

RESEARCH

Open Access

Path following control of planar snake robots using virtual holonomic constraints: theory and experiments

Ehsan Rezapour^{1*}, Kristin Y Pettersen¹, Pål Liljebäck^{1,2}, Jan T Gravdahl¹ and Eleni Kelasidi¹

Abstract

This paper considers path following control of planar snake robots using virtual holonomic constraints. In order to present a model-based path following control design for the snake robot, we first derive the Euler-Lagrange equations of motion of the system. Subsequently, we define geometric relations among the generalized coordinates of the system, using the method of virtual holonomic constraints. These appropriately defined constraints shape the geometry of a constraint manifold for the system, which is a submanifold of the configuration space of the robot. Furthermore, we show that the constraint manifold can be made invariant by a suitable choice of feedback. In particular, we analytically design a smooth feedback control law to exponentially stabilize the constraint manifold. We show that enforcing the appropriately defined virtual holonomic constraints for the configuration variables implies that the robot converges to and follows a desired geometric path. Numerical simulations and experimental results are presented to validate the theoretical approach.

Keywords: Path following; Snake robots; Biologically inspired robots; Virtual holonomic constraints

Introduction

Although wheels and legs are extensively used in mobile robots and are known as the conventional locomotion tools, we sometimes need more adaptable and flexible locomotion systems in order to carry out tasks in complex, narrow, and unstructured environments. In such situations, snake robots, which have significant adaptability and structural flexibility properties, are a potentially useful alternative to conventional types of mobile robots. Snake robots are relevant for applications where restriction of human involvement is important due to safety (e.g. in firefighting operations [1]) and in applications where human presence is impossible (e.g. in narrow pipe inspection tasks [2,3]).

This paper considers path following control of snake robots. In the path following problem, the goal is to ensure that the error between the system output and a desired geometric path is asymptotically less than any pre-specified constant, while guaranteeing a forward motion

along the path and boundedness of the states of the controlled system [4]. This control problem is particularly challenging for snake robots. This is due to the fact that such mechanisms are generally hyper-redundant, i.e. they have a large degree of kinematic redundancy, and this gives rise to a complicated dynamical behaviour for the system. Moreover, snake robots are underactuated, i.e. they have fewer independent control inputs than degrees of freedom (DOF), and this complicates the control design for these robots.

Background and literature review

In general, snake robots can be categorized into two classes: snake robots which are subject to nonholonomic velocity constraints and snake robots without nonholonomic velocity constraints. Path following control of both classes of snake robots has been considered in several previous works. The majority of these works consider snake robots with nonholonomic velocity constraints, which is inspired by the world's first snake robot developed in 1972 [5]. Nonholonomic constraints are in the form of sideslip constraints on the links of the robot, i.e. where each link is constrained from moving sideways. These constraints

*Correspondence: Ehsan.Rezapour@itk.ntnu.no

¹Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim 7491, Norway

Full list of author information is available at the end of the article

allow the control input to be specified directly in terms of the desired propulsion of the snake robot, which is employed in [6-8] for computed torque control of the position and heading of snake robots with nonholonomic velocity constraints. In [9], position and path following controllers are proposed for the case where some, but not all, of the snake robot links are subject to sideslip constraints. These constrained links can be lifted from the ground, which give the system more DOF that can be utilized to follow a trajectory while simultaneously maintaining a high manipulability. Similar approaches are considered in [10], where strategies for sinus lifting during the lateral undulatory motion are proposed. In [11], a path following controller for a snake robot with nonholonomic velocity constraints is proposed, and Lyapunov analysis is employed in order to analyse the controller.

Path following control of snake robots without nonholonomic velocity constraints is only considered in a few previous works. In [12], path following control of swimming snake robots is achieved by moving the joints according to a predetermined gait pattern while introducing an angular offset in each joint to steer the robot to some desired path. Methods based on numerical optimal control are considered in [13] for determining optimal gaits during positional control of snake robots. In [14], a control strategy is proposed for sinus lifting during lateral undulation by solving a quadratic optimization problem. In [15], numerical simulations are used to study the properties of lateral undulation that are related to the optimality of motion of the snake robot. In [16,17], cascaded systems theory is employed to achieve path following control of a snake robot described by a simplified model. In this simplified model of the snake robot, the motion of the links is approximated as translational motion instead of rotational motion, which is valid for small joint angles. In [18], a dynamic feedback control law is proposed which controls the body shape and orientation of the snake robot. Controllability analysis of planar snake locomotion is presented in [19], and a controller for straight line path following control of snake robots is proposed and a Poincare map is investigated to prove that the resulting state variables of the snake robot, except for the position in the forward direction, trace out an exponentially stable periodic orbit.

Research design and methodology

The contribution of this paper is a solution to the path following control problem for a snake robot without nonholonomic velocity constraints, by using virtual holonomic constraints, which is a particularly useful concept for control of oscillations (see, e.g. [20-24]). Using this approach, we constrain the state evolution of the system to an appropriately defined submanifold of the configuration space, which is called the constraint manifold. This

manifold is defined based on the specified geometric relations among the generalized coordinates of the system which are called virtual holonomic constraints. The proposed feedback control law is designed to exponentially stabilize the constraint manifold, i.e. to enforce the virtual holonomic constraints, which allows the convergence of the snake robot to the desired path.

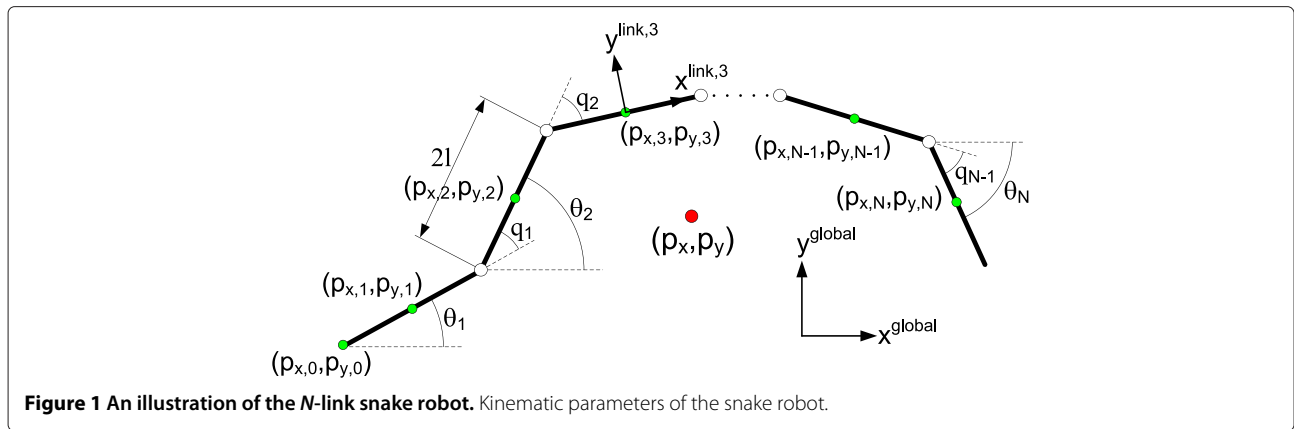
To our best knowledge, the application of model-based motion control approaches which rely on formal stability proofs for snake robots is very restricted in the previous literature. In particular, the only previous works which formally prove the stability of a path following controller for a snake robot without nonholonomic velocity constraints are presented in [17,18]. In [17], cascaded systems theory is used to stabilize a desired straight path for the position of the center of mass (CM) of a snake robot which is described based on simplified kinematic and dynamic models. In contrast with [17], in this paper, we consider a more complex and accurate model of the snake robot, where the motion of the links is not modelled based on the simplifying assumptions of [17]. Preliminary results of this paper are presented in [18], where the method of virtual holonomic constraints is used for path following control of a planar snake robot. In this paper, these results are extended with a new simulation study of a 11-link snake robot. Furthermore, we extend the results presented in [18] by an experimental investigation of the performance of the control approach. A back-to-back comparison between simulations and experimental results is given, in order to contribute to bridge the gap between advanced control theory and practice.

The paper is organized as follows. First, we derive the Euler-Lagrange equations of motion of the robot. Subsequently, we state the control design objectives. Afterwards, we use the virtual holonomic constraints approach for path following control design for the snake robot. Finally, we present the results of numerical simulations and real-time experiments which illustrate the performance of the theoretical control design.

Methods

In this section, we derive the kinematic model along with the dynamic equations of motion of the snake robot in a Lagrangian framework. Moreover, we use partial feedback linearization to write the model in a simpler form for model-based control design.

In order to perform control design, we need to write the governing equations of the system in an implementable way. This is often done by choosing a local coordinate chart and writing the system equations with respect to (w.r.t.) these coordinates. According to the illustration of the snake robot in Figure 1, we choose the vector of the generalized coordinates of the N -link snake robot as



$x = [q_1, q_2, \dots, q_{N-1}, \theta_N, p_x, p_y]^T \in \mathbb{R}^{N+2}$, where q_i with $i \in \{1, \dots, N-1\}$ denotes the i th joint angle, θ_N denotes the head angle, and the pair (p_x, p_y) describes the position of the CM of the robot w.r.t. the global $x-y$ axes. Since the robot is not subject to nonholonomic velocity constraints, the vector of the generalized velocities is defined as $\dot{x} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_{N-1}, \dot{\theta}_N, \dot{p}_x, \dot{p}_y]^T \in \mathbb{R}^{N+2}$. Using these coordinates, it is possible to specify the kinematic map of the robot. In this paper, we denote the first N elements of the vector x , i.e. $(q_1, \dots, q_{N-1}, \theta_N)$, as the angular coordinates, and the corresponding dynamics as the angular dynamics of the system.

The geometry of the problem

The $(N+2)$ -dimensional configuration space of the snake robot is denoted as $\mathcal{Q} = \mathcal{S} \times \mathcal{G}$, which is composed of the shape space \mathcal{S} and a Lie group \mathcal{G} which is freely and properly acting on the configuration space. In particular, the shape variables, i.e. $q_a = (q_1, \dots, q_{N-1})$, which define the internal configuration of the robot and which we have direct control on, take values in \mathcal{S} . Moreover, the position variables, i.e. $q_u = (\theta_N, p_x, p_y)$, which are passive DOF of the system, lie in \mathcal{G} . The velocity space of the robot is defined as the differentiable $(2N+4)$ -dimensional tangent bundle of \mathcal{Q} as $T\mathcal{Q} = \mathbb{T}^N \times \mathbb{R}^{N+4}$, where \mathbb{T}^N denotes the N -torus in which the angular coordinates live. The free Lagrangian function of the robot $\mathcal{L} : T\mathcal{Q} \rightarrow \mathbb{R}$ is invariant under the given action of \mathcal{G} on \mathcal{Q} . The coupling between the shape and the position variables causes the net displacement of the position variables, according to the cyclic motion of the shape variables, i.e. the *gait pattern*. Note that for simplicity of presentation, in this paper, we consider local representation of $T\mathcal{Q}$ embedded in an $(2N+4)$ -dimensional open subset of the Euclidean space \mathbb{R}^{2N+4} .

The forward kinematic map of the snake robot

Based on the kinematic parameters of the snake robot given in Figure 1, it is possible to write the coordinate

representation of the forward kinematic map. The map between the absolute link angles θ_i and the relative joint angles q_i is given by

$$\theta_i = \sum_{n=i}^{N-1} q_n + \theta_N \quad (1)$$

The position of the CM of the i th link w.r.t. the global $x-y$ axes can be, respectively, given as

$$p_{x,i} = p_{x,0} + 2l \sum_{j=1}^{i-1} \cos \theta_j + l \cos \theta_i \quad (2)$$

$$p_{y,i} = p_{y,0} + 2l \sum_{j=1}^{i-1} \sin \theta_j + l \sin \theta_i \quad (3)$$

where $2l$ denotes the length of each link, and $(p_{x,0}, p_{y,0})$ denotes the tail position (cf. Figure 1). The linear velocities of the CM of the i th link w.r.t. the global $x-y$ axes can be found by taking the time derivative of (2)-(3) which gives

$$\dot{p}_{x,i} = \dot{p}_{x,0} - 2l \sum_{j=1}^{i-1} \sin \theta_j \dot{\theta}_j - l \sin \theta_i \dot{\theta}_i \quad (4)$$

$$\dot{p}_{y,i} = \dot{p}_{y,0} + 2l \sum_{j=1}^{i-1} \cos \theta_j \dot{\theta}_j + l \cos \theta_i \dot{\theta}_i \quad (5)$$

Since all the links have equal length and mass, the position of the CM for the whole structure of the robot is defined as

$$(p_x, p_y) = \left(\frac{1}{N} \sum_{i=1}^N p_{x,i}, \frac{1}{N} \sum_{i=1}^N p_{y,i} \right) \quad (6)$$

To facilitate path following control of the CM of the snake robot, we replace the tail position $(p_{x,0}, p_{y,0})$ in (2)-(3) with

the position of the CM of the robot (p_x, p_y) using the following change of coordinates:

$$p_{x,0} = p_x - \frac{1}{N} \sum_{i=1}^N \left(2l \sum_{j=1}^{i-1} \cos \theta_j + l \cos \theta_i \right) \quad (7)$$

$$p_{y,0} = p_y - \frac{1}{N} \sum_{i=1}^N \left(2l \sum_{j=1}^{i-1} \sin \theta_j + l \sin \theta_i \right) \quad (8)$$

Substituting (7)-(8) along with their time derivatives into (2)-(5) completes the derivation of the forward kinematic map of the snake robot w.r.t. the desired specified coordinate chart (x, \dot{x}) .

Equations of motion of the snake robot

The majority of the previous literature on snake robots and similar mobile multi-body robotic structures, such as eel-like robots, have derived the equations of motion of these robots with a Newton-Euler formulation, i.e. where the equations describing the linear and angular motion of individual links are written separately (see, e.g. [15,16]). This is due to the fact that it is usually not straightforward to integrate the anisotropic external dissipative forces, i.e. ground friction forces, acting on these complex robotic structures into their Euler-Lagrange equations of motion. However, ground friction forces have been proved to play a fundamental role in snake robot locomotion (see, e.g. [16]). In this paper, we derive the equations of motion of the snake robot in a Lagrangian framework, i.e. treating the robot as a whole and performing the analysis using a Lagrangian function, which is simple to follow and better suited for studying advanced mechanical phenomena such as elastic link deformations [25], which might be insightful for future research challenges on snake robots. Moreover, we integrate the anisotropic friction forces into these equations using the Jacobian matrices of the links, which gives a straightforward mapping of these forces for the equations of motion.

Snake robots are a class of *simple* mechanical systems, where the Lagrangian $\mathcal{L}(q_a, \dot{x})$ is defined as the difference between the kinetic energy $\mathcal{K}(q_a, \dot{x})$ and potential energy $\mathcal{P}(x)$ of the system [26]. Since the planar snake robot is not subject to any potential field, i.e. $-\nabla \mathcal{P}(x) = 0$, we may write the Lagrangian equal to the kinetic energy, which is the sum of the translational and the rotational kinetic energy of the robot:

$$\mathcal{L}(q_a, \dot{x}) = \mathcal{K}(q_a, \dot{x}) = \frac{1}{2} m \sum_{i=1}^N (\dot{p}_{x,i}^2 + \dot{p}_{y,i}^2) + \frac{1}{2} J \sum_{i=1}^N \dot{\theta}_i^2 \quad (9)$$

where m and J denote the uniformly distributed mass and moment of inertia of the links, respectively. Using

the Lagrangian function (9), we write the Euler-Lagrange equations of motion of the controlled system as

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q_a, \dot{x})}{\partial \dot{x}_i} \right] - \frac{\partial \mathcal{L}(q_a, \dot{x})}{\partial x_i} = (B(x)\tau - \tau_f)_i \quad (10)$$

where $i \in \{1, \dots, N+2\}$, $B(x) = [e_j] \in \mathbb{R}^{(N+2) \times (N-1)}$ is the full column rank actuator configuration matrix, where e_j denotes the j th standard basis vector in \mathbb{R}^{N+2} . Moreover, $B(x)\tau \in \mathbb{R}^{N+2}$ with $\tau = [\tau_1, \dots, \tau_{N-1}]^T \in \mathbb{R}^{N-1}$ stands for the generalized forces resulting from the control inputs. Furthermore, $\tau_f = [\tau_f^1, \dots, \tau_f^{N+2}]^T \in \mathbb{R}^{N+2}$ denotes viscous and Coulomb friction forces acting on $(N+2)$ DOF of the system. The controlled Euler-Lagrange equations (10) can also be written in the form of a second-order differential equation as

$$M(q_a) \ddot{x} + C(x, \dot{x}) \dot{x} = B(x)\tau - \tau_f \quad (11)$$

where $M(q_a) \in \mathbb{R}^{(N+2) \times (N+2)}$ is the positive definite symmetric inertia matrix, $C(x, \dot{x}) \dot{x} \in \mathbb{R}^{N+2}$ denotes the generalized Coriolis and centripetal forces, and the right-hand side terms denote the external forces acting on the system. The fact that the inertia matrix is only a function of the directly actuated shape variables q_a is a direct consequence of the invariance of the Lagrangian function (9) w.r.t. the position variables. Moreover, since $\text{rank}[B(x)] < \dim(x)$, the system is underactuated. This underactuation represents the lack of direct control on the head angle and the position of the CM of the robot.

The dynamic model (11) perfectly agrees with the equations of motion which are derived based on the Newton-Euler formulation in previous works (see, e.g. [16]). In order to validate the model, in the last section of this paper, we present simulation results which are obtained using the dynamic model (11) together with experimental results for the locomotion of the robot which are obtained using a physical snake robot. The agreement between simulations and experiments shows that the dynamic model (11) accurately represents the motion of the robot.

The ground friction model

In this subsection, both viscous and Coulomb friction models are used for capturing the essential properties of the anisotropic ground friction forces. For modelling the friction, we first define the rotation matrix for mapping from the global frame to the local frame of link i (cf. Figure 1) as

$$R_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \quad (12)$$

Using (4)-(5) and (12), the velocities of the links in the local link frames can be written in terms of the velocities of the links in the global frame as

$$v_{\text{link},i}^T = \begin{bmatrix} v_t^{\text{link},i} & v_n^{\text{link},i} \end{bmatrix}^T = R_i^T \begin{bmatrix} \dot{p}_{x,i} & \dot{p}_{y,i} \end{bmatrix}^T \quad (13)$$

where $v_t^{\text{link},i}$ and $v_n^{\text{link},i}$ denote the linear velocity of the CM of the i th link in the tangential (along link x -axis) and normal (along link y -axis) direction of the link, respectively. The total friction force acting on link i is defined as the sum of the viscous and Coulomb friction forces, which are denoted by f_{v_i} and f_{c_i} , respectively, as

$$f^{\text{link},i} = f_{c_i} + f_{v_i} \quad (14)$$

Assuming equal friction coefficients for all the links, we write the model of the friction for each individual link i as

$$f_{c_i} = mg \begin{bmatrix} \mu_t \text{sgn}(v_t^{\text{link},i}) & \mu_n \text{sgn}(v_n^{\text{link},i}) \end{bmatrix}^T \in \mathbb{R}^2 \quad (15)$$

$$f_{v_i} = \begin{bmatrix} c_t v_t^{\text{link},i} & c_n v_n^{\text{link},i} \end{bmatrix}^T \in \mathbb{R}^2 \quad (16)$$

where $i \in \{1, \dots, N\}$, m denotes the mass of a link, g denotes the acceleration due to gravity, and μ_t and μ_n denote Coulomb friction coefficients in the tangential and normal direction of the link, respectively. Furthermore, c_t and c_n denote viscous friction coefficients in the tangential and normal direction of the link, respectively. Thus, we map the friction force acting on the i th link to the global x - y frame as

$$f_{\text{global}}^{\text{link},i} = R_i f^{\text{link},i} \quad (17)$$

Finally, we can write τ_f in (11) as

$$\tau_f = \sum_{i=1}^N \mathcal{J}_i^T(x) f_{\text{global}}^{\text{link},i} \quad (18)$$

where

$$\mathcal{J}_i^T(x) = \begin{bmatrix} \frac{\partial \dot{p}_{x,i}}{\partial \dot{x}_j} & \frac{\partial \dot{p}_{y,i}}{\partial \dot{x}_j} \end{bmatrix} \in \mathbb{R}^{(N+2) \times 2}, \quad j \in \{1, \dots, N+2\} \quad (19)$$

denotes the transpose of the Jacobian matrix of the CM of the i th link.

Remark 1. As argued in [16], the motion of a snake robot with anisotropic viscous ground friction is qualitatively (but not quantitatively) similar as with anisotropic Coulomb friction. However, a viscous friction model is less complex w.r.t. control design and analysis. Accordingly, we employ a viscous friction model for the control design in this paper.

Partial feedback linearization of the dynamic model

A common method for control of mechanical systems is full-state feedback linearization. This approach is not applicable for snake robots due to the underactuation. However, it is still possible to linearize the dynamics of the actuated DOF of the robot, which is called collocated partial feedback linearization, and can simplify the analysis as well as the control design. A similar approach is considered in [16], but for the sake of completeness, we present the approach here. To this end, we separate the dynamic equations of the robot given by (11) into two subsets by taking $x = [q_a, q_u]^T \in \mathbb{R}^{N+2}$, with $q_a \in \mathbb{R}^{N-1}$ and $q_u \in \mathbb{R}^3$ which were defined in the subsection describing the geometry of the problem:

$$m_{11}(q_a) \ddot{q}_a + m_{12}(q_a) \ddot{q}_u + h_1(x, \dot{x}) = \psi \in \mathbb{R}^{N-1} \quad (20)$$

$$m_{21}(q_a) \ddot{q}_a + m_{22}(q_a) \ddot{q}_u + h_2(x, \dot{x}) = 0_{3 \times 1} \in \mathbb{R}^3 \quad (21)$$

where $m_{11} \in \mathbb{R}^{(N-1) \times (N-1)}$, $m_{12} \in \mathbb{R}^{(N-1) \times 3}$, $m_{21} \in \mathbb{R}^{3 \times (N-1)}$, and $m_{22} \in \mathbb{R}^{3 \times 3}$ denote the corresponding submatrices of the inertia matrix, and $0_{3 \times 1} = [0, 0, 0]^T \in \mathbb{R}^3$. Furthermore, $h_1(x, \dot{x}) \in \mathbb{R}^{N-1}$ and $h_2(x, \dot{x}) \in \mathbb{R}^3$ include all the contributions of the Coriolis, centripetal, and friction forces. Moreover, $\psi \in \mathbb{R}^{N-1}$ denotes the non-zero part of the vector of control forces, i.e. $B(x)\tau = [\psi, 0_{3 \times 1}]^T \in \mathbb{R}^{N+2}$. From (21), we have

$$\ddot{q}_u = -m_{22}^{-1}(h_2 + m_{21}\ddot{q}_a) \in \mathbb{R}^3 \quad (22)$$

Substituting (22) into (20) yields

$$\left(m_{11} - m_{12}m_{22}^{-1}m_{21}\right)\ddot{q}_a - \left(m_{12}m_{22}^{-1}\right)h_2 + h_1 = \psi \quad (23)$$

For linearizing the dynamics of the directly actuated DOF, we apply the global transformation of the vector of control inputs as

$$\psi = \left(m_{11} - m_{12}m_{22}^{-1}m_{21}\right)\vartheta - \left(m_{12}m_{22}^{-1}\right)h_2 + h_1 \quad (24)$$

where $\vartheta = [\vartheta_1, \vartheta_2, \dots, \vartheta_{N-1}]^T \in \mathbb{R}^{N-1}$ is the vector of new control inputs. Consequently, the dynamic model (20)-(21) can be written in the following partially feedback linearized form

$$\ddot{q}_a = \vartheta \in \mathbb{R}^{N-1} \quad (25)$$

$$\ddot{q}_u = \mathcal{D}(x, \dot{x}) + \mathcal{C}(q_a) \vartheta \in \mathbb{R}^3 \quad (26)$$

with

$$\mathcal{D}(x, \dot{x}) = -m_{22}^{-1}(q_a) h_2(x, \dot{x}) = [f_{\theta_N}, f_x, f_y]^T \in \mathbb{R}^3 \quad (27)$$

$$\begin{aligned} \mathcal{C}(q_a) &= -m_{22}^{-1}(q_a) m_{21}(q_a) \\ &= [\beta_i(q_a), 0, 0]^T \in \mathbb{R}^{3 \times (N-1)} \end{aligned} \quad (28)$$

where $\beta_i(q_a) : \mathcal{Q} \rightarrow \mathbb{R}$ is a smooth scalar-valued function. It can be numerically shown that the value of β_i is negative at any configuration $q_a \in \mathcal{Q}$. Furthermore, f_{θ_N} , f_x , and f_y denote the friction forces acting on θ_N , p_x , and p_y , respectively (f_{θ_N} also contains Coriolis forces besides the friction forces). For the aim of analysis and model-based control design, we write (25)-(26) in a more detailed form:

$$\ddot{q}_a = \vartheta \in \mathbb{R}^{N-1} \quad (29)$$

$$\ddot{\theta}_N = f_{\theta_N}(x, \dot{x}) + \beta_i(q_a) \vartheta^i \in \mathbb{R} \quad (30)$$

$$\ddot{p}_x = f_x(x, \dot{x}) \in \mathbb{R} \quad (31)$$

$$\ddot{p}_y = f_y(x, \dot{x}) \in \mathbb{R} \quad (32)$$

where the summation convention is applied in (30), and henceforth, to all the equations which contain repeated upper-lower indices (i.e. whenever an expression contains a repeated index, one as a subscript and the other as a superscript, summation is implied over this index [26]). The dynamical system (29)-(32) is in the form of a control-affine system with drift. In particular, the term

$$\mathcal{A}(x, \dot{x}) = [\dot{q}_a, \dot{q}_u, 0_{(N-1) \times 1}, \mathcal{D}(x, \dot{x})]^T \in \mathbb{R}^{2N+4} \quad (33)$$

is called the drift vector field, which specifies the dynamics of the robot when the control input is zero. Furthermore, the columns of the matrix

$$\mathcal{B}(q_a) = \begin{bmatrix} 0_{(N+2) \times (N-1)} \\ I_{N-1} \\ [\beta_1(q_a), \dots, \beta_{N-1}(q_a)] \\ 0_{2 \times (N-1)} \end{bmatrix} \in \mathbb{R}^{(2N+4) \times (N-1)} \quad (34)$$

are called the control vector fields, which enable us to control the internal configuration and consequently the orientation and the position of the robot in the plane.

Remark 2. The last two rows of the control vectors in (34) are composed of zero elements. This implies that the control forces have no direct effect on the dynamics of the position of the CM of the robot, i.e. (31)-(32). Furthermore, the dynamics of the position of the CM are coupled with the dynamics of the directly actuated shape variables q_a , i.e. (29), only through the friction forces. Accordingly, in the absence of the friction forces, the linear momentum of the robot is a conserved quantity, and the position of the CM of the robot is not controllable.

Control design objectives and the track-follow problem formulation

In this section, we state our control design objectives which will be followed throughout the remaining sections of the paper. In particular, we stress that for a complex mobile multi-link robotic structure such as a snake robot, formulating a pure path following, trajectory tracking, or maneuvering problem is unusual (for definitions of these problem formulations, see [27]). This is due to the fact that for a part of the state variables of the system (particularly the shape variables and the head angle), it is most natural to formulate the control problem as a trajectory tracking problem, while for the other state variables (particularly the position of the CM), we may formulate the problem as a path following or a maneuvering one.

To formulate a combinational *track-follow* problem for the snake robot, which we define as a trajectory tracking formulation for a subset of the state variables, together with a path following formulation for the remaining subset, we introduce the error variable for the i th joint of the robot as

$$y_i = q_i - \Phi_i \quad (35)$$

where $i \in \{1, \dots, N-1\}$, and $\Phi_i \in \mathbb{R}$ denotes a function that defines the reference trajectory for the i th joint which will be chosen through the control design in the next section. The head angle error is defined as

$$y_N = \theta_N - \Phi_N \quad (36)$$

where $\Phi_N \in \mathbb{R}$ denotes the reference head angle for the robot.

We divide the control objectives into three main parts. In the first part, the goal is to make the shape variables of the robot track given bounded smooth time-varying references, i.e. asymptotic trajectory tracking problem, such that

$$\lim_{t \rightarrow \infty} \|y_i(t)\| = 0 \quad (37)$$

for all $i \in \{1, \dots, N-1\}$. Furthermore, we seek to control the head angle of the robot. The second part of the control objective is thus to make the head angle of the robot track a desired head angle such that

$$\lim_{t \rightarrow \infty} \|y_N(t)\| = 0 \quad (38)$$

Moreover, we define a desired straight path that we want the CM of the snake robot to follow. This is defined as a smooth one-dimensional manifold $\mathcal{P} \subset \mathbb{R}^2$, with coordinates in the x - y plane given by the pair (p_{xd}, p_{yd}) , which

are parameterized by a scalar time-dependent variable $\Theta(t)$ as

$$\mathcal{P} = \{(p_{xd}(\Theta), p_{yd}(\Theta)) \in \mathbb{R}^2 : \Theta \geq 0\} \quad (39)$$

We define the vector of the path following error variables for the position of the CM of the robot as $\tilde{p} = [p_x(t) - p_{xd}(\Theta), p_y(t) - p_{yd}(\Theta)]^T \in \mathbb{R}^2$. Subsequently, the third part of the control objectives is defined as practical convergence (see, e.g. [4]) of the position of the CM of the robot to the desired path such that

$$\lim_{t \rightarrow \infty} \sup \|\tilde{p}(t)\| \leq \varepsilon \quad (40)$$

where $\varepsilon \in \mathbb{R}_{>0}$ is an arbitrary positive scalar. Moreover, we require that $\dot{\Theta}(t) \geq 0$ and $\lim_{t \rightarrow \infty} \Theta(t) = \infty$ (forward motion along the path), and boundedness of the states of the controlled system.

Path following control with virtual holonomic constraints

The idea of virtual holonomic constraints is particularly a useful concept for control of oscillations (see, e.g. [20-24]). We will in this section show how this approach can be used to solve the path following control problem of snake robots. In particular, we will show how, by designing the joint reference trajectories in (35) using virtual holonomic constraints and by combining this with virtual holonomic constraints motivated by line-of-sight (LOS) guidance for the head angle in (36), we are able to solve the path following control problem, i.e. achieving (40). Our main motivation for using this approach is the fact that while performing the gait pattern lateral undulation which consists of fixed periodic body motions, all the solutions of the snake robot dynamics have inherent oscillatory behaviour. Moreover, we will show how this behaviour can be analytically and *constructively* controlled based on virtual holonomic constraints. In particular, we use the word ‘constructive’ in the sense that through the feedback action, we shape the dynamics of the system such that it possesses the desired structural properties, i.e. positive invariance and exponential stability of an appropriately defined constraint manifold. To this end, we define a constraint manifold for the system, and we design the control input of (29) to exponentially stabilize the constraint manifold. The geometry of this manifold is defined based on specified geometric relations among the generalized coordinates of the system which are called virtual holonomic constraints. In particular, we call them virtual constraints because they do not arise from a physical connection

between two variables but rather from the actions of a feedback controller [20].

Trajectory planning by virtual holonomic constraints

Virtual holonomic constraints are specified through C^1 coordinate-dependent functions $\Phi_i : \mathcal{Q} \rightarrow \mathbb{R}$ which are called the constraint functions, in the relations of the form $\Phi_i(x) = 0$, which can be enforced through the feedback action. In particular, for the snake robot, we define a vector-valued function

$$\Phi = [\Phi_1, \dots, \Phi_N]^T \in \mathbb{R}^N \quad (41)$$

in which every element defines one constraint function for the corresponding angular coordinate of the system.

At this point, we augment the state vector of the system with three new states that in the following will be used in the control design. The introduction of these new variables to the state vector of the system, which will be used as constraint variables, is inspired by the notion of dynamic virtual holonomic constraints [21], i.e. virtual holonomic constraints which depend on the solutions of a dynamic compensator. The idea is to make the virtual holonomic constraints to depend on the variations of a dynamic parameter, which is used for controlling the system on the constraint manifold. The purpose of these additional states is explained below.

1. We introduce two new states $[\phi_o, \dot{\phi}_o]^T \in \mathbb{R}^2$ where the second-order time derivative of ϕ_o will be used as an additional control input that drives the snake robot towards the desired path by modifying the orientation of the robot in accordance with a path following guidance law.
2. In the previous section, we defined the control objective for the joints and the head angle of the robot as a trajectory tracking problem. However, it is known that holonomic constraints are coordinate-dependent equality constraints of the form $\Phi_i(x) = 0$, where Φ_i is a time-independent function [25]. Thus, we remove this explicit time dependency from the reference joint trajectories by augmenting the state vector of the system with a new variable η , with $\dot{\eta} = 2\pi/T$ and $\eta(0) = 0$, where T denotes the period of the cyclic motion of the shape variables of the robot. Subsequently, we denote the augmented coordinate vector of the system by

$$\hat{x} = [q_1, \dots, q_{N-1}, \theta_N, p_x, p_y, \phi_o, \eta]^T \in \mathbb{R}^{N+4} \quad (42)$$

and the corresponding augmented state space by $T\hat{\mathcal{Q}}$.

Virtual holonomic constraints for the joint angles

A fundamental work in the area of snake robots was presented by Hirose [5]. In this work, Hirose considers empirical studies of biological snakes to derive a mathematical approximation of the most common gait pattern among biological snakes, known as lateral undulation. In particular, the shape of a snake conducting lateral undulation can be described by a planar curve (the serpenoid curve) with coordinates in the x - y plane along the curve at arc length s given by

$$x(s) = \int_0^s \cos(a \cos(bz) + cz) dz \quad (43)$$

$$y(s) = \int_0^s \sin(a \cos(bz) + cz) dz \quad (44)$$

where a , b , and c are positive scalars. Locomotion of a snake-like structure in accordance with the serpenoid curve, i.e. lateral undulation, is achieved if the joints of the robot move according to the reference joint trajectories in the form of a sinusoidal function with specified amplitude, frequency, and phase shift. In particular, using the foregoing defined new states, we define a constraint function for the i th joint of the snake robot by

$$\Phi_i = \alpha \sin(\eta + (i - 1)\delta) + \phi_o \quad (45)$$

where $i \in \{1, \dots, N - 1\}$, α denotes the amplitude of the sinusoidal joint motion, and δ is a phase shift that is used to keep the joints *out of phase*. Moreover, ϕ_o is an offset value that is identical for all of the joints. It was illustrated in [16] how the offset value ϕ_o affects the orientation of the snake robot in the plane. Building further on this insight, we consider the second-order time derivative of ϕ_o in the form of a dynamic compensator, which will be used to control the orientation of the robot. In particular, through this control term, we modify the orientation of the robot in accordance with a reference orientation. This will be done by adding an offset angle to the reference trajectory of each joint. We will show that this will steer the position of the CM of the robot towards the desired path. The constraint function (45) is dynamic, since it depends on the solution of a dynamic compensator.

Virtual holonomic constraint for the head link angle

In this subsection, we define a constraint function for the head angle of the robot. In particular, we use a line-of-sight (LOS) guidance law as the reference angle for the head link. LOS guidance is a much-used method in marine control systems (see, e.g. [27]). In general, guidance-based control strategies are based on defining a reference heading angle for the vehicle through a guidance law and designing a controller to track this angle [27]. Motivated by marine control literature, in [17] based

on a simplified model of the snake robot, using cascade systems theory, it was proved that if the heading angle of the snake robot was controlled to the LOS angle, then also the position of the CM of the robot would converge to the desired path. We will show that a similar guidance-based control strategy can successfully steer the robot towards the desired path. However, we perform the model-based control design based on a more accurate model of the snake robot which does not contain the simplifying assumptions of [17] which are valid for small joint angles.

To define the guidance law, without loss of generality, we assign the global coordinate system such that the global x -axis is aligned with the desired path. Consequently, the position of the CM of the robot along the y -axis, denoted by p_y , defines the shortest distance between the robot and the desired path, often referred to as the *cross-track error*. In order to solve the path following problem, we use the LOS guidance law as a virtual holonomic constraint, which defines the desired head angle as a function of the cross-track error as

$$\Phi_N = -\tan^{-1}\left(\frac{p_y}{\Delta}\right) \quad (46)$$

where $\Delta > 0$ is a design parameter known as the look-ahead distance. The idea is that steering the head angle of the snake robot such that it is headed towards a point located at a distance Δ ahead of the robot along the desired path will make the snake robot move towards the path and follow it.

Defining a constraint manifold

We collect all the foregoing defined constraint functions in the following vector-valued function

$$\Phi = \left[\alpha \sin(\eta) + \phi_o, \dots, \alpha \sin(\eta + (N - 1)\delta) + \phi_o, \tan^{-1}\left(\frac{p_y}{\Delta}\right) \right]^T \in \mathbb{R}^N \quad (47)$$

For trajectory planning using virtual holonomic constraints, we define the constraint manifold associated with the constraint functions (47) as

$$\Gamma = \left\{ (\hat{x}, \dot{\hat{x}}) \in T\hat{\mathcal{Q}} : q_i = \Phi_i(\eta, \phi_o), \theta_N = \Phi_N(p_y), \right. \\ \left. \dot{q}_i = \dot{\eta} \frac{\partial \Phi_i}{\partial \eta} + \dot{\phi}_o \frac{\partial \Phi_i}{\partial \phi_o}, \dot{\theta}_N = \dot{p}_y \frac{\partial \Phi_N}{\partial p_y} \right\} \quad (48)$$

where $i \in \{1, \dots, N - 1\}$. The constraint manifold (48) is a six-dimensional submanifold of $\hat{\mathcal{Q}}$, since we have

three different constraint variables, i.e. η, ϕ_o, p_y . The goal of the control input is to enforce the virtual holonomic constraints (47), by making Γ exponentially stable for the closed-loop system and thereby achieving the control objectives (37)-(38). To this end, we define the elements of a controlled output vector $y \in \mathbb{R}^N$ for the system (29)-(32) as the difference between the angular coordinates and their corresponding constraint functions as

$$y = [q_1 - \Phi_1(\eta, \phi_o), \dots, q_{N-1} - \Phi_{N-1}(\eta, \phi_o), \theta_N - \Phi_N(p_y)]^T \in \mathbb{R}^N \quad (49)$$

We will achieve our control design objectives which we defined in the previous section, by designing the control inputs ϑ and $\ddot{\phi}_o$ such that $(y_i, \dot{y}_i) \rightarrow (0, 0)$ for all $i \in \{1, \dots, N\}$. To this end, we first need to ensure that the given relations in (47) are stabilizable, i.e. a suitable choice of feedback can make the constraint manifold exponentially stable for the closed-loop system. For simplicity of notation, we denote the following differentials:

$$d\Phi_i = \dot{\eta} \frac{\partial \Phi_i}{\partial \eta} + \dot{\phi}_o \frac{\partial \Phi_i}{\partial \phi_o} \quad (50)$$

$$d\Phi_N = \dot{p}_y \frac{\partial \Phi_N}{\partial p_y} \quad (51)$$

$$d^2\Phi_i = \ddot{\eta} \frac{\partial \Phi_i}{\partial \eta} + \dot{\eta}^2 \frac{\partial^2 \Phi_i}{\partial \eta^2} + \ddot{\phi}_o \frac{\partial \Phi_i}{\partial \phi_o} + \dot{\phi}_o^2 \frac{\partial^2 \Phi_i}{\partial \phi_o^2} \quad (52)$$

$$d^2\Phi_N = \ddot{p}_y \frac{\partial \Phi_N}{\partial p_y} + \dot{p}_y^2 \frac{\partial^2 \Phi_N}{\partial p_y^2} \quad (53)$$

where $i \in \{1, \dots, N-1\}$. The Lie derivative of (49) along the solutions of (29)-(32) is of the form

$$\dot{y} = [\dot{q}_1 - d\Phi_1, \dots, \dot{q}_{N-1} - d\Phi_{N-1}, \dot{\theta}_N - d\Phi_N]^T \in \mathbb{R}^N \quad (54)$$

which lacks the control inputs. The Lie derivative of (54) along the solutions of (29)-(32) is of the form

$$\ddot{y} = [\vartheta_1 - d^2\Phi_1, \dots, \vartheta_{N-1} - d^2\Phi_{N-1}, f_{\theta_N} + \beta_i \vartheta^i - d^2\Phi_N]^T \in \mathbb{R}^N \quad (55)$$

which contains the control inputs. Consequently, the controlled output vector (49) yields a well-defined vector relative degree $\{2, 2, \dots, 2\}$ everywhere on the constraint manifold Γ . The virtual holonomic constraints satisfying this vector relative degree condition are called regular, and regular constraints are always feasible [22], i.e. there exists a smooth feedback such that Γ is positively invariant for the closed-loop system. Furthermore, regular constraints in parametric form (47) are always stabilizable [21].

The well-defined vector relative degree $\{2, 2, \dots, 2\}$ on Γ implies that the system (29)-(32) with the controlled output function (47) is input-output feedback linearizable. Consequently, we can stabilize Γ with an input-output feedback linearizing controller.

Output regulation via input-output linearization

In this subsection, we will derive a control law for (29) such that the constraint manifold (48) with the constraint functions defined in (47) is globally exponentially stable for the closed-loop system. In particular, we use input-output linearization to stabilize the constraint manifold (48).

To stabilize $\Phi_i(\eta, \phi_o)$ for the i th joint, i.e. to make $(y_i, \dot{y}_i) \rightarrow (0, 0)$ for all $i \in \{1, \dots, N-1\}$, we define an exponentially stabilizing joint control law. The second-order time derivative of the i th joint tracking error, i.e. the i th element of (55), is of the form

$$\ddot{y}_i = \vartheta_i - d^2\Phi_i \quad (56)$$

We define the control input for the i th joint in (29) as

$$\vartheta_i = d^2\Phi_i - k_p y_i - k_d \dot{y}_i \quad (57)$$

where $k_p > 0$ and $k_d > 0$ are the joint controller gains. These gains are chosen similar for all the joints since the links have similar inertial parameters. Inserting (57) into (56) yields

$$\ddot{y}_i + k_d \dot{y}_i + k_p y_i = 0 \quad (58)$$

The tracking error dynamics of the i th joint angle (58) clearly has a globally exponentially stable equilibrium at the origin $(y_i, \dot{y}_i) = (0, 0)$, which implies that every i th control input (57) exponentially stabilizes the constraint manifold for the i th joint, and the control objective (37) is achieved.

In the following, we aim to stabilize the constraint manifold for the head angle, i.e. to make $(y_N, \dot{y}_N) \rightarrow (0, 0)$. The head angle error corresponds to the N th element of the controlled output vector (49), and its second-order time derivative (i.e. the head angle error dynamics) is given by

$$\ddot{y}_N = f_{\theta_N} + \beta_i \vartheta^i - d^2\Phi_N \quad (59)$$

Inserting ϑ_i from (57) into (59) gives

$$\ddot{y}_N = f_{\theta_N} + \sum_{i=1}^{N-1} \beta_i (d^2\Phi_i - k_p y_i - k_d \dot{y}_i) - d^2\Phi_N \quad (60)$$

which is equivalent to

$$\ddot{y}_N = f_{\theta_N} + \sum_{i=1}^{N-1} \beta_i \left(\ddot{\eta} \frac{\partial \Phi_i}{\partial \eta} + \dot{\eta}^2 \frac{\partial^2 \Phi_i}{\partial \eta^2} + \ddot{\phi}_o \frac{\partial \Phi_i}{\partial \phi_o} + \dot{\phi}_o^2 \frac{\partial^2 \Phi_i}{\partial \phi_o^2} - k_p y_i - k_d \dot{y}_i \right) - d^2 \Phi_N \quad (61)$$

For simplicity of notation, we denote the constraint function for the i th joint angle of the robot by $\Phi_i = S_i + \phi_o$, where $S_i = \alpha \sin(\eta + (i-1)\delta)$. Subsequently, based on the specified constraint functions in (47), i.e. since $\ddot{\eta} = 0$ and $\frac{\partial^2 \Phi_i}{\partial \phi_o^2} = 0$, we may write (61) as

$$\ddot{y}_N = f_{\theta_N} + \sum_{i=1}^{N-1} \beta_i (-\dot{\eta}^2 S_i + \ddot{\phi}_o - k_p y_i - k_d \dot{y}_i) - d^2 \Phi_N \quad (62)$$

In order to stabilize the constraint function $\Phi_N(p_y)$ for the head angle, we define the second-order time derivative of the augmented coordinate ϕ_o in the form of a dynamic compensator as

$$\ddot{\phi}_o = \left(\sum_{i=1}^{N-1} \beta_i \right)^{-1} \left(\sum_{i=1}^{N-1} \beta_i (\dot{\eta}^2 S_i + k_p y_i + k_d \dot{y}_i) + d^2 \Phi_N - f_{\theta_N} - k_{p,\theta_N} y_N - k_{d,\theta_N} \dot{y}_N \right) \quad (63)$$

where $k_{p,\theta_N} > 0$ and $k_{d,\theta_N} > 0$ are the head angle controller gains. Notice that since β_i is negative-valued in any configuration, (63) is globally well-defined. Through numerical simulations, we show that the states of the dynamic compensator (63), i.e. $(\phi_o, \dot{\phi}_o)$, remain bounded. By inserting (63) into (62), the error dynamics of the head angle takes the form

$$\ddot{y}_N + k_{d,\theta_N} \dot{y}_N + k_{p,\theta_N} y_N = 0 \quad (64)$$

which clearly has a globally exponentially stable equilibrium at the origin $(y_N, \dot{y}_N) = (0, 0)$. Consequently, we have that $(y_N, \dot{y}_N) \rightarrow (0, 0)$ from any initial condition, and the control objective (38) will be achieved.

Finally, we conjecture that while the output trajectories are evolving on the constraint manifold (48), the internal dynamics (31)-(32), which has the form

$$\ddot{p}_x = f_x(\Phi, p_x, p_y, \Phi', \dot{p}_x, \dot{p}_y) \quad (65)$$

$$\ddot{p}_y = f_y(\Phi, p_x, p_y, \Phi', \dot{p}_x, \dot{p}_y) \quad (66)$$

converge to and follow the desired planar path. Analytically investigating the convergence of the snake robot

position to the desired path is a topic of future work. As a preliminary support of this conjecture, we provide simulation and experimental results which show that the snake robot successfully converges to and follows the desired path.

Simulation results

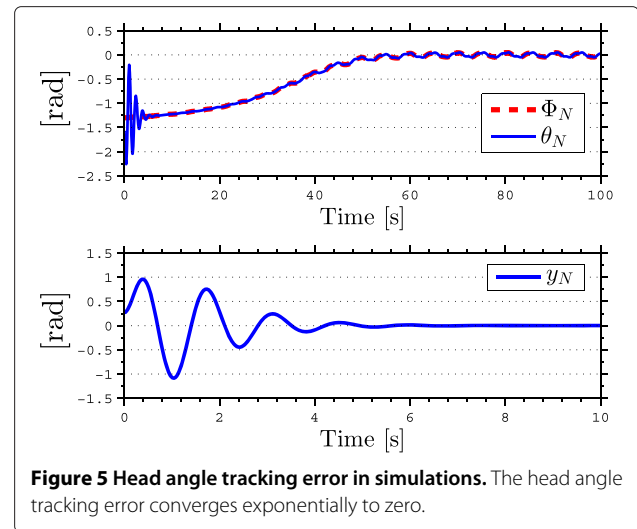
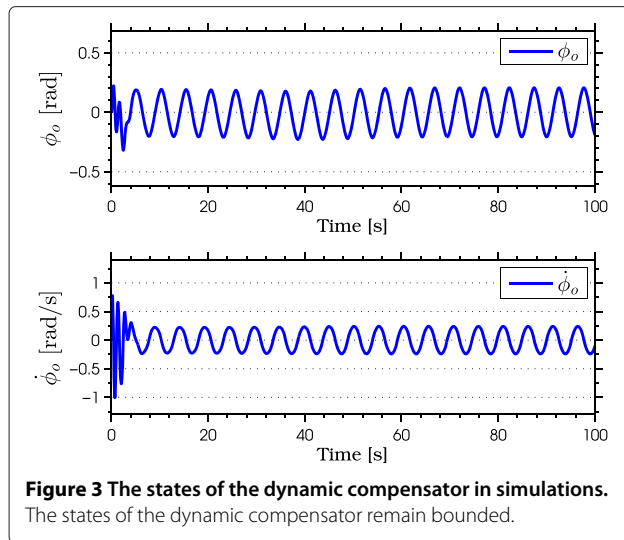
In this section, we present simulation results which illustrate the performance of the proposed path following controller. We considered a snake robot with $N = 11$ links, $m = 1$ kg, $l = 0.07$ m, and $J = 0.0016$ kg m² (Figure 2). The friction coefficients were $c_n = 10$ and $c_t = 1$. The parameters of the joint constraint functions (45) were $\alpha = \pi/6$ rad, $\eta = 70\pi t/180$ rad, and $\delta = 36\pi/180$ rad. The controller gains in (57) and (63) were tuned as $k_p = 10$, $k_d = 5$, $k_{p,\theta_N} = 20$, $k_{d,\theta_N} = 1$, and $\Delta = 1.4$ m. In order to calculate Φ_N and $\ddot{\phi}_o$, we employed the approach taken in [27] by passing Φ_N through a low-pass filter of the form

$$\dot{\Omega} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\psi_f \omega_n \end{bmatrix} \Omega + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \Phi_N \quad (67)$$

with natural frequency $\omega_n = \pi/2$ rad, damping ratio $\psi_f = 1$, and initial condition $\Omega(0) = [0, 0]^T$. As seen from the simulation results which are presented in Figures 3, 4, 5, 6, 7, the snake robot successfully converges to and follows the desired path. In particular, Figure 3 shows that the solutions of the dynamic compensator (63) remain bounded. Figure 4 shows that the joint angles track the reference angles provided by the constraint functions (45), while the tracking errors converge exponentially to zero. Figure 5 shows that the head angle tracks the reference head angle provided by the constraint function (46), while the tracking error converges to zero exponentially fast. Finally, Figure 6 shows that the CM of the robot converges to and follows the desired straight path. Moreover, in order to show the performance of the proposed tracking control law (57) in the presence of angular position measurement noise, we subjected every i th joint angle q_i to an additive noise by using Matlab function `randn()` which generates normally distributed pseudorandom numbers



Figure 2 An illustration of the experimental setup. Snake robot Wheeko was used for the experiments.



that can be considered as measurement noise for the joint angles. In particular, we added $\text{randn}(1)$ to each joint angle q_i in each integration step. The result of the simulation is presented in Figure 7, which shows that the joint tracking errors converge to a very small neighbourhood of zero in the presence of measurement noise.

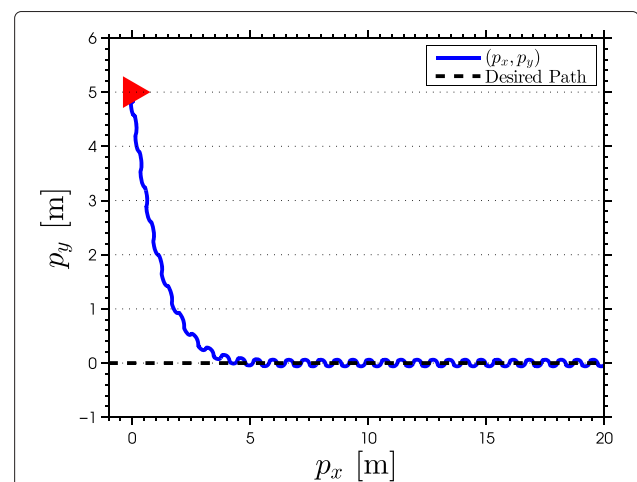
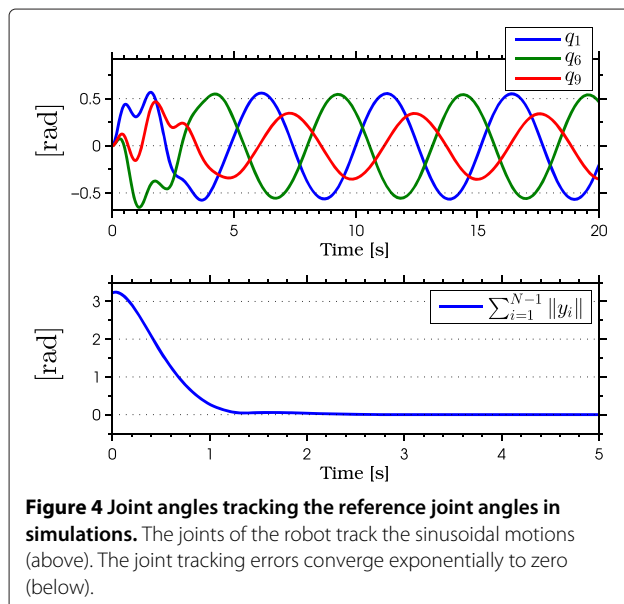
Experimental results

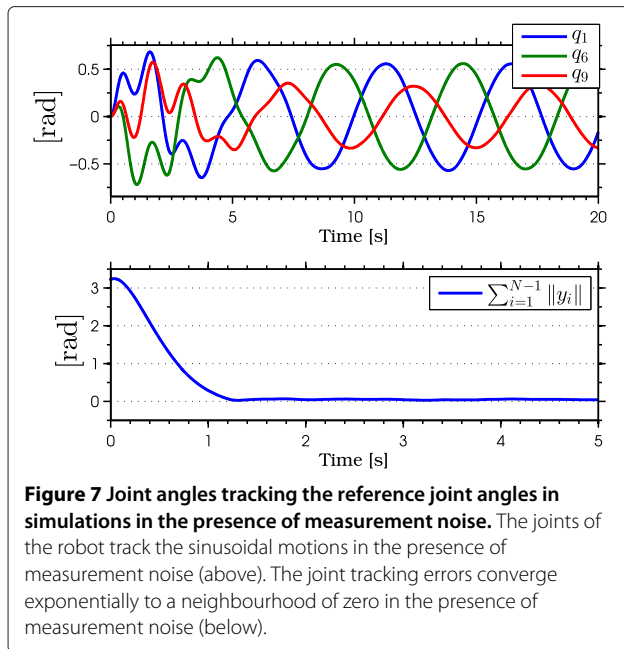
In this section, we present results from an experimental investigation of the real-time performance of the proposed control strategy using a mechanical snake robot.

Experimental setup

The experiment was carried out using the snake robot Wheeko [16]. The robot, which is shown in Figure 2, has

10 identical joint modules, i.e. $N = 11$ links. Each joint module is equipped with a set of passive wheels which give the robot anisotropic ground friction properties during motion on flat surfaces. The wheels are able to slip sideways and thus do not introduce nonholonomic velocity constraints in the system. Each joint is driven by a Hitec servo motor (HS-5955TG; Hitec RCD USA, Inc., Poway, CA, USA), and the joint angles are measured using magnetic rotary encoders. The motion of the snake robot was measured using a camera-based motion capture system from OptiTrack of type Flex 13 (NaturalPoint, Inc., Corvallis, OR, USA). The system consists of 16 cameras which are sampled at 120 frames per second and which allow reflective markers to be tracked on a submillimetre level. During the experiment, reflective markers were mounted on the head link of the snake robot in order to measure





the position $(p_{x,N}, p_{y,N})$ and orientation (θ_N) of the head. These measurements were combined with the measured joint angles (q_1, \dots, q_{N-1}) of the snake robot in order to measure the absolute link angles (1) and the position of the CM (p_x, p_y) of the robot. In order to obtain the derivatives of the reference head angle (46), we used the same technique as in the simulations, i.e. passing Φ_N through a low-pass filter of the form (67). The parameters of the low-pass filter were set to $\omega_n = \pi/2$ and $\psi_f = 1$.

In the following, we elaborate on a few adjustments that were made in the implemented path following controller in order to comply with the particular properties and capabilities of the physical snake robot employed in the experiment. We conjecture that these adjustments only marginally affected the overall motion of the robot. The successful path following behaviour of the robot demonstrated below supports this claim. Since the experimental setup only provided measurements of the joint angles and the position and orientation of the head link, we chose to implement the joint controller in (57) as

$$\ddot{\vartheta}_i = -k_p \dot{y}_i \quad (68)$$

where $i \in \{1, \dots, 10\}$. We conjecture that eliminating the joint angular velocity terms from (57) did not significantly change the dynamic behaviour of the system since the joint motion was relatively slow during the experiment. The main consequence of excluding the velocity terms from (57) is that we potentially introduce a steady-state error in the tracking of the joint angles. Consequently, since with the joint control law (68) the derivative terms in (63) are identically zero, they need not to be linearized in the head angle dynamics by the dynamic compensator.

As the result, we implemented the dynamic compensator of the form

$$\ddot{\phi}_o = \left(\sum_{i=1}^{N-1} \beta_i \right)^{-1} (-f_{\theta_N} + d^2 \Phi_N - k_{p,\theta_N} \dot{y}_N - k_{d,\theta_N} \dot{y}_N) - k_p \phi_o - k_d \dot{\phi}_o \quad (69)$$

where the controller gains were $k_{p,\theta_N} = 20$, $k_{d,\theta_N} = 1$, $k_p = 10$, and $k_d = 5$. We saturated the joint angle offset ϕ_o according to $\phi_o \in [-\pi/6, \pi/6]$, in order to keep the joint reference angles within reasonable bounds w.r.t the maximum allowable joint angles of the physical snake robot. Moreover, from Figure 2, it can be seen that the head link of the physical snake robot does not touch the ground since the ground contact points occur at the location of the joints. As a results, we implemented (69) with $f_{\theta_N} \equiv 0$. The solutions of the dynamic compensator (69) were obtained by numerical integration in *LabVIEW* which was used as the development environment.

We chose the look-ahead distance of the path following controller as $\Delta = 1.4$ m. The initial values for the configuration variables of the snake robot were $q_i = 0$ rad, $\theta_N = -\pi/2$ rad, $p_x = 0.3$ m, and $p_y = 1.7$ m, i.e. the snake robot was initially headed towards the desired path (the x -axis), and the initial distance from the CM to the desired path was 1.7 m. Furthermore, the parameters of the constraint functions for the joint angles, i.e. (45), were $\alpha = \pi/6$, $\eta = 70\pi t/180$, and $\delta = 36\pi/180$, while the ground friction coefficients were $c_t = 1$ and $c_n = 10$ (i.e identical to the simulation parameters).

Results and discussion

The results of the experiments are illustrated in Figures 8, 9, 10, 11, 12, 13. In particular, Figure 8 shows that the

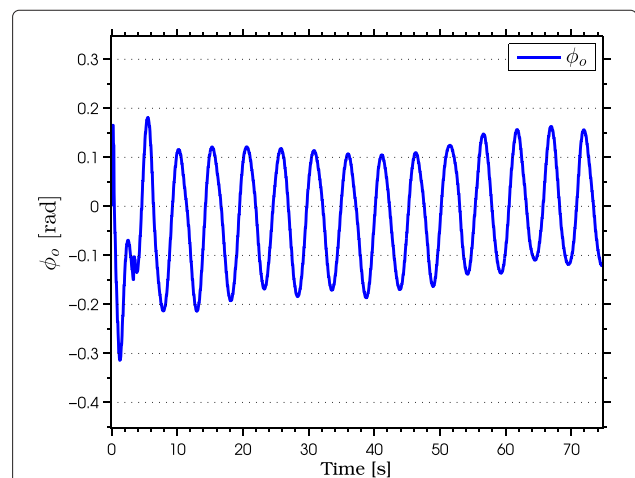
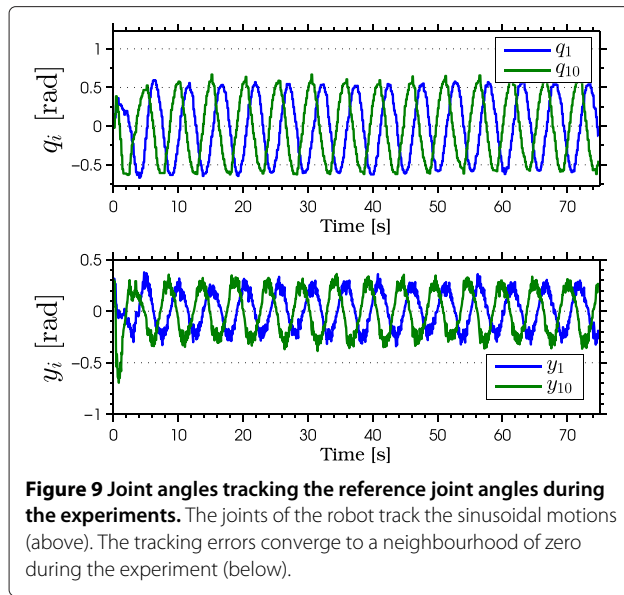
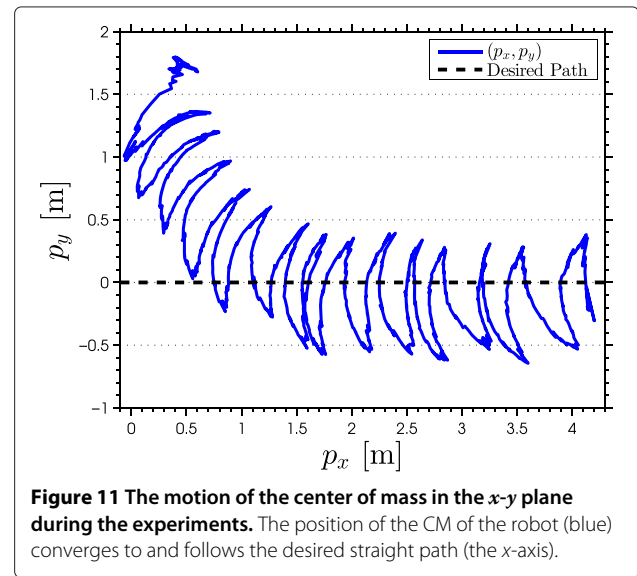


Figure 8 The solution of the dynamic compensator in experiments. The solution of the dynamic compensator remains bounded during the experiments.



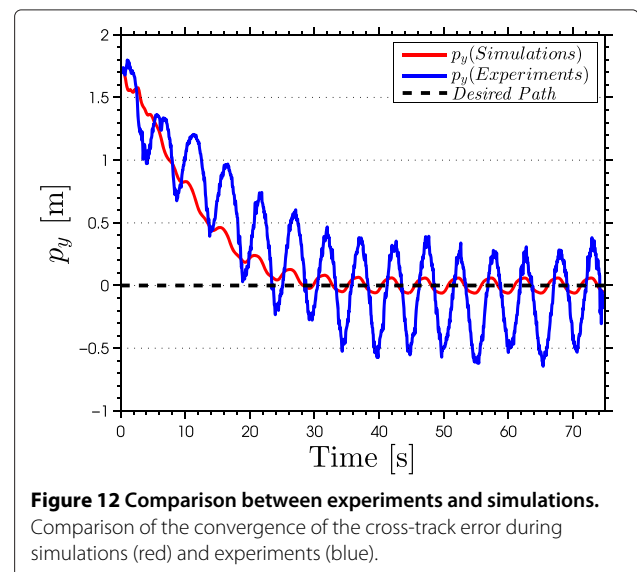
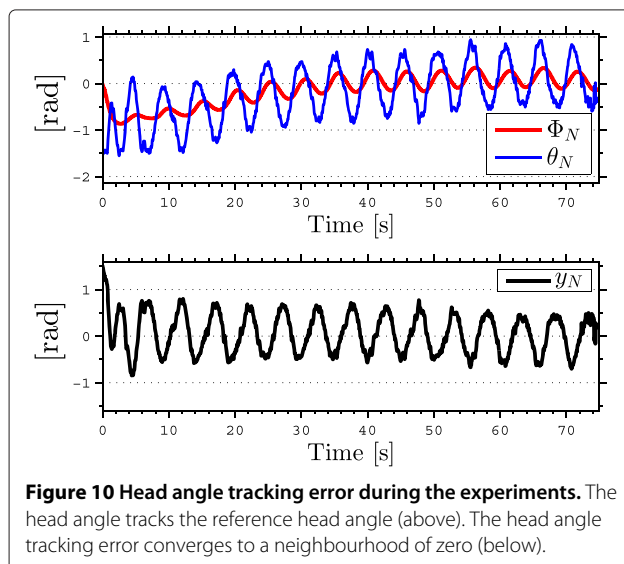
solution of the dynamic compensator remained bounded. Figure 9 shows that the joints of the robot tracked the sinusoidal reference angles provided by the constraint functions (45) and that the tracking error converged to a neighbourhood of the origin. As discussed above, this is probably due to the modification of the joint controller (68) due to the lack of velocity measurements in the lab. Figure 10 shows that the head angle of the robot tracked the reference head angle defined by the constraint function (46) and that the tracking error converged to a neighbourhood of the origin. Figure 11 shows the motion of the CM of the robot in the x - y plane, which converged to and followed the desired path. Figure 12 compares the motion of the CM during the simulation and the



experiment, which were performed using the same controller parameters in order to obtain comparable results. In particular, from Figure 12, it can be seen that the physical snake and the simulated snake follow almost the same path. However, due to precise measurement and a more accurate joint control law for the simulated snake, the path following error converges to a smaller neighbourhood of the origin. Figure 13 shows screenshots from a video recording of the experiment.

Conclusions

This paper has considered path following control of planar snake robots by using virtual holonomic constraints. The equations of motion of the snake robot were derived using



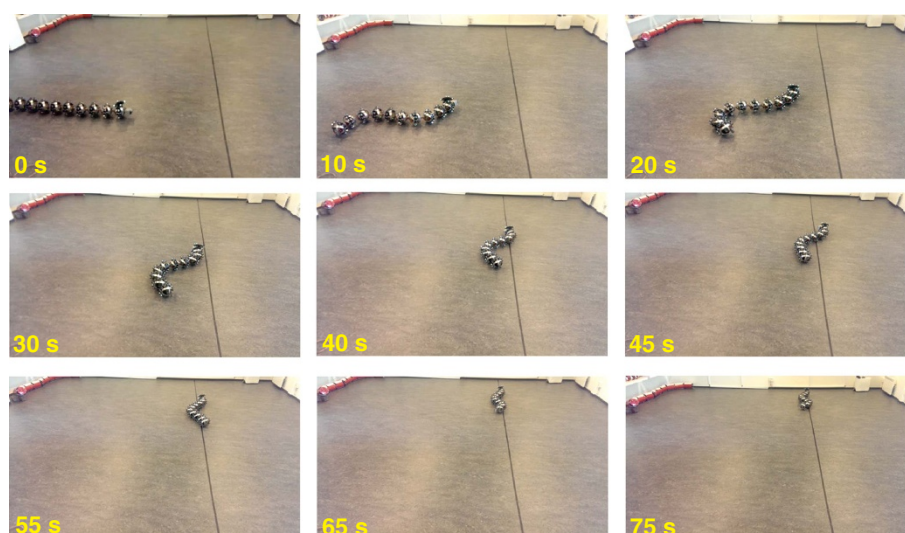


Figure 13 An image of the motion of the robot during the experiments. The robot converges to and follows the desired path.

a Lagrangian framework. We then introduced virtual holonomic constraints that defined the geometry of a constraint manifold for the robot. We showed that the constraint manifold can be made positively invariant by a suitable choice of feedback, and we designed an input-output feedback linearizing control law to exponentially stabilize the constraint manifold for the system. We presented simulation and experimental results which validated the theoretical design. In particular, the robot successfully converged to and followed a desired straight path.

As a topic of future work, we aim to prove the practical stability of the desired path with the proposed control approach. Furthermore, a formal proof for boundedness of the solutions of the dynamic compensator remains as a topic of future work. Moreover, application of the proposed control strategy for more complex paths such as curved paths, using different path following guidance laws, remains as a topic of future work.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors contributed equally and significantly in the writing of this paper. All authors read and approved the final manuscript.

Acknowledgements

This research was partly supported by the Research Council of Norway through the Centres of Excellence funding scheme, project no. 223254 AMOS and project no. 205622.

Author details

¹Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim 7491, Norway. ²Department of Applied Cybernetics, SINTEF ICT, Trondheim 7465, Norway.

Received: 14 April 2014 Accepted: 18 July 2014

Published: 28 August 2014

References

1. Liljebäck P, Stavdahl Ø, Beitnes A (2006) SnakeFighter - development of a water hydraulic fire fighting snake robot. In: Proc. IEEE international conference on control, automation, robotics, and vision ICARCV, Singapore
2. Wang Z, Appleton E (2003) The concept and research of a pipe crawling rescue robot. *Adv Robot* 17:4339–358
3. Fjerdigen SA, Liljebäck P, Transteth AA (2009) A snake-like robot for internal inspection of complex pipe structures (PIKo). In: Proc. IEEE/RSJ international conference on intelligent robots and systems, St. Louis, MO, USA
4. Dacic DB, Nesic D, Teel AR, Wang W (2011) Path following for nonlinear systems with unstable zero dynamics: an averaging solution. *IEEE Trans Automatic Control* 56:880–886
5. Hirose S (1993) Biologically inspired robots: snake-like locomotors and manipulators. Oxford University Press, Oxford, England
6. Matsuno F, Sato H (2005) Trajectory tracking control of snake robots based on dynamic model. In: Proc. IEEE international conference on robotics and automation, pp 3029–3034. 18–22 April 2005
7. Date H, Hoshi Y, Sampei M (2000) Locomotion control of a snake-like robot based on dynamic manipulability. In: Proc. IEEE/RSJ international conference on intelligent robots and systems, Takamatsu, Japan
8. Tanaka M, Matsuno F (2008) Control of 3-dimensional snake robots by using redundancy. In: Proc. IEEE international conference on robotics and automation, Pasadena, CA, pp 1156–1161
9. Ma S, Ohmameuda Y, Inoue K, Li B (2003) Control of a 3-dimensional snake-like robot. In: Proc. IEEE international conference on robotics and automation, vol. 2, Taipei, Taiwan, pp 2067–2072
10. Tanaka M, Matsuno F (2009) A study on sinus-lifting motion of a snake robot with switching constraints. In: Proc. IEEE international conference on robotics and automation, pp 2270–2275. 12–17 May 2009
11. Prautsch P, Mita T, Iwasaki T (2000) Analysis and control of a gait of snake robot. *Trans IEE J Ind Appl Soc* 120-D372–381
12. McIsaac K, Ostrowski J (2003) Motion planning for anguilliform locomotion. *IEEE Trans Robot Automation* 19:637–652
13. Hicks G, Ito K (2005) A method for determination of optimal gaits with application to a snake-like serial-link structure. *IEEE Trans Automatic Control* 50:1291–1306
14. Ma S, Ohmameuda Y, Inoue K (2004) Dynamic analysis of 3-dimensional snake robots. In: Proc. IEEE/RSJ international conference on intelligent robots and systems, pp 767–772. 28 Sept–2 Oct. 2004
15. Ma S (2001) Analysis of creeping locomotion of a snake-like robot. *Adv Robot* 15(2):205–224

16. Liljebäck P, Pettersen KY, Stavdahl Ø, Gravdahl JT (2013) Snake robots - modelling, mechatronics, and control. *Advances in industrial control*. Springer
17. Liljebäck P, Haugstuen IU, Pettersen KY (2012) Path following control of planar snake robots using a cascaded approach. *IEEE Trans Control Syst Technol* 20:11–126
18. Rezapour E, Pettersen KY, Liljebäck P, Gravdahl JT (2013) Path following control of planar snake robots using virtual holonomic constraints. Paper presented at the IEEE international conference on robotics and biomimetics, Shenzhen, China
19. Liljebäck P, Pettersen KY, Stavdahl Ø, Gravdahl JT (2013) Controllability and stability analysis of planar snake robot locomotion. *IEEE Trans Automatic Control* 56(6):1365–1380
20. Westervelt ER, Grizzle JW, Chevallereau C, Choi JH, Morris B (2007) *Feedback control of dynamic bipedal robot locomotion*. CRC press, Boca Raton
21. Maggiore M, Consolini L (2013) Virtual holonomic constraints for Euler-Lagrange systems. *IEEE Trans on Automatic Control* 58(4):1001–1008
22. Consolini L, Maggiore M (2010) Control of a bicycle using virtual holonomic constraints. In: *Proc. 49th IEEE conference on decision and control, Atlanta, Georgia, USA, December 15-17, 2010*
23. Shiriaev A, Perram JW, Canudas-de-Wit C (2005) Constructive tool for orbital stabilization of underactuated nonlinear systems: virtual constraints approach. *IEEE Trans Automatic Control* 50(8):1164–1176
24. Freidovich L, Robertsson A, Shiriaev A, Johansson R (2008) Periodic motions of the Pendubot via virtual holonomic constraints: theory and experiments. *Automatica* 44(3):785–791
25. Spong MW, Hutchinson S, Vidyasagar M (2006) *Robot modeling and control*. John Wiley and Sons, New York
26. Bullo F, Lewis A (2005) *Geometric control of mechanical systems*. Springer
27. Fossen TI (2002) *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. Trondheim, Norway, Marine Cybernetics

doi:10.1186/s40638-014-0003-6

Cite this article as: Rezapour *et al.*: Path following control of planar snake robots using virtual holonomic constraints: theory and experiments. *Robotics and Biomimetics* 2014 **1**:3.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com