\text{wave}}$ and the adaptation component $\phi_{\text{adapt}}$ to achieve control objective (12), and simultaneously use the heading component $\phi_{\text{heading}}$ to achieve control objectives (10) and (11). The path following controller is based on the following assumption:

*Assumption 8:* The control system has access to measurements of the cross-track error $p_y$, the joint angles $\phi$, the joint angle velocities $\dot{\phi}$, the contact forces $\boldsymbol{\rho}$, and at least one of the absolute link angles $\theta_i$ for some $i \in \{1, \cdots, n\}$.

Note that the remaining link angles, and thereby also the heading $\bar{\theta}$, can be calculated from $\phi$ and $\theta_i$.

*Remark 9:* The assumption that the snake robot can sense external contact forces is a realistic assumption. For instance, the authors have previously proposed a snake robot which demonstrates this capability [15].

## C. The body wave component

We begin by presenting the body wave component $\phi_{\text{wave}}$ of the joint reference angles. The hybrid controller in [10] produces body waves through a leader-follower approach, where the angle of the foremost (head) joint is propagated backwards along the snake body at a constant velocity and used as the reference angle for all subsequent joints. The drawback of this approach is that it relies on the assumption that the snake robot moves forward with the same speed as the head angle propagates backward.

In the present work, we therefore propose to employ a *predecessor-follower* scheme, where each joint follows the angle of the preceding joint ahead of itself with a specified time delay $\Delta t$. The angle of joint $i$ is always a suitable reference angle for joint $i-1$ since the current shape of the snake robot always represents a feasible reference trajectory. The resulting reference angle of joint $i \in \{1, \cdots, n-2\}$ in this predecessor-follower scheme can be written

$$\phi_{\text{wave},i}(t) = \phi_{i+1}(t - \Delta t) \tag{13}$$

In order to produce body wave motion, we introduce a sinusoidal reference angle for the heading-adjusted angle of the head link, $\widetilde{\theta}_n$, given by

$$\widetilde{\theta}_{n,\text{ref}}(t) = \alpha \sin(\omega t) \tag{14}$$

where $\alpha$ and $\omega$ are the amplitude and angular frequency, respectively, of the sinusoidal motion. Since the head joint (i.e. joint $n-1$) is at the front of the snake robot, the rotation of the head joint mainly affects the angle of the head link (i.e. link $n$) and not the angle of the subsequent links. From the relation $\phi_{n-1} = \widetilde{\theta}_{n-1} - \widetilde{\theta}_n$, we can therefore track the head link reference angle in (14) by controlling the head joint according to the reference

$$\phi_{\text{wave},n-1}(t) = \widetilde{\theta}_{n-1} - \alpha \sin(\omega t) \tag{15}$$

From (13) and (15), we can now write the complete body wave component $\phi_{\text{wave}}$ in matrix form as

$$\boldsymbol{\phi}_{\text{wave}} = \boldsymbol{S}_{\text{head}} \left( \widetilde{\theta}_{n-1} - \alpha \sin(\omega t) \right) + \boldsymbol{S}_{\text{joints}} \boldsymbol{\phi}(t - \Delta t) \tag{16}$$

where $\boldsymbol{\phi}(t - \Delta t)$ are the measured joint angles at time $t - \Delta t$ and where $\boldsymbol{S}_{\text{head}}$ and $\boldsymbol{S}_{\text{joints}}$ are, respectively, a selection vector and a selection matrix defined as

$$\boldsymbol{S}_{\text{head}} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \in \mathbb{R}^{N-1} \tag{17}$$

$$\boldsymbol{S}_{\text{joints}} = \begin{bmatrix} 0 & 1 & 0 & & & \\ & 0 & 1 & 0 & & \\ & & \ddots & \ddots & & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{(N-1)\times(N-1)} \tag{18}$$

## D. The environment adaptation component

The environment adaptation component $\phi_{\text{adapt}}$ is based on the jam resolution principle from [10]. However, whereas a complex hybrid formulation is employed in [10], the jam resolution motion in this paper is specified in terms of simple continuous equations. The idea behind the adaptation strategy is to rotate the links affected by contact forces so that the propulsive component of each contact force increases. Since the propulsive components of the contact forces are what propel the snake robot forward, we conjecture that rotating the contacted links to increase the total propulsive force will adapt the body shape to the environment in a way that maintains or increases the propulsion of the robot. Note that the adaptation strategy only aims at satisfying control objective (12), i.e. propelling the snake robot forward along its current heading direction.

The change of the propulsive force on link $i$ due to a change of the link angle is found by differentiating (9) with respect to $\widetilde{\theta}_i$:

$$\frac{\partial \rho_{\text{prop},i}}{\partial \widetilde{\theta}_i} = -\rho_i \cos \widetilde{\theta}_i \tag{19}$$

During adaptation, we choose to rotate links with a high propulsive force gradient with respect to the link angle, which suggests that link $i$ is rotated according to

$$\triangle \widetilde{\theta}_{i,\text{ref}} = k_\rho \frac{\partial \rho_{\text{prop},i}}{\partial \widetilde{\theta}_i} = -k_\rho \rho_i \cos \widetilde{\theta}_i \tag{20}$$

where $k_\rho > 0$ is a controller gain. We choose to change only the angle of link $i$ while leaving the angle of link $i-1$ and

$i+1$ unchanged. This means that $\triangle\widetilde{\theta}_{i-1,\mathrm{ref}} = \triangle\widetilde{\theta}_{i+1,\mathrm{ref}} = 0$. From the relation $\phi_i = \widetilde{\theta}_i - \widetilde{\theta}_{i+1}$, we may now write the desired change of the joint angles at each side of link $i$ as

$$\begin{aligned}\triangle\phi_{i-1,\mathrm{ref}} = \triangle\widetilde{\theta}_{i-1,\mathrm{ref}} - \triangle\widetilde{\theta}_{i,\mathrm{ref}} = k_\rho\rho_i\cos\widetilde{\theta}_i\\\triangle\phi_{i,\mathrm{ref}} = \triangle\widetilde{\theta}_{i,\mathrm{ref}} - \triangle\widetilde{\theta}_{i+1,\mathrm{ref}} = -k_\rho\rho_i\cos\widetilde{\theta}_i\end{aligned} \quad (21)$$

This means that the desired change of the angle of joint $i \in \{1, \cdots, n-1\}$ in the environment adaptation component $\phi_{\mathrm{adapt}}$ is given by

$$\phi_{\mathrm{adapt},i} = -k_\rho\left(\rho_i\cos\widetilde{\theta}_i - \rho_{i+1}\cos\widetilde{\theta}_{i+1}\right) \quad (22)$$

The complete environment adaptation component $\phi_{\mathrm{adapt}}$ can thereby be written in matrix form as

$$\phi_{\mathrm{adapt}} = -k_\rho\boldsymbol{D}\,\mathrm{diag}(\boldsymbol{\rho})\cos\widetilde{\boldsymbol{\theta}} \quad (23)$$

where $\mathrm{diag}(\cdot)$ produces a diagonal matrix with the elements of its argument along its diagonal, and where

$$\cos\widetilde{\boldsymbol{\theta}} = \begin{bmatrix}\cos\widetilde{\theta}_1 \cdots & \cos\widetilde{\theta}_n\end{bmatrix}^T \in \mathbb{R}^n \quad (24)$$

$$\boldsymbol{D} = \begin{bmatrix}1 & -1 & & \\ & \cdot & \cdot & \\ & & \cdot & \cdot \\ & & & 1 & -1\end{bmatrix} \in \mathbb{R}^{(n-1)\times n} \quad (25)$$

### E. The heading control component

The heading control component $\phi_{\mathrm{heading}}$ of the joint reference angles is similar to a guidance law presented in [11], which considers path following control of snake robots on flat surfaces without obstacles. In order to steer the snake robot towards the desired straight path, we employ the Line-of-Sight (LOS) guidance law

$$\overline{\theta}_{\mathrm{ref}} = -\arctan\left(\frac{p_y}{\Delta}\right) \quad (26)$$

where $p_y$ is the cross-track error and $\Delta > 0$ is a design parameter referred to as the *look-ahead distance* that influences the rate of convergence to the desired path. As illustrated in Fig. 3, the LOS angle $\overline{\theta}_{\mathrm{ref}}$ corresponds to the orientation of the snake robot when it is headed towards the point located a distance $\Delta$ ahead of itself along the desired path. To steer the heading $\overline{\theta}$ according to the LOS angle in (26), we offset the reference angle of the head joint according to

$$\phi_{\mathrm{heading},n-1} = k_\theta\left(\overline{\theta} - \overline{\theta}_{\mathrm{ref}}\right) \quad (27)$$

where $k_\theta > 0$ is a controller gain. Using (17), the heading component can be written in matrix form as

$$\phi_{\mathrm{heading}} = \boldsymbol{S}_{\mathrm{head}}k_\theta\left(\overline{\theta} - \overline{\theta}_{\mathrm{ref}}\right) \quad (28)$$

### F. Low-level joint angle controller

In order to make the joint angles $\phi$ track the reference angles given by $\phi_{\mathrm{ref}}$, we set the joint actuator torques $\boldsymbol{u}$ according to the PD-controller

$$\boldsymbol{u} = k_P\left(\phi_{\mathrm{ref}} - \phi\right) + k_D\left(\dot{\phi}_{\mathrm{ref}} - \dot{\phi}\right) \quad (29)$$

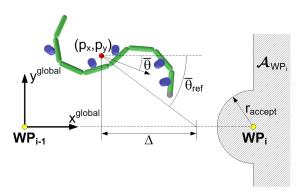where $k_P > 0$ and $k_D > 0$ are controller gains.



Fig. 3. Straight line path following control of the snake robot combined with waypoint guidance.

### G. Summary of the path following controller

The complete straight line path following controller is now summarized. In accordance with the general controller framework defined in (3), we conjecture that control objectives (10), (11), and (12) are achieved by employing the PD-controller in (29) to control the joint angles of the snake robot according to

$$\phi_{\mathrm{ref}} = \phi_{\mathrm{wave}} + \phi_{\mathrm{adapt}} + \phi_{\mathrm{heading}} \quad (30)$$

where

$$\phi_{\mathrm{wave}} = \boldsymbol{S}_{\mathrm{head}}\left(\widetilde{\theta}_{n-1} - \alpha\sin(\omega t)\right) + \boldsymbol{S}_{\mathrm{joints}}\phi(t - \Delta t) \quad (31)$$

$$\phi_{\mathrm{adapt}} = -k_\rho\boldsymbol{D}\,\mathrm{diag}(\boldsymbol{\rho})\cos\widetilde{\boldsymbol{\theta}} \quad (32)$$

$$\phi_{\mathrm{heading}} = \boldsymbol{S}_{\mathrm{head}}k_\theta\left(\overline{\theta} - \overline{\theta}_{\mathrm{ref}}\right) \quad (33)$$

*Remark 10:* Due to the complexity of the snake robot model in (2), we are currently unable to provide a formal proof of the achievement of objectives (10), (11), and (12) with the proposed controller. It is probably not possible to develop such a proof solely based on the model and the control strategy of the robot since it is difficult, if not impossible, to analytically predict the interaction between the robot and the obstacles in advance. However, it may be possible to develop logical arguments regarding the achievement of the control objectives by making assumptions regarding the obstacle interactions. Nevertheless, the controller performed well in the simulation study presented in Section VI.

## V. WAYPOINT GUIDANCE CONTROL

In this section, we combine the path following controller proposed in (30) with a guidance strategy presented in [12] for steering the snake robot between a set of reference locations, or *waypoints*. The waypoint guidance strategy, which was employed in [12] to steer a snake robot on a flat surface without obstacles, is included here to show how we can achieve obstacle-aided locomotion along arbitrary paths given by waypoints interconnected by straight lines.

As explained in [12], the reason for specifying the path of the robot in terms of waypoints is that future applications of snake robots will generally involve bringing sensors and/or tools to a single or several specified target location(s). In these situations, the exact path taken by the robot as it moves towards the target(s) is generally of less interest as long as

the robot reaches the target(s) within a reasonable amount of time. Specifying the motion of a snake robot in terms of waypoints supports this target-oriented control approach.

There are $k$ waypoints and the $i$th waypoint is denoted by $WP_i$, where $i \in \{1, \cdots, k\}$. As illustrated in Fig. 3, we interconnect the waypoints by straight lines and employ the controller in (30) to steer the robot towards the straight line leading to the next waypoint. The next waypoint is activated as soon as the position of the robot enters inside an *acceptance region* consisting of an *acceptance circle* (with radius $r_{\text{accept}}$) centered in the current waypoint and also the right half plane of a coordinate system with origo in the current waypoint and $x$ axis pointing away from the previous waypoint (see illustration in Fig. 3). The acceptance region of $WP_i$ is denoted by $\mathcal{A}_{WP_i}$. The goal of the path following controller (30) is to steer the robot into the acceptance circle of the current waypoint. However, in situations where the obstacle environment prevents the robot from entering inside the acceptance circle, the robot will still proceed towards the next waypoint whenever the position enters inside the right half plane contained in the acceptance region.

The above definitions are formalized in [12], where the waypoint guidance strategy is stated as follows:

*Algorithm 11:* **The waypoint guidance strategy.**
1) Define the initial position of the snake robot as $WP_0$.
2) Repeat for all $i \in \{0, \cdots, k-1\}$:
   a) Move the origin of the global frame to $WP_i$ and orient the global $x$ axis towards $WP_{i+1}$.
   b) Conduct path following according to (30) until $(p_x, p_y) \in \mathcal{A}_{WP_{i+1}}$.

## VI. SIMULATION STUDY

This section presents simulation results that show how the path following controller in (30) performs in combination with the guidance strategy in Algorithm 11.

*Remark 12:* In this section, we simulate path following control between five waypoints in an obstacle environment. Even though we only simulate a single scenario, the path to be followed changes each time a waypoint is reached, which essentially means that we evaluate the path following controller from five different sets of initial conditions. Moreover, the simulation is carried out both with and without environment adaptation.

### A. Simulation parameters

The model of the snake robot (2) and the guidance strategy in Algorithm 11 were implemented in *Matlab R2008b* on a laptop running *Windows XP*. The continuous dynamics of the model were calculated with the *ode45* solver in Matlab with a relative and absolute error tolerance of $10^{-3}$.

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 recently developed by the authors. Circular obstacles measuring 10 cm in diameter were placed in a random fashion in the environment of the snake robot. The ground and obstacle friction coefficients were $\mu = 0.3$ and $\mu_o = 0.25$, respectively. The initial link angles and position of the snake robot were $\boldsymbol{\theta}(0) = [-30, -10, 30, 60, 40, 0, -40, -60, -30, 0]^T$ [deg] and $\boldsymbol{p}(0) = [0, 0]^T$, respectively.

We defined $k = 5$ waypoints with global frame coordinates $(2.5, 0)$, $(2.5, 1)$, $(0, 1)$, $(1, 2)$, and $(3, 2)$, respectively. The radius of the acceptance circle enclosing each waypoint was $r_{\text{accept}} = 0.5$ m. The remaining controller parameters were $\Delta = 0.7$ m, $k_\theta = 1.3$, $k_\rho = 0.02$, $\Delta t = 0.7$ s, $\alpha = 60°$, $\omega = 40°$/s, $k_P = 20$, and $k_D = 5$. In order to prevent the measured contact forces in $\phi_{\text{adapt}}$ from producing steps in the the joint reference angles $\phi_{\text{ref}}$ in (30), the reference angles were filtered using a critically damped 2nd order low-pass filtering reference model with cutoff frequency at 0.75 Hz (see e.g. Chapter 5 in [16]). This filter also provided the derivative of $\phi_{\text{ref}}$ with respect to time, which is needed by the PD-controller in (29).

### B. Simulation results

To show the importance of environment adaptation, the waypoint guidance strategy was first simulated *without* adaptation ($\phi_{\text{adapt}} = \mathbf{0}$). The path of the center link of the snake robot (link 5) is shown in blue in Fig. 4(a), where black squares indicate the waypoints, the dotted black lines indicate the straight paths between the waypoints, and where the shape and position of the robot are shown in green at $t = 0$ s, $t = 65$ s, and $t = 125$ s, respectively. Furthermore, Fig. 4(b)-(c) show the forward velocity, $\bar{v}_t$, and the obstacle constraint force on link 5, $\rho_5$, respectively. The vertical dashed lines in the plots indicate time instants where the guidance strategy switched to the next waypoint. We see from Fig. 4(a) that the robot managed to reach the acceptance region of the two first waypoints. However, the motion was jammed about halfway to the third waypoint, as can be seen from Fig. 4(b), which shows that the forward velocity varied around zero after about 110 s. Note that there is a slight overlap between the path of link 5 and some of the obstacles. This is a consequence of modelling obstacle contact solely by a unilateral force on the contacted link (as explained in Section II), which means that there is nothing preventing the foremost link (the head) of the snake robot from penetrating an obstacle head-on along its tangential direction.

The same plots for the case where environment adaptation was present ($\phi_{\text{adapt}}$ was set according to (32)) are shown in Fig. 5 and Fig. 6. In addition to the forward velocity and the obstacle constraint forces, Fig. 6 also shows the cross-track error, $p_y$, and the heading angle, $\bar{\theta}$. With environment adaptation, the propulsion of the robot was maintained through all the waypoints. As seen from Fig. 6(c), the forward velocity varied between 5 - 10 cm/s, which suggests that control objective (12) was satisfied. Fig. 6(a)-(b) show that the cross-track error and the heading angle had an oscillatory behaviour around zero after each waypoint switch, which suggests that control objectives (10) and (11) were eventually achieved in *average* over each cycle of the gait pattern.

It is interesting to note that, in addition to improving the propulsion of the robot, the environment adaptation strategy reduces the constraint forces on the robot significantly, which is seen by comparing Fig. 6(d) with Fig. 4(c). This is also expected since the environment contact forces opposing the motion will naturally be larger when the motion is performed without considering the environment.
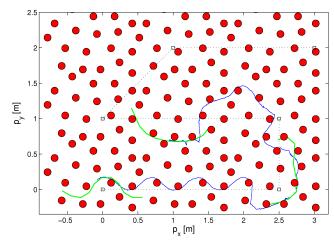
In summary, the proposed controller maintained the propulsion and steered the snake robot to each waypoint in the obstacle environment.
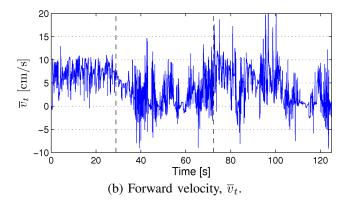
## VII. CONCLUSIONS

This paper has focused on adaptive motion control of snake robots in unstructured environments. The first contribution has been a general controller framework which allows the motion of a snake robot to be specified in terms of a *body wave* component, an *environment adaptation* component, and a *heading control* component. The second contribution has been a control law for straight line path following control in environments with obstacles, which was specified within the proposed controller framework. Simulation results have been presented where the path following controller, in combination with a waypoint guidance strategy, successfully steered the snake robot between waypoints in an obstacle environment. In future work, the authors will present an experimental investigation of the control strategy proposed in this paper.

## REFERENCES

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

[2] B. Moon and C. Gans, "Kinematics, muscular activity and propulsion in gopher snakes," *Journal of Experimental Biology*, vol. 201, pp. 2669–2684, 1998.

[3] 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.

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

[5] 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.

[6] Z. Y. Bayraktaroglu, "Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 591–602, 2008.

[7] A. M. Andruska and K. S. Peterson, "Control of a snake-like robot in an elastically deformable channel," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 2, pp. 219–227, april 2008.

[8] 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.

[9] H. Date and Y. Takita, "Adaptive locomotion of a snake like robot based on curvature derivatives," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, San Diego, CA, USA, Oct-Nov 2007, pp. 3554–3559.

[10] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Hybrid modelling and control of obstacle-aided snake robot locomotion," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 781–799, Oct 2010.

[11] P. Liljebäck, I. U. Haugstuen, and K. Y. Pettersen, "Path following control of planar snake robots using a cascaded approach," in *Proc. IEEE Conf. Decision and Control*, Atlanta, GA, USA, 2010, pp. 1969–1976.

[12] P. Liljebäck and K. Y. Pettersen, "Waypoint guidance control of snake robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, 2011, accepted.

[13] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Experimental investigation of obstacle-aided locomotion with a snake robot," *IEEE Trans. Robotics*, 2010, conditionally accepted.

[14] ——, "Controllability and stability analysis of planar snake robot locomotion," *IEEE Trans. Automatic Control*, 2011, to appear.

[15] P. Liljebäck, K. Y. Pettersen, and Ø. Stavdahl, "A snake robot with a contact force measurement system for obstacle-aided locomotion," in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 683–690.

[16] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.

(a) The path of the center link (link 5).



(b) Forward velocity, $\overline{v}_t$.
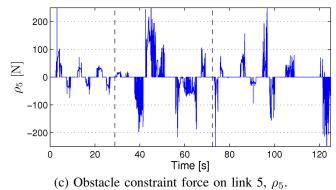


(c) Obstacle constraint force on link 5, $\rho_5$.

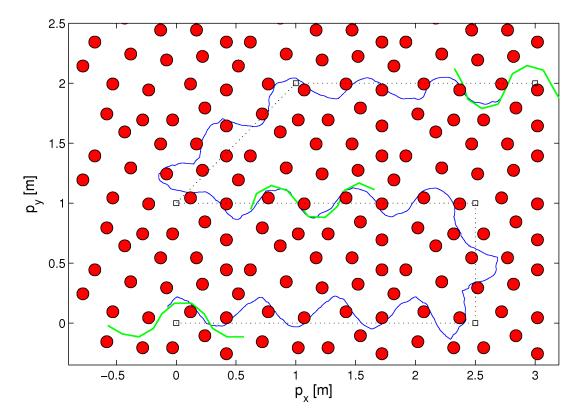Fig. 4. Simulation of the waypoint guidance strategy without environment adaptation.

Fig. 5. The path of the center link (link 5) of the snake robot during waypoint guidance with environment adaptation.



(a) Cross-track error, $p_y$.

(b) Heading angle, $\bar{\theta}$.

(c) Forward velocity, $\bar{v}_t$.

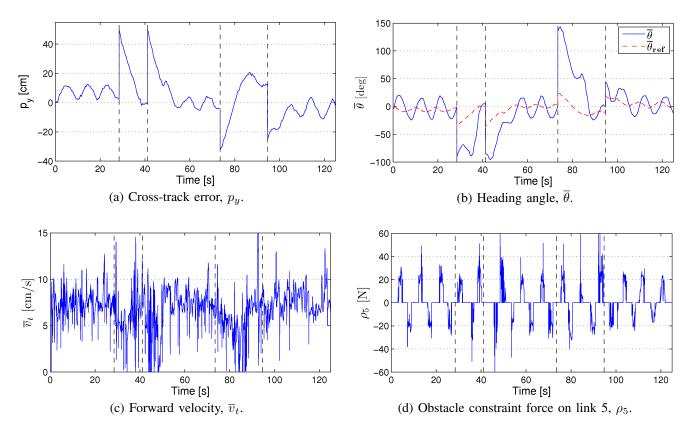(d) Obstacle constraint force on link 5, $\rho_5$.

Fig. 6. Simulation of the waypoint guidance strategy with environment adaptation.