

Velocity and orientation control of underwater snake robots using absolute velocity feedback

A. M. Kohl, K. Y. Pettersen, and J. T. Gravdahl

Abstract—This article presents a control system for velocity and orientation control of underwater snake robots using absolute velocity feedback. The control system is structured in a hierarchical way, where the highest priority is to enforce virtual constraints encoding a planar gait on the body shape of the robot. To this end, we propose an adaptive joint controller and show that it asymptotically stabilizes the constraint manifold. The virtual constraints are parametrized by the states of two dynamic compensators, which can be used to control the velocity and the orientation of the robot, the second and third control priority. We design an adaptive controller that asymptotically stabilizes the forward velocity to a reference and an orientation controller that utilizes the current estimate of the velocity controller. It is shown that the zero dynamics of the closed-loop system remains bounded, and we present simulation results that demonstrate the performance of the controller.

I. INTRODUCTION

In today's technology there is a large potential for improving efficiency and reducing costs by increasing autonomy. This includes improving traditional industrial robots as well as developing new robotic solutions for sectors that have so far relied on conventional technology. Subsea inspection, maintenance and repair (IMR) operations is a field of technology with a large potential for autonomous robotics to increase the efficiency, reduce operation costs, and provide safer and more environmentally friendly solutions. This requires the development of new systems for IMR operations that are more robust, agile, and versatile than existing technology. When designing solutions with these properties, inspiration can be found in nature, where a variety of species has adapted to subsea conditions over millions of years, and are thus specialised in propelling and maneuvering underwater. Underwater snake robots, a class of robots that propel their slender body by mimicking the swimming motion of eels, are therefore considered promising to provide autonomous IMR solutions in the future.

Research on snake robots started with land-based snake robots [1,2], and more recently broadened towards amphibious and swimming snake robots [3–5]. A closely related field is that of robotic fish, see for instance [6]. In previous studies on underwater snake and fish-like robots, researchers have often considered the velocity of the robot with respect to the surrounding fluid, i. e. the relative velocity of the robot.

This is an obvious assumption, since the relative velocity determines the drag forces, the dominating fluid force for low velocities [7], and thus the dynamic equations of the robot. Motion control approaches for underwater snake robots that rely on position and relative velocity measurements can for instance be found in [8,9]. Similarly, hardware solutions following this line of research have been presented, e. g. a fish robot that can sense the surrounding flow [10]. Absolute velocity measurements in the global frame have not been considered as much in the context of underwater snake robots, even though they are often easier to obtain than their relative counterparts. The absolute velocity can for instance be measured with an acoustic Doppler velocity log with bottom lock [11] or obtained from position measurements.

In this article, we take a first step towards the control of underwater snake robots using absolute velocity measurements for feedback. We will present a control system for velocity and orientation control of underwater snake robots, motivated by the direction following results using virtual holonomic constraints (VHCs) for terrestrial snake robots in [12]. Other than methods such as central pattern generators, using VHCs to encode the robotic gait has the advantage that it makes the control design amenable to hierarchical synthesis, where the gait is enforced at the lowest level and velocity and orientation control are done at a higher level. To this end, we will employ the hierarchical control design methodology from [13], that was used for control design for ships in [14] and terrestrial snake robots in [15]. In order to make the feedback independent of the unknown relative velocities that enter the dynamic equations via the fluid drag forces, we will make use of adaptive backstepping control [16]. More specifically, we propose a joint controller that asymptotically stabilizes the VHCs with absolute velocity measurements available for feedback, as opposed to a design based on relative velocity measurements in previous work [17]. The gait that is encoded in the VHCs is widely used for snake robots, and modified in this article to take into account actuator constraints. The velocity controller and the orientation controller are designed subsequently. In doing so, we prioritize the velocity control higher than in previous approaches for terrestrial snake robots, a change of paradigm that removes a singularity from the orientation controller that was an issue in [12,18]. The velocity controller is designed along the lines of adaptive backstepping. The design is not a straightforward application of the method, because the system is non-autonomous and unknown terms enter the equations at every stage of the backstepping procedure. By postponing the design of the adaptive update law until

This work was supported by the Research Council of Norway through its Centres of Excellence funding scheme, project no. 223254-NTNU AMOS. A. M. Kohl and K. Y. Pettersen are with the Centre for Autonomous Marine Operations and Systems (NTNU-AMOS), Dep. of Engineering Cybernetics, NTNU. anna.kohl@itk.ntnu.no. J. T. Gravdahl is with the Dep. of Engineering Cybernetics, NTNU*, *Norwegian University of Science and Technology, NO-7491 Trondheim, Norway.

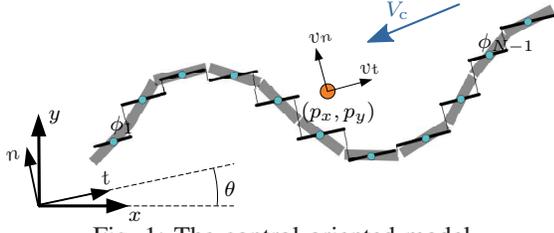


Fig. 1: The control oriented model.

the last step, we are able to compensate for the effect of these signals in the closed-loop system, and show asymptotic stability of the origin of the velocity error dynamics. In the last step, the orientation controller is designed as a feedback-linearizing controller utilizing the adaptive term of the velocity controller as a current estimate. We show that the zero dynamics of the closed-loop system is bounded for the proposed controller. Simulation results of a snake robot that is exposed to an unknown ocean current are presented. The results validate the theoretical analysis.

This article is structured as follows. In Sec. II, the control-oriented model, which the design is based on, is briefly reviewed. Sec. III presents the control objectives, the hierarchical design strategy that is employed to solve the control problem, and the proposed control system. Furthermore, the stability properties of each controller are discussed in the respective design step. An analysis of the zero dynamics and conditions for stability of the closed-loop system are presented in Sec. IV. Simulation results are presented in Sec. V and concluding remarks are given in Sec. VI.

II. THE CONTROL-ORIENTED MODEL

This section briefly reviews the control-oriented model that the hierarchical control design in this article is based on. The model was originally developed for land-based snake robots that conduct a planar, sinusoidal gait with limited link angles [2]. It was later shown that a similar formulation can be used to describe neutrally buoyant underwater snake robots as well, provided that they also move slowly with a planar, sinusoidal gait [17]. The general idea of the control-oriented modelling approach is to approximate the rotational joint motion of the snake robot by the corresponding translational motion, as displayed in Fig. 1. An extensive simulation study in [17] shows that this captures the behaviour of a more complex model with rotational joints quite well.

The model considers a snake robot that consists of N links of mass m and length $2l$ each. The links are connected with $N - 1$ translational joints, with the joint coordinates $\phi_i, i = 1, \dots, N - 1$ assembled into the vector $\phi = [\phi_1, \dots, \phi_{N-1}]^T \in \mathbb{R}^{N-1}$. The joints are actuated with the control input $\mathbf{u} \in \mathbb{R}^{N-1}$. The position of the robot in the plane is defined by the position of its center of mass, (p_x, p_y) , and the orientation of the robot is defined as θ . Furthermore, the velocity of the robot is described by the forward velocity component, v_t , and the sideways velocity component, v_n . These velocities are defined in the body-aligned $t-n$ -frame. The $t-n$ -frame, the global $x-y$ -frame, and the above kinematic definitions are visualized in Fig. 1. The analysis in [17] shows that it is a valid assumption to model the external

forces as linear drag forces, which depend on the relative velocity, i.e. the velocity of the robot with respect to the surrounding fluid $[v_{t,\text{rel}}, v_{n,\text{rel}}]^T = [v_t, v_n]^T - \mathbf{R}_\theta^T V_c$. The matrix $\mathbf{R}_\theta \in \mathbb{R}^{2 \times 2}$ denotes the rotation matrix with rotation angle θ . Furthermore, we make the following assumption regarding the ocean current:

Assumption 1: The unknown ocean current is irrotational and constant in the inertial frame, $V_c = [V_x, V_y]^T \in \mathbb{R}^2$. The magnitude of the current is bounded: $\sqrt{V_x^2 + V_y^2} \leq V_{\text{max}}$.

We define the state vector $\mathbf{x} = [\phi, \theta, p_x, p_y, \mathbf{v}_\phi, v_\theta, v_t, v_n]^T \in \mathbb{R}^{2N+4}$. The kinematic and dynamic equations of the robot are then given by

$$\dot{\phi} = \mathbf{v}_\phi, \quad (1a)$$

$$\dot{\theta} = v_\theta, \quad (1b)$$

$$\dot{p}_x = v_t \cos \theta - v_n \sin \theta, \quad (1c)$$

$$\dot{p}_y = v_t \sin \theta + v_n \cos \theta, \quad (1d)$$

$$\dot{\mathbf{v}}_\phi = -\frac{c_n}{m} \mathbf{v}_\phi + \frac{c_p}{m} v_{t,\text{rel}} \mathbf{A} \mathbf{D}^T \phi + \frac{1}{m} \mathbf{D} \mathbf{D}^T \mathbf{u}, \quad (1e)$$

$$\dot{v}_\theta = -\lambda_1 v_\theta + \frac{\lambda_2}{N-1} v_{t,\text{rel}} \bar{\mathbf{e}}^T \phi, \quad (1f)$$

$$\dot{v}_t = -\frac{c_t}{m} v_{t,\text{rel}} + \frac{2c_p}{Nm} \bar{\mathbf{e}}^T \phi v_{n,\text{rel}} - \frac{c_p}{Nm} \phi^T \mathbf{A} \bar{\mathbf{D}} \mathbf{v}_\phi, \quad (1g)$$

$$\dot{v}_n = -\frac{c_n}{m} v_{n,\text{rel}} + \frac{2c_p}{Nm} \bar{\mathbf{e}}^T \phi v_{t,\text{rel}}, \quad (1h)$$

with the drag coefficients in tangential and normal direction satisfying $c_t < c_n$, a propulsion coefficient c_p , and two constants λ_1 and λ_2 , which determine the rotational dynamics. In (1), we have used the following notation:

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

with $\mathbf{A}, \mathbf{D} \in \mathbb{R}^{(N-1) \times N}$, $\bar{\mathbf{D}} = \mathbf{D}^T (\mathbf{D} \mathbf{D}^T)^{-1} \in \mathbb{R}^{N \times (N-1)}$, $\bar{\mathbf{e}} = [1, \dots, 1]^T \in \mathbb{R}^{N-1}$. $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ denotes the unity matrix.

The relative velocities $v_{t,\text{rel}}$ and $v_{n,\text{rel}}$ enter the dynamics of the system in (1e),(1f),(1g),(1h). This happens because the external force, the fluid drag force, depends on the velocity of the robot relative to the surrounding fluid. In this paper, we focus on controlling the robot with absolute velocity feedback, which means that the relative velocities in the dynamic equations represent a design challenge.

III. CONTROL DESIGN

In this section, we present the proposed control system. In the first part, we will outline the control objectives and the hierarchical design approach that is employed. Subsequently, the control system is designed stage by stage, following the hierarchical structure. The stability properties of each stage are discussed in the respective sub-sections.

A. Control objectives and the hierarchical control approach

The control objectives can be categorized into three stabilization tasks that are prioritized in a hierarchical manner. The design is based on the model (1), i.e. the stabilization tasks will be formulated in terms of the dynamics in (1).

The highest priority is to stabilize VHCs that encode a sinusoidal gait, thus propelling the robot forward. We will design a feedback \mathbf{u} for the joints, described by (1a),(1e),

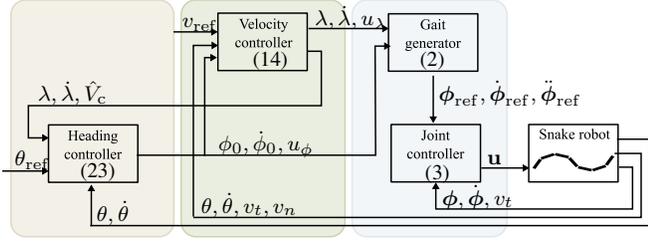


Fig. 2: A block diagram of the proposed control system with absolute velocity measurements for feedback.

that controls each joint i , for $i = 1, \dots, N-1$, to follow the reference signal

$$\phi_{\text{ref},i}(\lambda, \phi_0) = \alpha g_1(i) \sin(\lambda + (i-1)\delta) + g_2(\phi_0), \quad (2)$$

where α is the maximal amplitude and δ is the phase shift between adjacent joints. The VHC in (2) is an adaptation of the VHC used in [18] where $g_1 : \mathbb{Z} \mapsto [0, 1]$ is a scaling function that varies the amplitude along the body and a smooth, strictly increasing, and twice differentiable saturation function $g_2 : \mathbb{R} \mapsto [-\phi_{0,\text{max}}, \phi_{0,\text{max}}]$ has been added. The additional function g_2 ensures that the theoretical joint reference always respects the physical constraints of the robot and that the joint reference remains uniformly bounded by $|\phi_{\text{ref},i}| \leq \frac{\epsilon_\phi}{N-1} = \alpha + \phi_{0,\text{max}}$. The two states λ and ϕ_0 are generated by dynamic compensators that will be developed later in order to control the velocity and the orientation of the robot, respectively.

The second priority of the control system is to control the forward velocity v_t , the dynamics of which are given in (1g), to a constant reference v_{ref} using the dynamic compensator $\ddot{\lambda} = u_\lambda$. We employ the second derivative of λ as the control input, because in order to stabilize the constraints in (2), the time derivatives of $\phi_{\text{ref},i}$ up to $\ddot{\phi}_{\text{ref},i}$ are required, as we will see in Sec. III-B. Thus $\ddot{\lambda} = u_\lambda$ is a natural choice. In order to make sure that the robot does not just float with the ocean current, we will furthermore choose the reference v_{ref} larger than the current magnitude: $v_{\text{ref}} > V_{c,\text{max}}$.

Remark 1: As opposed to similar control approaches for snake robots in the literature [12,18], in this work, we consider the regulation of the forward velocity as a higher priority than the orientation control. Several motion control systems for snake robots in the literature (e.g. [2,5,17]) do not control the velocity of the robot, but propose to choose a constant $\dot{\lambda}$ to achieve some forward velocity, while controlling the orientation of the robot to follow the reference of a guidance system. We conjecture that making velocity control a higher priority than orientation control will provide a velocity controller that is a natural complement of motion control approaches such as in [2,5].

Controlling the orientation of the robot, θ , to a constant reference θ_{ref} is the third and lowest priority of the proposed control system. Based on (1b),(1f), we will design a feedback for the control input u_ϕ to the dynamic compensator $\ddot{\phi}_0 = u_\phi$ in order to achieve this task. The structure of the proposed control system is visualized in Fig. 2.

Remark 2: The robot described by (1) is an underactuated system. In particular, we cannot directly control the sideways

velocity v_n . However, we will design the control system such that v_n remains bounded. It was shown in [17] that the average of v_n converges to the sideways component of the current when the robot conducts steady forward motion. In the presence of a current, we will therefore expect the robot to drift sideways, as it travels in the direction θ_{ref} with the forward velocity v_{ref} . If this is not the desired behaviour of the robot, one typically designs a guidance system that provides the references θ_{ref} and v_{ref} in a way that implicitly controls the position and sideways velocity of the robot.

B. Joint control

The aim of the joint controller is to stabilize the joint coordinates to obey the VHCs given by (2), which is analogous to stabilizing the system to a constraint manifold. Details on the stabilization of sets can for instance be found in [13]. The dynamics of the joints are described by (1a),(1e) and the unknown relative velocity $v_{t,\text{rel}}$ enters the system in (1e). To this end, the error coordinates $\tilde{\phi}_i = \phi_i - \phi_{\text{ref},i}$ and $\dot{\tilde{\phi}}_i = \dot{\phi}_i - \dot{\phi}_{\text{ref},i}$, $i = 1, \dots, N-1$, will be stabilized to zero with the following adaptive backstepping controller:

$$\begin{aligned} \mathbf{u} &= m(\mathbf{D}\mathbf{D}^T)^{-1} \left[\frac{c_n}{m} \mathbf{v}_\phi - \mathbf{X}_t(\phi) v_t + \beta^T(\phi, \theta) \dot{\tilde{\phi}} \right. \\ &\quad \left. + \ddot{\phi}_{\text{ref}} - (k_1 + k_2) \dot{\tilde{\phi}} - (1 + k_1 k_2) \tilde{\phi} \right], \quad (3) \\ \dot{\tilde{\theta}} &= -\gamma \beta(\phi, \theta) (\dot{\tilde{\phi}} + k_1 \tilde{\phi}), \end{aligned}$$

with the control gains $k_1, k_2 > 0$ and the adaptive gain $\gamma > 0$. In (3), the joint errors and references are assembled in the vectors $\tilde{\phi}, \phi_{\text{ref}} \in \mathbb{R}^{N-1}$. Furthermore, $\mathbf{X}_t(\phi) = \frac{c_p}{m} \mathbf{A}\mathbf{D}^T \phi$ and $\beta^T(\phi, \theta) = \mathbf{X}_t(\phi) \mathbf{R}_\theta^T$.

Proposition 1: Consider a robot described by (1). The joint control law (3) asymptotically stabilizes the set

$$\begin{aligned} \Gamma_3 &= \{(\mathbf{x}, \dot{\mathbf{x}}, \lambda, \dot{\lambda}, \phi_0, \dot{\phi}_0) \in \mathbb{R}^{2N+8} : \\ &\quad \phi = \phi_{\text{ref}}, \mathbf{v}_\phi = \dot{\lambda} \frac{\partial \phi_{\text{ref}}}{\partial \lambda} + \dot{\phi}_0 \frac{\partial \phi_{\text{ref}}}{\partial \phi_0}\}, \quad (4) \end{aligned}$$

i. e. $(\tilde{\phi}, \dot{\tilde{\phi}}) \rightarrow (\mathbf{0}, \mathbf{0})$.

Proof: We define the new error coordinates $\xi_1 = \tilde{\phi}$, $\xi_2 = \dot{\tilde{\phi}} + k_1 \tilde{\phi}$, and $\tilde{\theta} = V_c - \dot{\theta}$. In the new coordinates, the closed-loop system (1a),(1e) with (3) transforms to

$$\begin{aligned} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} &= \begin{bmatrix} -k_1 \mathbf{I}_{N-1} & \mathbf{I}_{N-1} \\ -\mathbf{I}_{N-1} & -k_2 \mathbf{I}_{N-1} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \beta^T(\phi, \theta) \end{bmatrix} \tilde{\theta}, \\ \dot{\tilde{\theta}} &= -\gamma \beta(\phi, \theta) \xi_2. \quad (5) \end{aligned}$$

With the Lyapunov function $V = \frac{1}{2} \xi_1^T \xi_1 + \frac{1}{2} \xi_2^T \xi_2 + \frac{1}{2\gamma} \tilde{\theta}^T \tilde{\theta}$, we have

$$\dot{V} = -k_1 \xi_1^T \xi_1 - k_2 \xi_2^T \xi_2 \leq 0. \quad (6)$$

Therefore, \dot{V} is negative semi-definite, and thus the origin $(\xi_1, \xi_2, \tilde{\theta}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is uniformly globally stable and ξ_1, ξ_2 , and $\tilde{\theta}$ are bounded. The error system (5) is a non-autonomous system because it depends on time-varying parameters. Furthermore, V is lower bounded by zero and \dot{V} is finite because $\beta(\phi, \theta)$ in (5) is bounded, since $\beta(\phi, \theta) = \beta(\phi, \mathbf{R}_\theta)$, $\xi_1 = \tilde{\phi}$ is bounded, and ϕ_{ref} is bounded by design. With these conditions, it follows from

Barbalat's Lemma [19] that $(\xi_1, \xi_2) \rightarrow (0, 0)$ asymptotically as $t \rightarrow \infty$. This implies that also $\tilde{\phi}$ and $\tilde{\dot{\phi}}$ converge to zero asymptotically. ■

Remark 3: Since the regressor $\beta(\phi, \theta)$ is not persistently exciting, we cannot expect the adaptive estimate, $\hat{\theta}$ to converge to the true ocean current V_c .

C. Velocity control

For the design of the velocity controller, the reduced system dynamics on the invariant manifold Γ_3 is considered. On Γ_3 , the dynamics of the forward velocity (1g) reduces to

$$\dot{v}_t = -\frac{c_t}{m}v_{t,\text{rel}} + X_n(\lambda, \phi_0)v_{n,\text{rel}} - \mathbf{X}_\phi(\lambda, \phi_0)\mathbf{v}_\phi, \quad (7)$$

with $X_n(\lambda, \phi_0) = \frac{2c_p}{Nm}\bar{\mathbf{e}}^T \phi_{\text{ref}}(\lambda, \phi_0)$ and $\mathbf{X}_\phi(\lambda, \phi_0) = \frac{c_p}{Nm}\phi_{\text{ref}}^T(\lambda, \phi_0)\mathbf{A}\bar{\mathbf{D}}$. For the feedback control of the forward velocity, we propose an adaptive backstepping controller in order to stabilize the error coordinates $z_1 = \tilde{v}_t = v_t - v_{\text{ref}}$ and $z_2 = \dot{\lambda} - \zeta$, where ζ is a virtual control input. While the adaptive backstepping controller in Sec. III-B was derived using standard techniques (e.g. [16]), the adaptive velocity controller design is not as straightforward. Both the dynamics in Sec. III-B and in this section are non-autonomous, but for the joint controller in Sec. III-B, this did not affect the control design, it only complicated the analysis. For the velocity controller, however, the system structure is such that parameter-varying terms have to be compensated in the first design step. The dynamics of these terms depend on the unknown ocean current as well, which makes the control design different from the method for non-autonomous systems in [20]. The derivation of the controller is therefore presented step by step in the following.

To begin with, (7) is re-written and expanded to

$$\dot{v}_t = -\frac{c_t}{m}v_t + X_n(\lambda, \phi_0)v_n - \mathbf{X}_\phi(\lambda, \phi_0)\bar{\mathbf{e}}\frac{\partial g_2}{\partial \phi_0}\dot{\phi}_0 - \beta_{\lambda,1}^T(\lambda, \phi_0, \theta)V_c - X_\lambda(\lambda, \phi_0)\dot{\lambda}, \quad (8a)$$

$$\ddot{\lambda} = u_\lambda, \quad (8b)$$

with $\beta_{\lambda,1}^T(\lambda, \phi_0, \theta) = [-\frac{c_t}{m}, X_n(\lambda, \phi_0)]\mathbf{R}_\theta^T$ and $X_\lambda(\lambda, \phi_0) = \mathbf{X}_\phi(\lambda, \phi_0)[\alpha g_1(1)\cos(\lambda), \dots, \alpha g_1(N-1)\cos(\lambda + (N-2)\delta)]^T$. The first design step is then to use the virtual control input in (8a), $\zeta = \dot{\lambda}$, and the current estimate $\hat{\theta}_\lambda$ to stabilize $z_1 = \tilde{v}_t$ to zero. With the Lyapunov function $V_1 = \frac{1}{2}z_1^2$ and $\dot{z}_1 = \dot{v}_t$, we choose

$$\zeta = \frac{1}{X_\lambda(\lambda, \phi_0)}\zeta^*(\lambda, \phi_0, \dot{\phi}_0, v_n, \theta, z_1, \hat{\theta}_\lambda) \quad (9a)$$

$$\zeta^*(\cdot) = -\frac{c_t}{m}v_{\text{ref}} + X_n(\lambda, \phi_0)v_n - \mathbf{X}_\phi(\lambda, \phi_0)\bar{\mathbf{e}}\frac{\partial g_2}{\partial \phi_0}\dot{\phi}_0 - \beta_{\lambda,1}^T(\lambda, \phi_0, \theta)\hat{\theta}_\lambda + k_{\lambda,1}z_1 \quad (9b)$$

with the control gain $k_{\lambda,1} > 0$. Note that $X_\lambda(\lambda, \phi_0) > 0$ is bounded away from zero because of the phase shift δ [18]. However, when analysing \dot{V}_1 , we need to keep in mind that we do not directly control $\dot{\lambda}$, and that the current estimate $\hat{\theta}_\lambda$ is not necessarily exact, i.e. take into account the errors $z_2 = \dot{\lambda} - \zeta$ and $\hat{\theta}_\lambda = V_c - \hat{\theta}_\lambda$ in

$$\dot{z}_1 = -\left(\frac{c_t}{m} + k_{\lambda,1}\right)z_1 - X_\lambda(\lambda, \phi_0)z_2 - \tilde{\theta}_\lambda^T \beta_1(\lambda, \phi_0, \theta), \quad (10)$$

yielding

$$\dot{V}_1 = -\left(\frac{c_t}{m} + k_{\lambda,1}\right)z_1^2 - X_\lambda(\lambda, \phi_0)z_1z_2 - \tilde{\theta}_\lambda^T \beta_1(\lambda, \phi_0, \theta)z_1. \quad (11)$$

In order to stabilize z_2 to zero, we design the control law u_λ in the second step, using the Lyapunov function $V_2 = V_1 + \frac{1}{2}z_2^2 + \frac{1}{2\gamma_\lambda}\tilde{\theta}_\lambda^T \tilde{\theta}_\lambda$. For doing so, the time derivative $\dot{\zeta}$ is required. This complicates the control design significantly, because the dynamics of the uncontrolled state v_n enters the controller and it depends on the relative velocities, i.e. the unknown term. The time derivative of ζ is given by

$$\dot{\zeta} = \frac{\dot{\zeta}^*(\lambda, \dot{\lambda}, \phi_0, \dot{\phi}_0, \ddot{\phi}_0, v_n, \dot{v}_n, \theta, v_\theta, z_1, \dot{z}_1, \hat{\theta}_\lambda, \dot{\hat{\theta}}_\lambda)}{X_\lambda(\lambda, \phi_0)} - \frac{\zeta^*(\lambda, \phi_0, \dot{\phi}_0, v_n, \theta, z_1, \hat{\theta}_\lambda)\dot{X}_\lambda(\lambda, \dot{\lambda}, \phi_0, \dot{\phi}_0)}{X_\lambda^2(\lambda, \phi_0)}. \quad (12)$$

The second term in (12) only contains known signals and is not written down explicitly for compactness. However, $\dot{\zeta}^*$ in the first term contains unknown signals that have to be compensated for by the adaptive controller. It is obtained by taking the time derivative of ζ^* in (9b) and inserting (1h), (10) and $\tilde{\theta}_\lambda = V_c - \hat{\theta}_\lambda$:

$$\begin{aligned} \dot{\zeta}^* &= \left(\frac{\partial X_n}{\partial \lambda}\dot{\lambda} + \frac{\partial X_n}{\partial \phi_0}\dot{\phi}_0 - X_n\frac{c_n}{m}\right)v_n + X_n^2v_t \\ &+ \left[-X_n^2 - X_n\frac{c_n}{m}\right]\mathbf{R}_\theta^T V_c - \frac{\partial \mathbf{X}_\phi}{\partial \lambda}\bar{\mathbf{e}}\dot{\lambda}\frac{\partial g_2}{\partial \phi_0}\dot{\phi}_0 \\ &- \frac{\partial \mathbf{X}_\phi}{\partial \phi_0}\bar{\mathbf{e}}\frac{\partial g_2}{\partial \phi_0}\dot{\phi}_0^2 - \mathbf{X}_\phi\bar{\mathbf{e}}\frac{\partial^2 g_2}{\partial \phi_0^2}\dot{\phi}_0^2 - \mathbf{X}_\phi\bar{\mathbf{e}}\frac{\partial g_2}{\partial \phi_0}\ddot{\phi}_0 \\ &- \left(\frac{\partial \beta_{\lambda,1}^T}{\partial \lambda}\dot{\lambda} + \frac{\partial \beta_{\lambda,1}^T}{\partial \phi_0}\dot{\phi}_0 + \frac{\partial \beta_{\lambda,1}^T}{\partial \theta}v_\theta\right)\hat{\theta}_\lambda - \beta_{\lambda,1}^T\dot{\hat{\theta}}_\lambda \\ &+ k_{\lambda,1}\left(-\left(\frac{c_t}{m} + k_{\lambda,1}\right)z_1 - X_\lambda z_2 + \beta_{\lambda,1}^T\hat{\theta}_\lambda - \beta_{\lambda,1}^T V_c\right). \end{aligned} \quad (13)$$

In (13), the function arguments are omitted for better readability and the time derivative of the current estimate $\dot{\hat{\theta}}_\lambda$ has not been inserted yet, because it will be designed in the next step. From (13) we can see why backstepping methods from the literature [16,20] cannot be applied: the unknown term, V_c , enters the time derivative of the virtual control input ζ through $\dot{\zeta}^*$. With the Lyapunov function V_2 and $\dot{z}_2 = u_\lambda - \zeta$, the control input is chosen as

$$\begin{aligned} u_\lambda &= -(k_{\lambda,2} + k_{\lambda,1})z_2 + \left(X_\lambda + \frac{X_n^2}{X_\lambda} - \frac{k_{\lambda,1}}{X_\lambda}\left(\frac{c_t}{m} + k_{\lambda,1}\right)\right)z_1 \\ &+ \frac{X_n^2}{X_\lambda}v_{\text{ref}} - \frac{\zeta^*\dot{X}_\lambda}{X_\lambda^2} + \frac{1}{X_\lambda}\left(\frac{\partial X_n}{\partial \lambda}\dot{\lambda} + \frac{\partial X_n}{\partial \phi_0}\dot{\phi}_0 - X_n\frac{c_n}{m}\right)v_n \\ &- \frac{1}{X_\lambda}\left(\frac{\partial \mathbf{X}_\phi}{\partial \lambda}\bar{\mathbf{e}}\dot{\lambda}\frac{\partial g_2}{\partial \phi_0}\dot{\phi}_0 + \frac{\partial \mathbf{X}_\phi}{\partial \phi_0}\bar{\mathbf{e}}\frac{\partial g_2}{\partial \phi_0}\dot{\phi}_0^2 + \mathbf{X}_\phi\bar{\mathbf{e}}\frac{\partial^2 g_2}{\partial \phi_0^2}\dot{\phi}_0^2\right. \\ &+ \left.\mathbf{X}_\phi\bar{\mathbf{e}}\frac{\partial g_2}{\partial \phi_0}\ddot{\phi}_0\right) - \frac{1}{X_\lambda}\left(\frac{\partial \beta_{\lambda,1}^T}{\partial \lambda}\dot{\lambda} + \frac{\partial \beta_{\lambda,1}^T}{\partial \phi_0}\dot{\phi}_0 + \frac{\partial \beta_{\lambda,1}^T}{\partial \theta}v_\theta\right. \\ &\left.- k_{\lambda,1}\beta_{\lambda,1}^T\right)\hat{\theta}_\lambda - \frac{1}{X_\lambda}\beta_{\lambda,1}^T\dot{\hat{\theta}}_\lambda + \beta_{\lambda,2}^T\hat{\theta}_\lambda \end{aligned} \quad (14)$$

with the control gain $k_{\lambda,2} > 0$ and

$$\beta_{\lambda,2}^T = \frac{1}{X_\lambda}\left[k_{\lambda,1}\frac{c_t}{m} - X_n^2 - k_{\lambda,1}X_n + X_n\frac{c_n}{m}\right]\mathbf{R}_\theta^T. \quad (15)$$

Again, the function arguments in (14) and (15) have been omitted for compactness. The control law (14) yields the following dynamics of the error z_2 :

$$\dot{z}_2 = X_\lambda(\lambda, \phi_0)z_1 - k_{\lambda,2}z_2 - \tilde{\theta}_\lambda^T \beta_2(\lambda, \phi_0, \theta) \quad (16)$$

and thus

$$\begin{aligned} \dot{V}_2 = & -\left(\frac{c_t}{m} + k_{\lambda,1}\right)z_1^2 - k_{\lambda,2}z_2^2 - \tilde{\theta}_\lambda^T \beta_{\lambda,1}(\lambda, \phi_0, \theta)z_1 \\ & - \tilde{\theta}_\lambda^T \beta_{\lambda,2}(\lambda, \phi_0, \theta)z_2 - \frac{1}{\gamma_\lambda} \tilde{\theta}_\lambda^T \dot{\theta}_\lambda. \end{aligned} \quad (17)$$

Finally, the adaptive update law is designed as

$$\dot{\hat{\theta}}_\lambda = -\gamma_\lambda(\beta_{\lambda,1}z_1 + \beta_{\lambda,2}z_2), \quad (18)$$

thus cancelling the remaining indefinite terms in \dot{V}_2 and making

$$\dot{V}_2 = -\left(\frac{c_t}{m} + k_{\lambda,1}\right)z_1^2 - k_{\lambda,2}z_2^2 \quad (19)$$

negative semidefinite. Postponing the design of the adaptive update law until the last step is inspired by the tuning function procedure in [16]. Note, however, that the system structure in this article is fundamentally different from the autonomous system in [16].

Proposition 2: Suppose that the robot described by (1) moves according to (3) on the manifold Γ_3 . The adaptive control law (14),(18) asymptotically stabilizes the set

$$\Gamma_2 = \{(\mathbf{x}, \dot{\mathbf{x}}, \lambda, \dot{\lambda}, \phi_0, \dot{\phi}_0) \in \Gamma_3 : v_t = v_{\text{ref}}\}. \quad (20)$$

i. e. $\tilde{v}_t \rightarrow 0$.

Proof: The closed-loop system with the error coordinates z_1 and z_2 reads as given by (10) and (16). With the Lyapunov function $V_2 = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2 + \frac{1}{2\gamma_\lambda} \tilde{\theta}_\lambda^T \tilde{\theta}_\lambda$ we get the time derivative \dot{V}_2 in (19), which is negative semidefinite. We can conclude that the origin $(z_1, z_2, \tilde{\theta}_\lambda) = (0, 0, 0)$ is uniformly globally stable and z_1, z_2 , and $\tilde{\theta}_\lambda$ are bounded. Again, the analysed error system is a non-autonomous system because it depends on the time-varying parameters ϕ_0, θ . However, V_2 is lower bounded by zero and \dot{V}_2 is finite because all time-varying parameters enter the dynamics of z_1, z_2 within a sine, cosine, or saturation function. With these conditions, we can employ Barbalat's Lemma to conclude that $(z_1, z_2) \rightarrow (0, 0)$ as $t \rightarrow \infty$. This implies that $v_t \rightarrow v_{\text{ref}}$ and $\dot{\lambda} \rightarrow \zeta$. ■

Remark 4: Note that the result in Prop. 2 implies that $\tilde{\theta}_\lambda$ is bounded. However, in order to show convergence of the estimate $\hat{\theta}_\lambda \rightarrow V_c$, one has to check if the regressor defined by β_1, β_2 is persistently exciting (PE). In the particular case of snake robots, this is complicated by the fact that the regressor and thereby the PE property depend on the gait parameters, as will be discussed in Sec. V. A formal analysis of the regressor and rigorous conditions on the gait for PE remain a topic for future work.

D. Orientation control

In the last step of the hierarchical control design, we propose an orientation controller based on the reduced system dynamics on Γ_2 . The reduced system is obtained by evaluating (1b),(1f) on the invariant manifold Γ_2 :

$$\dot{\theta} = v_\theta, \quad (21a)$$

$$\dot{v}_\theta = -\lambda_1 v_\theta + \frac{\psi_1(\theta)}{N-1} \mathbf{e}^T \phi_{\text{ref}}(\lambda, \phi_0), \quad (21b)$$

where $\psi_1(\theta) = \lambda_2(v_{\text{ref}} - [\cos \theta, \sin \theta]V_c)$. The objective of the orientation controller is to use the control input $u_\phi = \ddot{\phi}_0$

to stabilize the error coordinate $\tilde{\theta} = \theta - \theta_{\text{ref}}$ to zero. Inspired by the control design in [12], this is achieved by taking the time derivatives of (21b) until the input u_ϕ shows up:

$$\begin{aligned} v_\theta^{(3)} = & -\lambda_1 \ddot{v}_\theta + \psi_2(\phi_0, \dot{\phi}_0, \lambda, \dot{\lambda}, \theta, v_\theta, \dot{v}_\theta) \\ & + \psi_1(\theta) \frac{\partial g_2}{\partial \phi_0} u_\phi + \psi_3(\theta, \lambda) \ddot{\lambda}. \end{aligned} \quad (22)$$

The expressions for $\psi_2(\cdot)$ and $\psi_3(\cdot)$ in (22) can be obtained by taking the second time derivative of (21b) and are omitted because of space restrictions. The single terms of (22) depend on the unknown ocean current, which complicates the control design. In particular, the function ψ_1 that is multiplied with the control input contains the unknown signal, which is why we cannot design an adaptive controller analogously to the joint and velocity controllers. We will therefore make the following assumption.

Assumption 2: For the design of the orientation controller, an exact current estimate $\hat{\theta} = V_c$ is available.

Remark 5: The adaptive term in the velocity controller (14) provides an estimate of the unknown current. More specifically, the analysis in Sec. III-C shows that $\hat{\theta}_\lambda$ converges and the estimation error $\tilde{\theta}_\lambda$ is bounded. If the regressor in (18) is PE, the remaining offset will converge to zero.

Based on Ass. 2 and (22), the control law

$$u_\phi = \frac{1}{\hat{\psi}_1(\theta) \frac{\partial g_2}{\partial \phi_0}} \left[\lambda_1 \tilde{\theta}^{(3)} - \hat{\psi}_2(\cdot) - k_{\phi,3} \tilde{\theta}^{(3)} - k_{\phi,2} \ddot{\tilde{\theta}} - k_{\phi,1} \dot{\tilde{\theta}} - k_{\phi,0} \tilde{\theta} \right] \quad (23)$$

is proposed. In (23), the control gains $k_{\phi,3}, k_{\phi,2}, k_{\phi,1}, k_{\phi,0} > 0$ are introduced and the superscript $\hat{\cdot}$ indicates that instead of the unknown current, the estimate $\hat{\theta}_\lambda$ is used to compute $\psi_1(\cdot)$ and $\psi_2(\cdot)$. The same is done for the computation of the time derivatives of θ . Note that the last term on the right hand side of (22) cannot be compensated by the feedback linearising controller (23). This is due to the design of u_λ in (14), which contains a term depending on $\ddot{\phi}_0$.

Remark 6: Because of prioritizing the velocity control higher than the orientation control, the choice $v_{\text{ref}} > V_{c,\text{max}}$, and Ass. 2, ψ_1 is bounded away from zero. Furthermore, $\frac{\partial g_2}{\partial \phi_0} \neq 0$, because g_2 is strictly increasing. The controller in (23) therefore improves previous results for terrestrial snake robots [12,18], where singularities were an issue in the orientation controller. However, care needs to be taken when implementing (23). When tuning the control system it needs to be made sure that the current estimate converges sufficiently fast in order to avoid a singularity. Similarly, the control gains in (23) should be tuned such that ϕ_0 remains sufficiently small to not drive g_2 into saturation. Even though $\frac{\partial g_2}{\partial \phi_0} \neq 0$, numerical problems can occur for large ϕ_0 .

Under Ass. 2, the following result holds for the orientation controller in (23):

Proposition 3: Suppose that the robot described by (1) moves according to (3) and (14) on the manifold Γ_2 . If the control input of the velocity controller has no finite escape times and goes to zero, $u_\lambda \rightarrow 0$, the control law (23) asymptotically stabilizes the manifold

$$\Gamma_1 = \{(\mathbf{x}, \dot{\mathbf{x}}, \lambda, \dot{\lambda}, \phi_0, \dot{\phi}_0) \in \Gamma_2 : \|\theta - \theta_{\text{ref}}\| = 0\}, \quad (24)$$

i. e. $\theta \rightarrow \theta_{\text{ref}}$.

Proof: The closed-loop system (22) with (23) is a linear system with the parameter-varying input matrix $\mathbf{b}(\lambda, \theta) = [\mathbf{0}_{1 \times 3}, \frac{\lambda_2(v_{\text{ref}} - |\cos \theta, \sin \theta| V_c)}{N-1} \sum_{i=0}^{N-1} \alpha g_1(i) \cos(\lambda + (i-1)\delta)]^T$ and input $\ddot{\lambda}$. With $\Theta = [\tilde{\theta}, \dot{\tilde{\theta}}, \ddot{\tilde{\theta}}, \tilde{\theta}^{(3)}]^T$ we have that

$$\dot{\Theta} = \underbrace{\begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -k_{\phi,0} & -k_{\phi,1} & -k_{\phi,2} & -k_{\phi,3} \end{bmatrix}}_{\mathbf{H}} \Theta + \mathbf{b}(\lambda, \theta) \ddot{\lambda}. \quad (25)$$

The matrix \mathbf{H} is Hurwitz by design and it can be verified with the matrix exponential of \mathbf{H} and the bound $\|\mathbf{b}(\lambda, \theta)\| \leq B$ that (25) is ISS and disturbed by the input $\ddot{\lambda}$:

$$\|\Theta(t)\| \leq ce^{-\tilde{\lambda}(t-t_0)} \|\Theta(t_0)\| + \frac{cB}{\tilde{\lambda}} \ddot{\lambda}. \quad (26)$$

If the finite bound $\|\ddot{\lambda}\| \leq \epsilon_\lambda$ exists and $\tilde{\lambda} \rightarrow 0$ as $t \rightarrow \infty$, the orientation will therefore converge to $\theta \rightarrow \theta_{\text{ref}}$. ■

Note that even if $\tilde{\lambda}$ does not converge to zero, the disturbance by $\ddot{\lambda}$ can be made small by making $\tilde{\lambda}$ in (26) large, i. e. placing the eigenvalues of \mathbf{H} such that they are far in the left hand plane by choosing the gains accordingly. In this case, $\tilde{\theta}$ is practically stabilized to zero.

IV. STABILITY ANALYSIS

In Sec. V, simulation results will demonstrate that the proposed control system (3),(14),(23) can stabilize all error coordinates to zero for an underwater snake robot that is exposed to an unknown, constant ocean current. In this section we will present an analysis that shows boundedness of the zero dynamics. Furthermore, we sketch how to show uniform global asymptotic stability of the closed-loop system under conditions that guarantee PE in the velocity controller.

As was pointed out in Remark 2, the sideways velocity v_n of the robot cannot be controlled directly. However, the following property holds for the closed-loop system.

Proposition 4: Suppose that the body shape, forward velocity, and orientation of an underwater snake robot described by (1) are controlled according to (3),(14), and (23). Then, the sideways velocity, v_n , is uniformly bounded.

Proof: With the Lyapunov function $V_n = \frac{1}{2}v_n^2$ and (1h) the time derivative \dot{V}_n is given by

$$\dot{V}_n = -\frac{c_n}{m}v_n^2 + \frac{2c_p}{Nm}\mathbf{e}^T\phi v_t v_n + (\frac{c_n}{m}V_n - \frac{2c_p}{Nm}\mathbf{e}^T\phi V_t)v_n, \quad (27)$$

where V_n and V_t are the components of the current velocity in the body-aligned frame. The second and third term on the right hand side of (27) are indefinite. However, we know that v_t and $\mathbf{e}^T\phi$ are bounded since $v_t \rightarrow v_{\text{ref}}$ and $\phi \rightarrow \phi_{\text{ref}}$, and $\|\mathbf{e}^T\phi_{\text{ref}}\| \leq \epsilon_\phi$. We denote these bounds by $\|v_t\| \leq \bar{v}_t$ and $\|\mathbf{e}^T\phi\| \leq \bar{\epsilon}_\phi$. Furthermore we conclude from the bound on the current magnitude in Ass. 1, $V_{c,\text{max}}$, that $\|V_n\| \leq V_{c,\text{max}}$ and $\|V_t\| \leq V_{c,\text{max}}$.

Therefore we have that

$$\dot{V}_n \leq -\frac{c_n}{m}v_n^2 + v_n k. \quad (28)$$

We can now use Young's inequality [21]

$$ab \leq \frac{\xi a^2}{2} + \frac{b^2}{2\xi}, \quad \xi > 0 \quad (29)$$

to reformulate (28) to

$$\dot{V}_n \leq (-\frac{c_n}{m} + \frac{\xi}{2})v_n^2 + \frac{k^2}{2\xi}. \quad (30)$$

By choosing $\xi < \frac{2c_n}{m}$ we can make sure that the coefficient of v_n^2 is negative and conclude from the Comparison Lemma [19] that

$$V_n(t) \leq e^{-c_1 t} V_n(0) + c_2, \quad (31)$$

which implies that v_n is bounded. ■

Remark 7: Note that the result in Prop. 4 does not depend on Ass. 2.

Future work will establish conditions on the gait in (2) under which the regressor in (18) is PE, and the convergence of the current estimate to the true value can be shown. With these conditions, the orientation controller can be shown to asymptotically stabilize the set

$$\Gamma_1 = \{(\mathbf{x}, \dot{\mathbf{x}}, \lambda, \dot{\lambda}, \phi_0, \dot{\phi}_0) \in \Gamma_2 : \|\theta - \theta_{\text{ref}}\| \leq \epsilon\}, \quad (32)$$

if all states are bounded. In previous studies for terrestrial snake robots [12,18], numerical simulations were used to show that the states generated by u_λ and u_ϕ remain bounded. Similarly, in the next section of this article, we will see simulation results that indicate that all states remain bounded, and the error coordinates converge to zero.

V. SIMULATION STUDY

This section presents simulation results that illustrate the performance of the control system.

A. Simulation set-up

The model (1) and the control system (3),(14),(23) were implemented in Matlab and simulated using the *ode23t* solver with an absolute and relative error tolerance of 10e-6. The model parameters and control gains are displayed in Table I. The gains were obtained by tuning the control system stage by stage. The gait parameters of (2) were set to $\alpha = 7$ cm and $\delta = 40^\circ$, and the gait functions were chosen as $g(i) = 1$ and $g_2(\phi_0) = \phi_{0,\text{max}} \tanh(\frac{\phi_0}{\phi_{0,\text{max}}})$ with $\phi_{0,\text{max}} = \alpha$. The references for the velocity and orientation controller were set to $v_{\text{ref}} = 8$ cm/s and $\theta_{\text{ref}} = 45^\circ$. All states were initialized at zero, i. e. the robot was initially straight, aligned with the x -axis, and fixed in the origin. As a first step, only the joint and velocity controllers were activated, and the orientation control input u_ϕ was set to zero in order to investigate Ass. 2. In a second step, the entire control system was simulated.

TABLE I: Parameters of the simulation.

N	m	c_t	c_n	c_p	λ_1	λ_2	V_x	V_y	
10	1.56 kg	4.45	17.3	35.7	6	120	- 4 cm/s	- 1 cm/s	
k_1	k_2	γ_1	$k_{\lambda,1}$	$k_{\lambda,2}$	γ_2	$k_{\phi,0}$	$k_{\phi,1}$	$k_{\phi,2}$	$k_{\phi,3}$
3	6	1	0.1	10	0.04	2	21	60	20

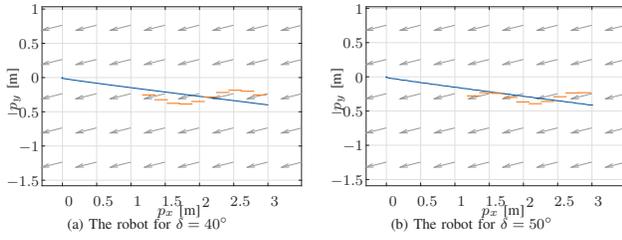


Fig. 3: The path of the robot during velocity control.

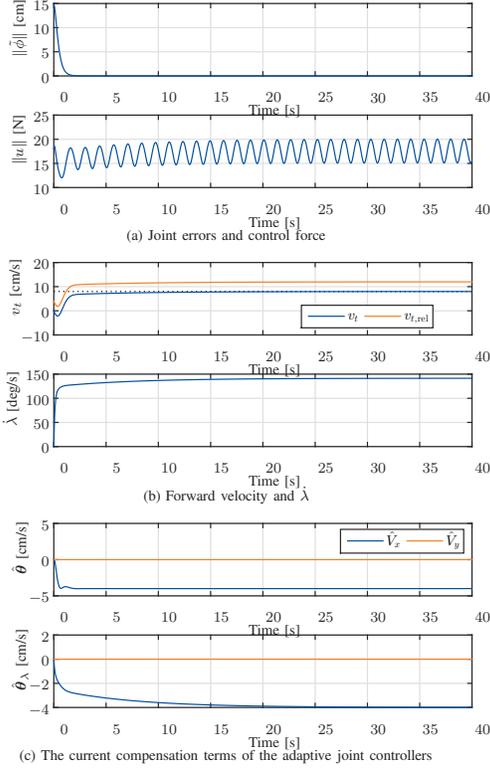


Fig. 4: Simulation results of the velocity controller, $\delta = 40^\circ$.

B. Simulation results

Fig. 3a shows the path of the robot during the first simulation. The error signal of the joint controller and the control torque are displayed in Fig. 4a, the forward velocity and the state related to the body frequency, λ , in Fig. 4b, and the adaptive term of both the joint and the velocity controller in Fig. 4c. It becomes clear from the figures that all error coordinates converge to zero and all signals remain bounded. However, for the current estimate of both control stages, only one component converges to the correct value. Clearly, the regressors used in the adaptive control laws are not PE for the particular choice of gait parameters.

The simulation was therefore re-run with a phase shift of $\delta = 50^\circ$. The path of the robot during this simulation is displayed in Fig. 3b. Fig. 5 shows the analogous signals of Fig. 4 for the second simulation. All controlled states still converge, all signals remain bounded, and the current term of the joint controller still does not provide an exact estimate. However, it can now be seen from Fig. 5c that the current estimate of the velocity controller, $\hat{\theta}_\lambda$, converges to

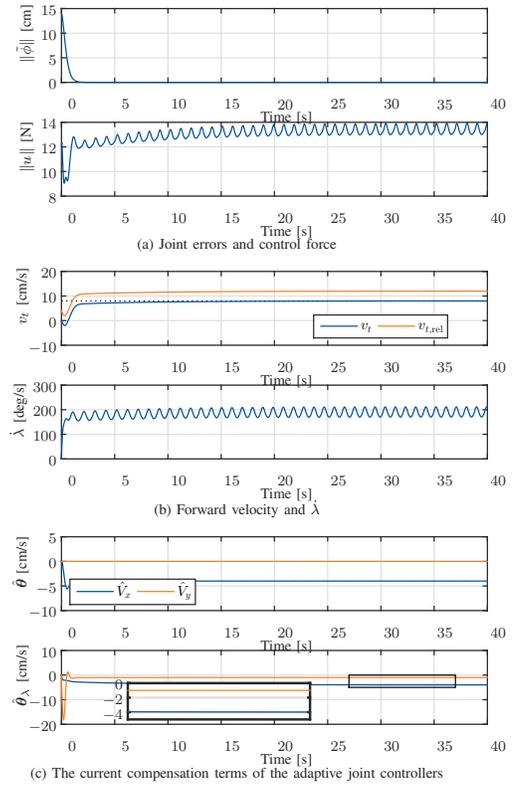


Fig. 5: Simulation results of the velocity controller, $\delta = 50^\circ$.

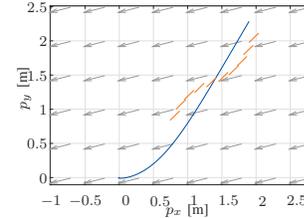


Fig. 6: The path of the robot with the complete controller.

the correct values after an initial overshoot. This overshoot is a result of not tuning the control gains for the new gait parameters. The parameter convergence in Fig. 5c indicates that Ass. 2 is a valid assumption if the gait parameters are chosen accordingly.

The first simulation was repeated with the orientation controller. The path of the robot is shown in Fig. 6, where we see that the robot is now turning towards its reference orientation. The signals of the single control stages are displayed in Fig. 7, which show that all error signals converge to zero while all signals remain bounded. In addition to the signals presented in the above discussion, the orientation of the robot converges towards its reference, even though Ass. 2 is violated in the first scenario. This is a result of the turning of the robot, which provides additional information to the adaptive controller. The same effect can be observed in Fig. 7d, where the current estimate of the velocity controller now converges to the correct value.

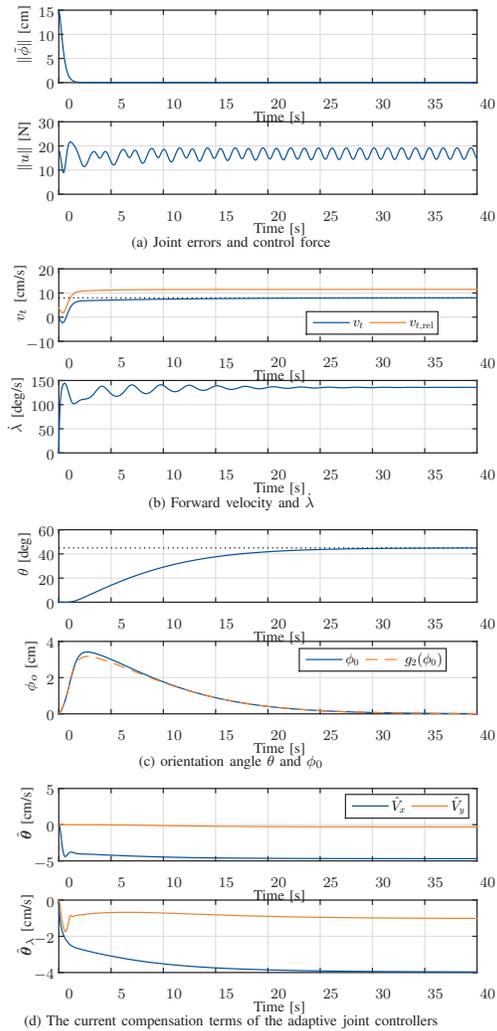


Fig. 7: Simulation results of the complete controller.

VI. CONCLUSIONS

We have presented a control system for velocity and orientation control of underwater snake robots using absolute velocity feedback. The control system has a hierarchical structure, where the highest priority is to stabilize virtual joint constraints that encode a planar gait. In order to do so, we designed an adaptive joint controller and showed that the controller asymptotically stabilizes the constraint manifold. The second and third priority of the control system were to control the velocity and the orientation of the robot using dynamic compensators whose states parametrize the virtual joint constraints. We proposed an adaptive controller that asymptotically stabilizes the forward velocity error to zero and an orientation controller that utilizes the current estimate of the velocity controller. It was shown that the zero dynamics of the closed-loop system remains bounded, and simulation results demonstrated the performance of the controller.

In future work we will formulate conditions for the gait under which we can show PE in the velocity controller. This will enable a formal stability analysis of the entire closed-loop system. We will furthermore extend the orientation

controller to take into account a time-varying reference, and combine the system with a guidance scheme in order to do maneuvering control.

REFERENCES

- [1] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford: Oxford University Press, 1993.
- [2] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake Robots: Modelling, Mechatronics, and Control*, ser. Advances in Industrial Control. Springer London, 2013.
- [3] A. Crespi and A. J. Ijspeert, “AmphiBot II: An amphibious snake robot that crawls and swims using a central pattern generator,” in *Proc. 9th Int. Conf. Climbing and Walking Robots*, Brussels, Belgium, Sep. 2006.
- [4] M. Porez, F. Boyer, and A. J. Ijspeert, “Improved lighthill fish swimming model for bio-inspired robots: Modeling, computational aspects and experimental comparisons,” *Int. J. Robot. Res.*, vol. 33, no. 10, pp. 1322–1341, 2014.
- [5] E. Kelasidi, P. Liljebäck, K. Y. Pettersen, and J. T. Gravdahl, “Innovation in underwater robots: Biologically inspired swimming snake robots,” *IEEE Robotics Automation Magazine*, vol. 23, no. 1, pp. 44–62, March 2016.
- [6] A. Raj and A. Thakur, “Fish-inspired robots: design, sensing, actuation, and autonomy – a review of research,” *Bioinspiration & Biomimetics*, vol. 11, no. 031001, 2016.
- [7] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.
- [8] A. M. Kohl, K. Y. Pettersen, E. Kelasidi, and J. T. Gravdahl, “Planar path following of underwater snake robots in the presence of ocean currents,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 383–390, 2016.
- [9] A. M. Kohl, E. Kelasidi, A. Mohammadi, M. Maggiore, and K. Pettersen, “Planar maneuvering control of underwater snake robots using virtual holonomic constraints,” *Bioinspiration & Biomimetics*, vol. 11, no. 065005, 2016.
- [10] O. Akanyeti, J. C. Brown, L. D. Chambers, H. el Daou, M.-C. Fiazza, P. Fiorini, J. Ježov, D. S. Jung, M. Kruusmaa, M. Listak, A. Liszewski, J. L. Maud, W. M. Megill, L. Rossi, A. Quattieri, F. Rizzi, T. Salumäe, G. Toming, R. Venturelli, F. Visentin, and M. D. Vittorio, “FILOSE for Svenning: A flow sensing bioinspired robot,” *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 51–62, 2014.
- [11] R. Christ and R. Wernli, *The ROV Manual: A User Guide for Remotely Operated Vehicles*. Butterworth-Heinemann, Oxford, 2014.
- [12] E. Rezapour, A. Hofmann, K. Y. Pettersen, A. Mohammadi, and M. Maggiore, “Virtual holonomic constraint based direction following control of planar snake robots described by a simplified model,” in *Proc. IEEE Conf. Control Applications*, Antibes, France, Oct. 2014.
- [13] M. El-Hawwary and M. Maggiore, “Reduction theorems for stability of closed sets with application to backstepping control design,” *Automatica*, vol. 49, pp. 214–222, 2013.
- [14] D. J. W. Belleter, C. Paliotta, M. Maggiore, and K. Y. Pettersen, “Path following for underactuated marine vessels,” in *Proc. 10th IFAC Symposium on Nonlinear Control Systems*, Monterey, CA, Aug. 2016.
- [15] A. Mohammadi, E. Rezapour, M. Maggiore, and K. Y. Pettersen, “Maneuvering control of planar snake robots using virtual holonomic constraints,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 884 – 899, 2015.
- [16] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. Wiley, 1995.
- [17] A. M. Kohl, K. Y. Pettersen, E. Kelasidi, and J. T. Gravdahl, “Analysis of underwater snake robot locomotion based on a control-oriented model,” in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, Zhuhai, China, Dec. 2015.
- [18] E. Rezapour, A. Hofmann, and K. Y. Pettersen, “Maneuvering control of planar snake robots based on a simplified model,” in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, Bali, Indonesia, Dec. 2014.
- [19] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
- [20] T. I. Fossen, A. Loría, and A. Teel, “A theorem for UGAS and ULES of (passive) nonautonomous systems: robust control of mechanical systems and ships,” *International Journal of Robust and Nonlinear Control*, vol. 11, pp. 95–108, 2001.
- [21] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, ser. Graduate Texts in Mathematics, S. Axler, F. Gehring, and K. Ribet, Eds. Springer, 1989, vol. 60.