Data-driven control of planar snake robot locomotion

M. L. Scarpa, B. Nortmann, K. Y. Pettersen and T. Mylvaganam

***Abstract*— A direct data-driven strategy for snake-robot locomotion control is proposed in this paper. The approach leads to a time-varying state feedback controller with robustness guarantees. Instead of relying on exact model knowledge - which is often not available in practice - the proposed control strategy requires only input-state data collected during offline experiments. The efficacy of the proposed strategy is demonstrated via simulations. Notably, by using data to compensate for inaccurate models, the proposed control strategy can lead to significant improvements in closed-loop performance compared to existing (model-based) control strategies, while also eliminating the need for manual tuning of control parameters.**

I. Introduction

Snake robots have gained interest for a wide range of applications. Mimicking the motion of biological snakes (see *e.g.* [1], [2]), notably their *undulatory locomotion* pattern, snake robots have excellent mobility and maneuverability, even in hard-to-reach, challenging environments. With these capabilities, snake robots are well suited for performing monitoring and intervention tasks in the context of ground-based, subsea and space applications (see, for instance, [3], [4], [5], [6], [7], [8] and references therein). However, obtaining accurate mathematical models to describe the dynamics of snake robots is challenging. The complex friction forces acting between the robot and its surroundings, for instance, play a fundamental role in the locomotion of snake robots, but are very difficult to model [6]. Existing models (see *e.g.* [6], [9], [10]) are often based to some degree on empirical studies, and the presence of parameters that are difficult to estimate accurately may necessitate that control parameters are tuned via ad-hoc procedures (see, *e.g.* [10]). The fact that snake robots are underactuated poses a further challenge in terms of control design. Thus, with the aim of easing the control design task, a simplified, *control-oriented* model has been introduced in [4]. The model provides a good *qualitative* representation of the dynamical behaviour of snake robots and renders the task of designing control laws ensuring such robots follow a desired locomotion readily solvable. More precisely, in the presence of *complete knowledge* of the underlying model, it has been demonstrated in [4], [6] that a relatively simple control law can be designed, via partial feedback linearisation, to render a desired locomotion globally exponentially stable (GES). In this paper, we use data to derive a similar control law in the absence of accurate model parameter knowledge.

Direct data-driven control, see *e.g.* [11], [12], allows to design controllers directly using measured data, bypassing any modelling step. Snake robots, which are complex nonlinear systems, achieve a desired motion via undulatory locomotion by tracking a *repetitive*, sinusoidal joint angle trajectory. By approximating the behaviour of the actuated degrees of freedom of the robot around the undulatory motion trajectory as a linear time-varying (LTV) system, and hinging upon a data-driven framework for LTV systems [13], [14], we implement a direct data-driven tracking controller. Differently from existing techniques, the proposed data-driven control design method does not require accurate prior knowledge of model parameters or manual tuning. An additional benefit of the approach, which results in a time-varying controller, is that it is naturally amenable to settings in which model parameters are time-varying. For instance, varying friction coefficients as the snake robot "slithers" across different surfaces. We demonstarte (via simulations) that by using data to compensate for lack of accurate model knowledge, the proposed control strategy outperforms alternative control laws employed in the literature.

The remainder of this paper is organised as follows. In Section II the considered problem is introduced along with some preliminaries on data-driven control of unknown LTV systems. A brief overview of existing results on modelling and locomotion of snake robots is provided in Section III. The main result of the paper, namely a data-driven control design strategy to ensure a snake robot follows a desired locomotion pattern is presented in Section IV. Finally, the proposed control scheme is illustrated via simulations in Section V and concluding remarks are provided in Section VI.

Notation. Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable function, then $\partial f/\partial x$ denotes the Jacobian matrix of f with respect to the variable $x \in \mathbb{R}^n$. When $m = 1$, $\partial f/\partial x = [\partial f/\partial x_1, ..., \partial f/\partial x_n]$ denotes the row vector of partial derivatives of f with respect to $x = [x_1, x_2, \dots, x_n]^\top$. I_n represents the $n \times n$ identity matrix, and 0 the zero matrix of appropriate dimension. Given a matrix $A \in \mathbb{R}^{n\times n}$, $A > 0$ ($A \geq 0$) indicates that the matrix is positive definite (positive semi-definite). We use square brackets to denote a discrete-time signal $s[\cdot]$ and round brackets to denote a continuous-time signal $s(\cdot)$. A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class $\mathcal{K}$-function if it is continuous, strictly increasing and $\gamma(0) = 0$.

M. L. Scarpa, B. Nortmann and T. Mylvaganam are with the Department of Aeronautics, Imperial College London, London SW7 2AZ, UK (e-mail: {maria.scarpa18, benita.nortmann15, t.mylvaganam}@imperial.ac.uk).

K. Y. Pettersen is with the Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (email: Kristin.Y.Pettersen@ntnu.no).

The work of K. Y. Pettersen has been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, through the ERC Advanced Grant 101017697-CRÈME, and by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 – NTNU AMOS.

Given a vector $v \in \mathbb{R}^n$, $||v||$ denotes its Euclidean norm.

II. Problem formulation and preliminaries

The dynamic behaviour of snake robots is highly dependent on friction forces - which are difficult to model or estimate accurately - posing a major challenge in terms of control design. Motivated by this, we consider the problem of designing - without requiring the knowledge of the coefficient of frictions in the model - a feedback control law, which ensures the snake robot follows a desired reference path, and we provide a data-driven solution to the considered problem.

A. Preliminaries on data-driven control of LTV systems

We introduce here some preliminaries concerning the direct data-driven control technique, which will be used to design a controller for the snake robot (see [14], [13] for more details). Consider a discrete-time system subjected to process noise (denoted by $d[k]$) described by

$$x[k+1] = A[k]x[k] + B[k]u[k] + d[k] \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the state of the system, $u \in \mathbb{R}^m$ denotes the control input, and $A[k]$ and $B[k]$ of appropriate dimension denote the unknown (time-varying) dynamics and input matrices, respectively. With the aim of designing a state feedback control law for system (1) of the form

$$u[k] = K[k]x[k]\,, \tag{2}$$

for $k \in [0, N]$, consider the following standing assumption.

Assumption 1. *It is possible to collect an ensemble of input-state data sequences*[1] $u_{d,j,[0,N-1]}, x_{d,j,[0,N]}$, *for* $j = 1, \ldots, L$, *with* $L \geq n+m$, *capturing the same time-varying behaviour for* $k = 0, \ldots, N$, *with* $N \in \mathbb{N}$.

The L data sequences are combined to form the matrices

$$X[k] = [x_{d,1}[k], x_{d,2}[k], \ldots, x_{d,L}[k]]\,, \tag{3}$$

for $k = 0, \ldots, N$, and

$$U[k] = [u_{d,1}[k], u_{d,2}[k], \ldots, u_{d,L}[k]] \tag{4}$$

for $k = 0, \ldots, N-1$. A similar matrix can be assembled for the (unknown) process noise $D[k]$, for $k = 0, \ldots, N-1$, corresponding to the measured data. Note that $D[k]$ is not measured. In [14] it has been shown that (3) and (4) can be used to design controllers of the form (2), which guarantee a decreasing bound on the closed-loop trajectories in the presence of process noise, as recalled in the following statement.

Lemma 1. *[14] Consider the system* (1). *Suppose that data matrices* (3) *and* (4) *are available and that these are such that the rank condition* (5) *holds.*

$$\textit{rank} \begin{bmatrix} X[k] \\ U[k] \end{bmatrix} = n+m\,. \tag{5}$$

Finally, suppose $D[k]$ satisfies a quadratic bound, that is

$$\begin{bmatrix} I_n \\ D[k]^\top \end{bmatrix}^\top \begin{bmatrix} Q_r[k] & S_r[k] \\ S_r[k]^\top & R_r[k] \end{bmatrix} \begin{bmatrix} I_n \\ D[k]^\top \end{bmatrix} \geq 0, \tag{6}$$

with $Q_r[k] \in \mathbb{R}^{n\times n}$, $S_r[k] \in \mathbb{R}^{n\times L}$ and $R_r[k] \leq 0 \in \mathbb{R}^{L\times L}$ for $k = 0, \ldots N-1$. Then a control input of the form (2) *with*

$$K[k] = U[k]Y[k]P[k]^{-1}, \tag{7}$$

is such that the trajectories of the closed-loop system (1), (2) *satisfy the bound*

$$\|x[k]\| \leq \sqrt{\frac{\rho}{\eta}}\left(1 - \frac{1}{\rho}\right)^{\frac{k}{2}} \|x[0]\| + \gamma_1(|d|_{k-1}, k),$$

for $k = 0, \ldots, N$, where

$$\gamma_1(|d|_{k-1}, k) = \left(\sum_{j=0}^{k-1} \sqrt{\frac{\rho}{\eta}}\left(1 - \frac{1}{\rho}\right)^{\frac{k-1-j}{2}}\right) |d|_{k-1},$$

with $|d|_{k-1} = sup\,\{\|d(j)\|, 0 \leq j \leq k-1\} \leq \infty$, is a class $\mathcal{K}$ function, and where $Y[k]$, $P[k]$ are a solution of

$$\begin{bmatrix} P[k+1] - I_n - Q_r[k] & -S_r[k] & X[k+1]Y[k] \\ -S_r[k]^\top & -R_r[k] & Y[k] \\ Y[k]^\top X[k+1]^\top & Y[k]^\top & P[k] \end{bmatrix} \geq 0, \tag{8a}$$

$$X[k]Y[k] = P[k], \tag{8b}$$

for $k = 0, \ldots, N-1$, and

$$\eta I_n \leq P[k] \leq \rho I_n, \tag{8c}$$

for $k = 0, \ldots N$, and $\eta \geq 1$, $\rho > \eta$.

The result in Lemma 1 entails that the control design involves data matrices only, *i.e.* it does not require any knowledge of the (time-varying) matrices $A[k]$ and $B[k]$.

In the following sections, we first recall some insights related to the dynamics and locomotion of snake robots, before showing how the described data-driven control method can be applied to the snake robot system.

III. Dynamics and locomotion of snake robots

Some preliminaries related to modeling and control of snake robots (in the presence of full model knowledge) are provided in this section (see [4], [5], [6] for more details).

A. Equations of motion

We consider a planar snake robot consisting of $N_l > 1$ links and $N_l - 1$ joints, where each joint is actuated. For simplicity we assume that all links have the same length $l_i = l$ and the same mass $m_i = m$, for $= 1, \ldots, N_l$, hence the total mass of the robot is $\sum_{i=1}^{N_l} m_i = N_l m$.

We consider the simplified control-oriented model of snake robots introduced in [4], [6]. Namely, we let the joint angles ϕ_i and the joint velocity $v_{\phi,i}$ for $i = 1, \ldots, N_l - 1$ denote, respectively, the normal direction distance between links i and $i+1$ and the relative velocity between links i and $i+1$. Let $\phi = [\phi_1, \ldots, \phi_{N_l-1}]^T \in \mathbb{R}^{N_l-1}$ and $v_\phi = [v_{\phi_1}, \ldots, v_{\phi_{N_l-1}}]^T \in \mathbb{R}^{N_l-1}$. The coordinates of the centre

[1]The subscript d indicates measured data samples.

of gravity of the snake robot in the global frame is denoted by $p = (p_x, p_y)$ (with the subscripts denoting the x- and y-coordinates), whereas its tangential and normal direction velocity in the body frame are denoted by (v_t, v_n). Finally, θ and v_θ denote the global orientation of the snake, *i.e.* the angle between the body frame and the global frame, and the corresponding angular velocity, respectively. We utilise the following notation (similar to that of [5], [9], [10]). Let $e = \left[\ 1, \cdots, 1\ \right]^\top \in \mathbb{R}^{N_l}$, $\bar{e} = \left[\ 1, \cdots, 1\ \right]^\top \in \mathbb{R}^{N_l - 1}$, let $A \in \mathbb{R}^{(N_l-1)\times N_l}$ and $D \in \mathbb{R}^{(N_l-1)\times N_l}$ denote the matrices

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

and let $\bar{D} = D^T(DD^T)^{-1} \in \mathbb{R}^{N_l \times (N_l - 1)}$. We consider an anisotropic viscous model to represent the friction forces acting on the snake model (see *e.g.* [4], [6]). In what follows c_n and c_t denote the coefficients of friction in the normal and tangential directions, respectively, whereas c_p denotes the *propulsion* coefficient given by $c_p = (c_n - c_t)/(2l)$. The equations of motion of the snake robot are then given by

$$\dot{\phi} = v_\phi, \tag{9a}$$

$$\dot{\theta} = v_\theta, \tag{9b}$$

$$\dot{p}_x = v_t \cos\theta - v_n \sin\theta, \tag{9c}$$

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

$$\dot{v}_\phi = -\frac{c_n}{m} v_\phi + \frac{c_p}{m} v_t A D^T \phi + \frac{1}{m} D D^T u, \tag{9e}$$

$$\dot{v}_\theta = -\lambda_1 v_\theta + \frac{\lambda_2}{N_l - 1} v_t \bar{e}^T \phi, \tag{9f}$$

$$\dot{v}_t = -\frac{c_t}{m} v_t + \frac{2c_p}{N_l m} v_n \bar{e}^T \phi - \frac{c_p}{N_l m} \phi^T A \bar{D} v_\phi, \tag{9g}$$

$$\dot{v}_n = -\frac{c_n}{m} v_n + \frac{2c_p}{N_l m} v_t \bar{e}^T \phi, \tag{9h}$$

where λ_1 and λ_2 are defined as *rotational* parameters, capturing the rotational dynamics of the snake robot (see [4], [6]). The control input is $u = [u_1, \ldots, u_{N_l - 1}]^\top$, where u_i is the actuator force at joint i, for $i = 1, \ldots, N_l - 1$. Finally, let $x = \left[\phi^\top, \theta, p_x, p_y, v_\phi^\top, v_\theta, v_t, v_n\right]^\top \in \mathbb{R}^{2N_l + 4}$.

B. Snake locomotion

A difficulty in designing a controller for the snake robot described by equation (9) lies in the fact that the system is underactuated. Common techniques are aimed at "steering" the snake robot's heading in particular directions to follow a desired path ([6], [10]) by following a certain gait pattern. We focus our attention on the *inertial shape motion* and, in particular, we consider *undulatory gait patterns* (see, *e.g.* [2], [4], [5], [6], [9], [10]) that can be utilised to generate forward propulsive foces. That is, the snake robot is required to follow a serpenoid curve which is achieved when the joint angles ϕ_i follow a reference signal of the form

$$\phi_{i,\text{loc}}(t) = \alpha \sin(\omega t + (i-1)\beta) + \gamma \tag{10}$$

where α represents the amplitude of the motion, ω corresponds to the frequency and, β and γ are the phase shift between the joints and the joint offset, respectively, for $i = 1, \ldots, N_l - 1$. The corresponding locomotion is referred to as *lateral undulation*. The parameters α, β, ω and γ can be used to achieve trajectory control (see *e.g.* [5]). Considering the overall robot, lateral undulation for the ensemble of joint angles is defined as

$$\phi_{\text{loc}} = [\phi_{1,\text{loc}}, \phi_{2,\text{loc}}, \ldots, \phi_{N_l-1,\text{loc}}]^\top \in \mathbb{R}^{N_l - 1}. \tag{11}$$

The corresponding reference for the relative velocity between links is given by

$$v_{\phi,\text{loc}} = [\dot{\phi}_{1,\text{loc}}, \dot{\phi}_{2,\text{loc}}, \ldots, \dot{\phi}_{N_l-1,\text{loc}}]^\top \in \mathbb{R}^{N_l - 1}. \tag{12}$$

The following result provides a control law that ensures - in the presence of full model knowledge (including the friction coefficients) - that the desired locomotion (11) is achieved.

Lemma 2. *[4, Section V] Consider the system* (9)*. If the system parameters m, c_n and c_p are known exactly, then the choice of control law $u = u_{\text{fl}}^\star$, where*

$$u_{\text{fl}}^\star = m(DD^T)^{-1}\left(\bar{u} + \frac{c_n}{m}\dot{\phi} - \frac{c_p}{m} v_t(t) A D^T \phi\right), \tag{13}$$

partially feedback linearises the dynamics. Namely, (9e) *becomes $\dot{v}_\phi = \bar{u}$, where $\bar{u} \in \mathbb{R}^{N_l - 1}$ is the input of the partially feedback linearised snake robot model and can be designed to ensure the joint angles ϕ track a desired trajectory. The choice*

$$\bar{u} = \ddot{\phi}_{\text{loc}} + k_p(\phi_{\text{loc}} - \phi) + k_d(\dot{\phi}_{\text{loc}} - v_\phi), \tag{14}$$

with $k_p > 0$ and $k_d > 0$, ensures that $\lim_{t\to\infty}(\phi_{\text{loc}}(t) - \phi(t)) = 0$, for any initial condition $x(0) = x_0$.

IV. Control of the actuated degrees of freedom without exact knowledge of the model

The result in Lemma 2 is such that the desired locomotion (10), for $i = 1, \ldots, N_l - 1$, is globally attractive for the system (9) in closed loop with the control law (13)-(14). However, such result is not guaranteed to hold in the absence of complete knowledge of the model (9) (*e.g.* in the absence of accurate estimates of the friction coefficients). We consider instead the case in which only certain *nominal values* for the normal and tangential friction coefficients, denoted by $\tilde{c}_n$ and $\tilde{c}_t$ (potentially different from the actual friction coefficients c_n and c_t), are available for control design. The corresponding nominal propulsion coefficient is given by $\tilde{c}_p = (\tilde{c}_n - \tilde{c}_t)/(2l)$. Letting $u = u_{\text{fl}}$ in (9), with

$$u_{\text{fl}} = m(DD^T)^{-1}\left(\bar{u} + \frac{\tilde{c}_n}{m}\dot{\phi} - \frac{\tilde{c}_p}{m} v_t A D^T \phi\right), \tag{15}$$

designed on the basis of the nominal friction coefficients (as opposed to the control law (13) which is designed based on the actual friction coefficients), the terms including the friction coefficients no longer cancel out and (9e) becomes $\dot{v}_\phi = \frac{\tilde{c}_n - c_n}{m} v_\phi + \frac{c_p - \tilde{c}_p}{m} v_t A D^T \phi + \bar{u}$. To streamline the presentation, let $B_{fl} = m(DD^\top)^{-1}$, and $u_{fl,\text{nom}} =$

$B_{fl}\left(\frac{\tilde{c}_n}{m}\dot{\phi} - \frac{\tilde{c}_p}{m}v_t AD^T\phi\right)$. We will provide a mechanism to design a control law similar to (13)-(14), using data in place of exact knowledge of the friction coefficients. The result is achieved by considering the "actuated subsystem" of (9) with $u = u_{\mathrm{fl}}$, described by the dynamics

$$\begin{cases} \dot{\phi} = v_\phi \,, \\ \dot{v}_\phi = \frac{\tilde{c}_n - c_n}{m} v_\phi + \frac{c_p - \tilde{c}_p}{m} v_t AD^T\phi + \bar{u}\,, \end{cases} \tag{16}$$

with state $x_a = \left[\phi^\top, v_\phi^\top\right]^\top = Cx$, where

$$C = \begin{bmatrix} I_{N_l-1} & 0 & 0 & 0 \\ 0 & 0 & I_{N_l-1} & 0 \end{bmatrix}.$$

Remark 1. *Note that the evolution of* (16) *is dependent on* (9)*, via the tangential velocity* v_t*.*

Our objective is to provide a strategy to design a control law taking the place of (14), *i.e.* ensuring the joint angles of the snake robot (9) track the desired reference trajectory corresponding to undulatory locomotion, directly using data. Let the reference trajectory for the actuated states associated with lateral undulation be

$$x_{a,\mathrm{ref}} = \begin{bmatrix} \phi_{\mathrm{loc}}(t) \\ v_{\phi,\mathrm{loc}}(t) \end{bmatrix}, \tag{17}$$

with ϕ_{loc} and $v_{\phi,\mathrm{loc}}$ defined in (11) and (12), respectively, and introduce the error coordinates

$$\delta x_a(t) = x_a(t) - x_{a,\mathrm{ref}}(t). \tag{18}$$

In what follows, we assume that T (recall that we are interested in controlling the system over the interval $t \in [0, T]$) is a multiple of the sampling time T_s, such that $N = T/T_s$ and the discrete-time interval $k \in [0, N]$ corresponds to uniform samples of the continuous-time interval of interest. Consider the snake robot dynamics (9) and the controller $u = u_{\mathrm{cl}}$ with

$$u_{\mathrm{cl}}(t) = u_{fl,\mathrm{nom}}(t) + B_{fl}(\ddot{\phi}_{\mathrm{loc}}(t) + \delta\tilde{u}(t)), \tag{19}$$

where $\delta\tilde{u}(t)$ is obtained from a discrete-time feedback control law $\delta\tilde{u}[k]$ using zero-order hold (ZOH). Note that u_{cl} corresponds to the control law (15) with $\bar{u} = \ddot{\phi}_{\mathrm{loc}}(t) + \delta\tilde{u}(t)$.

In the following subsections we will derive a discrete-time LTV representation of the actuated subsystem (16) and demonstrate how to apply the results on direct data-driven control recalled in Section II-A to design the discrete-time control input $\delta\tilde{u}[k]$ (in the form of a state feedback). A block diagram representation of the control strategy is provided in Figure 1. The subsystem indicated by the green dashed line represents the (open-loop) system (9) with $u = u_{\mathrm{fl}}$ as defined in (15). The blue solid line highlights the actuated subsystem as defined in (16). The overall system considered for control design is indicated by the red solid line and is referred to as the DT actuated error subsystem. Note that the output of the DT actuated error subsystem is obtained by sampling the difference between the actuated states x_a and the reference signal $x_{a,\mathrm{ref}}$ at a (uniform) rate $1/T_s$, *i.e.* $\delta x_a[k] = \delta x_a(kT_s)$.

A. A LTV approximation of the actuated subsystem

To apply the direct data-driven control design method recalled in Section II-A, we linearise the actuated subsystem (16) about the reference trajectory (17) corresponding to lateral undulation. Consider the trajectories for $\bar{u}$ and v_t, labelled $\bar{u}^*_{\mathrm{ref}}$ and $v_{t,\mathrm{ref}}$ respectively, which ensure that (9) in closed loop with (15) is such that the actuated subsystem (16) follows the desired trajectory (17). Note that

$$\begin{aligned} \bar{u}^*_{\mathrm{ref}}(t) = \ddot{\phi}_{\mathrm{loc}}(t) - \frac{\tilde{c}_n - c_n}{m}\dot{\phi}_{\mathrm{loc}}(t) \\ - \frac{c_p - \tilde{c}_p}{m} v_{t,\mathrm{ref}}(t) AD^T \phi_{\mathrm{loc}}(t)\,, \end{aligned} \tag{20}$$

which is obtained by solving (16) in terms of $\bar{u}$ for $x_a = x_{a,\mathrm{ref}}$.

Introducing the error coordinates, $\delta\bar{u}(t) = \bar{u}(t) - \bar{u}^*_{\mathrm{ref}}(t)$, a linear approximation of the system (16) about the reference trajectories (17), (20) and $v_{t,\mathrm{ref}}$ yields the LTV system

$$\delta\dot{x}_a(t) = A(t)\cdot\delta x_a(t) + B(t)\cdot\delta\bar{u}(t) + \bar{d}(t), \tag{21}$$

where

$$A(t) = \frac{\partial f}{\partial x_a}\Big|_{x_{a,\mathrm{ref}}(t),\bar{u}^*_{\mathrm{ref}}(t),v_{t,\mathrm{ref}}(t)},$$

and

$$B(t) = \frac{\partial f}{\partial \bar{u}}\Big|_{x_{a,\mathrm{ref}}(t),\bar{u}^*_{\mathrm{ref}}(t),v_{t,\mathrm{ref}}(t)},$$

with

$$f = \begin{bmatrix} v_\phi \\ \frac{\tilde{c}_n - c_n}{m} v_\phi + \frac{c_p - \tilde{c}_p}{m} v_t AD^T\phi + \bar{u} \end{bmatrix},$$

and $\bar{d}(t)$ contains the higher order terms, as well as the effects of the deviation $v_t - v_{t,\mathrm{ref}}$, which is not explicitly accounted for since v_t is not considered a state of the actuated subsystem (16).

The feasible reference control input $\bar{u}^*_{\mathrm{ref}}(t)$, defined in (20), depends on both the actual and the nominal friction coefficients. Since we consider the actual friction coefficients c_n, c_p to be unknown, we introduce another change of coordinates, to a *known* reference $\bar{u}_{\mathrm{ref}}(t)$. Let $\delta\tilde{u}(t) = \bar{u}(t) - \bar{u}_{\mathrm{ref}}(t)$ and

$$d(t) = \bar{d}(t) + B(t)(\bar{u}_{\mathrm{ref}}(t) - \bar{u}^*_{\mathrm{ref}}(t))\,,$$

then (21) can be written as

$$\begin{aligned} \delta\dot{x}_a(t) = \begin{bmatrix} 0 & I_{N_l-1} \\ \frac{c_p - \tilde{c}_p}{m} v_{t,\mathrm{ref}}(t) AD^T & \frac{\tilde{c}_n - c_n}{m} I_{N_l-1} \end{bmatrix} \delta x_a(t) \\ + \begin{bmatrix} 0 \\ I_{N_l-1} \end{bmatrix} \delta\tilde{u}(t) + d(t). \end{aligned} \tag{22}$$

Note that $\bar{u}_{\mathrm{ref}}(t)$ is not required to be feasible (*i.e.* $\bar{u}_{\mathrm{ref}}$, $x_{a,\mathrm{ref}}$ and $v_{t,\mathrm{ref}}$ are not required to satisfy the actuated subsystem dynamics (16)), but it can be freely defined by the user.

To apply the results of Lemma 1 we discretise (22) using ZOH, resulting in a DT LTV system of the form

$$\delta x_a[k+1] = A_d[k]x[k] + B_d[k]\delta\tilde{u}[k] + d_d[k]\,. \tag{23}$$

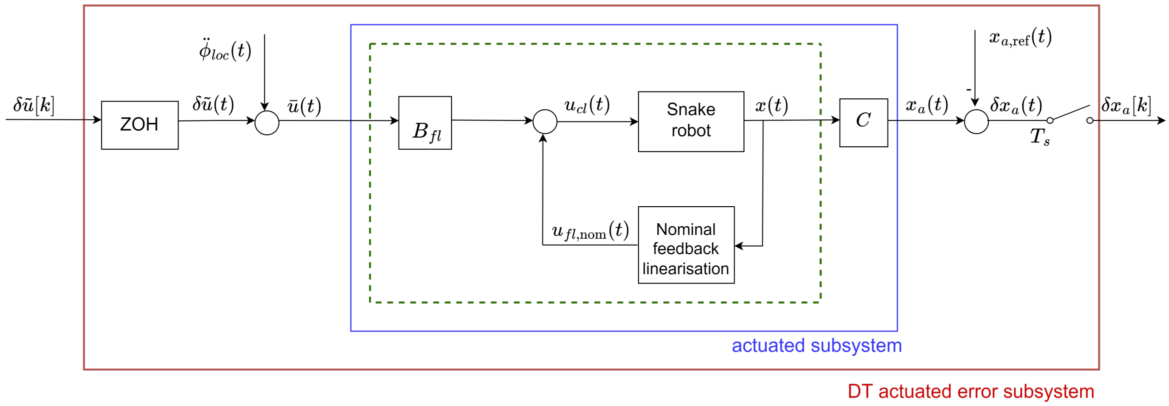

Fig. 1. Block diagram representation of the overall data-driven control strategy for the snake robot.

where the matrices $A_d[k]$, $B_d[k]$ are regarded as unknown. The process noise $d_d[k]$ captures the noise $d(t)$ as well as the effects of ZOH and sampling. In the following subsections, we will present a method to design (discrete-time) feedback control laws of the form

$$\delta\tilde{u}[k] = K[k]\delta x_a[k]\,, \tag{24}$$

guaranteeing a bound on the error trajectories δx_a, without knowledge of the LTV dynamics (22), using data. Exploiting the result in Lemma 1 the time-varying gain $K[k]$ can be designed *strategically*, without relying on manual tuning.

B. Data-driven control design

In the following we provide a direct data-driven strategy for designing $\delta\tilde{u}[k]$, of the form (24). The strategy relies on two steps: data collection and control design.

1) Data collection: The first step of the control strategy is to gather input-state data to form matrices similar to (3) and (4), which are used to represent the DT actuated error subsystem. To this end, we assume it is possible to perform multiple (L) simulations/experiments of the system (9) (where, as depicted in Figure 1, the input to the continuous-time system (9) is obtainend via a ZOH and the state data is obtained by uniformly sampling the difference bewteen the actuated states and the reference, *i.e.* δx_a). For the data collection, we choose $\delta\tilde{u}[k]$ of the form

$$\begin{aligned}\delta\tilde{u}[k] = k_p(\phi_{\text{loc}}[k] - \phi[k]) + k_d(\dot{\phi}_{\text{loc}}[k] - v_\phi[k]) \\ + u_{\text{exp}}[k]\,,\end{aligned} \tag{25}$$

where $k_p > 0$ and $k_d > 0$, and u_{exp} is a randomly generated "exploring input" such that $u_{\text{exp}}[k] \in [0,1]$, for $k = 0,\dots,N-1$. Note that (25) is similar to a discrete-time version of the PD controller utilised in (14) (introduced in Lemma 2), with the addition of the "exploring input" term $u_{\text{exp}}[k]$. The role of the feedback term is to ensure that the collected state data trajectories do not diverge rapidly (which may cause subsequent numerical issues), whereas the role of $u_{\text{exp}}[k]$ is to ensure that the rank condition (5) is satisfied for all $k \in [0,N]$. In practice such a randomly generated exploring input typically results in that the (easily verifiable) rank condition is satisfied. Proceeding as per Section II-A, the data collected through the L experiments are combined to form the data matrices

$$X[k] = [\delta x_{a,1}[k], \delta x_{a,2}[k], \dots, \delta x_{a,L}[k]]\,, \tag{26}$$

for $k = 0,\dots,N$, and

$$U[k] = [\delta\tilde{u}_1[k], \delta\tilde{u}_2[k], \dots, \delta\tilde{u}_L[k]], \tag{27}$$

for $k = 0,\dots,N-1$, where the subscripts i indicate data collected from a specific experiment, $i = 1,\dots,L$. As in Section II-A, consider also the matrix $D[k]$, for $k = 0,\dots,N-1$, containing the (unknown) process noise samples corresponding to the collected data in (26) and (27).

Remark 2. *A necessary condition for* (5) *to hold is that the number of experiments* $L \geq 3(N_l - 1)$*. Recalling Assumption 1, it is further required that the open-loop experiments capture the same time-varying behaviour (of the LTV approximation). In the current context, the evolution of* v_t *will differ from one experiment to another due to the different sequences of inputs* $\delta\tilde{u}_i[k]$*,* $k = 0,\dots,N-1$*,* $i = 1,\dots,L$*, applied during each experiment. This deviation* $(v_t - v_{t,\text{ref}})$ *is accounted for in the disturbance term of* (22).

2) Control design: The second step consists in using the data matrices (26), and (27) to design a control law for the DT actuated error subsystem, which ensures that the actuated states of the snake robot (9) stay within a given bound around the desired reference trajectories corresponding to undulatory locomotion (17). The control design for the snake robot is detailed in the following statement that utilises Lemma 1.

Lemma 3. *Consider a snake robot described by the dynamics* (9)*. Suppose data matrices* (26) *and* (27) *are available and these are such that the rank condition* (5) *holds. Further suppose D[k] satisfies a quadratic bound of the form* (6) *for* $k = 0,\dots,N-1$*. Then a control input of the form* (24)*, with* $K[k]$ *as given in* (7)*, where* $Y[k]$*,* $P[k]$ *are a solution of the feasibility problem* (8)*, is such that the trajectories of* *the closed-loop DT actuated error subsystem satisfy*

$$\|\delta x_a[k]\| \le \sqrt{\frac{\rho}{\eta}}\left(1-\frac{1}{\rho}\right)^{\frac{k}{2}}\|\delta x_a[0]\| + \gamma_1(|d|_{k-1},k), \tag{28}$$

for $k = 0,\dots,N$, where $\eta \ge 1$, $\rho > \eta$ are constants and $\gamma_1(\cdot,k)$ is a class $\mathcal{K}$ function.

V. Simulation

Consider a snake robot with $N_l = 3$ links of mass $m = 1$ kg and of length $l = 0.14$ m. The coefficients of frictions are taken to be $c_t = 1$ and $c_n = 3$ (such that $c_p = 7.1429$) and are considered to be unknown. The values of λ_1 and λ_2 are set to, respectively, 0.5 and 20. The aforementioned parameters are similar to those considered in [6, Chapter 6.10]. The parameters for the undulatory motion are chosen as $\alpha = 0.045$ m, $\omega = 2.0944$ rad/s, $\beta = 0.6981$ rad and $\gamma = 0$ rad. We consider the case in which the coefficients of friction are unknown, with only certain nominal values, *i.e.* $\tilde{c}_t = 1.2$ and $\tilde{c}_n = 4$ (resulting in the nominal propulsion coefficient $\tilde{c}_p = 10$), available for control design.

In the following the performance of two controllers is compared via simulations. Namely, considering the overall control law $u_{\rm fl}$ given by (15), we compare the performance of the closed-loop system for two different selections of $\bar{u}$. Specifically, we consider the case in which $\bar{u}$ is the (continuous-time) controller given by (14), and the case in which $\bar{u} = \ddot{\phi}_{\rm loc} + \delta\tilde{u}(t)$, as in (19), where $\delta\tilde{u}(t)$ is obtained (via ZOH) from the data-driven (discrete-time) controller (24), with the time-varying gains designed according to Lemma 3. The overall control law $u_{\rm fl}$ corresponding to the former selection is denoted by $u_{\rm PD}$, whereas the one corresponding to the latter (data-driven) selection is denoted by $u_{\rm DD}$. We emphasise again that only nominal values for the coefficients of friction are utilised to design the two controllers. The gains of $u_{\rm PD}$ are chosen to be $k_p = 20$ and $k_d = 5$ (as in [6]). The same values of k_p and k_d are selected for data collection (*i.e.* in (25)). For the data-driven controller the sample time is taken to be $T_s = 0.02$ s and the time horizon $N = 800$ s. The values of ρ and η in (8) are chosen to be $\rho = 60$ and $\eta = 1$. The system is simulated in MATLAB using `ode45`, whereas the feasibility problem (8) is solved using `CVX` [15]. In the following we consider the initial conditions of (9) to be $\phi_1(0) = 0.1$, $\phi_2(0) = -0.1$, whereas those of the remaining states, namely the initial conditions of $\theta, p_x, p_y, v_{\phi,1}, v_{\phi,2}, v_\theta, v_t, v_n$, have been randomly generated in $[0,1]$. The time histories of the norm of the error coordinate δx_a corresponding to $u_{\rm fl} = u_{\rm PD}$ (dashed, grey line) and $u_{\rm fl} = u_{\rm DD}$ (solid, blue line) are shown in Figure 2, along with the time history of the bound (28) (solid, red line). It is clear that (28) is satisfied by the trajectory corresponding to the controller $u_{\rm DD}$, but not for that corresponding to the controller $u_{\rm PD}$.

VI. Conclusion

A direct data-driven approach to control the locomotion of a snake robot has been presented. The result uses data in place of an accurate model to construct time-varying state feedback control laws that ensure the snake robot follows an undulatory locomotion. Using nominal values for friction coefficients and data collected from offline experiments, the proposed control design relies on a LTV approximation of the actuated subsystem about the desired trajectory, and the subsequent solution of a purely data-dependent convex feasiblity problem. The resulting control law is demonstrated, via simulations, to result in improved performance with respect to an alternative model-based control law (designed purely on the basis of the nominal parameter values available).

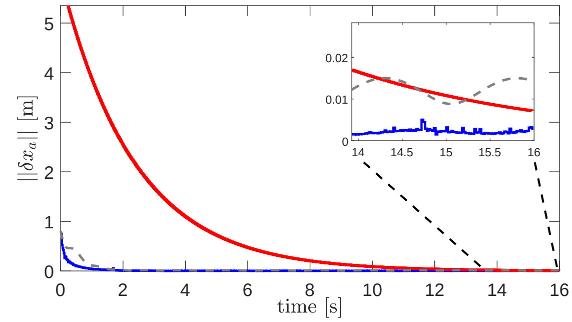

Fig. 2. Time histories of the bound (28) (solid, red line) and of the norm of δx_a for $u_{\rm fl} = u_{\rm PD}$ (dashed, grey line) and $u_{\rm fl} = u_{\rm DD}$ (solid, blue line).

References

[1] J. Gray, "The mechanism of locomotion in snakes," *Journal of Experimental Biology*, vol. 23, no. 2, p. 101, 1946.

[2] S. Hirose, *Biologically inspired robots*. Oxford Univ. Press, 1993.

[3] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A review on modelling, implementation, and control of snake robots," *Robotics and autonomous systems*, vol. 60, no. 1, pp. 29–40, 2012.

[4] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A simplified model of planar snake robot locomotion," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2868–2875.

[5] K. Y. Pettersen, "Snake robots," *Annual Reviews in Control*, vol. 44, pp. 19–44, 2017.

[6] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control*. Springer, 2013.

[7] R. Bogue, "Snake robots: a review of research, products and applications," *Industrial Robot*, vol. 41, no. 3, pp. 253–258, 2014.

[8] M. M. Tonapi, I. S. Godage, A. Vijaykumar, and I. D. Walker, "A novel continuum robotic cable aimed at applications in space," *Advanced Robotics*, vol. 29, no. 13, pp. 861–875, 2015.

[9] M. Sato, M. Fukaya, and T. Iwasaki, "Serpentine locomotion with robotic snakes," *IEEE Control Syst. Mag.*, vol. 22, no. 1, pp. 64–81, 2002.

[10] E. Kelasidi and A. Tzes, "Serpentine motion control of snake robots for curvature and heading based trajectory - parameterization," in *20th Mediterranean Conference on Control Automation*, 2012, pp. 536–541.

[11] C. D. Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[12] H. J. van Waarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: Nonconservative design via a matrix s-lemma," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 162–175, 2022.

[13] B. Nortmann and T. Mylvaganam, "Data-driven control of linear time-varying systems," in *59th IEEE Conference on Decision and Control*, 2020, pp. 3939–3944.

[14] ——, "Direct data-driven control of linear time-varying systems," Nov 3, 2021. [Online]. Available: https://arxiv.org/abs/2111.02342

[15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, 2014.