Path Following Control of Planar Snake Robots Using Virtual Holonomic Constraints

Ehsan Rezapour, Kristin Y. Pettersen, Pål Liljebäck, and Jan T. Gravdahl

Abstract— **This paper considers path following control of planar snake robots using virtual holonomic constraints. We first derive the Euler-Lagrange equations of motion of the snake robot. Moreover, we integrate the effects of friction forces into these equations. 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. In particular, we show that the constraint manifold can be made invariant by a suitable choice of feedback. Furthermore, we analytically design a smooth feedback control law to render the constraint manifold exponentially stable for the controlled system. We show that enforcing the appropriately defined virtual holonomic constraints implies that the robot converges to and follows a desired geometric path. Numerical simulations are presented to support the theoretical design.**

I. 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), and in applications where human presence is impossible (e.g. in narrow pipe inspection tasks).

This paper considers path following control of snake robots. The goal of the path following problem, is to ensure that the error between the system output and a desired geometric path is eventually less than any prespecified constant, while guaranteeing a forward motion along the path and boundedness of all states. 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.

Path following control of snake robots has been considered in several previous works. The majority of these works consider snake robots with passive wheels, which is inspired by the world's first snake robot developed in 1972 [1], and which introduces sideslip constraints on the links of the robot. These constraints allow the control input to be specified directly in terms of the desired propulsion of the snake robot, which is employed in e.g. [2-5] for computed torque control of the position and heading of wheeled snake robots. Path following control of wheel-less (i.e. without velocity constraints) snake robots is only considered in a few previous works. In [6], 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 [7] for determining optimal gaits during positional control of snake robots. In [8,9] cascaded systems theory is employed to achieve path following control of a snake robot described by a simplified model.

The contribution of this paper is a solution to the path following control problem for a wheel-less snake robot by use of virtual holonomic constraints, which is a particularly useful concept for control of oscillations, (see e.g. [13,14]). In our solution, we constrain the state evolution of the system to an appropriately defined submanifold of the configuration space, which is called the constraint manifold. The geometry of this manifold is defined based on the specified geometric relations among the generalized coordinates of the system. 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 only previous work which formally proves the stability of a path following controller for a snake robot is presented in [8,9]. This previous work is based on a simplified model of the snake robot, where the motion of the links is approximated as translational motion instead of rotational motion, which is valid for limited joint angles. In this paper, however, we consider a more complex and accurate model of a snake robot, where the motion of the links is not modelled based on such simplifying assumptions.

The paper is organized as follows. In Section II, we derive the Euler-Lagrange equations of motion of the robot. In Section III, we state our control design objectives. In Section IV, we use the virtual holonomic constraints approach for path following control design for the snake robot. Finally, Section V presents the simulation results which illustrate the performance of our theoretical control design method.

Ehsan Rezapour, Kristin Y. Pettersen, and Jan T. Gravdahl are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. emails: {ehsan.rezapour, kristin.y.pettersen, jan.tommy.gravdahl}@itk.ntnu.no. The affiliation of Pål Liljebäck is shared between the Department of Engineering Cybernetics and the Department of Applied Cybernetics, SINTEF ICT, NO-7465 Trondheim, Norway. email: pal.liljebaeck@sintef.no.

II. MODELLING

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 w.r.t. these coordinates. According to the illustration of the snake robot in Fig. 1, we choose the vector of the generalized coordinates of the N-link snake robot as $x = [q_1, q_2, ..., q_{N-1}, \theta_N, p_x, p_y]^T \in \mathbb{R}^{N+2}$, where q_i with $i \in \{1, ..., N-1\}$ denotes the i-th joint angle, θ_N denotes the head angle of the robot, and the pair (p_x, p_y) describes the position of the center of mass (CM) of the robot w.r.t. the global $x - y$ axes. The vector of the generalized velocities is defined as $\dot{x} = [\dot{q}_1, \dot{q}_2, ..., \dot{q}_{N-1}, \dot{\theta}_N, \dot{p}_x, \dot{p}_y]^T \in \mathbb{R}^{N+2}$. 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. $(q_1, ..., q_{N-1}, \theta_N)$, as the angular coordinates, and the corresponding dynamics as the angular dynamics of the system.

A. 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 of the robot, i.e. $q_a = (q_1, ..., q_{N-1})$, 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 $\mathcal{L} : T\mathcal{Q} \to \mathbb{R}$ is invariant under the given action of $\mathcal{G}$ on $\mathcal{Q}$. The coupling between the shape and position variables causes the net displacement of the position variables, according to the cyclic motion of the shape variables which is known as 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 $\mathbb{R}^{2N+4}$.

B. The Forward Kinematic Map of the Snake Robot

Based on the kinematic parameters of the snake robot given in Fig. 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

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

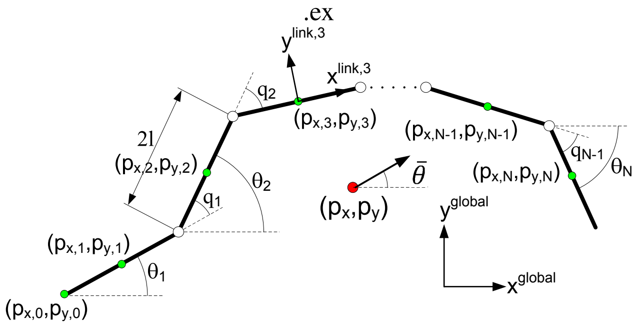

Fig. 1: Kinematic parameters of the snake robot

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

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

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

where $2l$ denotes the length of the links, and $(p_{x,0}, p_{y,0})$ is the tail position, cf. Fig. 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

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

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

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

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

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

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

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

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.

C. 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. 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. [8]). 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 [11], which is simple to follow and better suited for studying advanced mechanical phenomena such as elastic link deformations [11], 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 give an straightforward mapping of these forces to 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 [10]. 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 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 mass and moment of inertia of a link, respectively. Using the definition of the Lagrangian from (9), one can write the Euler-Lagrange equations of motion for the controlled system as

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

where $i \in \{1, ..., N+2\}$, $B(x) = [e_j] \in \mathbb{R}^{(N+2)\times(N-1)}$ 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 $\tau = [\tau_1, ..., \tau_{N-1}]^T \in \mathbb{R}^{N-1}$ stands for the generalized forces resulting from the control inputs. Furthermore, $\tau_f = [\tau_f^1, ..., \tau_f^{N+2}]^T \in \mathbb{R}^{N+2}$ denotes viscous and Coulomb friction forces acting on $(N+2)$ DOF of the system. The controlled Euler-Lagrange equations can also be written in the form of a second-order differential equation

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

where $M(q_a) \in \mathbb{R}^{(N+2)\times(N+2)}$ is the positive definite symmetric inertia matrix, $C(x, \dot{x}) \in \mathbb{R}^{(N+2)\times(N+2)}$ is the matrix of Coriolis and centrifugal forces, and the right-hand side terms denote the external forces acting on the system. Since $\mathrm{rank}[B(x)] < \dim(x)$ 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.

D. 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. Fig. 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

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

where $v_t^{\text{link},i}$ and $v_n^{\text{link},i}$ denote the linear velocity of the 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 for that link, 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

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

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

where $i \in \{1, ..., N\}$, m denotes the mass of a link, g is 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. Finally, we map the friction force acting on the i-th link to the global $x-y$ frame as

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

and 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

$$\mathcal{J}_i^T(x) = \left[\frac{\partial\dot{p}_{x,i}}{\partial\dot{x}_j}, \frac{\partial\dot{p}_{y,i}}{\partial\dot{x}_j}\right] \in \mathbb{R}^{(N+2)\times 2}, j \in \{1, ..., N+2\} \quad (19)$$

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

Remark I. *As argued in [8], 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.*

E. Partial Feedback Linearization

A common method for control of mechanical systems is full-state feedback linearization. This approach is not applicable for snake robots due to 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 [8] 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 Section II.A:

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

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

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 $0_{3\times 1}$ denotes a column vector composed of three zero elements. Furthermore, $h_{1,2}(x,\dot{x})$ include all the contributions of the Coriolis, centrifugal and friction forces, and $\psi \in \mathbb{R}^{N-1}$ 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

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

For linearizing the dynamics of the directly actuated DOF (20), we apply the global transformation of the vector of control inputs as

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

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

$$\begin{aligned} \ddot{q}_a &= \vartheta \in \mathbb{R}^{N-1} & (25) \\ \ddot{q}_u &= \mathcal{D}(x,\dot{x}) + \mathcal{C}(q_a)\vartheta \in \mathbb{R}^3 & (26) \end{aligned}$$

with

$$\mathcal{D}(x,\dot{x}) = -m_{22}^{-1}(q_a)h_2(x,\dot{x}) = [f_{\theta_N}, f_x, f_y]^T \in \mathbb{R}^3$$
$$\mathcal{C}(q_a) = -m_{22}^{-1}(q_a)m_{21}(q_a) = [\beta_i(q_a), 0, 0]^T \in \mathbb{R}^{3\times(N-1)}$$

where $\beta_i(q_a) : \mathcal{Q} \to \mathbb{R}$ 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:

$$\begin{aligned} \ddot{q}_a &= \vartheta \in \mathbb{R}^{N-1} & (27) \\ \ddot{\theta}_N &= f_{\theta_N}(x,\dot{x}) + \beta_i(q_a)\vartheta^i \in \mathbb{R} & (28) \\ \ddot{p}_x &= f_x(x,\dot{x}) \in \mathbb{R} & (29) \\ \ddot{p}_y &= f_y(x,\dot{x}) \in \mathbb{R} & (30) \end{aligned}$$

where the summation convention is applied in (28), and henceforth, to all the equations which contain repeated upper-lower indices. The dynamical system (27-30) is in the form of a control-affine system with drift. In particular, the term

$$\mathcal{A}(x,\dot{x}) = [\dot{q}_a, \dot{q}_u, 0_{(N-1)\times 1}, \mathcal{D}(x,\dot{x})]^T \in \mathbb{R}^{2N+4} \quad (31)$$

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

$$\mathcal{B}(q_a) = \begin{bmatrix} 0_{(N+2)\times(N-1)} \\ I_{N-1} \\ [\beta_1(q_a), ..., \beta_{N-1}(q_a)] \\ 0_{2\times(N-1)} \end{bmatrix} \in \mathbb{R}^{(2N+4)\times(N-1)} \quad (32)$$

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 II. *The last two rows of the control vectors in (32) 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. (29-30), and the dynamics of the position of the CM are coupled with the dynamics of the directly actuated shape variables q_a, i.e. (27), 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.*

III. 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 please see [15]). This is due to the fact that for a part of the state variables of the system (particularly the shape variables and the heading angle) the control problem is usually formulated as a trajectory tracking problem, while for the other states 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 along with a path following formulation for the remaining subset, we introduce the error variable for the i-th joint of the robot as

$$y_i = q_i - \Phi_i \quad (33)$$

where $i \in \{1, ..., N-1\}$, 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 Section IV. The head angle error is defined as

$$y_N = \theta_N - \Phi_N \quad (34)$$

where $\Phi_N \in \mathbb{R}$ denotes the desired 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, which implies

$$\lim_{t\to+\infty} \|y_i(t)\| = 0 \quad (35)$$

for all $i \in \{1, ..., N-1\}$. 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

$$\lim_{t\to+\infty} \|y_N(t)\| = 0 \tag{36}$$

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

$$\mathcal{P} = \left\{ \ (p_{xd}(\Theta), p_{yd}(\Theta)) \in \mathbb{R}^2 : \Theta \geq 0 \right\} \tag{37}$$

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 objective is defined as practical convergence to the path such that [16]

$$\lim_{t\to+\infty} \sup \|\tilde{p}(t)\| \leq \varepsilon \tag{38}$$

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

IV. 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. [13,14]). 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 designing the joint reference trajectories in (33) using virtual holonomic constraints, and combining this with virtual holonomic constraints motivated by Line-of-Sight (LOS) guidance for the head angle error (34), we are able to solve the path following control problem, i.e. achieving (38). Our main motivation for using this approach is the fact that while performing the gait pattern lateral undulation which constitutes of fixed periodic body motions, all the solutions of the snake robot dynamics have inherent oscillatory behaviour along any desired trajectory. 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 (27) 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 [12].

A. Trajectory Planning by Virtual Holonomic Constraints

Virtual holonomic constraints are specified through C^1 coordinate-dependent functions $\Phi_i : \mathcal{Q} \to \mathbb{R}$ 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, we define a vector-valued function

$$\Phi = [\Phi_1, ..., \Phi_N]^T \in \mathbb{R}^N \tag{39}$$

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, i.e. virtual holonomic constraints which depend on dynamic variables, in [13]. There, the idea is to make the virtual holonomic constraints to depend on the variations of a dynamic parameter, which affects the dynamics of the system on the constraint manifold. The purpose of these additional states is explained below.

1) We introduce two new states $[\phi_0, \dot{\phi}_0]^T \in \mathbb{R}^2$ where the second order time-derivative of ϕ_0 will be used as an additional control term that drives the snake robot towards the desired path by modifying the orientation of the robot in the plane.

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 the holonomic constraints are coordinate-dependent equality constraints of the form $\Phi_i(x) = 0$, where Φ_i is a time-independent function [11]. 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

$$\hat{x} = [q_1, ..., q_{N-1}, \theta_N, p_x, p_y, \phi_0, \eta]^T \in \mathbb{R}^{N+4} \tag{40}$$

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

B. Virtual Holonomic Constraints for the Joint Angles

A fundamental work in the area of snake robotic was presented by Hirose [1]. 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

$$x(s) = \int_0^s \cos(a \cos(bz) + cz) dz \tag{41}$$

$$y(s) = \int_0^s \sin(a \cos(bz) + cz) dz \tag{42}$$

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. Subsequently, using the foregoing defined new states, we define a dynamic constraint function for the i-th joint of the snake robot by

$$\Phi_i = \alpha \sin\left(\eta + (i-1)\delta\right) + \phi_0 \tag{43}$$

where $i \in \{1, ..., N-1\}$, 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_0 is an offset value that is identical for all of the joints. It was illustrated in [8] how the offset value ϕ_0 affects the orientation of the snake in the plane. Building further on this insight, we consider the second-order time-derivative of ϕ_0 as an additional control term for our underactuated control design. In particular, through this control term we modify the orientation of the robot 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.

C. Virtual Holonomic Constraint for the Head Link Angle

In this subsection, we define a LOS guidance law as the reference head angle for the snake robot. LOS guidance is a much-used method in marine control systems, (see e.g. [15]). 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 [15]. Motivated by marine control literature, in [8] based on 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. 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

$$\Phi_N = -\tan^{-1}\left(p_y/\Delta\right) \tag{44}$$

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 in a distance Δ ahead of the robot along the desired path, will make the snake robot move towards the path.

D. Defining a Constraint Manifold

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

$$\begin{aligned}\Phi = [\alpha \sin\left(\eta\right) + \phi_0, ..., \alpha \sin\left(\eta + (N-1)\delta\right) + \phi_0 \\ , -\tan^{-1}\left(p_y/\Delta\right)]^T \in \mathbb{R}^N\end{aligned} \tag{45}$$

For trajectory planning using virtual holonomic constraints, we define the constraint manifold associated with the constraint functions (45) as

$$\begin{aligned}\Gamma = \{(\hat{x}, \dot{\hat{x}}) \in T\hat{\mathcal{Q}} : q_i = \Phi_i(\eta, \phi_0), \theta_N = \Phi_N(p_y), \dot{q}_i \\ = \dot{\eta}\frac{\partial \Phi_i}{\partial \eta} + \dot{\phi}_0 \frac{\partial \Phi_i}{\partial \phi_0}, \dot{\theta}_N = \dot{p}_y \frac{\partial \Phi_N}{\partial p_y}\}\end{aligned} \tag{46}$$

where $i \in \{1, ..., N-1\}$. The constraint manifold (46) is a 6-dimensional submanifold of $\hat{\mathcal{Q}}$, since we have three different constraint variables, i.e. (η, ϕ_0, p_y). The goal of the control input is to enforce the virtual holonomic constraints (45), by making Γ exponentially stable for the closed-loop system, and thereby achieving the control objectives (35-36). To this end, we define the elements of a controlled output vector $y \in \mathbb{R}^N$ for the system (28-31) as the difference between the angular coordinates and their corresponding constraint functions in the form

$$y = [q_1 - \Phi_1(\eta, \phi_0), ..., q_{N-1} - \Phi_{N-1}(\eta, \phi_0), \theta_N - \Phi_N(p_y)]^T \tag{47}$$

We will achieve our control design objectives which we defined in Section III by designing the control inputs ϑ and $\ddot{\phi}_0$ such that $(y_i, \dot{y}_i) \to (0, 0)$ for all $i \in \{1, \ldots, N\}$. First, we need to ensure that the given relations in (45) are stabilizable, i.e. a suitable choice of feedback can make the constraint manifold asymptotically stable for the closed-loop system. For simplicity, we denote the following differentials:

$$\begin{aligned}d\Phi_i &= \dot{\eta}\frac{\partial \Phi_i}{\partial \eta} + \dot{\phi}_0 \frac{\partial \Phi_i}{\partial \phi_0} \\ d\Phi_N &= \dot{p}_y \frac{\partial \Phi_N}{\partial p_y} \\ d^2\Phi_i &= \ddot{\eta}\frac{\partial \Phi_i}{\partial \eta} + \dot{\eta}^2 \frac{\partial^2 \Phi_i}{\partial \eta^2} + \ddot{\phi}_0 \frac{\partial \Phi_i}{\partial \phi_0} + \dot{\phi}_0^2 \frac{\partial^2 \Phi_i}{\partial \phi_0^2} \\ d^2\Phi_N &= \ddot{p}_y \frac{\partial \Phi_N}{\partial p_y} + \dot{p}_y^2 \frac{\partial^2 \Phi_N}{\partial p_y^2}\end{aligned} \tag{48}$$

The first derivative of (47) is in the form

$$\dot{y} = [\dot{q}_1 - d\Phi_1, ..., \dot{q}_{N-1} - d\Phi_{N-1}, \dot{\theta}_N - d\Phi_N]^T \tag{49}$$

which lacks the control inputs. The second derivative of (47) is in the form

$$\ddot{y} = [\vartheta_1 - d^2\Phi_1, ..., \vartheta_{N-1} - d^2\Phi_{N-1}, f_{\theta_N} + \beta_i \vartheta^i - d^2\Phi_N]^T \tag{50}$$

which contains the control inputs. Consequently, the controlled output vector (47) yields a well-defined vector relative degree $\{2, 2, ..., 2\}$ everywhere on Γ. The virtual holonomic constraints satisfying this vector relative degree condition are called regular, and regular constraints are always feasible [13], i.e. there exists a smooth feedback such that Γ is positively invariant for the closed-loop system. Furthermore, regular constraints in parametric form (45) are always stabilizable [13]. The well-defined vector relative degree $\{2, 2, ..., 2\}$ on Γ implies that the system (27-30) with the controlled output function (47) is input-output feedback linearizable. Consequently, we can stabilize Γ with an input-output feedback linearizing controller.

E. Output Regulation via Input-Output Linearization

In this subsection, we will derive a control law for (27) such that the constraint manifold (46) with the constraint functions defined in (45) is globally exponentially stable for the closed-loop system. In particular, we use input-output linearization to stabilize the constraint manifold (46).

To stabilize $\Phi_i(\eta, \phi_0)$ for the i-th joint, i.e. to make $(y_i, \dot{y}_i) \rightarrow (0, 0)$ for all $i \in \{1, \ldots, N-1\}$, 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 the vector (50), is in the form

$$\ddot{y}_i = \vartheta_i - d^2\Phi_i \tag{51}$$

We define the control input for the i-th joint in (27) as

$$\vartheta_i = d^2\Phi_i - k_p y_i - k_d \dot{y}_i \tag{52}$$

where $k_p > 0$ and $k_d > 0$ are the joint controller gains, which are similar for all the joints. Inserting (52) into (51) yields

$$\ddot{y}_i + k_d \dot{y}_i + k_p y_i = 0 \tag{53}$$

The tracking error dynamics of the i-th joint angle (53), clearly has a globally exponentially stable equilibrium at the origin $(y_i, \dot{y}_i) = (0, 0)$, which implies that the i-th control input (52), exponentially stabilizes the constraint manifold for every i-th joint angle, and the control objective (35) 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 corresponds to the N-th element of (47), and its second-order time-derivative (i.e. the error dynamics for the head angle) is given by

$$\ddot{y}_N = f_{\theta_N} + \beta_i \vartheta^i - d^2\Phi_N \tag{54}$$

Inserting ϑ_i from (52) into (54) gives

$$\ddot{y}_N = f_{\theta_N} + \sum_{i=1}^{N-1} \beta_i (d^2\Phi_i - k_p y_i - k_d \dot{y}_i) - d^2\Phi_N \tag{55}$$

which is equivalent to

$$\begin{aligned}\ddot{y}_N =& f_{\theta_N} + \sum_{i=1}^{N-1} \beta_i (\ddot{\eta}\frac{\partial \Phi_i}{\partial \eta} + \dot{\eta}^2 \frac{\partial^2 \Phi_i}{\partial \eta^2} + \ddot{\phi}_0 \frac{\partial \Phi_i}{\partial \phi_0} + \\ & \dot{\phi}_0^2 \frac{\partial^2 \phi_i}{\partial \phi_0^2} - k_p y_i - k_d \dot{y}_i) - d^2\Phi_N\end{aligned} \tag{56}$$

Let us denote the constraint function for the i-th joint angle of the robot as $\Phi_i = S_i + \phi_0$, where S_i denotes the sinusoidal part of (43). Subsequently, based on the specified constraint functions in (45), we may write (56) in a simpler form as

$$\ddot{y}_N = f_{\theta_N} + \sum_{i=1}^{N-1} \beta_i (-\dot{\eta}^2 S_i + \ddot{\phi}_0 - k_p y_i - k_d \dot{y}_i) - d^2\Phi_N \tag{57}$$

For stabilizing the constraint function $\Phi_N(p_y)$ for the head angle, we define the second-order time-derivative of the augmented coordinate ϕ_0 as

$$\begin{aligned}\ddot{\phi}_0 =& (\sum_{i=1}^{N-1} \beta_i)^{-1} (\sum_{i=1}^{N-1} \beta_i (\dot{\eta}^2 S_i + k_p y_i + k_d \dot{y}_i) + d^2\Phi_N - f_{\theta_N} \\ & - k_{p,\theta_N} y_N - k_{d,\theta_N} \dot{y}_N)\end{aligned} \tag{58}$$

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, (58) is globally well-defined. Through numerical simulations we show that the states of the dynamic compensator (58), i.e. $(\phi_0, \dot{\phi}_0)$, remain bounded, however, a formal proof of this boundedness remains as a topic of future work. By inserting (58) into (57), the error dynamics of the head angle takes the form

$$\ddot{y}_N + k_{d,\theta_N} \dot{y}_N + k_{p,\theta_N} y_N = 0 \tag{59}$$

which clearly has a globally exponentially stable equilibrium at the origin $(y_N, \dot{y}_N) = (0, 0)$. Consequently, the control objective (36) is achieved.

Finally, we conjecture that while the output trajectories are evolving on the constraint manifold (46), the internal dynamics (29-30), which has the form

$$\ddot{p}_x = f_x(\Phi, p_x, p_y, \Phi^{'}, \dot{p}_x, \dot{p}_y) \tag{60}$$

$$\ddot{p}_y = f_y(\Phi, p_x, p_y, \Phi^{'}, \dot{p}_x, \dot{p}_y) \tag{61}$$

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 in Section V simulation results which show that the snake robot successfully converges to and follows the desired path.

V. 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 $N = 4$ links, $m = 0.5$ kg, $l = 0.15$ m, and $J = 0.0016$ kgm^2. The friction coefficients were $c_n = 10$, $c_t = 1$. The parameters of the joint constraint functions (43) were $\alpha = \pi/6$ rad, $\eta = 80\pi t/180$ rad, and $\delta = \pi/2$ rad. The controller gains in (52) and (58) were tuned as $k_p = 5$, $k_d = 4$, $k_{p,\theta_N} = 5$, $k_{d,\theta_N} = 5$, and $\Delta = 3$ m. In order to calculate $\dot{\Phi}_N$ and $\ddot{\Phi}_N$, we employed the approach taken in [15] by passing Φ_N through a low-pass filter of the form

$$\dot{\Omega} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\psi_f \omega_n \end{bmatrix} \Omega + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \Phi_N \tag{62}$$

with natural frequency $\omega_n = 0.8$ rad, damping ratio $\psi_f = 0.8$. As seen from the simulation results, the snake robot successfully converges to and follows the desired path.

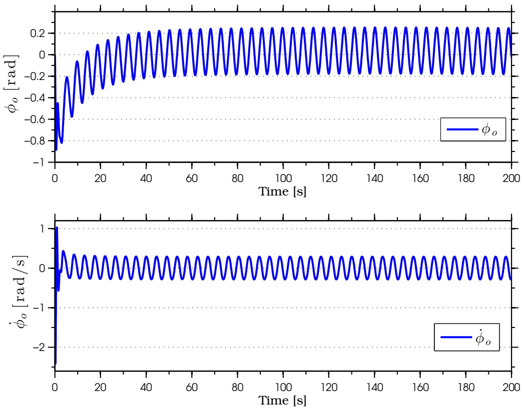

Fig. 2: The states of the dynamic compensator remain bounded

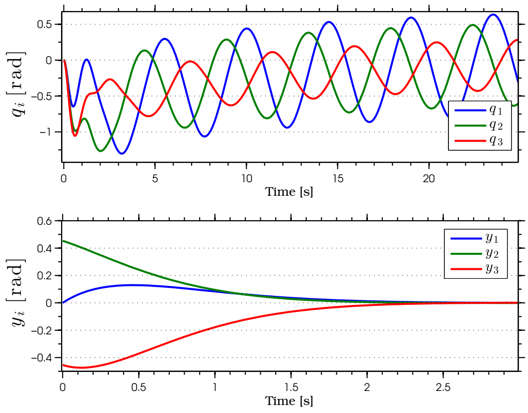

Fig. 3: The joints of the robot track the serpenoid motions (above). The joint tracking errors converge exponentially to zero (below)

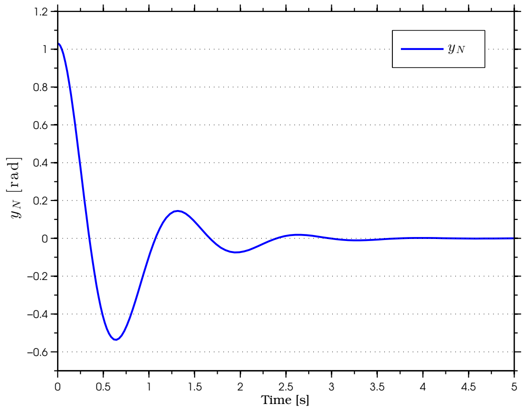

Fig. 4: The head angle tracking error converges exponentially to zero

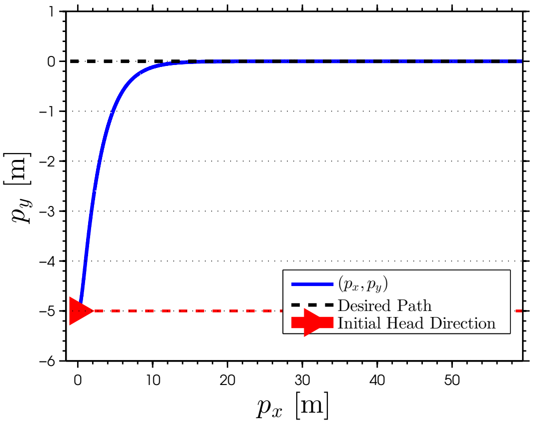

Fig. 5: The position of the CM of the robot (blue) converges to and follows the desired straight line path (the x-axis)

VI. CONCLUSIONS

This paper has considered path following control of a planar snake robot by use of virtual holonomic constraints. The equations of motion of the snake robot were derived using 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 linearzing control law to exponentially stabilize the constraint manifold for the system. The numerical simulations supported the theoretical results.

REFERENCES

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

[2] P. Prautsch, T. Mita, and T. Iwasaki. ”Analysis and control of a gait of snake robot”, *Transactions-Institute of Electrical Engineers of Japan*, D-120.3. pp. 372-381. 2000.

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

[4] 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.

[5] M. Tanaka and F. Matsuno, ”Control of 3-dimensional snake robots by using redundancy”, in Proc. IEEE Int. Conf. Robotics and Automation, pp. 1156-1161, Pasadena, CA, 2008.

[6] K. McIsaac and J. Ostrowski, ”Motion planning for anguilliform locomotion”, *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 637-652, 2003.

[7] 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.

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

[9] 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.

[10] F. Bullo, A. Lewis, ”Geometric Control of Mechanical Systems”, Springer, 2005.

[11] M. W. Spong, S. Hutchinson, and M. Vidyasagar, ”Robot modeling and control”, New York: John Wiley and Sons, 2006.

[12] 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.

[13] 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.

[14] A. Shiriaev, J. W. Perram, and C. Canudas-de-Wit, ”Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach”, *IEEE Transactions on Automatic Control*, 50.8: 1164-1176, 2005.

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

[16] D. B. Dacic, D. Nesic, A. R. Teel, and W. Wang, ”Path following for nonlinear systems with unstable zero dynamics: an averaging solution”, *IEEE Transactions on Automatic Control*. 56, 880-886. 2011.