Maneuvering Control of Planar Snake Robots Based on a Simplified Model

Ehsan Rezapour, Andreas Hofmann, and Kristin Y. Pettersen

***Abstract*— This paper considers maneuvering control of planar snake robots, for which the equations of motion are described based on a simplified model. In particular, we aim to stabilize a desired straight line path for the position of the center of mass of the robot, and to regulate the forward velocity of the robot along the path to a constant reference velocity. In order to solve this problem, we first stabilize a desired gait pattern for the fully-actuated body shape variables of the robot. Furthermore, we use the parameters of this gait pattern in the form of two dynamic compensators which control orientation and position of the robot in the plane. In particular, by solving the maneuvering problem, we control the body shape, orientation, and planar position of the robot.**

I. INTRODUCTION

The vast potential of snake robots for performing tasks in narrow and unstructured environments has made them an interesting alternative for many emerging industrial [1], search and rescue [2], and medical [3] applications. These robots have many degrees-of-freedom, which give them a robust motion even during the failure of some of their actuators. Moreover, they are characterized by a slender body, which enables them to traverse narrow environments. Snake robots are a class of underactuated mechanical systems, which are characterized by the lack of direct independent control input for some degrees-of-freedom (DOF) of the system. As a result, motion control of these robots propose many challenging control problems which require fundamental nonlinear control approaches. This makes them an interesting testbed for theoretical developments.

This paper considers maneuvering control of planar snake robots. In general, maneuvering control consists of two tasks [4]. The primary task is called the geometric task in which we aim to stabilize a desired planar path for the position variables of the robot. The secondary task is called the dynamic task in which we make the robot satisfy some dynamical constrains, e.g. a desired velocity profile, along the path. This problem is particularly interesting for snake robots, since it can automate their applications in the environments where the human presence is unsafe or impossible. However, to our best knowledge, the maneuvering control problem has never been considered for snake robots. This is probably due to the fact that the complicated dynamical behaviour of these robots gives rise to complex dynamic models. Consequently, the application of the model-based control approaches which rely on formal stability proofs is very restricted in the snake robots literature. In this paper, however, we carry out a model-based control design, and present formal stability proofs for the proposed maneuvering controller. The design is based on a simplified model of the snake robot dynamics, which was proposed in [9] where it was shown to provide an adequate approximation of the complex dynamic model.

Some previous literature investigate locomotion control problems for snake robots. Position and orientation control for a class of snake robots which are subject to nonholonomic velocity constraints is considered in [5-7]. In particular, these constraints allow the control input to be specified directly in terms of the desired propulsion of the snake robot. Locomotion control of snake robots without nonholonomic velocity constraints is only considered in a few previous works. Methods based on numerical optimal control are considered in [8] for determining optimal gaits during positional control of snake robots without nonholonomic velocity constraints. In [9,10], cascaded systems theory is employed to achieve path following control of a snake robot without nonholonomic velocity constraints which is described by a simplified model. Thus, [9,10] solves the primary task of the maneuvering problem but does not address the secondary task. In particular, [9,10] do not present any result regarding feedback control of the forward velocity or the position of the robot along the path. In [11], a virtual holonomic constraints (VHC) approach is used to control the orientation of the robot to a desired angle defined by a path following guidance law. In [12], direction following control of snake robots is considered, where the objective is to regulate the orientation and forward velocity of the robot to a constant reference. In contrast with [12], we use a similar idea for velocity control, however, we use a different dynamic model which is more amenable to model-based control design, which resolves the singularity problem in the control law derived in [12]. Furthermore, we present results regarding the position control for the robot along with the direction of motion.

In this paper we show that it is possible to combine the path following control approach of [10] and the velocity control approach in [20] in order to solve the maneuvering control problem. To this end, we replace the reference signal of [20] with a path following guidance law, which enables the robot to follow a time-varying reference angle, in contrast with the constant reference angle in [20]. Furthermore, we develop a new dynamic compensator which controls the position of the robot along the path in addition to its velocity as in [20]. Using this dynamic compensator also enables us to relax a restricting assumption on the forward velocity of the robot in the path following control approach given in [10].

E. Rezapour, and K. Y. Pettersen are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. emails: {ehsan.rezapour, kristin.y.pettersen}@itk.ntnu.no

The affiliation of A. Hofmann is shared between the Department of Engineering Cybernetics, Norwegian University of Science and Technology, and the Institute for Systems, Theory and Automatic Control, University of Stuttgart. email: andreas.hofmann@stud.uni-stuttgart.de

The paper is organized as follows. In Section II, we present a simplified dynamic model for the snake robot. In Section III, we state the control design objectives. In Section IV, propose a dynamic feedback control law in order to stabilize a desired gait pattern for the robot. In Section V, we design an orientation controller for the robot. In Section VI, we design a position and velocity controller for the robot. Finally in Section VII, simulation results are presented to validate the theoretical results.

II. MODELLING

In this section, we present a simplified model of the snake robot dynamics that can effectively be used for the model based control design in the subsequent sections. This model is previously presented in [9], where it is validated both through numerical simulations and real time experiments. Furthermore, in [9] it is shown that the fundamental properties of the simplified model such as stabilizability and controllability, are essentially the same as the more complex models presented in several previous works, see e.g. [5,11].

A. Notation

The following notations will be used in the model equations.

$$0_{N-1} = [0, \ldots, 0]^T \in \mathbb{R}^{N-1} \tag{1}$$

$$\bar{e} = [1, \ldots, 1]^T \in \mathbb{R}^{N-1} \tag{2}$$

$$A = \begin{bmatrix} 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N} \tag{3}$$

$$D = \begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N} \tag{4}$$

$$\overline{D} = D^T(DD^T)^{-1} \in \mathbb{R}^{N\times(N-1)} \tag{5}$$

Furthermore, N denotes the number of links, l denotes the length of each link, and m denotes the uniformly distributed mass of each link.

B. A simplified model of the snake robot locomotion

Kinematic and dynamic models of snake robots are previously derived in several works (see e.g. [5,9,11]). All these models share the same property that they are very complex for analytical investigations. The derivation of the simplified model of snake robot dynamics in [9] is motivated by the attractive idea that these complex dynamic models contain some nonlinear dynamics that are not essential to the overall locomotion of the robot. Moreover, proper approximations of these nonlinear dynamics with simpler mathematical descriptions can significantly simplify the analysis and model-based control design for snake robots. In particular, it is seen in [9] that lateral undulation mainly consists of link displacements which are transversal to the direction of motion. Moreover, it is this transversal link displacement that induces the forward motion of snake robots, cf. Fig. 2. The main idea behind the simplified model of the snake robot dynamics is to map the periodic body shape changes to forward propulsion, through mapping the rotational joint motion to translational link displacements, cf. Fig. 2. Since the translational displacements are in general less complex than rotational motion, this will simplify the resulting dynamic model of the robot.

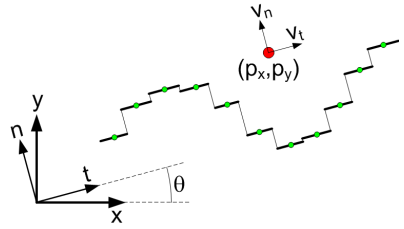

Fig. 1: Illustration of two coordinate frames used in the simplified model. The $x-y$ frame is fixed, and the $t-n$ frame is always aligned with the snake robot.

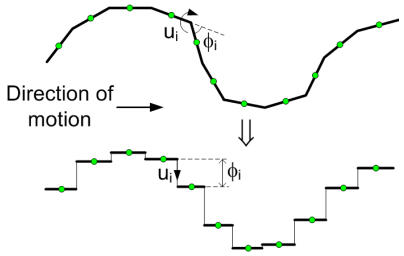

Fig. 2: The snake robot is modelled using a series of prismatic joints which move the robot forward by translational displacements. u_i is the exerted torque or force in the i-th joint of the robot.

C. Simplified kinematics and dynamics of the snake robot

In this subsection, we present the simplified kinematic and dynamic models of a snake robot without nonholonomic velocity constraints which moves on a horizontal and flat surface. Based on the illustrations of the robot in Fig. 1-2, we choose the elements of the vector of the generalized coordinates, which represent the configuration space $\mathcal{Q}$ of the robot, as

$$x = [\phi_1, \ldots, \phi_{N-1}, \theta, p_x, p_y]^T \in \mathbb{R}^{N+2} \tag{6}$$

where ϕ_i denotes the i-th joint coordinate, θ denotes the orientation, and (p_x, p_y) denotes the planar position of the center of mass (CM) of the robot. We denote the vector of the joint coordinates of the robot by $\phi = [\phi_1, \ldots, \phi_{N-1}]^T \in \mathbb{R}^{N-1}$. The elements of ϕ are called the body shape variables, which define the internal configuration of the robot. The vector of the generalized velocities is defined as the time-derivative of (6) as

$$\dot{x} = [v_{\phi_1}, \ldots, v_{\phi_{N-1}}, v_\theta, \dot{p}_x, \dot{p}_y]^T \in \mathbb{R}^{N+2} \tag{7}$$

We denote the vector of the joint velocities by $v_\phi = [v_{\phi_1}, \ldots, v_{\phi_{N-1}}]^T \in \mathbb{R}^{N-1}$. Since we aim to control the forward and normal velocities of the robot, we map the the inertial velocity of the CM of the robot to the $t-n$ frame which is always aligned with the robot, cf. Fig. 1, as

$$\dot{p}_x = v_t \cos(\theta) - v_n \sin(\theta) \tag{8}$$

$$\dot{p}_y = v_t \sin(\theta) + v_n \cos(\theta) \tag{9}$$

where $v_t \in \mathbb{R}$ and $v_n \in \mathbb{R}$ denote the tangential and normal components of the inertial velocity of the CM mapped into the direction of motion of the robot, respectively. The

simplified model of the robot w.r.t. $(x, \dot{x})$ is [9]

$$\dot{\phi} = v_\phi \quad (10)$$
$$\dot{\theta} = v_\theta \quad (11)$$
$$\dot{p}_t = v_t \quad (12)$$
$$\dot{p}_n = v_n \quad (13)$$
$$\dot{p}_x = v_t \cos(\theta) - v_n \sin(\theta) \quad (14)$$
$$\dot{p}_y = v_t \sin(\theta) + v_n \cos(\theta) \quad (15)$$
$$\dot{v}_\phi = -\frac{c_n}{m} v_\phi + \frac{c_p}{m} v_t A D^T \phi + \frac{1}{m} D D^T u \quad (16)$$
$$\dot{v}_\theta = -\lambda_1 v_\theta + \frac{\lambda_2}{N-1} v_t \bar{e}^T \phi \quad (17)$$
$$\dot{v}_t = -\frac{c_t}{m} v_t + \frac{2c_p}{Nm} v_n \bar{e}^T \phi - \frac{c_p}{Nm} \phi^T A \overline{D} v_\phi \quad (18)$$
$$\dot{v}_n = -\frac{c_n}{m} v_n + \frac{2c_p}{Nm} v_t \bar{e}^T \phi \quad (19)$$

where $c_n \in \mathbb{R}_{>0}$ and $c_t \in \mathbb{R}_{>0}$ denote the viscous friction coefficients in the normal and tangential direction of motion of the links, respectively. Furthermore, $\lambda_1 \in \mathbb{R}_{>0}$ and $\lambda_2 \in \mathbb{R}_{>0}$ are used to describe the mapping from the rotational motion to the prismatic motion (see [9]). Moreover, $c_p \in \mathbb{R}_{>0}$ is defined as $c_p = \frac{c_n - c_t}{2l}$. In order to linearize the dynamics of the fully-actuated DOF of the robot, i.e. the joint angles ϕ, we use the following change of the vector of the control inputs:

$$u = m(DD^T)^{-1}(\bar{u} + \frac{c_n}{m} v_\phi - \frac{c_p}{m} v_t A D^T \phi) \quad (20)$$

where $\bar{u} = [\bar{u}_1, \ldots, \bar{u}_{N-1}]^T \in \mathbb{R}^{N-1}$ is the new set of control inputs. Inserting (20) into (16), transforms the dynamics of the joint angles into

$$\dot{v}_\phi = \bar{u} \quad (21)$$

D. Model Transformation

In this subsection, we present a coordinate transformation which can simplify the model-based maneuvering control design for the snake robot. In particular, we note that the joint angles ϕ are present in both the dynamics of the angular velocity v_θ and the sideways velocity v_n. This coupling complicates the control design. In order to remove this coupling, we use the following coordinate transformation [9]

$$\bar{p}_y = p_y + \epsilon \sin(\theta), \quad \bar{v}_n = v_n + \epsilon v_\theta \quad (22)$$

where

$$\epsilon = -\frac{2(N-1)c_p}{Nm\lambda_2} \quad (23)$$

is a negative constant. This change of coordinates transforms the dynamics of the position of the robot into

$$\dot{\bar{p}}_y = v_t \sin(\theta) + v_n \cos(\theta) \quad (24)$$
$$\dot{\bar{v}}_n = X v_\theta - Y \bar{v}_n \quad (25)$$

where

$$X = \epsilon(\frac{c_n}{m} - \lambda_1), \quad Y = \frac{c_n}{m} \quad (26)$$

The joint angle coupling is removed from the dynamic model, and the resulting model is suitable for model-based maneuvering control design which is the subject of the subsequent sections.

III. CONTROL DESIGN OBJECTIVES

In this section, we formulate the maneuvering control objective for the proposed controllers in the subsequent sections. In general, the maneuvering problem consists of two tasks (see e.g. [4]). The first task is to converge to and follow a desired geometric path. This task is called the geometric task. The second task consists of satisfying dynamical constraints, e.g. a desired velocity profile, along the desired path. This task is called the dynamic task. In order to solve the maneuvering problem for the snake robot, we need to control the body shape, orientation, position, and velocity of the robot in the plane.

We start the control design formulation by defining a desired gait pattern, which is given by the vector function $\phi_{\text{ref}}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^{N-1}$, for the fully-actuated body shape variables of the robot. In particular, we aim to asymptotically stabilize $\phi \to \phi_{\text{ref}}$ such that

$$\lim_{t \to \infty} \|\phi(t) - \phi_{\text{ref}}(t)\| = 0 \quad (27)$$

which is equivalent to body shape control of the robot.

The second control objective concerns the orientation of the robot in the plane. In particular, given a reference time-varying orientation $\theta_{\text{ref}}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}$, we aim to asymptotically stabilize $\theta \to \theta_{\text{ref}}$ such that

$$\lim_{t \to \infty} \|\theta(t) - \theta_{\text{ref}}(t)\| = 0 \quad (28)$$

which is equivalent to orientation control of the robot.

The third control objective concerns the planar position and linear velocity of the CM of the robot, i.e. the maneuvering control. In order to formulate the maneuvering control objectives, we first define a desired straight line path, as a one dimensional manifold $\mathcal{P} \subset \mathbb{R}^2$, with coordinates in the $x - y$ plane given by the pair (p_{xd}, p_{yd}). These coordinates are parametrized by a time-dependent variable $\Theta(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Consequently, the desired path is defined as

$$\mathcal{P} = \{(p_{xd}(\Theta), p_{yd}(\Theta)) \in \mathbb{R}^2 : \Theta \geq 0\} \quad (29)$$

Furthermore, without loss of generality, we assume that the global x-axis is always aligned with the desired straight line path, i.e. $p_{yd}(\Theta) \equiv 0$. Thus, the geometric task is formulated as the convergence to the desired path such that

$$\lim_{t \to \infty} \|p_y(t)\| = 0 \quad (30)$$

To formulate the dynamic task, i.e. to regulate the linear velocity of the robot along the desired path to a desired constant velocity profile $v_{t,\text{ref}} \in \mathbb{R}_{\geq 0}$, we define a reference position along the desired path $p_{t,\text{ref}}(t) \in \mathbb{R}$ with $\dot{p}_{t,\text{ref}} = v_{t,\text{ref}}$. The dynamic task is defined as

$$\lim_{t \to \infty} \sup \|p_t(t) - p_{t,\text{ref}}(t)\| \leq \varepsilon_v \quad (31)$$

where $\varepsilon_v \in \mathbb{R}_{>0}$ is any constant. Achieving (30) and (31) is equivalent to position and velocity control for the robot. This can also be regarded as an output trajectory tracking objective, but since we focus on controlling the motion *along the path* we think that maneuvering control is the most adequate term. Finally, we require that all the solutions of the controlled system remain uniformly bounded.

IV. BODY SHAPE CONTROL

In this section, we propose a feedback control law for the body shape of the snake robot. In particular, we stabilize a desired gait pattern for the body shape variables, which induces lateral undulatory forward locomotion on the robot.

It is well-known [5] that the gait pattern lateral undulation for an N-link snake robot will be achieved if the i-th joint of the robot moves in accordance with the reference joint trajectory given by

$$\phi_{\text{ref},i}(t) = \alpha \sin(\omega t + (i-1)\delta) + \phi_o \tag{32}$$

where α denotes the amplitude of the sinusoidal joint motion, ω denotes the frequency of the joint oscillations, and δ denotes a phase shift which is used to keep the joints *out of phase*. Furthermore, ϕ_o is an offset term which can be used for controlling the orientation of the robot in the plane.

In [9], based on analytical investigations using the averaging theory, it was shown that the forward velocity of a snake robot which moves based on the lateral undulatory gait induced by (32), is affected by the gait parameters (α, ω, δ). Consequently, inspired by the work of [5,9], we introduce the following reference for the joint angles of the snake robot:

$$\phi_{\text{ref},i}(\lambda, \phi_o) = \alpha \sin(\lambda + (i-1)\delta) + \phi_o \tag{33}$$

where λ and ϕ_o are the solutions of two dynamic compensators which we will use to control the position and orientation of the robot, respectively. A similar idea for the body shape control is previously presented in [20], however, in contrast, here we use a more complex dynamic compensator for λ which enables us to control the position of the robot along the desired path in addition to the forward velocity of the robot, and thus to solve the maneuvering control problem.

A. Virtual Holonomic Constraints

VHC, see e.g. [15,16], are relations of the form $\Phi : \mathcal{Q} \to \mathbb{R}$ that are called constraint functions, which can be made invariant by the actions of a feedback controller [16]. In this case we say that the VHC are enforced. 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 [15].

Inspired by the idea of VHC that has effectively been used for motion control of mechanical systems (see e.g. [15,16] for various examples), we consider (33) as a VHC for the body shape variables of the snake robot. Furthermore, these VHC will be enforced through the control input $\bar{u}$ in (21). In particular, (33) is a dynamic VHC in that it depends on the state evolution of two dynamic compensators.

Associated with constraint functions (33), is the following constraint manifold

$$\begin{aligned}\Gamma = \{(x, \dot{x}, \phi_o, \dot{\phi}_o, \lambda, \dot{\lambda}) \in \mathbb{R}^{2N+8} : \\ \phi_i = \phi_{\text{ref},i}(\lambda, \phi_o),\ v_{\phi_i} = \dot{\lambda}\frac{\partial \phi_{\text{ref},i}}{\partial \lambda} + \dot{\phi}_o \frac{\partial \phi_{\text{ref},i}}{\partial \phi_o}\}\end{aligned} \tag{34}$$

where $i \in \{1, \ldots, N-1\}$. Our control design approach for the snake robot is given in the following four steps:

1) In the first step, we use the control input $\overline{u}$ to stabilize the solutions of the joint angles dynamics (21) to the constraint manifold (34). This will induce a forward motion based on the gait pattern lateral undulation on the robot.
2) In the second step, we reduce the dynamics of the system to the invariant constraint manifold (34), where we use ϕ_o as an additional control term, which will be used to control the orientation of the robot.
3) In the third step, we use the frequency of the periodic body motion, i.e. the gait pattern, as an additional control term to control forward velocity of the robot. This solves the dynamic task.
4) In the fourth step, we use the reference orientation such that the convergence of the position of the robot to the desired path is guaranteed. This solves the geometric task.

B. Enforcing the VHC for the Shape Variables of the Robot

In order to stabilize the constraint manifold for the shape variables ϕ, we define the following controlled output vector

$$\tilde{\phi} = [\phi_1 - \phi_{\text{ref},1}, \ldots, \phi_{N-1} - \phi_{\text{ref},N-1}]^T \in \mathbb{R}^{N-1} \tag{35}$$

which yields a well-defined vector relative degree $\{2, \ldots, 2\}$ everywhere on the configuration space. Consequently, we can stabilize the constraint manifold using an input-output linearizing feedback control law [16] which we define as

$$\bar{u} = \ddot{\phi}_{\text{ref}} - K_d \dot{\tilde{\phi}} - K_p \tilde{\phi} \tag{36}$$

where $K_p = \text{diag}\{k_{p_i}\}_{i=1}^{N-1}$ and $K_d = \text{diag}\{k_{d_i}\}_{i=1}^{N-1}$ denote the positive definite diagonal matrices of the joint proportional and derivative controller gains, respectively. Note that (36) is a dynamic feedback control law, in that it depends on the state evolution of two dynamic compensators which will be defined later. By inserting (36) into (21), the error dynamics equation for the joint angles takes the form

$$\ddot{\tilde{\phi}} + K_d \dot{\tilde{\phi}} + K_p \tilde{\phi} = 0 \tag{37}$$

which clearly has a globally exponentially stable equilibrium at the origin $(\tilde{\phi}, \dot{\tilde{\phi}}) = (0_{N-1}, 0_{N-1})$. This implies that joint angle errors converge to zero exponentially, i.e. the constraint manifold is a globally exponentially stable manifold for (21), and the control objective (27) will be achieved.

V. ORIENTATION CONTROL

In this section, we control the orientation of the robot to a reference angle defined by a path following guidance law by using ϕ_o as an additional control term on the exponentially stable constraint manifold. The major contribution of the orientation controller w.r.t. [20] is that here we stabilize the orientation of the robot to a time-varying reference which allows convergence of the robot to a desired path rather than a constant angle as in [20].

A. The Path Following Guidance Law

In this subsection, we define a reference orientation for the robot through a Line-of-Sight (LOS) guidance law. Guidance-based control strategies are a common approach for e.g. marine control systems, (see e.g. [13]). These control strategies are based on defining a reference orientation angle for the vehicle through a guidance law, and subsequently designing a controller to track this angle.

Since we assumed that the global x-axis is always aligned with the desired path (29), then the position $\bar{p}_y$ of the robot along the y-axis defines the shortest distance between the robot and the desired path, which is often referred to as the *cross-track* error. We then define the LOS path following guidance law, giving the reference orientation for the robot, as a function of the cross-track error:

$$\theta_{\text{ref}} = -\text{atan2}(\frac{\bar{p}_y}{\Delta}) \tag{38}$$

where $\Delta > 0$ is a design parameter that is called the look-ahead-distance. The idea of the LOS guidance law (38) is that steering the orientation of the snake robot such that the robot is headed towards a point which is located at a distance Δ ahead of the robot on the desired path, will make the position of the CM of the robot converge to the desired straight path. A similar guidance law for snake robots is used in [9] where the path following control of snake robots is considered. In contrast, here we relax a restricting assumption in [9] on the forward velocity of the robot by regulating the forward velocity of the robot by using a dynamic compensator, and we solve the maneuvering control problem.

B. Stabilizing the Reference Orientation

In this section, we control the orientation of the robot by using $\ddot{\phi}_o$ as an additional control input on the exponentially stable constraint manifold. To this end, we define the orientation error as

$$\tilde{\theta} = \theta - \theta_{\text{ref}} \tag{39}$$

Furthermore, we derive the orientation error dynamics of the robot evaluated on the constraint manifold. This can be done by writing (11,17) in the error coordinates $(\tilde{\phi}_1, \dots, \tilde{\phi}_{N-1}, \tilde{\theta})$, and then reducing them to the invariant manifold where $(\tilde{\phi}, \dot{\tilde{\phi}}) = (0_{N-1}, 0_{N-1})$. The resulting error dynamics has the form

$$\ddot{\tilde{\theta}} = -\lambda_1 \dot{\tilde{\theta}} - \lambda_1 \dot{\theta}_{\text{ref}} + \frac{\lambda_2}{N-1} v_t \bar{e}^T S + \lambda_2 v_t \phi_o - \ddot{\theta}_{\text{ref}} \tag{40}$$

where $S \in \mathbb{R}^{N-1}$ denotes the following vector:

$$S = [\alpha \sin(\lambda), \dots, \alpha \sin(\lambda + (N-1)\delta)]^T \in \mathbb{R}^{N-1} \tag{41}$$

In order to control the orientation of the robot, we use $\ddot{\phi}_o$ as a dynamic compensator. To this end, we take the time-derivatives of (40) until the control input $\ddot{\phi}_o$ appears. The resulting dynamics is of the form

$$\tilde{\theta}^{(4)} = -\lambda_1 \tilde{\theta}^{(3)} + \psi_1(v_t, \phi_o)\ddot{\phi}_o + \psi_2(v_t, v_n, \phi_o, \dot{\phi}_o, \lambda, \dot{\lambda}, \ddot{\lambda}) \tag{42}$$

Note that it is straightforward to derive $\psi_1(\cdot)$ and $\psi_2(\cdot)$ by taking the time-derivatives of (40), but due to space restrictions, we write them in the symbolic form. We define the input-output linearizing control law

$$\ddot{\phi}_o = \frac{1}{\psi_1}\left(\lambda_1 \tilde{\theta}^{(3)} - \psi_2 + \sigma\right) \tag{43}$$

where $\sigma \in \mathbb{R}$ is a new control input which we define as

$$\sigma = -k_3 \tilde{\theta}^{(3)} - k_2 \tilde{\theta}^{(2)} - k_1 \tilde{\theta}^{(1)} - k_0 \tilde{\theta} \tag{44}$$

where $k_0, k_1, k_2, k_3 > 0$ denote the orientation controller gains. It can be numerically verified that ψ_1 is bounded away from zero except for very small values of the forward velocity v_t, and this agrees well with the fact that the orientation is not controllable if the forward velocity of the snake robot is zero [9]. We stabilize the origin, i.e. $\tilde{\theta}^{(i)} = 0$ for all $i \in \{0, \dots, 4\}$, of the orientation error dynamics by properly choosing the gains k_i. Furthermore, we show the boundedness of the solutions of the dynamic compensator (43) through numerical simulations, however, a formal proof of this boundedness remains as a topic of future work. We denote this bound by $\|[\phi_o, \dot{\phi}_o]\| \leq \varepsilon$ where $\varepsilon \in \mathbb{R}_{>0}$. In particular, we denote the upper-bound on each i-th reference joint angle, which is composed of a bounded sinusoidal part and the offset term ϕ_o, as

$$\|\phi_{\text{ref},i}\| \leq \varepsilon^* \tag{45}$$

where $\varepsilon^* \in \mathbb{R}_{>0}$ is a constant.

VI. MANEUVERING CONTROL

In this section, we perform the dynamic task for the maneuvering problem by utilizing the idea of velocity control given in [12]. To this end, we derive a dynamic compensator which controls the velocity and position of the robot along the desired path by using the frequency of the joint oscillations as an additional control term. Moreover, following [9], we perform the geometric task by using the look-ahead-distance as a control term. However, using our velocity controller we relax the restricting assumption in [9] on forward velocity of the robot.

As a preliminary, in the next theorem we show that the normal velocity v_n of the robot is uniformly bounded.

Theorem I. *Under the controllers (36,43), the normal velocity v_n of the robot is uniformly bounded.*

Proof: In order to show the boundedness of normal velocity v_n, we select the Lyapunov function

$$V = \frac{1}{2}\overline{v}_n^2 \tag{46}$$

Using (25), the time-derivative of (46) is given by

$$\dot{V} = \overline{v}_n \dot{\overline{v}}_n = \overline{v}_n \left(X v_\theta - Y \overline{v}_n\right) = X \overline{v}_n v_\theta - Y \overline{v}_n^2 \tag{47}$$

For the first RHS term, we apply Young's inequality [18] where we have that

$$ab \leq \frac{\gamma a^2}{2} + \frac{b^2}{2\gamma} \tag{48}$$

where $\gamma \in \mathbb{R}_{>0}$ is any positive constant. Consequently, for (47) we have that

$$\dot{V} \leq -Y\overline{v}_n^2 + |X|(\frac{\gamma \overline{v}_n^2}{2} + \frac{v_\theta^2}{2\gamma}) = (-Y + \frac{\gamma |X|}{2})\overline{v}_n^2 + \frac{|X| v_\theta^2}{2\gamma} \tag{49}$$

From the stability result of the previous section and assuming that v_t has no finite escape time (the proof follows from the linear growth condition on the closed-loop system with the velocity dynamic compensator which will be given in (56)), we conclude that the second term in RHS of (49) is uniformly bounded, i.e. since $v_\theta \to \dot{\theta}_{\text{ref}}$ which is bounded. We denote this bound by

$$\frac{|X| v_\theta^2}{2\gamma} \leq \beta_1 \quad \forall t \geq 0 \tag{50}$$

where $\beta_1 \in \mathbb{R}_{>0}$ is a constant. By (26) Y is positive. Suppose

now that we choose γ sufficiently small so that the coefficient of $\overline{v}_n^2$ is negative. In this case we conclude that there is a positive constant β_2 such that

$$\dot{V} \leq -\beta_2 V + \beta_1 \tag{51}$$

Therefore, it is straightforward to conclude from the Comparison Lemma [14] that

$$V(t) \leq e^{-\beta_2 t} V(0) + (\beta_1/\beta_2) \tag{52}$$

From (52) we conclude that $\overline{v}_n$ remains bounded, and converges to a ball of radius $\sqrt{2\frac{\beta_1}{\beta_2}}$. Since $v_n = \overline{v}_n + \epsilon v_\theta$, and v_θ is bounded, this also implies that v_n remains bounded. We denote this bound by $\|v_n\| \leq \epsilon_n$. ■
We use this fact in the stability proofs in the next subsection.

A. Dynamic Task

In this subsection, we solve the dynamic task by controlling the position and velocity of the robot along the path. In particular, we use the frequency of the joint angle oscillations as an additional control term in order to regulate the forward velocity of the robot to a constant reference. To this end, we define the tangential position and velocity errors as

$$\tilde{p}_t = p_t - p_{t,\text{ref}}, \quad \tilde{v}_t = v_t - v_{t,\text{ref}} \tag{53}$$

Using (53), we derive the position and velocity error dynamics evaluated on the constraint manifold (34) as

$$\dot{\tilde{p}}_t = \tilde{v}_t \tag{54a}$$

$$\dot{\tilde{v}}_t = -\frac{c_t}{m}(\tilde{v}_t + v_{t,\text{ref}}) + \frac{2c_p}{Nm} v_n \bar{e}^T \Phi_{\text{ref}} + \eta \left(\dot{\lambda} C + \dot{\phi}_o \bar{e}\right) \tag{54b}$$

where C, Φ_{ref}, and η denote the following vector-valued functions, respectively,

$$C = [\alpha\cos(\lambda), \ldots, \alpha\cos(\lambda + (i-1)\delta)]^T \in \mathbb{R}^{N-1} \tag{55a}$$

$$\Phi_{\text{ref}} = [\phi_{\text{ref},1}, \ldots, \phi_{\text{ref},N-1}]^T \in \mathbb{R}^{N-1} \tag{55b}$$

$$\eta = -\frac{c_p}{Nm} \Phi_{\text{ref}}^T A \overline{D} \in \mathbb{R}^{N-1} \tag{55c}$$

In the following, we use

$$u_\lambda = \ddot{\lambda} \tag{56}$$

as a control input to stabilize the origin $(\tilde{p}_t, \tilde{v}_t) = (0, 0)$ of (54). In particular, we iteratively introduce control-Lyapunov functions (CLF) borrowing the techniques of backstepping (see e.g. [14]). We select the first CLF of the form

$$V_1 = \frac{1}{2}\tilde{p}_t^2 \tag{57}$$

The time-derivative of (57) along the solutions of (54) is

$$\dot{V}_1 = \tilde{p}_t \tilde{v}_t \tag{58}$$

We take $\tilde{v}_t$ as a virtual control input which we utilize to make (58) negative. In particular, we define

$$\tilde{v}_t = -k_{z_0} \tilde{p}_t \tag{59}$$

where $k_{z_0} > 0$ is a constant gain. We define the error variable

$$z_1 = \tilde{v}_t + k_{z_0} \tilde{p}_t \tag{60}$$

that we aim to drive to zero. Thus, we can rewrite (58) as

$$\dot{V}_1 = -k_{z_0} \tilde{p}_t^2 + z_1 \tilde{p}_t \tag{61}$$

To perform backstepping for z_1, we write the error dynamics for the error variable which has the form

$$\dot{z}_1 = \dot{\tilde{v}}_t + k_{z_0} \tilde{v}_t \tag{62}$$

We choose an augmented CLF of the form

$$V_2 = V_1 + \frac{1}{2} z_1^2 \tag{63}$$

The time-derivative of V_2 along the solutions of (54) is

$$\begin{aligned} \dot{V}_2 =& -k_{z_0} \tilde{p}_t^2 + z_1 \left(\tilde{p}_t + \dot{\tilde{v}}_t + k_{z_0} \tilde{v}_t\right) \\ =& -k_{z_0} \tilde{p}_t^2 + z_1 (\tilde{p}_t - \frac{c_t}{m} z_1 - \frac{c_t}{m} v_{t,\text{ref}} + \frac{c_t}{m} k_{z_0} \tilde{p}_t \\ &+ \frac{2c_p}{Nm} v_n \bar{e} \Phi_{\text{ref}} + \eta C \dot{\lambda} + \eta \bar{e} \dot{\phi}_o + k_{z_0} \tilde{v}_t) \end{aligned} \tag{64}$$

We denote

$$\delta_1(\phi_o, \lambda) = \eta C \tag{65}$$

It can be numerically verified that $\delta_1(\cdot)$ is uniformly bounded away from zero, and this is because of the phase shift between the link references in (33). We take $\dot{\lambda}$ as a virtual control input that we use to make (64) negative:

$$\begin{aligned} \dot{\lambda} =& \frac{1}{\delta_1}(-\tilde{p}_t + \frac{c_t}{m} v_{t,\text{ref}} - \frac{c_t}{m} k_{z_0} \tilde{p}_t - \frac{2c_p}{Nm} v_n \bar{e} \Phi_{\text{ref}} \\ &- k_{z_0} \tilde{v}_t - k_{z_1} z_1) \end{aligned} \tag{66}$$

where $k_{z_1} > 0$ is a constant gain. For simplicity, we denote

$$\begin{aligned} \delta_2(\phi_o, \lambda, \tilde{p}_t, \tilde{v}_t) = \frac{1}{\delta_1}(&-\tilde{p}_t + \frac{c_t}{m} v_{t,\text{ref}} - \frac{c_t}{m} k_{z_0} \tilde{p}_t - \\ &\frac{2c_p}{Nm} v_n \bar{e} \Phi_{\text{ref}} - k_{z_0} \tilde{v}_t - k_{z_1} z_1) \end{aligned} \tag{67}$$

We define the second error variable as

$$z_2 = \dot{\lambda} - \delta_2 \tag{68}$$

which we aim to drive to zero. Inserting $\dot{\lambda} = z_2 + \delta_2$ into (64) yields

$$\dot{V}_2 = -k_{z_0} \tilde{p}_t^2 - (\frac{c_t}{m} + k_{z_1}) z_1^2 + z_1 z_2 \delta_1 + z_1 \eta \bar{e} \dot{\phi}_o \tag{69}$$

To perform backstepping for z_2, we write the error dynamics for the error variable z_2 which has the form

$$\dot{z}_2 = u_\lambda - \dot{\delta}_2 \tag{70}$$

We choose the augmented CLF in the form

$$V_3 = V_2 + \frac{1}{2} z_2^2 \tag{71}$$

The time-derivative of V_3 along the solutions of (54) is

$$\dot{V}_3 = -k_{z_0} \tilde{p}_t^2 - (\frac{c_t}{m} + k_{z_1}) z_1^2 + z_2 (z_1 \delta_1 + u_\lambda - \dot{\delta}_2) + z_1 \eta \bar{e} \dot{\phi}_o \tag{72}$$

We define the control input u_λ in the form

$$u_\lambda = -z_1 \delta_1 + \dot{\delta}_2 - k_{z_2} z_2 \tag{73}$$

where $k_{z_2} > 0$ is a constant gain. Inserting (73) into (72) yields

$$\dot{V}_3 = -k_{z_0} \tilde{p}_t^2 - (\frac{c_t}{m} + k_{z_1}) z_1^2 - k_{z_2} z_2^2 + z_1 \eta \bar{e} \dot{\phi}_o \tag{74}$$

Only the last term in (74) has indefinite sign. For this term, we apply Young's inequality and we can write

$$|z_1||\eta\bar{e}\dot{\phi}_o| \leq \zeta\left(\frac{\gamma_1}{2}+\frac{z_1^2}{2\gamma_1}\right) \tag{75}$$

where $\gamma_1 > 0$ and

$$\zeta = |-\frac{c_p}{Nm}\overline{e}^T A\overline{D}\overline{e}|\varepsilon\varepsilon^* \tag{76}$$

is a positive constant. Consequently, (74) will be of the form

$$\dot{V}_3 \leq -k_{z_0}\tilde{p}_t^2 - \left(\frac{c_t}{m}+k_{z_1}-\frac{\zeta}{2\gamma_1}\right)z_1^2 - k_{z_2}z_2^2 + \frac{\gamma_1\zeta}{2} \tag{77}$$

From (77) we conclude that there exists a sufficiently small positive constant $\beta \in \mathbb{R}_{>0}$ such that

$$\dot{V}_3 \leq -\beta V_3 + \frac{\gamma_1\zeta}{2} \tag{78}$$

Consequently, a straightforward application of the Comparison Lemma, see e.g. [14], implies that

$$V_3(t) \leq V_3(0)e^{-\beta t} + \frac{\gamma_1\zeta}{2\beta} \tag{79}$$

From (79) we conclude that V_3 converges to a ball of radius $\frac{\gamma_1\zeta}{2\beta}$. Furthermore, because of (71), z_2, z_1, and $\tilde{p}_t$ converge to a ball of radius $\sqrt{\frac{\gamma_1\zeta}{2\beta}}$, and because of (52,68,79) $\dot{\lambda}$ remains bounded. Moreover, we can drive the position and velocity errors to any arbitrarily small neighbourhood of the origin $\epsilon_v \in \mathbb{R}_{>0}$ by taking

$$\gamma_1 = (2\beta\epsilon_v^2/\zeta) \tag{80}$$

and a sufficiently large k_{z_1}, i.e. such that the coefficient of z_1^2 in (77) will be negative. This implies that the origin $(\tilde{p}_t, \tilde{v}_t) = (0, 0)$ of (54) is practically stable, and the control objective (31) will be achieved.

B. Geometric Task

In this subsection, we use the look-ahead-distance Δ as an additional control term to stabilize the desired path $\mathcal{P}$ such that $(\overline{p}_y, \overline{v}_n) \to (0, 0)$. To this end we define the position and velocity cross track errors

$$\tilde{p}_y = \overline{p}_y, \quad \tilde{v}_n = \overline{v}_n \tag{81}$$

which we would like to drive to zero and thereby achieve control objective (30). This will imply that the robot will converge to the global x-axis which we defined as the desired straight line path. A similar approach is used in [9], where cascaded systems theory is used to show that for a properly chosen look-ahead distance Δ, the desired path is globally asymptotically stable for (24-26). However, [9] does not provide any results regarding the forward velocity of the robot, and only assumes that the tangential velocity of the robot is inside a positive constant range, i.e. $v_t \in [V_{\max}, V_{\min}]$ where $V_{\max}$ and $V_{\min}$ denote the maximum and minimum of the forward velocity, respectively. However, this is a restricting assumption since the velocity can change sign or go unstable. In the previous subsection, we used a dynamic compensator which enables us to control the forward velocity of the robot, and thus we relax this assumption. The following theorem is a reformed version of the theorem given in [9].

Theorem II. *The controller defined by (36,43,56) globally asymptotically and locally exponentially stabilizes the desired straight path $\mathcal{P}$ (29) for the dynamic system (10-19), provided that the look-ahead distance Δ satisfies*

$$\Delta > \frac{|X|}{|Y|}\left(1+\frac{V_{\max}}{V_{\min}}\right) \tag{82}$$

The proof of Theorem II follows from the proof of [9, Th. 8.2], together with the proof of stabilization of $\tilde{v}_t$ given in Section VI.A.

VII. SIMULATION RESULTS

To illustrate the performance of the proposed maneuvering controller, we present simulation results in this section. We considered a snake robot with $N = 10$ links of length $l = 0.14$ m, mass $m = 1$ kg, and anisotropic ground friction coefficients $c_t = 1$ and $c_n = 3$. We chose the rotation parameters such that the simplified model qualitatively and quantitatively behaves similar to the complex model derived in [9]. In particular, we defined $\lambda_1 = 0.5$ and $\lambda_2 = 20$. The gait parameters were $\alpha = 4.5$ cm and $\delta = 40\pi/180$, and the joint controller gains were $k_p = 20$ and $k_d = 5$. To stabilize the orientation error dynamics we employed a quadratic cost function and the state-feedback law (43) as a linear quadratic regulator in which gains were obtained with the solution of the corresponding Riccati equation, see e.g. [19]. The gains of the position tracking controller were tuned to $k_{z_0} = 0.5$, $k_{z_1} = 0.5$ and $k_{z_2} = 0.1$. The reference forward velocity was $v_{t,\text{ref}} = 0.2$ m/s, and the position reference was $p_{t,\text{ref}} = \int v_{t,\text{ref}}dt$. The look ahead distance was chosen as twice the length of the robot $\Delta = 2.8$ m. To avoid singularities the the initial tangential velocity was set to $v_t(0) = 0.1$ m/s, see the arguments after (44). All the other states were set initially to zero. The results of the simulation are shown in Figs. 3-7. Fig. 3 illustrates the exponential stability of the joint error, hence the joints move in accordance with the lateral undulatory gait provided by (33). The convergence of the orientation to the reference orientation and the corresponding error can be seen in Fig. 4. Fig. 5 shows that the controller regulates the velocity to the desired velocity. Fig. 6 shows that the joint oscillation frequency $\dot{\lambda}$ converges to a positive constant and the solution of the orientation controller ϕ_o is bounded and becomes zero when the robot is on the path. Finally, Fig. 7 illustrates that the snake robot approaches the desired path.

VIII. CONCLUSION

We considered maneuvering control of planar snake robots. First we enforced VHC for the body shape variables of the robot. These constraints were inspired by the well-known reference joint angle trajectories which induce lateral undulatory gait pattern on snake robots. Furthermore, we removed the explicit time-dependence of the reference joint angles, and rather, made them a function of the states of two compensators. Subsequently, we reduced the dynamics of the system to the invariant constraint manifold, and we used the dynamic compensators to control the velocity and orientation of the robot on this manifold. Simulations results were presented to show the performance of the control approach.

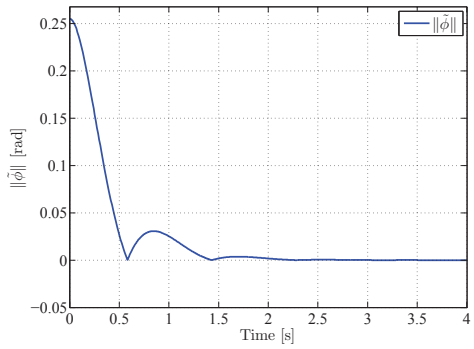

Fig. 3: Exponential stability of the joints tracking errors.

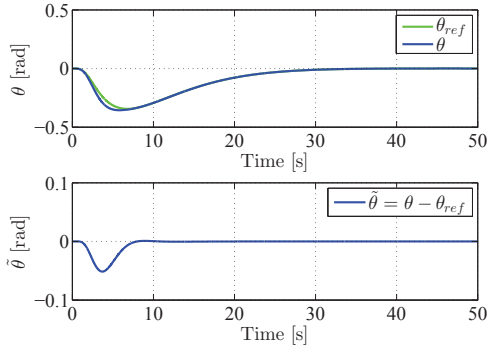

Fig. 4: Orientation reference tracking and orientation error.

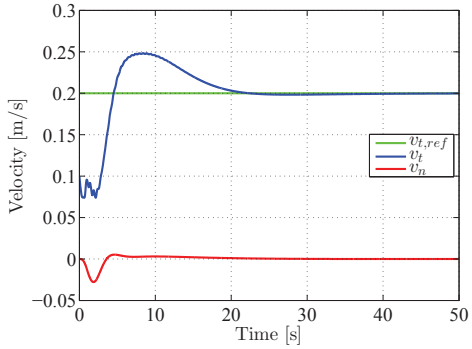

Fig. 5: Forward and sideways velocities with references.

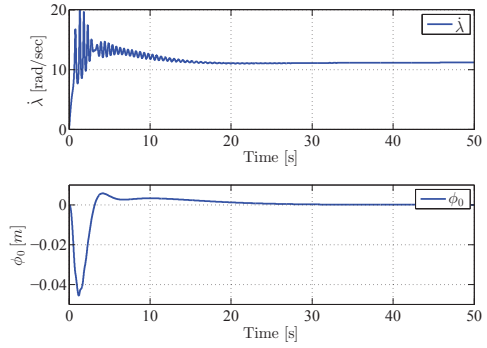

Fig. 6: The frequency of the joint oscillation converges to a positive constant (above), the joint offset remains uniformly bounded.

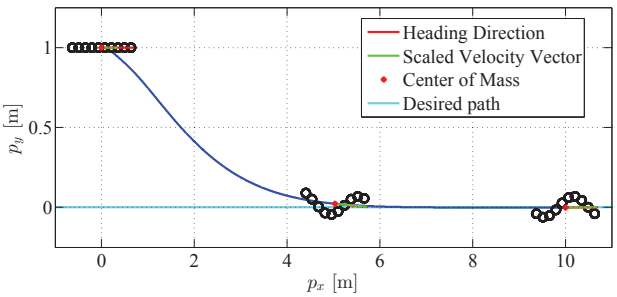

Fig. 7: A snake robot following the desired path.

REFERENCES

[1] P. Chatzakos, Y.P. Markopoulos, K. Hrissagis, and A. Khalid, "On the development of a modular external-pipe crawling omni-directional mobile robot", *Industrial Robot: An International Journal*, Vol. 33, no. 4, pp. 291-297, 2006.

[2] Z. Wang, and E. Appleton, "The concept and research of a pipe crawling rescue robot.", *Advanced Robotics*, 17.4, 339-358, 2003.

[3] S. Tully, G. Kantor, M.A. Zenati, and H. Choset, "Shape Estimation for Image-Guided Surgery with a Highly Articulated Snake Robot", Proc. 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Francisco, CA, USA, Sep. 2011.

[4] R. Skjetne, T.I. Fossen, P.V. Kokotovic, "Robust output maneuvering for a class of nonlinear systems", Automatica 40.3, 373-383, 2004.

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

[6] H. Date, Y. Hoshi, and M. Sampei, "Locomotion control of a snakelike robot based on dynamic manipulability", in Proc. the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Takamatsu, Japan, 2000.

[7] S. Ma, Y. Ohmameuda, K. Inoue, and B. Li, "Control of a 3-dimensional snake-like robot", in Proc. IEEE Int. Conf. Robotics and Automation, vol. 2, Taipei, Taiwan, pp. 2067-2072, 2003.

[8] G. Hicks and K. Ito, "A method for determination of optimal gaits with application to a snake-like serial-link structure", *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1291-1306, 2005.

[9] P. Liljebäck, K.Y. Pettersen, Ø. Stavdahl, and J.T. Gravdahl, "Snake Robots - Modelling, Mechatronics, and Control", Advances in Industrial Control, Springer, 2013.

[10] P. Liljebäck, I.U. Haugstuen, and K.Y. Pettersen, "Path following control of planar snake robots using a cascaded approach", *IEEE Transactions on Control Systems Technology*, vol. 20, 111-126, 2012.

[11] E. Rezapour, K.Y. Pettersen, P. Liljebäck, J.T. Gravdahl, and E. Kelasidi, "Path following control of planar snake robots using virtual holonomic constraints: theory and experiments", *Robotics and Biomimetics*, 1:3, SpringerOpen, 2014.

[12] A. Mohammadi, E. Rezapour, M. Maggiore, and K.Y. Pettersen, "Direction Following Control of Planar Snake Robots Using Virtual Holonomic Constraints", Proc. 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, Dec 2014.

[13] T.I. Fossen, "Marine control systems: Guidance, navigation and control of ships, rigs and underwater vehicles", Trondheim, Norway: Marine Cybernetics, 2002.

[14] H.K. Khalil, "Nonlinear Systems", Third ed., Englewood cliffs, NJ: Prentice-Hall, 2002.

[15] E.R. Westervelt, J.W. Grizzle, C. Chevallereau, J.H. Choi, and B. Morris, "Feedback control of dynamic bipedal robot locomotion", Boca Raton: CRC press, 2007.

[16] M. Maggiore, and L. Consolini, "Virtual Holonomic Constraints for Euler-Lagrange Systems," *IEEE Transactions on Automatic Control*, vol.58, no.4, pp.1001-1008, 2013.

[17] X. Yang, "Practical stability in dynamical systems", *Chaos, Solitons and Fractals*, vol.11 no.7, 1087-1092, 2000.

[18] V. I. Arnol'd, "Mathematical methods of classical mechanics", Vol. 60. Springer, 1989.

[19] P. Dorato, C. T. Abdallah, and V. Cerone. Linear-quadratic control: an introduction. Prentice Hall, 1995.

[20] E. Rezapour, A. Hofmann, K.Y. Pettersen, A. Mohammadi, and M. Maggiore, "Virtual holonomic constraints based direction following control of planar snake robots described by a simplified model", IEEE Multi-Conf. on systems and control, Antibes, France, 2014.