Waypoint guidance control of snake robots

Pål Liljebäck and Kristin Y. Pettersen

Abstract— **This paper considers path following control of snake robots and has two contributions. The first contribution is a description of how a straight line path following controller previously proposed by the authors can be extended to path following of general curved paths. The second contribution of this paper is a waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines. The waypoint guidance strategy builds on the straight line path following controller previously proposed by the authors. The paper presents simulation results that illustrate the performance of the proposed guidance strategy.**

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.

Research on snake locomotion has been conducted for several decades. Gray [1] performed empirical and analytical studies of snake locomotion already in the 1940s, and Hirose [2] studied biological snakes and developed mathematical relationships characterizing their motion, such as the *serpenoid curve*. The emphasis in literature so far has mainly been on achieving forward and turning locomotion. The next step will be not only to achieve forward locomotion, but also to make the snake robot follow a desired path. The research results reported on this control problem are still limited. The work in [3] proposes a path following controller for a wheeled snake robot (i.e. with nonholonomic constraints on the links) aimed at making the head track a reference trajectory. A similar approach is presented in [4], where a measure of dynamic manipulability that takes the constraint forces on the wheels into account is employed in the formulation of a path following controller. The work in [5] considers trajectory tracking of snake robots where some, but not all, of the links are assumed to be wheeled. This gives the system more degrees of freedom and is utilized to follow a trajectory while simultaneously maintaining a high manipulability. The authors employed *Poincaré maps* in [6] to study the stability properties of a wheelless snake robot during motion along a straight path, and employed cascaded systems theory in [7] to propose a path following controller that $\mathcal{K}$-exponentially stabilizes a snake robot to a straight path.

Research on robotic fish and eel-like mechanisms is relevant to research on snake robots since these mechanisms are very similar. The works in [8]–[10] synthesize gaits for translational and rotational motion of various fish-like mechanisms and propose controllers for tracking straight and curved trajectories. However, an analysis that formally proves convergence to the desired path still remains.

This paper extends previous work by the authors on straight line path following presented in [7] and has two contributions. The first contribution is a description of how the straight line path following controller can be extended to path following of general curved paths by employing an approach previously presented in [11] in the context of path following control of marine vessels.

The second contribution of this paper is a waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines. The waypoint guidance strategy builds on the straight line path following controller proposed in [7] and represents an operator-friendly framework for motion control of snake robots. The paper presents simulation results that illustrate the performance of the proposed guidance strategy.

The paper is organized as follows. Section II presents a complex model of a snake robot that is included for simulation purposes, while Section III presents a simplified model of a snake robot that the path following controller is derived from. The controller for tracking straight paths is presented in Section IV, while Section V describes how the straight line path following controller can be extended to path following of general curved paths. The waypoint guidance strategy is proposed in Section VI and simulation results are presented in Section VII. Finally, Section VIII presents concluding remarks.

II. A complex model of a planar snake robot

This section summarizes a complex model of a planar snake robot previously presented in [6]. We will use this model to simulate the motion of the snake robot in Section VII.

We consider a planar snake robot consisting of N links of length l interconnected by $N-1$ active joints. The kinematics of the robot is defined in terms of the symbols illustrated in Fig. 1. All N links have the same mass m and moment of inertia J. The total mass of the 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. 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 $\boldsymbol{p} = (p_x, p_y) \in \mathbb{R}^2$. 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 $i+1$ (i.e. the angle of joint i) is given by $\phi_i = \theta_i - \theta_{i+1}$. Each

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 is with the Department of Engineering Cybernetics at the Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. E-mail: Kristin.Y.Pettersen@itk.ntnu.no

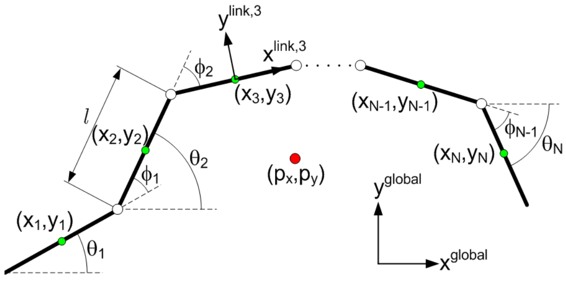


Fig. 1. Kinematic parameters of the snake robot.

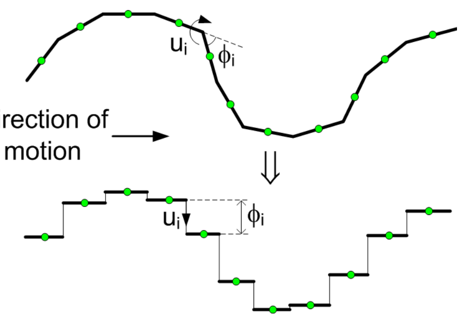


Fig. 2. The revolute joints of the snake robot are modelled as prismatic joints that displace the CM of each link transversal to the direction of motion.

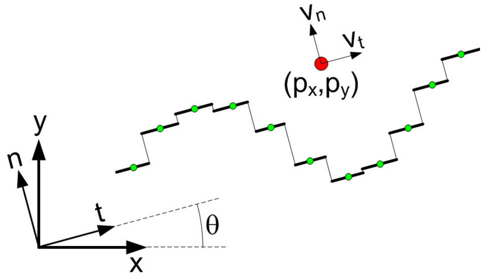


Fig. 3. Illustration of the two coordinate frames employed in the model. The global x-y frame is fixed. The t-n frame is always aligned with the snake robot.

link is subjected to an anisotropic viscous ground friction force. Since the friction is anisotropic, a link has two viscous friction coefficients, c_t and c_n, describing the friction force in the tangential and normal direction of the link, respectively.

It is shown in [6] that the equations of motion of the snake robot in terms of the joint angles, $\boldsymbol{\phi} \in \mathbb{R}^{N-1}$, the absolute angle of the head link, $\theta_N \in \mathbb{R}$, the position of the CM of the snake robot, $\boldsymbol{p} = (p_x, p_y) \in \mathbb{R}^2$, and the joint torques, $\boldsymbol{u} \in \mathbb{R}^{N-1}$, can be written as

$$\begin{gathered} \ddot{\boldsymbol{\phi}} = \boldsymbol{u}, \quad \ddot{\theta}_N = g(\boldsymbol{\phi}, \theta_N, \dot{\boldsymbol{\phi}}, \dot{\theta}_N, \dot{p}_x, \dot{p}_y, \boldsymbol{u}), \\ Nm\ddot{p}_x = \textstyle\sum_{i=1}^{N} f_{x,i}, \quad Nm\ddot{p}_y = \textstyle\sum_{i=1}^{N} f_{y,i} \end{gathered} \tag{1}$$

where $g(\boldsymbol{\phi}, \theta_N, \dot{\boldsymbol{\phi}}, \dot{\theta}_N, \dot{p}_x, \dot{p}_y, \boldsymbol{u}) \in \mathbb{R}$ is a function of the state vector and the joint torques, and where $f_{x,i}$ and $f_{y,i}$ are the viscous friction force components on link i in the global x and y direction, respectively.

III. A SIMPLIFIED MODEL OF A PLANAR SNAKE ROBOT

This section summarizes a simplified model of a planar snake robot which the straight line path following controller in Section IV is based upon. For a more detailed presentation of the model, the reader is referred to [12].

A. Overview of the model

The idea behind the simplified model is illustrated in Fig. 2 and motivated by an analysis presented in [12], which shows that:

- The forward motion of a planar snake robot is produced by the link velocity components that are *normal* to the forward direction.
- The change in body shape during forward locomotion primarily consists of relative displacements of the CM of the links *normal* to the forward direction of motion.

Based on these two properties, the simplified model describes the body shape changes of a snake robot as *linear displacements* of the links with respect to each other instead of rotational displacements. The linear displacements occur *normal* to the forward direction of motion and produce friction forces that propel the robot forward. This essentially means that the revolute joints of the snake robot are modelled as prismatic (translational) joints and that the rotational link motion during body shape changes is disregarded. However, the model still captures the *effect* of the rotational link motion during body shape changes, which is a linear displacement of the links normal to the forward direction of motion. Note that the relative link displacements transversal to the direction of motion will *not* dominate over the relative link displacements tangential to the direction of motion when the amplitudes of the link angles become large. The simplified model is therefore a valid description of snake robot locomotion only as long as the link angles are limited.

B. Equations of motion

The snake robot has N links of length l and mass m interconnected by $N-1$ prismatic joints. As seen in Fig. 4, the normal direction distance from link i to link $i+1$ is denoted by ϕ_i and represents the coordinate of joint i. The global frame orientation, $\theta \in \mathbb{R}$, and the CM position, $(p_x, p_y) \in \mathbb{R}^2$, of the snake robot are illustrated in Fig. 3. Each link is influenced by a ground friction force and constraint forces that hold the joints together. A model of these forces is presented in [12], where it is also shown that

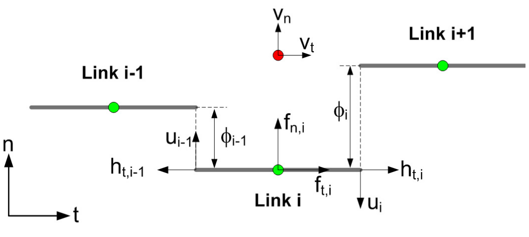


Fig. 4. Symbols characterizing the kinematics and dynamics of the snake robot.

the complete model of the snake robot can be written as

$$\dot{\boldsymbol{\phi}} = \boldsymbol{v}_\phi \tag{2a}$$

$$\dot{\theta} = v_\theta \tag{2b}$$

$$\dot{p}_x = v_t \cos\theta - v_n \sin\theta \tag{2c}$$

$$\dot{p}_y = v_t \sin\theta + v_n \cos\theta \tag{2d}$$

$$\dot{\boldsymbol{v}}_\phi = -\frac{c_1}{m}\boldsymbol{v}_\phi + \frac{c_2}{m} v_t \boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi} + \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} \tag{2e}$$

$$\dot{v}_\theta = -c_3 v_\theta + \frac{c_4}{N-1} v_t \overline{\boldsymbol{e}}^T \boldsymbol{\phi} \tag{2f}$$

$$\dot{v}_t = -\frac{c_1}{m} v_t + \frac{2c_2}{Nm} v_n \overline{\boldsymbol{e}}^T\boldsymbol{\phi} - \frac{c_2}{Nm}\boldsymbol{\phi}^T\boldsymbol{A}\overline{\boldsymbol{D}}\boldsymbol{v}_\phi \tag{2g}$$

$$\dot{v}_n = -\frac{c_1}{m} v_n + \frac{2c_2}{Nm} v_t \overline{\boldsymbol{e}}^T\boldsymbol{\phi} \tag{2h}$$

where $\boldsymbol{\phi} = \left(\phi_1, \cdots, \phi_{N-1}\right) \in \mathbb{R}^{N-1}$ are the joint coordinates, $\theta \in \mathbb{R}$ is the absolute orientation, $(p_x, p_y) \in \mathbb{R}^2$ is the CM position, $\boldsymbol{v}_\phi = \dot{\boldsymbol{\phi}} \in \mathbb{R}^{N-1}$ are the joint velocities, $v_\theta = \dot{\theta} \in \mathbb{R}$ is the angular velocity, $(v_t, v_n) \in \mathbb{R}^2$ are the tangential and normal direction velocity of the snake robot, $\boldsymbol{u} \in \mathbb{R}^{N-1}$ are the joint actuator forces, and

$$\overline{\boldsymbol{e}} = \begin{bmatrix}1 & \cdot & \cdot & 1\end{bmatrix}^T \in \mathbb{R}^{N-1},$$

$$\overline{\boldsymbol{D}} = \boldsymbol{D}^T\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1} \in \mathbb{R}^{N\times(N-1)},$$

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

where $\boldsymbol{A} \in \mathbb{R}^{(N-1)\times N}$ and $\boldsymbol{D} \in \mathbb{R}^{(N-1)\times N}$. The parameters c_1, c_2, c_3, and c_4 are positive scalar friction coefficients that characterize the external forces acting on the snake robot. In particular, the coefficient c_1 determines the magnitude of the friction forces resisting the link motion, c_2 determines the magnitude of the induced friction forces that propel the snake robot forward, c_3 determines the friction torque opposing the rotation of the snake robot, while c_4 determines the induced torque that rotates the snake robot. This torque is induced when the forward direction velocity and the average of the joint coordinates are nonzero. The role of each coefficient is explained in more detail in [12].

IV. Path following control along straight lines

This section presents a straight line path following controller for snake robots that has previously been proposed in [7]. The waypoint guidance strategy proposed in Section VI is based on this controller.

A. 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 and the orientation of the snake robot, θ, is the angle that the robot forms with the desired path. The control problem is thereby to design a feedback control law for $\boldsymbol{u} \in \mathbb{R}^{N-1}$ such that the following control objective is reached:

$$\lim_{t\to\infty} p_y(t) = 0 \quad , \quad \lim_{t\to\infty} \theta(t) = 0 \tag{3}$$

During path following with a snake robot, 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. The authors consider it less important to accurately control the forward velocity of the robot. In accordance with the work by the authors in [13], which shows that the forward velocity of a snake robot during lateral undulation oscillates around a positive nonzero average velocity, we choose to base the path following controller on the following assumption:

Assumption 1: The snake robot conducts *lateral undulation* and has a forward velocity which is always nonzero and positive, i.e. $v_t \in [V_{\min}, V_{\max}]\ \forall\ t \geq 0$ where $V_{\max} \geq V_{\min} > 0$.

B. Model transformation

As seen in (2f) and (2h), the joint coordinates $\boldsymbol{\phi}$ are present in the dynamics of both the angular velocity v_θ and the sideways velocity v_n of the snake robot. This complicates the controller design since the body shape changes will affect both the heading and the sideways motion of the robot. Motivated by [14], we therefore remove the effect of $\boldsymbol{\phi}$ on the sideways velocity by the coordinate transformation:

$$\overline{p}_x = p_x + \epsilon\cos\theta \tag{4a}$$

$$\overline{p}_y = p_y + \epsilon\sin\theta \tag{4b}$$

$$\overline{v}_n = v_n + \epsilon v_\theta \tag{4c}$$

where ϵ is a constant parameter defined as

$$\epsilon = -\frac{2\,(N-1)}{Nm}\frac{c_2}{c_4} \tag{5}$$

The coordinate transformation (4) is illustrated to the left in Fig. 5 and can be interpreted as moving the point that determines the position of the snake robot to the location where the body shape changes of the robot generate a pure rotational motion and no sideways force. The new coordinates in (4) transform the model (2) into

$$\dot{\boldsymbol{\phi}} = \boldsymbol{v}_\phi \tag{6a}$$

$$\dot{\theta} = v_\theta \tag{6b}$$

$$\dot{\overline{p}}_y = v_t\sin\theta + \overline{v}_n\cos\theta \tag{6c}$$

$$\dot{\boldsymbol{v}}_\phi = -\frac{c_1}{m}\boldsymbol{v}_\phi + \frac{c_2}{m} v_t \boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi} + \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} \tag{6d}$$

$$\dot{v}_\theta = -c_3 v_\theta + \frac{c_4}{N-1} v_t \overline{\boldsymbol{e}}^T\boldsymbol{\phi} \tag{6e}$$

$$\dot{\overline{v}}_n = X v_\theta + Y\overline{v}_n \tag{6f}$$

where $X = \epsilon\left(\frac{c_1}{m} - c_3\right)$, $Y = -\frac{c_1}{m}$, and $v_t \in [V_{\min}, V_{\max}]$ by Assumption 1.

C. The path following controller

The path following controller of the snake robot consists of two main components. The first component is the gait pattern controller, which propels the snake robot forward according to the gait pattern *lateral undulation* (as stated in Assumption 1). The second component is the heading controller, which steers the snake robot towards and subsequently along the desired path. The two components of the path following controller are now presented.

1) Gait pattern controller: As proposed in [2], lateral undulation is achieved by controlling joint $i \in \{1, \cdots, N-1\}$ of the snake robot according to the sinusoidal reference

$$\phi_{i,\mathrm{ref}} = \alpha \sin\left(\omega t + (i-1)\,\delta\right) + \phi_o \tag{7}$$

where α and ω are the amplitude and frequency, respectively, of the sinusoidal joint motion and δ determines the phase shift between the joints. The parameter ϕ_o is a joint offset coordinate that we will use to control the direction of the locomotion. In order to make the joints track the joint reference coordinates given by (7), we set the actuator forces according to the linearizing control law

$$\boldsymbol{u} = m\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\left(\overline{\boldsymbol{u}} + \frac{c_1}{m}\dot{\boldsymbol{\phi}} - \frac{c_2}{m}v_t\boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi}\right) \tag{8}$$

where $\overline{\boldsymbol{u}} \in \mathbb{R}^{N-1}$ is a new set of control inputs. This control law transforms the joint dynamics (6d) into $\dot{\boldsymbol{v}}_\phi = \ddot{\boldsymbol{\phi}} = \overline{\boldsymbol{u}}$. Subsequently, we choose the new control input $\overline{\boldsymbol{u}}$ as

$$\overline{\boldsymbol{u}} = \ddot{\boldsymbol{\phi}}_{\mathrm{ref}} + k_{v_\phi}\left(\dot{\boldsymbol{\phi}}_{\mathrm{ref}} - \dot{\boldsymbol{\phi}}\right) + k_\phi\left(\boldsymbol{\phi}_{\mathrm{ref}} - \boldsymbol{\phi}\right) \tag{9}$$

where $k_\phi > 0$ and $k_{v_\phi} > 0$ are scalar controller gains and $\boldsymbol{\phi}_{\mathrm{ref}} = \left(\phi_{1,\mathrm{ref}}, \cdots, \phi_{N-1,\mathrm{ref}}\right) \in \mathbb{R}^{N-1}$ are the joint reference coordinates given by (7).

2) Heading controller: In order to steer the snake robot towards the desired straight path, we employ the Line-of-Sight (LOS) guidance law

$$\theta_{\mathrm{ref}} = -\arctan(\frac{\overline{p}_y}{\Delta}) \tag{10}$$

where $\overline{p}_y$ is the cross-track error and $\Delta > 0$ is a design parameter referred to as the *look-ahead distance* that determines the rate of convergence to the desired path. As illustrated to the right in Fig. 5, the LOS angle θ_{ref} corresponds to the orientation of the snake robot when it is headed towards the point located a distance Δ ahead of the snake robot along the desired path. To steer the heading θ according to the LOS angle given by (10), we choose the joint offset ϕ_o as (see [7] for a detailed derivation)

$$\begin{aligned}\phi_o = \tfrac{1}{c_4 v_t}\Big(&\ddot{\theta}_{\mathrm{ref}} + c_3\dot{\theta}_{\mathrm{ref}} - k_\theta(\theta - \theta_{\mathrm{ref}}) \\ &-\tfrac{c_4}{N-1}v_t\sum_{i=1}^{N-1}\alpha\sin(\omega t + (i-1)\,\delta)\Big)\end{aligned} \tag{11}$$

where $k_\theta > 0$ is a scalar controller gain. Note that the inverse of the forward velocity in (11) does not represent a problem since $v_t > 0$ by Assumption 1.

The complete path following controller, whose structure is summarized in Fig. 6, satisfies the following theorem:

Theorem 2: Consider a planar snake robot described by the model (6) and suppose that Assumption 1 is satisfied. If the parameter Δ of the LOS guidance law (10) is chosen such that

$$\Delta > \frac{|X|}{|Y|}\left(1 + \frac{V_{\max}}{V_{\min}}\right) \tag{12}$$

then the path following controller defined by (7), (8), (9), (10), and (11) guarantees that the control objective in (3) is achieved for any set of initial conditions satisfying $v_t \in [V_{\min}, V_{\max}]$.

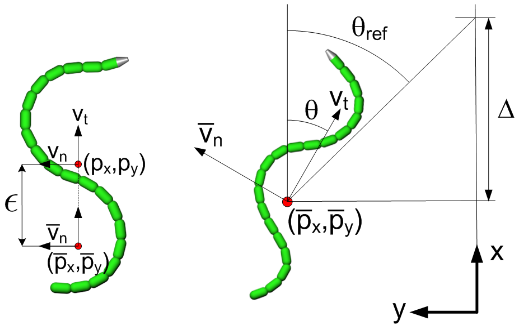


Fig. 5. Left: The coordinate transformation of the snake robot. Right: The Line-of-Sight (LOS) guidance strategy.

Proof: The theorem is proved in [7] using cascaded systems theory and is not included due to space restrictions. ■

Remark 3: As explained in Section III-A, the assumptions underlying the simplified model are only valid as long as the link angles with respect to the forward direction are limited. The stability result in Theorem 2 is therefore claimed only for snake robots conducting lateral undulation with limited link angles.

V. Path following control along curved paths

In this section, we describe how the straight line path following controller presented in the previous section can be extended to path following of general curved paths. A similar controller was presented in [11] for path following control of marine vessels. In this paper, we show how the approach from [11] can be modified for curved path following control of snake robots by taking into account a few differences between the model of the snake robot and the model of the vessel considered in [11], and also adapting the controller development accordingly.

The desired path that the snake robot should follow is a continuously differentiable curve denoted by $\mathcal{C}$ (see Fig. 7). The idea behind the controller is to steer the snake robot towards a *virtual* particle that moves along the path. The distance travelled by the particle along the curve is denoted by s, which means that $\dot{s}$ is the instantaneous speed of the particle along the curve. Furthermore, we define a moving coordinate frame with axes denoted by T and N such that the origin of the frame coincides with the particle and the T axis is always tangential to the curve. This is called a *Serret-Frenet* coordinate frame [15]. As visualized in Fig. 7, the angle of the T axis with respect to the global x axis is denoted by θ_T and the position of the snake robot in the T-N frame is denoted by (p_T, p_N).

Since the goal is to make the snake robot converge to and follow the desired path $\mathcal{C}$, we state the control objective as

$$\lim_{t\to\infty} p_T(t) = 0 \quad , \quad \lim_{t\to\infty} p_N(t) = 0 \tag{13}$$

In order to achieve this control objective, we steer the heading θ of the robot according to the guidance law

$$\theta_{\mathrm{ref}} = \theta_T - \arctan(\frac{\overline{v}_n}{v_t}) - \arctan(\frac{p_N}{\sqrt{\Delta^2 + p_T^2}}) \tag{14}$$

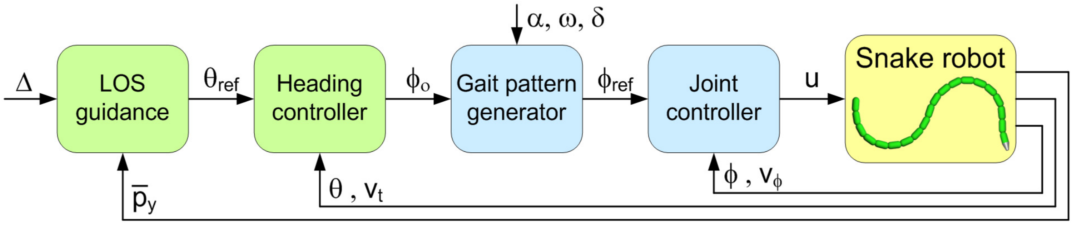


Fig. 6. The structure of the straight line path following controller.

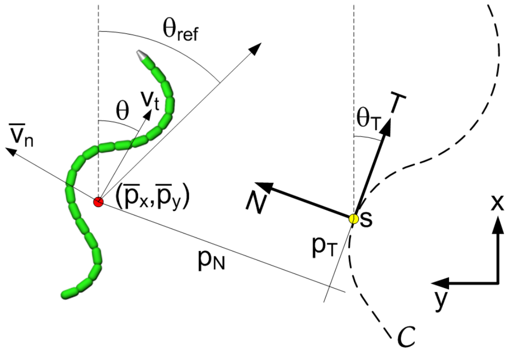


Fig. 7. The guidance strategy for path following of general curved paths.

and update the position of the virtual particle along the curve according to

$$\dot{s} = U \frac{\sqrt{\Delta^2 + p_T^2} + p_T}{\sqrt{\Delta^2 + p_T^2 + p_N^2}} \;,\; U = \sqrt{v_t^2 + \bar{v}_n^2} \tag{15}$$

Furthermore, we assume that the minimum forward velocity of the robot $V_{\min}$, the maximum forward acceleration of the robot $A_{\max} = \max_t |\dot{v}_t|$, the maximum curvature along the path $\kappa_{\max} = \max_s |\kappa(s)|$, and the look-ahead distance Δ satisfy

$$-V_{\min} + \sigma \;<\; X \tag{16a}$$

$$A_{\max} \;<\; \frac{Z_{\min}|Y|}{|X|} V_{\min}^2 \tag{16b}$$

$$\kappa_{\max} \;<\; \frac{1}{6}\left(\frac{Z_{\min}|Y|}{|X|} - \frac{A_{\max}}{V_{\min}^2}\right) \tag{16c}$$

$$\Delta \;>\; \frac{\frac{3}{2}|X|}{Z_{\min}|Y| - |X|\left(6\kappa_{\max} + \frac{A_{\max}}{V_{\min}^2}\right)} \tag{16d}$$

where σ is a positive constant, $Z_{\min} = \nu / (V_{\min} + |X|)$, and ν is a constant chosen such that $0 < \nu < \sigma$. Note that the conditions in (16) are slightly different from the conditions in [11] due to model differences. The above controller satisfies the following theorem:

Theorem 4: Consider a planar snake robot described by the model (6). Suppose that Assumption 1 is satisfied and that the joints of the robot are controlled according to (7), (8), (9), and (11), where θ_{ref} is given by (14) and where s is updated according to (15). Then control objective (13) is achieved for any set of initial conditions satisfying $v_t \in [V_{\min}, V_{\max}]$ if the conditions in (16) are satisfied.

Proof: The proof of this theorem is developed by following similar steps as the proof presented in [11], and is not included here due to space restrictions. ■

VI. WAYPOINT GUIDANCE CONTROL

In this section, we employ the straight line path following controller presented in Section IV in order to propose a guidance strategy for steering a snake robot between a set of reference locations, or *waypoints*, in the environment of the robot. The waypoint guidance strategy proposed in this section represents an operator-friendly framework for motion control of snake robots.

A. Description of the approach

Future applications of snake robots will generally involve motion in challenging and unstructured environments where the aim is to bring 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 in terms of waypoints supports this target-oriented control approach. Waypoint guidance is a commonly used approach for control of e.g. marine surface vessels (see e.g. [16]), but has, to the authors' best knowledge, not been considered for motion control of snake robots.

In accordance with the target-oriented approach discussed above, we choose to interconnect the waypoints by straight lines and employ the path following controller presented in Section IV in order to steer the snake robot towards the straight line leading to the next waypoint. This approach is illustrated in Fig. 8. The reason for considering straight lines instead of curved paths is our long-term goal of also employing the guidance strategy in unstructured environments. The idea of requiring a snake robot to follow a nice and smooth curved path in an unknown and unstructured environment seems unrealistic, while a straight line reference path between each waypoint basically tells the robot to take the shortest possible path to the next waypoint.

A common rule for switching between the waypoints is to proceed towards the next waypoint as soon as the position of the system enters inside an *acceptance circle* enclosing the current waypoint [16]. In the present work, we propose that the acceptance circle is replaced by an *acceptance region* composed of an acceptance circle 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. 8). With this definition, we are guaranteed that the robot will reach the

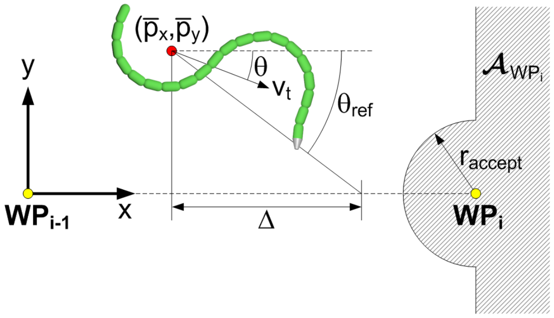


Fig. 8. The waypoint guidance strategy.

acceptance region of the current waypoint no matter how the waypoints are defined. With only acceptance circles enclosing each waypoint, there would be the risk that the robot misses a waypoint which is placed too close to the previous waypoint, which would make the robot proceed indefinitely along the path away from the waypoint. Note that although the acceptance region is infinitely large, the path following controller presented in Section IV guarantees rapid convergence to the straight path between two waypoints.

B. The waypoint guidance strategy

In the following, we formalize the guidance strategy described in the previous subsection.

Definition 5: **Waypoint.**
A *waypoint* is a reference position along the path of the snake robot. There are k waypoints and the ith waypoint is denoted by WP_i, where $i \in \{1, \cdots, k\}$.

Definition 6: **Acceptance region.**
The *acceptance region* of WP_i, denoted $\mathcal{A}_{WP_i}$, is the union of all points inside a circle centered in WP_i with radius r_{accept} and the right half plane of a coordinate system with origo in WP_i and x axis aligned with the vector from WP_{i-1} to WP_i.

Definition 7: **The waypoint guidance problem.**
Given a set of k waypoints $WP_1, \ldots, WP_k$, the *waypoint guidance problem* is the task of steering the position of the snake robot into the acceptance region of each of the waypoints $WP_1, \ldots, WP_k$ in consecutive order.

In accordance with the above definitions and the description in the previous subsection, we now state the proposed waypoint guidance strategy for the snake robot as follows:

Algorithm 8: **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 the controller from Section IV until $(p_x, p_y) \in \mathcal{A}_{WP_{i+1}}$.

The guidance strategy proposed in Algorithm 8 satisfies the following result:

Proposition 9: The waypoint guidance problem presented in Definition 7 is solved by Algorithm 8 for a planar snake robot described by the model (6) under the conditions of Theorem 2.

Proof: Given any waypoint WP_i that the snake robot is crawling towards, where $i \in \{1, \cdots, k\}$, Definition 6 ensures that the desired straight path of the snake robot points into the acceptance region of WP_i. By Theorem 2, the snake robot will eventually reach the desired straight path and progress along the path indefinitely, which means that the position of the snake robot will eventually reach the acceptance region of WP_i. This completes the proof. ■

VII. Simulation Study

This section presents simulation results in order to investigate the performance of the guidance strategy proposed in Algorithm 8. In addition to simulation results from the simplified model (2), we also include simulation results from the complex model (1) to show that the applicability of the guidance strategy does not rely on the simplifications of the simplified model. The two models were implemented in *Matlab R2008b* and the dynamics were calculated using the *ode45* solver in Matlab.

A. Implementation of the guidance strategy with the simplified model

We considered a snake robot with $N = 10$ links of length $l = 0.14$ m, mass $m = 1$ kg, and moment of inertia $J = 0.0016$ kgm^2. These parameters characterize a physical snake robot recently developed by the authors. The initial values of all states of the snake robot were set to zero. Furthermore, we chose the friction coefficients as $c_1 = 0.4$, $c_2 = 2.2$, $c_3 = 0.5$ and $c_4 = 20$.

The radius of the acceptance circle enclosing each waypoint was $r_{\text{accept}} = 0.5$ m. The path following controller was implemented according to (7), (8), (9), (10), and (11), and with the coordinate transformation distance in (5) set to $\epsilon = -19.8$ cm. The controller gains were $k_\phi = 20$, $k_{v_\phi} = 5$, and $k_\theta = 0.06$, and the gait parameters were $\alpha = 0.1$ m, $\omega = 70°$/s, and $\delta = 40°$. We chose the look-ahead distance as $\Delta = 1.4$ m, which corresponds to the length of the snake robot, and conjecture that this value is well above the lower bound given by (12).

Note that the calculation of the control input in (9) and (11) requires the derivative of θ_{ref} and ϕ_o with respect to time. During the simulations, we generated these signals by using a critically damped 3rd order low-pass filtering reference model with cutoff frequency at 0.25 Hz (see e.g. Chapter 5 in [16]). The joint angle offset was saturated according to $\phi_o \in [-0.08 \text{ m}, 0.08 \text{ m}]$ in order to avoid the singularity in (11) at $v_t = 0$.

B. Implementation of the guidance strategy with the complex model

We chose the friction coefficients of the complex model as $c_t = 0.55$ and $c_n = 3$ since previous simulation studies have shown that the two models have similar behaviour with this choice. The guidance strategy was implemented in accordance with the guidance strategy for the simplified model except for the following modifications, which all concern the difference between the shape variables in the simplified model (i.e. the transversal distance between the links) and the complex model (i.e. the joint angles).

We estimated the orientation $\overline{\theta}$ of the robot as the average of the link angles, i.e. as $\overline{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i$, and the forward velocity $\overline{v}_t$ as the component of the CM velocity $\dot{p}$ in the $\overline{\theta}$

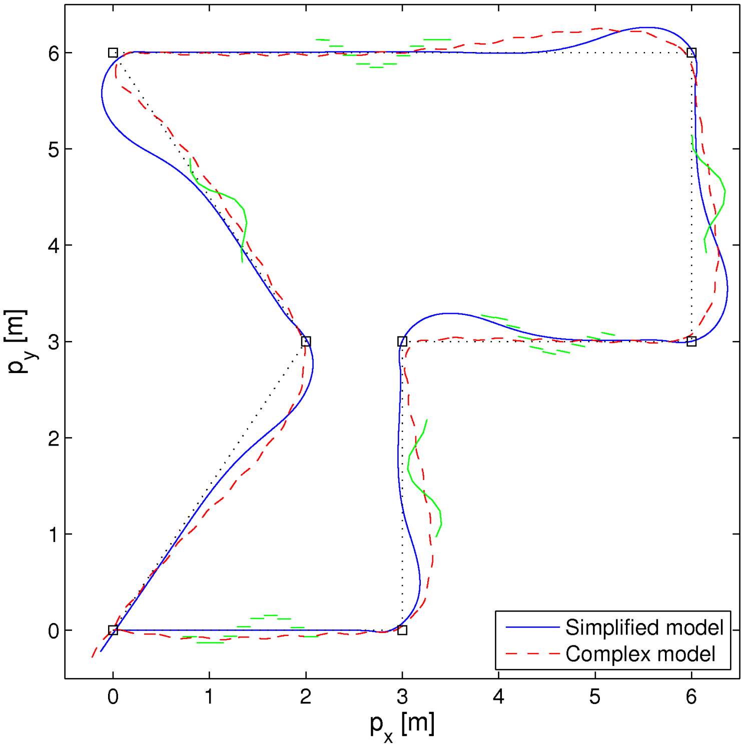


Fig. 9. The path of the CM of the snake robot from the simplified model (solid line) and the complex model (dashed line).

direction. The coordinate transformation distance ϵ in (5) was set to zero (i.e. we measured the cross-track error as $\overline{p}_y = p_y$) since, in contrast to the orientation θ in the simplified model, the orientation $\overline{\theta}$ of a snake robot with revolute joints oscillates during lateral undulation. These oscillations would cause the transformed position, and thereby also the cross-track error, to oscillate transversal to the forward direction of motion defined by $\overline{\theta}$, which would likely degrade the performance of the path following controller.

With the gait parameters employed for the simplified model, it is shown in [12] that $30°/0.1$ m is a suitable scaling factor between the joint coordinates of the two models. We therefore set $\alpha = 30°$, $k_\theta = 0.06\frac{30\pi/180}{0.1} = 0.3$, and saturated the joint offset according to $\phi_o \in [-0.08\frac{30°}{0.1}, 0.08\frac{30°}{0.1}] = [-25°, 25°]$. The remaining controller parameters were set equal to the controller parameters of the simplified model.

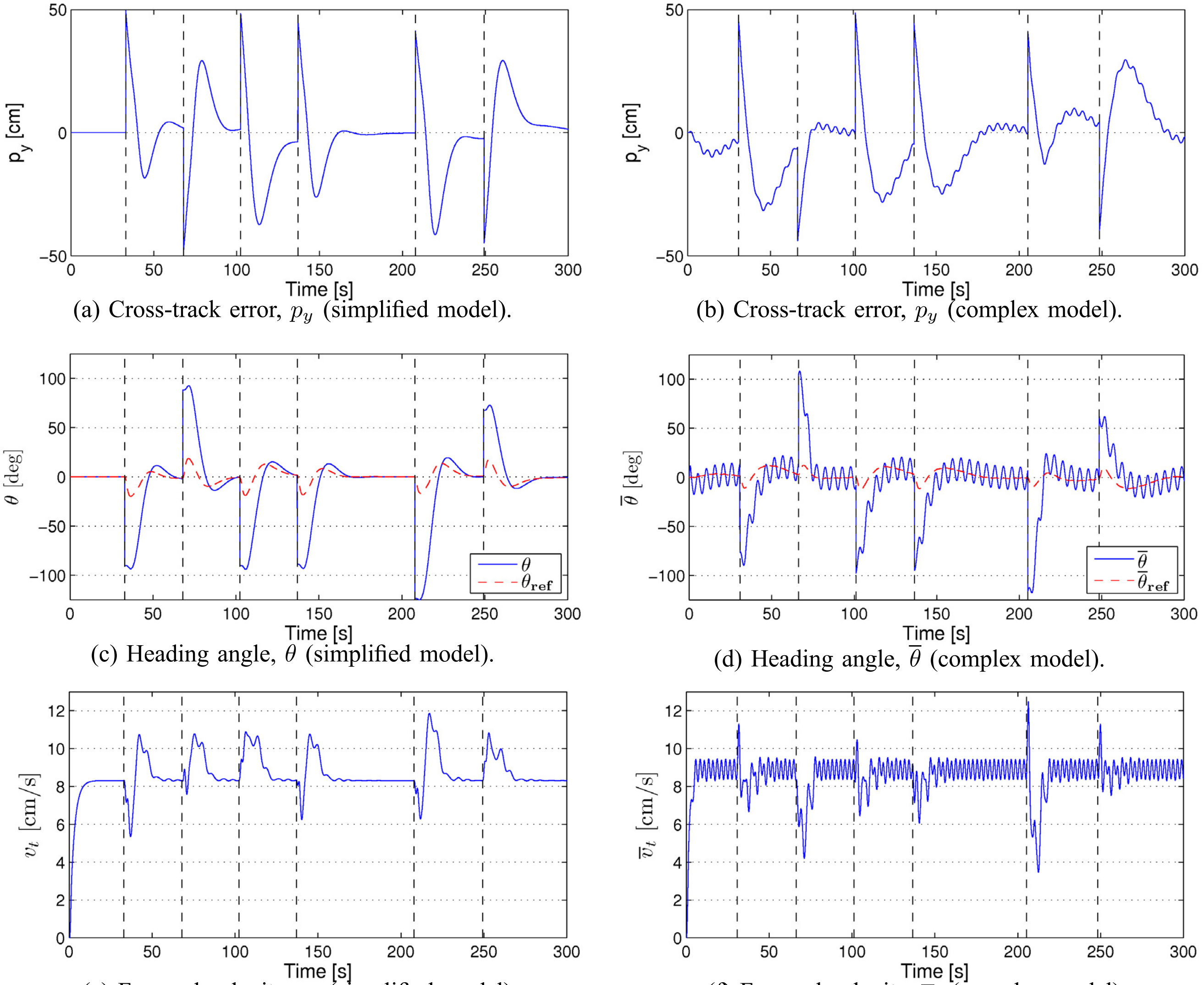
C. *Simulation results*

We defined $k = 7$ waypoints with global frame coordinates $(3, 0)$, $(3, 3)$, $(6, 3)$, $(6, 6)$, $(0, 6)$, $(2, 3)$, and $(0, 0)$, respectively. Fig. 9 shows the motion of the CM of the snake robot from the simplified model (solid line) and the complex model (dashed line), where each waypoint is indicated with a black square. The figure also shows the shape and position of the robot at $t = 20$ s, $t = 90$ s, and $t = 180$ s for the simplified model, and at $t = 55$ s, $t = 125$ s, and $t = 235$ s for the complex model. Furthermore, Fig. 10 shows the cross-track error (in terms of the y axis coordinate of the CM of the robot), the heading angle, and the forward velocity from the two models. The vertical lines in the plots indicate time instants where the guidance strategy switches to the next waypoint. We see that the state of the robot experiences a jump at each waypoint switch since, by Algorithm 8, the global frame is redefined at a waypoint switch.

As seen in Fig. 9, the snake robot has a nice and smooth motion towards each waypoint. The plotted paths, in particular the path near the waypoint at coordinate $(0, 6)$, indicate that the snake robot is able to turn more rapidly in the complex model compared to the simplified model. The qualitative behaviour of the two models are, however, similar. Fig. 10(a)-(b) shows that the cross-track error converges nicely to zero after each waypoint switch. The heading of the snake robot, shown in Fig. 10(c)-(d), also converges nicely to zero, i.e. to the direction of the desired path. In summary, the simulation results illustrate that the proposed waypoint guidance strategy successfully steers the snake robot towards each of the specified waypoints.

VIII. Conclusions

This paper considered path following control of snake robots. The first contribution was a description of how a straight line path following controller previously proposed by the authors can be extended to path following of general curved paths by employing an approach previously proposed in the marine control literature for path following control of marine vessels. The second contribution of this paper was a waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines. The waypoint guidance strategy builds on the straight line path following controller previously proposed by the authors.



(a) Cross-track error, p_y (simplified model).

(b) Cross-track error, p_y (complex model).

(c) Heading angle, θ (simplified model).

(d) Heading angle, $\overline{\theta}$ (complex model).

(e) Forward velocity, v_t (simplified model).

(f) Forward velocity, $\overline{v}_t$ (complex model).

Fig. 10. Simulation of the waypoint guidance strategy with the simplified (left) and the complex (right) model of the snake robot.

The paper presented simulation results that illustrated the performance of the proposed guidance strategy.

References

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

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

[3] P. Prautsch, T. Mita, and T. Iwasaki, "Analysis and control of a gait of snake robot," *Trans. IEE J. Ind. Appl. Soc.*, vol. 120-D, pp. 372–381, 2000, mer utfyllende versjon av "Control and Analysis of the Gait of Snake Robots" (Prautsch1999).

[4] H. Date, Y. Hoshi, and M. Sampei, "Locomotion control of a snake-like robot based on dynamic manipulability," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2000.

[5] F. Matsuno and H. Sato, "Trajectory tracking control of snake robots based on dynamic model," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3029–3034.

[6] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Controllability and stability analysis of planar snake robot locomotion," *IEEE Trans. Automatic Control*, 2011, to appear.

[7] 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.

[8] P. A. Vela, K. A. Morgansen, and J. W. Burdick, "Underwater locomotion from oscillatory shape deformations," in *Proc. IEEE Conf. Decision and Control*, vol. 2, Dec. 2002, pp. 2074–2080 vol.2.

[9] K. McIsaac and J. Ostrowski, "Motion planning for anguilliform locomotion," *IEEE Trans. Rob. Aut.*, vol. 19, no. 4, pp. 637–625, 2003.

[10] K. Morgansen, B. Triplett, and D. Klein, "Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles," *IEEE Trans. Robotics*, vol. 23, no. 6, pp. 1184–1199, Dec 2007.

[11] E. Børhaug, "Nonlinear control and synchronization of mechanical systems," Ph.D. dissertation, Norwegian University of Science and Technology, 2008.

[12] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A simplified model of planar snake robot locomotion," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 2868–2875.

[13] ——, "Stability analysis of snake robot locomotion based on averaging theory," in *Proc. IEEE Conf. Decision and Control*, Atlanta, GA, USA, 2010, pp. 1977–1984.

[14] K. Do and J. Pan, "Global tracking control of underactuated ships with off-diagonal terms," in *IEEE Conf. Decision and Control*, vol. 2, 2003, pp. 1250–1255.

[15] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*. Trondheim, Norway: Marine Cybernetics, 2002.

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